Evaluation of Software License Management Frameworks for Grid Environments: The Four Ts for Agile Systems

Filip Eriksson (filer663) & Joel Östlund (joeos007)
2/15/2014

Contains a Case Study and Experiences from:

Master’s thesis
Economic Information Systems
Industrial Engineering and Management
Department of Management and Engineering
Linköping University, Sweden
LIU-IEI-TEK-A--14/01801--SE
Evaluation of Software License Management Frameworks for Grid Environments:
The Four Ts for Agile Systems

Filip Eriksson (filer663) & Joel Östlund (joeos007)
2/15/2014
Examiner: Alf Westelius
Mentor: Carl-Johan Petri

Master’s thesis
Economic Information Systems
Industrial Engineering and Management
Department of Management and Engineering
Linköping University, Sweden
LIU-IEI-TEK-A--14/01801—SE
Abstract

Business Agility could be the means to survive in a competitive environment of continuous and unanticipated change and to respond quickly to rapidly changing, fragmenting global markets that are catered by competitors. To leverage the potential benefits of Business Agility, many companies use Information Technology as a major force for augmenting their agility. The topic of increasing Business Agility through Information Technology has been pervasively studied in research papers. And in the context of this thesis, we will investigate research papers in order to develop an evaluation model for facilitating making informed decisions regarding agile Information Systems in computational Grid environments. The Information Systems in this thesis will focus on license management systems, which is an intricate but coherently integrated blend of many different modules compiled as a technical solution that enables software license management. We will use our evaluation model to compare license management frameworks, easily grasped as a draft for a license management system, for Grid environments and also underpin it with a case study on an in-use license management system in order to strengthen its business applicability.

The research was conducted as a qualitative comparison and a case study, combining both research papers and a real business context in order to fulfill the goals and purposes of this thesis.

The results of the analysis implies that most of the license management frameworks for Grid environments lacks various details needed for making a feasible implementation in a certain business context, whilst those that have been trialed in a real business situation fulfilled almost every aspect of our evaluation model.

The conclusion drawn is that our evaluation model is a seemingly good start for making informed decisions when choosing or designing an agile license management system for Grid environments.
Acknowledgements

We give our special thanks and regards to the following persons involved in contributing to our essay:

Carl-Johan Petri, our supervisor at Linköping University: for tireless efforts, valuable insights, and countless hours of devotion to making this essay possible.

Andreas Ehrlund, our supervisor at Ericsson: for the daily guidance in the field of software licensing and always effortlessly helping us out when needed.

Johan R. Olsson, our contact person for arranging interviews: for the deep and nuanced insights and participation in making our essay possible.

Alf Westelius, our examiner at Linköping University: for interesting angles, discussions, and suggestions.

Axel Crona and Jonna Bertilsson, our opponents at Linköping University: for rigorous participation and well thought through suggestions.

All the interview subjects and other people participating in making our essay possible: it would not have been possible without your participation.
Table of Contents
1. Introduction ..................................................................................................................... 7
   1.1. Background ................................................................................................................. 7
   1.2. Problem discussion ..................................................................................................... 8
   1.3. Purpose and research questions ................................................................................. 9
   1.4. Delimitations .............................................................................................................. 10
   1.5. Basic terminology ..................................................................................................... 10
   1.6. Report structure ......................................................................................................... 13
2. Methodology .................................................................................................................... 14
   2.1. Background of the authors ......................................................................................... 14
   2.2. The Study Object ...................................................................................................... 14
   2.3. The Evaluation Model ............................................................................................... 15
   2.4. Approach .................................................................................................................. 16
   2.5. Approach criticism .................................................................................................... 21
   2.6. Scientific approach ................................................................................................... 22
   2.7. The literature study ................................................................................................... 24
   2.8. The case study .......................................................................................................... 27
3. Theoretical framework .................................................................................................... 34
   3.1. Introduction ................................................................................................................ 34
   3.2. Business Agility ........................................................................................................ 35
   3.3. From theory to evaluation model .............................................................................. 44
   3.4. The Evaluation Model ............................................................................................... 51
4. Our interpretation of Ericsson’s criteria in relation to the Evaluation Model ................. 52
   4.1. Evaluation Model comparison .................................................................................. 55
5. A Spectrum of License Management Frameworks ............................................................ 56
6. Analysis ............................................................................................................................ 85
   6.1. SmartLM ................................................................................................................... 85
   6.2. GenLM ...................................................................................................................... 93
   6.3. Service Oriented License Providing (SOLP) ............................................................. 99
   6.4. On-demand software license provisioning in Grid and Cloud computing (OSGC) .... 105
   6.5. Floating License Sharing System (FLSS) ................................................................. 111
   6.6. The LMF comparison ............................................................................................... 116
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMS</td>
<td>License Management System</td>
</tr>
<tr>
<td>LMF</td>
<td>License Management Framework for Grid environments</td>
</tr>
<tr>
<td>ISV</td>
<td>Independent Software Vendor</td>
</tr>
<tr>
<td>ASP</td>
<td>Application Service Provider</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
</tbody>
</table>
1. Introduction

The first subchapter will provide the background to the problem. This leads to the next subchapter which discusses the core of the problem and why there is need for further research. Later, by clarifying the purpose, research questions, and delimitations of the study facilitates to reduce the abstraction of the scope. Subsequently, the purpose of the basic terminology section is to clarify important concepts in order to prevent misunderstanding and instead give the reader the prerequisites to fully understand this thesis, this section is recommended to read first for readers that are not fully experienced within the field of this thesis. Finally, the outline of this thesis is presented.

1.1. Background

At present, companies have to survive and develop competitive advantage in a dynamic and turbulent environment of global competition and rapid business change (Sambamurthy, et al., 2003; van Oosterhout, 2010); hence augmenting the urge to proactively predict the changes before they impact business operations (Thao, et al., 2012). Surviving in an ever changing environment has since the last decades been crucial, owing to that the lifespan of products has dramatically declined since the beginning of the 1990s (Melarkode, et al., 2004). A company’s ability to deal with these kinds of issues is denoted Business Agility and have been comprehensively summarized by van Oosterhout (2010) as:

“[...] the ability of an organization to swiftly change businesses and business processes beyond the normal level of flexibility to effectively manage highly uncertain and unexpected but potentially consequential internal and external events, based on the capabilities to sense, respond and learn.”

Software licensing mechanisms are controls put in software to grant or deny the use of the whole or parts of the software, even though it might be fully installed on one hardware device and already capable of bringing the whole functionality. It has played an important part in the distribution and control of software, especially commercial software. (Manoharan & Wu, 2007) The importance for commercial software to use software licensing mechanisms is because these vendors often, beside the acquisition price, use license fees to sustain and increase profits. (Mazhelis, et al., 2013; Ferrante, 2006) It is therefore important for these vendors that the licensing shall be strong enough to mitigate cracking and piracy (Manoharan & Wu, 2007). From an enterprise perspective, the biggest issue is inadvertently application of the product in a way that violates the software license agreement. Software vendors can protect their assets with the act of auditing the customer and enforcing the legal agreement by building mechanisms into the product that disable the use when the purchaser violates any terms of the agreement. (Ferrante, 2006)

In a McKinsey Quarterly article, Persson et al. (1999) claimed that for most of the industries producing software, their transition towards software licensing has not been easily grasped. An example is traditional providers of telecom equipment that have long been hardware oriented; this industry failed to seize emerging software opportunities and the new services they made possible. As hardware prices are declining, the importance of capturing value from software becomes even more important as it offers economies of scale. In traditionally hardware oriented industries, the transition from hardware to
software orientation is a potential growth area, and they should therefore develop a clear strategy for the parts of their businesses that depend on software. (Persson, et al., 1999)

This research paper is conducted on the international telecommunications company Ericsson, who had the problem described in the previous paragraph. Ericsson began to seize the emerging opportunities for software licensing during the early 1990s. (Process Architect ELIS & BNET, 2013) Since then, a lot of new software licensing revenue models has appeared on the market, which in some cases demands new technologies for software licensing practices (Robinson, 2006). Ericsson have been reluctant in this field, and are therefore curious to explore new software license management techniques that adheres to contemporary technical and customer demands (Process Architect ELIS & BNET, 2013). A well-known problem of sticking to an old computer system is posed by the IT-flexibility paradox that claims that strategic benefits of Information Technology are not sustainable over time. This is due to that technology can contribute to organizational flexibility since (new) IT is inherently more flexible than its predecessors. However, since technology ages so rapidly and becomes hard to maintain, flexibility is quickly lost, which affects the companies’ ability to reactively adapt to their environment. (Lukas Jr & Olson, 1994) This causes a problem since executives often find it difficult to justify large expenditures on IT, especially if they are not tied to a specific business benefit and, thus, causes the expenditure to be seen as an unaffordable overhead or even as a dispensable IT “toy”. (Duncan, 1995) Ericsson hopes that a new license management system could bring greater profits due to the new abilities (and flexibility) that it might unleash (Process Architect ELIS & BNET, 2013); hence augmenting the company’s need for increased Business Agility (Ericsson’s Strategic Direction, 2013; Director of Software and Product Packaging, 2013). Due to the nature of telecom networks, Grid computing is a feasible choice for the business environment. (Berde, et al., 2009; Robertazzi, 2007; Future Generation Grids, 2004) Since a Grid is a specialization of a distributed computing environment (Grid café, 2013), the following authors agree to that Grid computing is the most feasible choice for working with telecom networks. (Gregory, 2000; Sukumar, 2007; Peleg, 2000)

1.2. Problem discussion
As the case of Ericsson in the previous section mentions, new ways of working with software licensing brings the hope of sustaining and increasing profits (which Mazhelis et al. (2013) and Ferrante (2006) also mentioned was the purpose of software licensing).

As the IT-flexibility paradox states: old systems lead to diminishing returns on the strategic value of Information Technology. A way to still keep the systems flexible could be to period-wisely implement new computer systems. Enhancing the flexibility of a system could reasonably increase the Business Agility of the company; hence augmenting the ability to proactively predict the changes before they impact business operations. A shortcut for implementing a new software licensing management system in order to upgrade or implement a new system to support contemporary technologies and maintain flexibility in Grid environments is to use a proposed license management framework adjusted to the computational environment.

As this thesis was conducted at Ericsson, the assignment was to investigate feasible software license management frameworks for their business environment and contemporary technical demands. As
Ericsson’s Strategic Direction (2013) urges for boosting the company’s Business Agility, this research paper will be based upon elaborating a framework to evaluate different software license management frameworks from a Business Agility perspective. To the best of our knowledge, there has been no previous research conducted on the topic of comparing license management frameworks for Grid environments, which makes this paper unique in its field of study.

This thesis, reports on our investigation of research articles that elicit criteria for computational Grid environments that augment Business Agility. The criteria are compiled into an evaluation model that is used to evaluate the different licensing management frameworks for Grid environments that are proposed by the scientific literature. The next step is to check the applicability of the Evaluation Model with regard to the case study. After this, the license management frameworks for Grid environments are compared with the Evaluation Model, without intention of making recommendations for choosing a framework. After this, we discuss the various frameworks and what would be suitable for Ericsson. At last, we draw conclusions regarding our evaluation model concerning its generalizability and applicability.

1.3. Purpose and research questions
The purpose of this study is to develop an evaluation model for license management frameworks for Grid environments, validate this evaluation model through expert interviews, and apply it on a set of license management frameworks for Grid environment. The Evaluation Model development leads to the first research question:

**RQ1:** Which aspects have previous authors argued for regarding systems in grid environments?

These aspects were synthesized into a number of categories. The Evaluation Model is then validated by licensing experts after answering the second research question:

**RQ2:** Which criteria do five licensing experts at Ericsson consider important when evaluating a license management system?

In order to apply the Evaluation Model we must discover which license management frameworks for Grid that has been described by previous research which leads to the third research question:

**RQ3:** Which license management frameworks for Grid environment are mentioned in previous research?

After applying the Evaluation Model on the license management frameworks for Grid environment the last two research questions can be answered:

**RQ4:** To which extent do license management frameworks for Grid environment, presented in previous research, match the aspects of our evaluation model?

**RQ5:** Which license management frameworks for Grid environments seem to be the most feasible for the particular case study?
1.4. Delimitations
When creating the Evaluation Model, criteria were gathered from previous research in the fields of Business Agility, systems in Grid environment, and licensing systems in Grid environment.

In order to validate the Evaluation Model, Ericsson’s criteria regarding an LMS were gathered. It would have been interesting to use licensing experts from more companies in order to increase the generalizability of the Evaluation Model, but we will leave that aspect as a case for further research. Also five employees were interviewed for these criteria and it might have been useful to talk to more people at Ericsson, but this was also not prioritized due to time limitations. There were limitations regarding the information received from Ericsson, where criteria could be discarded for the following three reasons (these are more thoroughly described in section 2.4.2):

- The criterion is considered too fundamental
- The criterion is considered too implementation specific
- The criterion is not within the scope of being applicable on an LMF

There was one limitation regarding the LMFs that were chosen. They had to support Grid environment since it, according to Berde et al. (2009), Robertazzi (2007), and Future Generation Grids (2004) is feasible in the telecom industry. The information regarding the LMFs could include the following (which can be found in section 2.2):

1. **The authors’ knowledge**: The aggregated expertise of the authors of the frameworks articles, regarding the framework, which refers to knowledge that has not been documented.
2. **The documentation of the framework and its references**: Apart from the actual framework articles, there might be other texts written by the authors themselves or somebody else that clarifies or completes its content.
3. **General method**: The general method explains on a conceptual level how the framework can be used. This method can be used regardless of the chosen implementation.
4. **The requirement specification**: The requirement specification describes how the framework can be implemented with regard to technical aspects.
5. **System implementation**: The system implementation is an actual implementation of the framework.

Only the documentation of the framework and its references were used (bullet 2) and an explanation of this choice can be found in section 2.2. The information presented regarding the LMFs was limited to give a brief overview of the framework and then give the information that was available regarding the criteria of the Evaluation Model.

1.5. Basic terminology
This section is meant to provide the reader with general knowledge to fully understand the concepts in this thesis. All the details in this section will not be mentioned in the rest of the thesis, but will be useful for the reader to grasp for his own understanding of important concepts.
1.5.1. License
German & Hassan (2009) defines the license as a set of grants that each has their set of conditions, which must be achieved for the grant to be given. A license in turn consists of one or more rights that, in turn, have zero or more obligations. This is visualized in the meta-model by German & Hassan (2009) which is shown below:

Furthermore, Jena & Thompson (2005) explains that the purpose of a license is to protect the owner’s property rights. This determines how the content can be sold or used, in order to regulate the compensation.

1.5.2. Software licensing
Software licensing is any procedure that lets an enterprise or user purchase, install, and use software on a machine or network in accordance with the agreement specified by the software producer or as negotiated between the parties. All software licensing aims to protect both the vendor’s investment by minimizing the risk of attempts to bypass a product’s licensing protections and the enterprise’s investment by minimizing the risk of auditing fines from inadvertently apply the product in a way that violates the software license agreement. Software vendors can protect their products by building in mechanisms that enforces the legal agreement and disable unauthorized use when the purchaser violates any aspects of the agreement. (Ferrante, 2006)

Software licensing mechanisms are controls put in software to grant or deny the use of the whole or parts of the software, even though it might be fully installed on one computer and already capable of bringing the whole functionality (Manoharan & Wu, 2007). Software licensing is changing how organizations and individuals purchase and use software. For enterprises that rely on software to maintain a market share, the software licensing model can strongly influence the return on software investment (Ferrante, 2006).
1.5.3. Grid computing
In the article “The Grid Economy”, Buyya, et al. (2005), comprehensively summarizes the background and definition of a Grid computational system:

“A grid enables, the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organization for solving large-scale resource intensive problems in science, engineering, and commerce.”

1.5.4. Resources
Open Grid Forum (2007), which is a community of users, developers, and vendors for standardization of grid computing, defines a resource in a Grid environment as:

“A resource is a physical or logical entity that supports use or operation of a computing application or environment. In a grid environment, those resources are often accessed through services and encompass entities that provide a capability or capacity (e.g., servers, networks, disks, memory, applications, databases, IP addresses and software licenses). Dynamic entities such as processes, print jobs, database query results, and virtual organizations may also be represented and handled as resources.”

1.5.5. License Management System (LMS)
In this thesis we define License Management System as:

“a technical solution that enables software license management.”

1.5.6. License Management Framework
In this thesis we define License Management Framework as:

“established practices for creating, managing, interpreting, analyzing, and using systems for software license management.”

A License Management System could easily be grasped as an implementation of a License Management Framework, but is not always the case. In the context of this thesis we will only take License Management Framework for Grid environments in consideration. Hereafter, we will denote License Management Framework for Grid environments as LMF.
1.6. Report structure

The structure of the report is relatively standard for a research report. The structure is presented to provide the reader with an overview of what can be expected of the coming chapters.

- **Chapter 2**: presents the methodology of this study, which serves the purpose of presenting the methods used for this study.
- **Chapter 3**: contains the theoretical frame of reference.
- **Chapter 4**: comprises our interpretation of Ericsson’s criteria in relation to the Evaluation Model
- **Chapter 5**: contains a presentation of the different study objects, namely the LMFs.
- **Chapter 6**: is the analysis chapter.
- **Chapter 7**: is the discussion and conclusion chapter.
- **Appendix**: contains information regarding key word used in this essay and the material used for interviews.
2. Methodology

In this chapter the methodology of the thesis is presented. First the background of the authors is presented, followed by an explanation of the Study Object and the Evaluation Model. The overall approach of the thesis is then explained, which will give a summary of how the work was structured in order to fulfill the purpose and answer the research questions. After that the scientific approach towards knowledge is presented. Then, the structure of the literature and empirical study is more thoroughly described under their respective headlines. The generalizability of the thesis is then presented, which implies explaining to what extent in which the result of this thesis can be used.

2.1. Background of the authors

We are Master students at Linköping University, studying Industrial Engineering and Management with specialization in Computer Science and Economic Information System. We did not have any licensing or telecom experience since before we began this research which also was a choice from Ericsson since they wanted their Research Interns to be able to think “outside of the box”. The thesis implied interaction with interviewees with technical knowledge and the technical specialization in Computer Science was important in order to understand them and the Study Object. Interviews were also held with interviewees with a more overall and strategic role, and for those, both the overall economic background was useful but also the specialization towards Economic Information Systems.

2.2. The Study Object

In this thesis the Study Object is the LMFs, which are analyzed using the Evaluation Model that is specified in section 2.3., the Evaluation Model. The purpose of this section is to clearly define an LMF from the perspective of this thesis.

At first, the extent of the LMF will be explained, which in our thesis, can include all the five aspects explained below:

1. The authors’ knowledge: The aggregated expertise of the authors of the frameworks articles, regarding the framework, which refers to knowledge that has not been documented.
2. The documentation of the framework and its references: Apart from the actual framework articles, there might be other texts written by the authors themselves or somebody else that clarifies or completes its content.
3. General method: The general method explains on a conceptual level how the framework can be used. This method can be used regardless of the chosen implementation.
4. The requirement specification: The requirement specification describes how the framework can be implemented with regard to technical aspects.
5. System implementation: The system implementation is an actual implementation of the framework.

Even though LMFs theoretically could include all those aspects, they will in this thesis only include the second aspect, the documentation of the framework and its references.
Since the frameworks can be implemented in a lot of different ways it would be too time consuming to compare different implementations instead of the overall concepts, and hence aspect four and five are discarded.

Furthermore, aspect three is discarded since the method, for how the framework shall be used, is not within the scope of this thesis; it is more important what is included in the frameworks rather than how it can be done.

The first aspect will not be regarded either because it is assumed that all important aspects, that are supported by research, have been documented; the scientists of the LMFs want the readers to use their frameworks and their advantages should hence be mentioned. When focusing on the second aspect the most important texts were the descriptions of the frameworks in academic articles and on their respective websites (in case the websites were available). These articles and websites presented both conceptual ideas and more implementation specific aspects of the frameworks. The implementation specific aspects were discarded unless they brought clarity to general ideas. The references of the framework articles were used for clarification which in some cases implied going even further back among references.

2.3. The Evaluation Model

The purpose of this thesis is to elaborate a framework for evaluating the study object. This framework is referred to as the Evaluation Model. It consists of criteria, from previous research, which have been categorized in order to make the Evaluation Model more perspicuous. The criteria and sub-criteria of the Evaluation Model are used for evaluating LMFs. Hence, the difference between LMF criteria and LMS criteria are explained in the next section.

2.3.1. Clarification of criteria for LMFs and LMS

In this section the difference between an LMF and LMS criteria is clarified, using figure 2-1. First of all, an LMS can be regarded as an implementation of an LMF, but it does not have to be based on an LMF. An LMF can enable several different LMSs (as shown in the figure); a criterion valid for an LMF is therefore valid for its LMS. Criteria for an LMS might, however, be too implementation specific and are therefore not always relevant for an LMF. The criteria from previous research used were general and could therefore hold for both an LMS and LMF. Ericsson has both requirements that are general, and directly applicable for an LMF, and more implementation specific, that either could confirm a general criterion or else be discarded. Hence, Ericsson’s requirements are regarded as LMF criteria.
2.4. Approach

In this section the approach of this thesis is presented. This is done by dividing it into the three sequential phases: initial study, Establish the Evaluation Model and the Study Object, and the analysis, which are presented separately, and this is visualized in figure 2-2.

![Diagram of LMF and LMS relationship](image)

**Figure: The relation between an LMF and an LMS**

2.4.1. The initial study

Klein & Myers (1999) suggest in their first principle (The Fundamental of the Hermeneutic Circle) of their article A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems, that:

“[...] all human understanding is achieved by iterating between considering the independent meaning of parts and the whole that they form”.

The initial study has been an iterative process which has mainly consisted of understanding the problem background at Ericsson and gathers previous research in order to understand what has been accomplished in the field of licensing. In the end of the initial study the scope was specified.

**Understand the problem background**

The first part of the thesis implied understanding the licensing situation at Ericsson. This included both a general presentation of Ericsson, but also an introduction to the licensing situation at Ericsson. This introduction consisted of a licensing history introduction, an explanation of the current licensing situation, and some requirements regarding an alternative licensing solution.
Gather previous research

The first step, when looking through the literature, was to get familiar with the field of licensing and the accomplishment of previous research. This implied reading about licensing and understand its advantages and challenges. Thereafter the literature was used to find and understand important terminology and technologies regarding licensing. It was during this step that solutions for handling the licensing were found, which in this thesis are referred to as LMFs and LMSs.

Define the scope

The scope of the thesis was not agreed upon before the start, but was about to be more clearly specified. The objective and important steps for it to be accomplished, was determined. Furthermore, what was included in a license management solution is diffuse and needed to be explained (which is done in section 2.2. The Study Object).

2.4.2. Establish the Evaluation Model and the Study Object

This phase consisted of four separate steps: create the Evaluation Model, make our interpretation of Ericsson’s criteria in relation to the Evaluation Model, compare it with the evaluation, and present the Study Object.

The establishment of the Theoretical Evaluation Model and the Study Object implied a literature study, whilst creating our interpretation of Ericsson’s criteria in relation to the Evaluation Model induced a case study (how these are performed are described thoroughly in section 2.7. The literature study and 2.8. The case study).

The data collection for the four steps was not performed sequentially due to the limited time where the interviews of the case study were the limiting factor. The employees of Ericsson were not always available, whilst the literature study could be performed at any time, and this implied that the empirical study had a higher priority.

There was however some dependencies regarding the descriptions of the four steps where the Evaluation Model was created first. Ericsson’s criteria were supposed to follow the structure of the Evaluation Model and were hence created after it. After that the Evaluation Model could be compared with Ericsson’s criteria in order to confirm its applicability for the particular case study.

The description of the Study Object followed the structure of the Evaluation Model and was established after the comparison, since the outcome of the comparison affected the analysis. In case the Evaluation Model would have diverged from Ericsson’s criteria then another structure might have been more feasible for describing the Study Object.
Create the Evaluation Model

In order to be able to fulfill the purpose of this thesis (which is to elaborate an evaluation model for license management frameworks for Grid environments) and answer RQ1 (Which aspects have previous authors argued for regarding systems in grid environments?) the Evaluation Model was created using criteria from previous research.

In order to avoid too niched criteria and thereby increase the likelihood for the chosen criteria to be generic and relevant for the case study, their importance had to be emphasized by more than one author. Furthermore, sub-criteria were identified and described and the intention of using them was to facilitate the discussion of the criteria.

The criteria that were gathered were important aspects related to Business Agility IT-architecture, Grid, and licensing, accomplished by previous research. These have been synthesized and categorized in order to create the Evaluation Model.

The criteria within a category and whole categories are supposed to be mutually independent, in order to avoid redundant criteria (and in worst case redundant categories). This was however hard to accomplish, but similar criterion were merged into one, in order to avoid redundant criterion. These categories and their respective criteria became the Evaluation Model.

Make our interpretation of Ericsson’s criteria in relation to the Evaluation Model

This thesis was performed at Ericsson and their LMF requirements were gathered in order to strengthen the perception that the Evaluation Model is feasible for the particular case study (RQ2), but also for discussing the LMFs applicability for the particular case study (RQ5).

Ericsson’s criteria were gathered in two ways: either by reading internal documents or by interviewing Ericsson’s employees and this is more thoroughly described in section 2.8.3 Collection of Ericsson’s criteria.

An internal document shows awareness and accordance regarding the criteria, and is thereby considered reliable. In case there is no formal documentation the interpretation of interviewees must be regarded.

If several interviewees emphasize the significance of a criterion it decreases the likelihood for it to be irrelevant.

However, in order not to miss any aspects, no criteria, presented by employees at Ericsson or written in internal documents, were discarded because they were not confirmed by another source. Some criteria were however discarded which could be for the following reasons:

- The criterion is considered too fundamental
- The criterion is considered too implementation specific
- The criterion is not within the scope of being applicable on an LMF
Too fundamental criteria were considered tacit and do not need to specified. An example is presented in Focal Point report where Product Manager License Management, BNET (2013) writes the following:

“The access to software functionality and capacity, licensed by Ericsson to customers, shall be provided by means of software keys used by a License Manager which is a software key and lock functionality used by our platforms and applications”.

Criteria considered too implementation specific might not rely on the LMFs. Ericsson have for instance several criteria in Focal Point Report that emphasizes the importance of their LMS to be safe and robust. This should not depend on the LMF chosen but on the way in which it is implemented. Furthermore, there are criteria in Focal Point Report saying that each hardware module shall have a unique identification (a so called fingerprint) and this also is something that does not rely on the LMF chosen.

The criteria that have been considered out of the scope of the LMFs have been rules regarding the people who are interacting with the LMS, and this includes for instance tasks of the regions of Ericsson.

After gathering these criteria, they were categorized in the same manner as the Evaluation Model and hence became our interpretation of Ericsson’s criteria in relation to the Evaluation Model. Since no further categories or criteria were needed the structure of the Evaluation Model could be followed. If that would not have been the case, categories or criteria would have been added.

**Compare the Evaluation Model with our interpretation of Ericsson’s criteria in relation to the Evaluation Model**

After creating the Evaluation Model and our interpretation of Ericsson’s criteria in relation to the Evaluation Model these were compared in order to investigate whether the Evaluation Model is applicable for this case study.

Due to the accordance between Ericsson’s criteria and the Evaluation Model, the Evaluation Model is used for analyzing the LMFs. Hence, we can answer RQ5 (Is our evaluation model feasible for the particular case study?). This analysis is however complemented with a shorter extension which is specific for the particular case study, in order to answer RQ5 (Which license management frameworks for Grid environments seem to be the most feasible for the particular case study?).

**Present the Study Object**

The information used in order to describe the LMFs were gathered from previous research. This included reading the framework article, its references, and its website, and this is more thoroughly described in section 2.7.3 Relevant literature.

The only reason not to present a framework was if it was not designed for Grid environments, since the Grid network structure is a feasible choice in the telecom industry. (Berde, et al., 2009; Robertazzi, 2007; Future Generation Grids, 2004)
When describing the LMFs the reader is first given an overview of the different frameworks and then more details regarding the different criteria. These criteria were grouped as in the Evaluation Model and a description was made for each criterion. The reason for this structure is that it gives the reader enough information to understand the LMF analysis later on. When this step was done RQ3 (Which license management frameworks for Grid environment are mentioned in previous research?) was answered.

2.4.3. The analysis

The analysis can be divided into three steps which were performed sequentially and these implied discussing the LMFs using the Evaluation Model, comparing the LMFs with each other, and thereafter analyze the applicability of the LMFs for the particular case study.

Analyze the Study Object using the Evaluation Model

The LMFs were analyzed individually using the Evaluation Model. This implied structuring the analysis after the categories and criteria of the Evaluation Model.

The analysis induced evaluating the Study Object by discussing the LMFs’ characteristic regarding the different criteria of the Evaluation Model. In some cases it was possible to determine that an LMF satisfied a particular criterion or sub-criterion. It is however difficult to say that an LMF does not satisfy a criterion (unless it is explicitly written) due to their flexibility.

This analysis enabled the LMF comparison, which implied answering RQ4 (To which extent do license management frameworks for Grid environment, presented in previous research, match the aspects of our proposed framework?) and by extending this comparison for the particular case study also answer RQ5 (Which license management frameworks for Grid environments seem to be the most feasible for the particular case study?).

Compare the Study Objects

After analyzing the LMFs individually, as described in the previous section they were compared with each other.

This comparison was visualized in a table where the performance of the LMFs was commented for each criterion and sub-criterion. This comparison gave a good overview of the different LMFs. By comparing whether the aspects of the Evaluation Model were covered by the different LMFs RQ4 (To which extent do license management frameworks for Grid environment, presented in previous research, match the aspects of our proposed framework?) and by extending this comparison for the particular case study also answer RQ5 (Which license management frameworks for Grid environments seem to be the most feasible for the particular case study?). This gave a hint of the general feasibility of the Evaluation Model.

Perform the Ericsson analysis

Since the Evaluation Model and Ericsson’s criteria cohered the LMF comparison is valid for the case study as well. However, one sub-criterion was missing in the Evaluation Model which was present among Ericsson’s criteria. Furthermore, one criterion was present in the Evaluation which could not be found in among Ericsson’s criteria. Both these aspects were discussed together with the result of the
Study Object comparison, and this led to the answer of RQ5 (Which license management frameworks for Grid environments seem to be the most feasible for the particular case study?)

2.5. Approach criticism

In this section the thoughts behind the approach is discussed, and this is done for each of the three phases: initial study, establish the Evaluation Model and the Study Object, and the analysis.

The initial phase was necessary in order to understand the situation at Ericsson, get an overview of what has been written in the field of licensing and thereby determine the scope of the thesis.

The phase, establish the Evaluation Model and the Study Object, was designed to be straightforward regarding satisfying RQ1 (Which aspects have previous authors argued for regarding systems in grid environments?), RQ2 (Which criteria do five licensing experts at Ericsson consider important when evaluating a license management system?), and RQ3 (Which license management frameworks for Grid environment are mentioned in previous research?), where the Evaluation Model was established, Ericsson’s criteria were gathered, and the Study Object was created. The Evaluation Model was also validated by licensing experts at Ericsson during this phase.

The Study Object was created during this phase in order to be able to answer RQ4 (To which extent do license management frameworks for Grid environment, presented in previous research, match the aspects of our proposed framework?) and RQ5 (Which license management frameworks for Grid environments seem to be the most feasible for the particular case study?).

The analysis implied first analyzing the LMFs separately before comparing them. This comparison implied answering RQ4 (To which extent do license management frameworks for Grid environment, presented in previous research, match the aspects of our proposed framework?) and by some further Ericsson analysis answer RQ5 (Which license management frameworks for Grid environments seem to be the most feasible for the particular case study?).

The approach has been designed to answer the research questions and fulfill the purpose, in a structured and straightforward manner.

Something that could have been interesting to do differently is that the comparison between the Evaluation Model and Ericsson’s criteria could have been on sub-criteria level and not only on criteria level. These sub-criteria were hard to gather for Ericsson because they tended to be too implementation specific and were hence not feasible on an LMF level. Since the LMF comparison is clearly visualized an Ericsson reader can easily neglect criteria in case they would disagree, which is unlikely considering that the Evaluation Model is supposed to be general.

Klein & Myers (1999) are in their second principle, the Principle of contextualization, suggesting a critical reflection regarding the contextualization of the research.

When creating the Evaluation Model, criteria were gathered regarding license management systems in Grid environments, Grid environment systems, systems but also regarding Business Agility.
The contextualization of using criteria for Grid environment systems was made in order to get a wider perspective. It is important to be aware of that these are not specifically for licensing but since they are supposed to be valid for systems in Grid environment they should hold for license management systems in Grid environments as well.

The main reason for using Business Agility as an approach for developing the Evaluation Model is that several authors claim that the concept is important for success (e.g., Goldman, et al., 1991) but also because Ericsson, through Ericsson Strategic Direction (2013), opted the importance of being business agile. There is also a pervasive amount of research papers that link the concept of Business Agility to Grid computing and Information Systems. The contextualization was hence made that Business Agility was important for licensing and aspects for Business Agility could therefore be used as criteria for the Evaluation Model. Focusing on Business Agility might imply missing important licensing aspects but since the literature found, that included criteria for Grid environment, also linked to Business Agility, no criterion was discarded for not having a connection to Business Agility.

2.6. Scientific approach
In this section the scientific approach of this thesis is presented. First, the relation between theory and empiricism is discussed, and therefore the approach regarding deductive, inductive, and abductive reasoning is presented.

This section also includes discussions regarding whether hermeneutics or positivism reflects the perception of knowledge.

Furthermore, it implies discussions about whether the approach of this thesis is qualitative or quantitative.

2.6.1. Deductive, inductive, and abductive reasoning
Deductive reasoning implies that conclusions are reached regarding a case, by the use of generalized principles and established theory. It is common to come up with hypotheses using theory that later on are tested empirically. Due to the use of theory this approach is supposed to be objective, but the limitations of previous research might prevent the scientist from discoveries. (Patel & Davidson, 2007)

An inductive approach implies that the research can be performed without the use of established theory. The discovery, however, implies formulating a theory, but a negative aspect with this approach is that it is hard to determine the generalizability of the research. (Patel & Davidson, 2007)

Abductive reasoning is a combination of deductive and inductive reasoning where a theory is formulated from an observation, and this implies that hypothesis is created. This hypothesis is tested for more cases and it might be modified in order to be more general. Then a theory is established. The advantage with this approach, according to Patel & Davidson (2007), is that the scientist is less limited compared to the previous two approaches. On the other hand, scientists using this approach are, according to Patel & Davidson (2007), affected by their experience and previous research, and make choices that exclude other interpretations. (Patel & Davidson, 2007)
The approach chosen in this thesis is mainly deductive since the Evaluation Model relies on generalized principles and established theories. Furthermore, the criteria of the Evaluation Model can be regarded as a hypothesis where the assumption is that they are suitable for the particular case study. This is however tested empirically by comparing the criteria of Ericsson’s criteria.

Due to our experience regarding the field of licensing it would have been difficult to present a reliable evaluation model using an inductive approach. Using an abductive approach, where the Evaluation Model would have been tested empirically, would have required more than one case study for this thesis (it would not have been sufficient to perform this thesis only at Ericsson), due to the nature of abductive reasoning.

2.6.2. Hermeneutics or positivism

Patel & Davidson (2007) describe positivism as an objective scientific approach whilst hermeneutics implies interpretations.

The scientific approach of positivism is according to Patel & Davidson (2007) performed as the deductive approach where a hypothesis based on theory is empirically tested. One characteristic for positivism is the perception of reductionism which implies that a problem can be broken down to different parts which are studied separately. (Patel & Davidson, 2007)

In hermeneutics the scientists are examining the study object using their thoughts, impressions, feelings, and knowledge, and this regarded as an asset and not a hinder in order to interpret and understand the study object. (Patel & Davidson, 2007)

The main approach of this thesis is positivism since it has been chosen since it uses a deductive approach. Furthermore the perception of reductionism has been used in order to analyze the LMFs; this implies dividing the Evaluation Model into criteria which in turn often are broken down to several sub-criteria. The criteria are grouped into categories; if all criteria within a category are satisfied then that category is satisfied.

Hermeneutics, which we think seems more useful for an inductive approach, have not been used in the same extent. However, when analyzing the Study Object our interpretations affected the analysis which implies a hermeneutic approach, but in order to increase the objectivity we have both critically examined and approved everything that has been written; it did not imply any major changes since our opinions have been concurrent. According to Klein & Myers (1999) first principle, The Fundamental Principle of Hermeneutic Circle, A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems:

“human understanding is achieved by iterating between considering the independent meaning of parts and the whole as they form”.

This is how the initial phase was performed and the scope of this thesis determined, and the approach of this thesis was therefore partly hermeneutical.
2.6.3. A qualitative or quantitative approach

In this thesis a qualitative approach has been chosen even though a quantitative approach could have been used in which the LMFs are given points that corresponds to their performance regarding the different criteria. Thereafter the points could have been weighted and summarized, and the LMF with the highest score would have been the preferable one. This would have been an interesting approach but there are some obstructions in order to get a reliable result from it. First of all, the frameworks presented are general and can be implemented in different ways and this makes it hard to give them a fair grade since the performance of the actual implementation is not revealed. Furthermore, just because no solution is presented for a certain criterion (for an LMF) it does not mean that it cannot be supported. Thirdly, it might be hard to compare different ways of handling a criterion which makes it hard to give the LMFs a correct and objective grade. Not only is the grading part complicated but also to weight the scores of each criterion before summing them. No literature has been found that weights the importance of the different criteria. One approach would be to let a lot of experts determine the weights and take the average of their results, this would have implied that the result of the Study Object analyze would have been objective but it would be hard to assure the academic result.

Due to the limitations with the quantitative approach a qualitative has been chosen, for analyzing the Study Object, in which logical interpretations have been performed. However, the interpretations that have been made are clearly motivated and shall be backed up with facts.

2.7. The literature study

There were four purposes with the literature study, where the first was to get a basic knowledge in relevant fields. Secondly, it was used to find criteria in order to evaluate the different LMFs. Thirdly; LMFs were gathered in order to have different alternatives for license handling. The last purpose was to improve the structure of the report and improve the interview technique and this was done by reading research methodology literature.

2.7.1. Literature search

The literature used in this thesis was found on internet using the following search engines: LiU UniSearch, Springer, Google, Google Scholar and IEEExplore. Almost all of those search engines were used for every particular search word.

The challenge of this thesis was that Ericsson were not satisfied with their way of handling their licenses and in order to improve the knowledge in that field the following two search words were initially used: software license and software license control.

After that more literature was found in two different ways where the first was to use the references of the literature already found. Secondly, keywords or important concepts were used to expand the search. This resulted in searches regarding different aspects of IT management and important technical concepts.

The following searches was performed regarding IT management: IT management, IT asset management, software asset management, license audit, product management, asset management,
technology management, IT governance, IT audit, software architecture, IT architecture, Business Agility, Business Agility architecture, access control, semantic modeling, and design theory.

The search for relevant technical concepts included the following search words: SaaS, Cloud, Grid, Grid computing, Grid environment, distributed environment, distributed computing, virtual servers, virtualization, and virtual machine.

### 2.7.2. Literature categorization

When searching for literature the list of search results were regarded for each search engine. A first selection was performed which implied first reading the title and if it seemed to be irrelevant for this thesis the literature was discarded. The abstract was read (for the literature passing “the title test”) and the article could once again be discarded. The articles passing “the title test” and “the abstract test” included at least one of following bullets:

- Clarification of challenges and benefits with licensing.
- Explanation of ways of managing software licensing.
- Criteria for managing software licenses.
- Clarification of the relation between Business Agility and IT
- Explanation of important concepts regarding Business Agility
- Definitions of terminology that is relevant for previous bullets.

The articles passing “the title test” and “the abstract test” were summarized in a reference document. The summary contained the title, author, search engine, search word, and a short explanation of its usefulness.

After reading the literature it was classified as follows:

- Relevant: The literature contained LMFs or criteria for LMSs to be used.
- Relevant facts: The literature included definitions of relevant terminology and clarification of challenges and benefits with licensing.
- Potentially relevant: The literature consisted of potentially useful terminology definitions, facts that might be useful, or presentation of LMFs that might be used.
- Irrelevant: The literature does not include any important LMFs, criteria for software licensing, or important facts.

Literature that had not been read was classified as not read, but categorization changed when the literature was read. Furthermore, literature could be reread and hence reclassified, but literature that was classified as relevant, relevant facts or irrelevant did generally not change. After limiting the thesis not to include pricing models, literature in that field went from potentially relevant to irrelevant. No irrelevant literature was reclassified and the reason for that was that if there was any chance that the literature later on would be useful it was classified as potentially relevant. Potentially relevant literature could be reread and basically be reclassified to any of the three categories depending on its usefulness.
2.7.3. Relevant literature

The literature classified as relevant consisted of data for theoretical criteria and empirical data for the LMFs and these are about to be explained. This data was as explained earlier gathered in the same way as any other literature but were important for achieving the purpose of this thesis and will therefore be explained.

Data collection for the Evaluation Model

The data collection for the Evaluation Model implied gathering literature from previous research and this was done in the same manner as the rest of the literature. The criteria found for systems in Grid environments were related to Business Agility, and hence no criterion was discarded for not being connected to Business Agility. The only reason for discarding criteria were not within the scope of an LMF (for instance too implementation specific). Some of the criteria found were directly adapted for license management systems in Grid environments whilst others were for Grid environment systems in general they were all used in the Evaluation Model. The authors of these articles were coherent regarding the criteria, but the sub-criteria could in some cases differ. They were, however, not contradictory and they seemed plausible for evaluating the Study Object and were hence included in the Evaluation Model.

Data collection for the Study Object

The empirical LMF collection was found in the same manner as the rest of the literature and did not imply any specific search words. When an article included a presentation of an LMF its references was used in case it would give more information about the LMF. Google was also used to search for the websites of the LMFs. In case they were available they were searched through for complementing information. In order to understand how this LMF information was used one must read section 2.4.2. Establish the Evaluation Model and the Study Object.

2.7.4. Literature handling

Before reading the literature summarized in the reference list it had to be gathered, which could be done in three ways: download it online, physically borrow it from the library or purchase it.

The majority of the literature was downloadable and these files were stored in folders named according to the different classifications in section 2.7.2. Literature categorization. Downloaded files that were about to be read were also printed and put into binders that were structured in the same way.

The literature that was not downloaded could not be stored in this way and that literature had the same categorization written on Post-it notes that were attached to it. Furthermore, Post-it notes could also be used to identify important sections of the literature.

After reading literature the reference list was updated which implied modifying the comments and in some cases the classification. In case downloaded literature was reclassified the printed copy had to be moved to the correct binder section, and the file to the correct folder.
In the beginning of the thesis the literature was used to get familiar with the research regarding software licensing, whilst it later became more important to gather LMFs, criteria for software license handling and understanding relevant terminology. The reference document turned out to be an efficient way to find the information that was currently needed.

### 2.7.5. Source criticism

There are some limitations regarding the literature study. Six search engines have been used in order to find literature, and using more search engines might have led to finding of more relevant literature. Due to the limited time, and considering that a lot of relevant literature was found in this way, no more search engines were used. Saturation was, however, reached for those search engines, in the literature study since the literature towards the end referenced to articles that had already been regarded.

Secondly, in the process of choosing literature articles and books were discarded after reading their title and abstract. Some of the discarded might have contained interesting information, but it is unlikely that any of them would contain something as important as an LMF considered that the LMFs found are the main contribution of its articles. In a perfect world it would have been interesting to read all this literature, but this was not possible within the scope of this thesis.

One important issue is if important search words are missing which is difficult to say. However basic search words like software license and software license control were used and the literature found gave new important search words, which also led to new search words and so forth.

The original source was used as long as it was available in either English or Swedish.

### 2.8. The case study

There were three different purposes with the case study which were: get the licensing problem background of Ericsson, gathering Ericsson’s criteria, and gather information regarding Ericsson’s LMS. The techniques used for all this are described in the subsequent section 2.8.1. Case study gathering and thereafter more specified for each of the three purposes, and thereafter the case study criticism is presented.

#### 2.8.1. Case study data gathering

The different techniques for gathering the data for the case study were basically interviews and Ericsson’s documentation and this is presented in this section.

A part of the interviews were carried out as presentations held by the interviewee, during which questions could be asked at any time. The presentations were held on a certain topic that the interviewee or the audience thought to be interesting for this research. The other interviews were carried out in a semi-structured manner, where questions prepared beforehand were the basis for discussion with the interview subjects, and can for each session be found in appendix 2. Notes were taken during the interview and the conversation was recorded if the interviewee allowed. Some interviews were performed face to face with the interviewee, whilst others have been performed using the software Microsoft Lync. The face to face recordings only included the audio while the recording on Microsoft Lync also included a video of the interviewee’s computer screen. The purpose of the two
interview forms is mainly gathering information that is needed for the essay, and in some cases consulting a person of special expertise to get answers on various topics.

Documents received from Ericsson were also used in this thesis. This included both slides that the employees used during lectures but also internal documentation from Ericsson’s Intranet.

2.8.2. Understanding Ericsson’s problem background
In the Ericsson data collection during the initial phase the information was gathered through interviews carried out as presentations and PowerPoint slides that belonged to the different presentations.

2.8.3. Collection of Ericsson’s criteria
The collection of Ericsson’s criteria implied reading internal documents and performing interviews with employees of Ericsson. These interviews were performed as presentations where the interviewee explained their criteria for an LMS, but also the criteria for Ericsson’s LMS.

When collecting these requirements triangulation was used which, according to Malterud (2009), means that phenomenon are studied using several observations. This makes it, according to Patton (2002), possible to exclude inaccuracy in the process.

This meant that several employees were interviewed in order to gather Ericsson’s criteria. Using triangulation was one way to discover contrarious criteria, but that did not occur, which increases the relevance of the criteria.

In case more than one person suggest the same criteria its importance is emphasized, but in order to get a wide perspective no criterion was discarded.

One challenge when gathering the criteria was not to get too implementation specific criteria but more general that can be used in order to evaluate LMFs. This was not done by explaining that the criteria had to work for LMFs but to emphasize that more general criteria were desirable.

2.8.4. The interviewees
The interviewees were employees at Ericsson but with different tasks and expertise. Some had a broad and overall perspective, whilst others had more specific and detailed knowledge. Some of the interviewees were directly working with the LMS where some were responsible for the licensing strategy for product areas or for whole business units and others were experts for parts of the LMS. Furthermore employees with a more indirect relation to the LMS, and this included expertise within product packaging or the pricing strategies were also interviewed.

The main interviewee of this thesis was the Process Architect ELIS & BNET and Strategic Supply Manager BNET. He has worked with licensing as a requirement analyst, and after observing the drawbacks with the current License Management System he took the initiative to let Research Interns investigate if there were better solutions for the license management of Ericsson.

The Director of Software and Product Packaging within the company division BNET (hereby denoted Director of Software and Product Packaging) focuses on strategic portfolio management, and is currently
integrating all Ericsson’s pricing models with their services for the business unit BNET. The Process Architect ELIS & BNET (2013) and Strategic Supply Manager BNET (2013) emphasizes that the requirements of BNET regarding licensing reflect the requirements of Ericsson in general (since it covers the requirements of the other Business Units), but that there are no explicit requirements for the whole company. The licensing aspects are important for the Director of Software and Product Packaging and due to his position in BNET he has a good overview regarding them.

The Product Manager Licensing Management within the company division BNET works with licensing on a strategic level and is responsible for Ericsson License Register (ELIS).

The Price Manager within the company division BNET has a commercial responsibility and determines the price models of BNET, which requires knowledge regarding the licensing at Ericsson.

Since we had no contacts on Ericsson before starting this thesis, the Process Architect ELIS & BNET has been important for finding interviewees. He had contacts in different positions and with different experience regarding license management. Depending on which information that was needed different contacts were recommended and these were asked for further interviewee recommendations.

2.8.5. Case study criticism

The case study has basically implied performing interviews and gathers lecture material and internal documents from Ericsson. The reliability of the internal documents is considered high since errors or ambiguity can be rectified since they, according to Process Architect ELIS & BNET (2013) are continually edited and updated in order to be relevant, and errors and ambiguity can therefore be rectified. The reliability of the lecture material and the interviews will however be discussed, and those will be discussed in the same manner since they both only reveal one person’s interpretation.

The Principle of Multiple Interpretations (Principle six) implies sensitivity to different interpretations among participants of the case study (Klein & Myers, 1999).

The interpretation for Ericsson’s criteria has differed between the sources, which include internal documents and the interviews. The difference gave a wider perspective and depended probably on different knowledge and experience among the participants. Since none of the criteria were contradictory they were all accepted.

The attitude towards the current LMS at Ericsson has varied among the participants, when some have expressed that even though problems have been faced, they can be overcome with the current solution. Others have pointed out its weaknesses and think that the current solution is not feasible for Ericsson.

This can be further discussed using the Principle of Suspicion (Principle seven) which implies sensitivity to biases and distortions in the data collected from participants of a case study (Klein & Myers, 1999).

This different interpretation among the participants does not necessarily depend on biases or distortions. The employees have different knowledge and experience regarding Ericsson’s LMS which therefore can affect their opinion in this matter.
No matter if the current solution is feasible or not, there might still be LMFs that are better and hence can be introduced.

Klein & Myers (1999) suggested the Principle of Interaction between the Researchers and the Subject (Principle three), which implies a reflection on how data was gathered through the interviews.

In order to affect the interviewee in as small extension as possible the questions asked were neutrally asked not to lead the interviewee in a certain direction. Furthermore, due to the semi-structure of the interviews, the interviewees could affect the direction of the discussion due to his or their answers. Something that might have affected the interviewee was that some of them saw our criteria before telling theirs. The reason was to make sure that they understood the level of the criteria and avoid getting all too implementation specific criteria. A risk when making these interviews is that the answers of the interviewee might depend on what they think that our employer wants to hear. Since our employer wants a problem solved there is no reason to hold back criticism, and considering the fact that the opinions of the interviewees regarding the Study Object criteria were consistent it is likely that the interviewees have been honest.

In their fifth principle, the Principle of Dialogical Reasoning, Klein & Myers (1999) discuss the sensitivity for misunderstandings. The techniques for avoiding misunderstandings with the interviewees, was to ask follow up questions in order to verify that the information was correctly understood. Furthermore, the interviews were recorded which was used to reduce the misconception. Another way to avoid misunderstanding was for us to discuss the content with each other and in case of different perception the recording was used.

Since the company used in the case study is a large company with approximately 110,000 employees, a limitation of the respondents used for collecting the empirical data had to be used. For most of the cases only people in a management or certain expertise positions was questioned. People in management positions were considered as the individuals with the broadest, but not certainly the deepest, expertise in their fields. People of certain expertise were contacted for consulting on a variety of topics. Due to the relatively small sample of interviews (in comparison the total amount of employees), there is always a risk of the material collected being biased.

The interview material was used regardless of the number of interviews emphasizing a certain aspect. Since the interviews had different expertise this helped to give us a wide perspective. The interviewees were concurrent but they could bring up different aspects as important for license management; however, none of the criteria or aspects was contradictitious which strengthens our perception of the criteria being relevant.

Because of the freedom of analyzing the LMF with regard to our theoretical criteria, there is a possibility that if anyone else did the same research, they could come to another conclusion than we did.
**Generalizability**

The fourth principle (The Principle of Abstraction and Generalization) of Klein & Myers (1999) concerns generalizability is discussed in this section.

Lee & Baskerville (2003) discusses, in their article Generalizing Generalizability in Information Systems Research from 2003, different types of generalization in their generalizability framework, which is shown in figure 2-3.

The author classifies the different types of generalizability using as follows (Lee & Baskerville, 2003):

- **EE**: Generalizing from data to description (from empirical statements to empirical statements).
- **ET**: Generalizing from description to theory (from empirical statements to theoretical statements).
- **TE**: Generalizing from theory to description (from theoretical statements to empirical statements).
- **TT**: Generalizing from concepts to theory (from theoretical statements to theoretical statements).

![Figure 2-2: Four types of Generalizing and Generalizability (Lee & Baskerville, 2003)](image-url)
When establishing the Evaluation Model theory from different sources was gathered, this is a case of TT. It is an assumption that the criteria together characterize what is important for an LMF. The criteria where all regarding systems in Grid environments and the literature which they were gathered from were all connected to Business Agility. Since these criteria were gathered from different observations with different empirical settings they were supposed to complete each other. A first step in order for the Evaluation Model to be valid is that it cannot contain contradictious criteria or sub-criteria, but its generalizability can, however, not be guaranteed without being tested for the empirical settings in which it shall be used.

One generalization that has been made in this thesis is a TE where the Evaluation Model has been used to evaluate the LMFs individually. The criteria of the Evaluation have been determined through other studies with other settings than the one in this thesis. The thought was that they would be relevant for the particular case study as well, since they are supposed to be generic. The only way the Evaluation Model can be claimed to be generalizable and hold for the particular case study is to test it in the new setting (Lee & Baskerville, 2003).

In order to increase the validity of the Evaluation Model is by letting licensing experts from Ericsson, who works with licensing in a real business context, determine what is important for License management. This implies using empiricism in order to strengthen a theory which is ET. By validating the Evaluation Model using Ericsson’s criteria can give a hint of the usefulness of the Evaluation Model for this case study. In order to do this comparison Ericsson’s criteria have been structured in the same manner as the Evaluation Model. This implied discarding too implementation specific criteria but also interpret Ericsson’s criteria in order to either cover the criteria of the Evaluation Model or come up with new criteria that do not exist in the Evaluation Model. They correspond to each other to a great extent, which strengthens the perception that the Evaluation Model is valid for the particular case study. It is hard to definitely say that the Evaluation Model is feasible in other contexts but it is, however, more likely that it does since it holds for the case study.

When gathering Ericsson’s criteria licensing experts are chosen to represent the whole company which is a case of EE. The generalization of letting a few people represent a whole company was sufficient for this study due to their expertise. We also got access to Ericsson’s internal documentation (Focal Point Report), which contained requirements of Ericsson’s license management system, which strengthened the validity of the statements made by the interviewees.

Another EE generalization that shall be discussed is whether the fact that the Ericsson empiricism strengthens the validity of the Evaluation Model in other business contexts then the one for this case study. This is hard to determine due to different prerequisites, but if it holds in one business context it is not unlikely that it holds in another. This validation strengthens the usefulness of the Evaluation Model but it can as previously mentioned not be determined.

It is interesting to discuss whether the LMF comparison in this thesis is generalizable in other contexts than for the particular case study. The criteria of the Evaluation Model are supposed to be generic and
work for different settings but their validity should be investigated for the context in which they are used.

The comparison of the LMFs using the Evaluation should be interesting for other cases than Ericsson, since the Evaluation Model and the LMFs are general. It is difficult to determine which LMF that is the most suitable for the case study due to the difficulty of weighting the criteria and the flexibility of the LMFs which makes it hard to say that a certain criterion cannot be supported. The comparison should however be useful both for the particular case, but also for other companies that are interested in license handling. The evaluation and comparison and evaluation of LMFs can therefore be regarded as a support for a decision about how they should handle their licenses. How the LMF can be implemented is, as previously mentioned, not within the scope of this thesis.
3. Theoretical framework

In order to answer the research questions of this study, a thorough walk-through of related research is required. This chapter provides a summary of a selection of previous research conducted in the fields of Business Agility, agile Information Systems, and Grid computing. The purpose of this chapter is to define what criteria that are going to be used for analyzing the different LMFs and apply to our case study. The criteria are categorized in the terms of: technology, tract, trade, and transparent. This chapter starts by giving a brief introduction to the chapter and after that discussing and defining important concepts that will be used in this study. After that, literature surrounding criteria for agile Information Systems and Grid computing are expressed. Finally, our evaluation model is presented.

3.1. Introduction

“Business Agility is the ability to thrive in a competitive environment of continuous and unanticipated change and to respond quickly to rapidly changing, fragmenting global markets that are served by networked competitors with routine access to a worldwide production system and are driven by demand for high-quality, high performance, low-cost, customer-configured products and services.” (Goldman, et al., 1991)

“As the industry moves quicker, the key to success is agility. Operators need to be fast, flexible and in control to deliver great experiences to their customers, operate efficiently and offer innovative new offerings.” (Ericsson’s Strategic Direction, 2013)

The foundation of this theoretical framework will be based upon the theory and concept of Business Agility. The background and main reason for using Business Agility as an approach for developing the Evaluation Model is that several authors claim that the concept is important for success (e.g., Goldman, et al., 1991; Sambamurthy, et al., 2003; van Oosterhout, 2010). Furthermore Ericsson, through Ericsson Strategic Direction (2013), opted the importance of being business agile. There are also a pervasive amount of research papers that link the concept of Business Agility to Grid computing and Information Systems. We think that Business Agility is an important concept for organizations to adapt and boost, due to the market-oriented nature of urging to cater the customers’ demand. However, depending on the case, some aspects in the theoretical framework might be weighed differently.

The next section will start by giving an introduction to the concept of Business Agility and its relation to Information Technology, and then present articles with relation to IT-architecture and commercial aspects that augment Business Agility. Each of these topics in the text will be connected to the topic of licensing and license management and will therefore ensure agile systems feasibility in a licensing context. The research presented will be sufficient for developing an evaluation model, based upon theory with association to Business Agility, fit for the purpose of this thesis and fulfill the goal of this chapter.
3.2. Business Agility

3.2.1. Concept introduction
At present, companies have to survive and develop competitive advantage in a dynamic and turbulent environment of global competition and rapid business change (Sambamurthy, et al., 2003; van Oosterhout, 2010); hence augmenting the urge to proactively predict the changes before they impact business operations (Thao, et al., 2012). Surviving in an ever changing environment has since the last decades been crucial, owing to that the lifespan of products has dramatically declined since the beginning of the 1990s (Melarkode, et al., 2004). A company’s ability to deal with this issue is denoted Business Agility and has been comprehensively summarized by van Oosterhout (2010) as:

“ [...] the ability of an organization to swiftly change businesses and business processes beyond the normal level of flexibility to effectively manage highly uncertain and unexpected but potentially consequential internal and external events, based on the capabilities to sense, respond and learn.”

In high competitive industries such as electronics, financial services, pharmaceuticals, and telecommunications: the need to be business agile is critical for continued success, due to many environmental changes and uncertain events. (Fine, 1998; Sambamurthy, et al., 2000) For an actor, the degree of environmental change is a combination of both the composure of internal conditions (changes within the actor’s organization) and changes in the external business environment (van Oosterhout, 2010).

3.2.2. The impact of Information Technology on Business Agility

“Information Technology plays a key role in transforming the entire enterprise in order to make it more agile, flexible and more capable of adapting to a constantly changing market environment.” (Capgemini, 2002)

On the topic of Information Technology contribution to Business Agility, van Oosterhout (2010) claims that IT is a major force driving the need for Business Agility and at the same time an important capability, which can hinder or enable a firm’s level of Business Agility. However, Boden (2004) claims that organizations keep Information Technology budgets depressed as they try to exploit their existing systems, forcing out the remaining capabilities of the system to bolster their financial cases. This creates a problem according to Lukas Jr and Olson (1994) due to, at some point, the law of diminishing returns for Information Technology (IT-flexibility paradox) will cut in as IT-assets gets older and the needs of Business Agility stretch beyond the capabilities of the current systems (Lukas Jr & Olson, 1994). As a result, companies will start to look for contemporary technology that will help them both upgrade their capacity and augment their Business Agility (Sambamurthy, et al., 2003). For a good alignment between IT and Business Agility, the system must be as agile as its environment forces it to be (Dove, 2005); thus organizations that operate in a dynamic environment require agility more strongly than organizations that operate in a less fierce business environment (Moitra & Ganesh, 2005).
“[IT infrastructure is] the system of hardware, software, facilities and service components that support the delivery of business systems and IT-enabled processes.” (Gartner, 2013)

Information Technology infrastructure has been identified in some businesses as having a critical impact on the enterprise’s ability to use IT competitively (Duncan, 1995). Despite of efforts to increase flexibility of corporate IT, most successful corporate IT still involves processes and applications that are unable to change rapidly. As a result of the slow change process, chances in the IT-infrastructure results in long lead times; hence affecting the support for new business processes and product concepts. (Stanojevska-Slabeva et al., 2010) Stanojevska-Slabeva et al. (2010), therefore claims that an agile company is only possible with an agile IT-infrastructure that can swiftly and efficiently be adjusted to new business ideas.

Thus, building a system that enhances Business Agility, could possibility help the company leverage the benefits of flexibility, renewal, and avoiding long lead times. The upcoming subchapter will further discuss concrete IT-architectural properties that augment Business Agility. Subjects as computer security in a grid environment, which is necessary in a business context, and literature that connects several of the mentioned aspects to the topic of licensing are expressed.

3.2.3. IT-architecture

“IT-architecture is a formal description of a system, or a detailed plan of the system at component level to guide its implementation. It is the structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time.” (The Open Group, 2013)

In the article “The improvisation-efficiency paradox in inter-firm electronic networks: governance and architecture considerations”, Konsynski & Tiwana (2004) claims that certain properties in an IT-architecture enhances Business Agility, whilst some are focused on delivering efficiency and often trivialize its capacity to reactively adapt to changing market conditions. The authors therefore developed the key idea that a network and IT-architecture can be improvisational and, yet, efficient if it achieves complementarities in its architecture, even though it is an intricate but coherently integrated blend of many different entities. To achieve an effective market practice includes attention to for example loose coupling and relevant modularization, increased visibility, and a self-organizing architecture. (Konsynski & Tiwana, 2004)

With loose coupling, Konsynski & Tiwana (2004) claims that it enhances the flexibility of the components in the implementation but requires modularization. Modularity refers to the attribute of a system to shift interdependence between components to the interfaces between those components. The key principle of modularity is that all interdependence must be moved to the inter-entity interfaces and complete independence must simultaneously be maintained within each entity (which gives a plug-and-play ability of the components). These two concepts of loose coupling and modularization augment flexibility in two ways (Konsynski & Tiwana, 2004):

- **Makes the entities semiautonomous:** by moving all coordination dependencies to the interface; consequently standard definitions of the interfaces are thus required to make the different
resources self-describing, which allows them to be mixed and matched with new or old modules.

- **Shifts focus away from hierarchies of functional structure:** to thinking in terms of organizing around processes; thus augment the systems role of being more adapted for its use.

The loose coupling and modularity concept brings robustness to the architecture and facilitate easy modifications and scalability of the system. (Konsynski & Tiwana, 2004)

With visibility, Konsynski & Tiwana (2004) means that the information visibility in the system is enhancing coordination and the trust among the partners involved in the system.

By self-organizing capabilities, Konsynski & Tiwana (2004) claims that the capability of self-organizing is embodied in two features of the IT architecture: shared meaning and swarm intelligence. Shared meaning refers to the agreed upon “vocabulary” to describe specialized concepts such as Web Services or other standards and open protocols. These standards and open protocols play a key role in the architecting modularity. With swarm intelligence, Konsynski & Tiwana (2004) means a decision making module shall be based on simple rules and, yet, at the same time support improvisation. Forming of a proper community of decision makers and decision influencers will have a positive effect on the computational collaborative environment. This is due to that the improvisational network has the ability to bring together those that have information and opinions and those that have decision rights and authorities when needed in order to make an optimal decision. Another alternative to swarm intelligence is to allot the authority to entities that have the best expertise for making the associated decisions. (Konsynski & Tiwana, 2004)

Boden (2004) suggests in his article “The grid enterprise — structuring the agile business of the future”, several architectural properties of a Grid system in order to make the organization using the system more agile. In the article Boden (2004) stresses several technical properties such as Web Services and Grid computing.

With Web Services, Boden (2004) claims that they are based upon open standards and as a result a broad set of tools have been developed to make their implementation relatively simple. It is certainly far easier to construct new applications by connecting together a set of existing services than it is to build these applications from no solid ground. A Web Service takes some specific business functionality and packages it so that any system can make use of its abilities of a common set of standards and also abilities to connect to various secure platforms. These standards are widely accessible and simple to implement, and are used in many widespread software packages. Web Services provides agility in integration, enabling the organization to be faster, and cheaper to market. Furthermore, they allow some aspects of integration to be performed more cheaply and more effectively than using other standards. In technical terms, a Web Service (Boden, 2004):

- Is a loosely coupled and reusable component.
- Encapsulates specific business functionality.
- Is accessible via stable interfaces defined by Internet standards.
Moreover, it brings agility to a computerized system and its utilizing organization by (Boden, 2004):

- Linking the process layer to the underlying systems in a decoupled manner.
- Simplifying integration across organizational boundaries, between application silos or partner enterprises.

In both these cases they encapsulate the complexity of integration by defining the interfaces required, not their implementation; hence enabling easier modification and extension of the existing software modules. (Boden, 2004)


According to Boden (2004), Grid computing augments Business Agility in the company’s infrastructure by allowing the pooling of resources across multiple systems and their allocation on demand, with spare capacity utilized to build transient services. It works by creating a resource access framework which applications can call for allocation of resources; hence bringing robustness and giving the flexibility necessary to expand and build new functionality into the system. This gives the same Quality of Service as a single large mainframe supercomputer at a lower price and performance ratio, but available for a larger group of users which may provide significant benefit to compute-intensive applications. When it comes to a Grid environment, there are two architectural options. First, the hub-and-spoke option opts that all requests are centralized by routing the network packages through a resilient central server, where the elicited controls are applied. Second, the local application server stresses the architectural idea of a managed proxy server linked to central management service for control purposes, but where the associated decisions are done at the point of invocation, which provides a more complete security solution. Boden (2004) claims that of these two options the most successful implementations have utilized the local application server alternative since it provide better control at the point of invocation. Furthermore, the Grid environment is inherently distributed, and thus, provides several key features, notably (Boden, 2004):

- Flexible resource allocation and management.
- Security (authorization, integrity, and accountability)
- Delivery of non-trivial Quality of Service (reliability and availability).
- Provision of information about resources and their availability.

In “Mechanisms for controlling access in the global grid environment”, Angelis, et al. (2004) expresses the two main concerns in grid environment security: the ability to assert and confirm identity of a participant in an operation or request (authentication); and determining whether a client should have access to an object or resource and to what extent this access should be allowed (authorization). These two enumerated aspects are denoted access control, and is the base security mechanism around which all other security aspects are developed. The tasks of the Authentication and Authorization processes are obligating the participants in a computing environment to obey a set of rules (policies), that define the security subjects (e.g., users), the security objects (e.g., resources), and relationships among them. Specifically, in a Grid computing environment, security policies could be described by a list of valid
entities (authentication), allowed some specific access rights (authorization) over the available resources in the Grid environment. (Angelis, et al., 2004)

Authentication, in a Grid environment, is more concretely the process of granting access to a resource or service by first verifying the identity of a participant in an operation and may be required for any entity in the network. Because of the particularity of the Grid environment, Angelis, et al. (2004) suggests several extensions to be comprised under the scope of the authentication processes in the Grid environment:

- **Delegation of identity**: Is the act of giving away the necessary authority so that another resource or service can assume the identity of another principle for completion of some task. For example, a user must be able to grant to a program the ability to run on the user’s behalf, so that the program is able to access the sources on which the user is authorized. This aspect is also mentioned in Humphrey & Thompson (2002).

- **Single sign-on**: While the computing package is traversing the Grid to locate the resource or service where the computation is going to be executed, the single sign-on aspect addresses the problem of the user signing-on to all diverse individually controlled domains. Users should be able to authenticate only once when accessing the Grid and then access resources that they are authorized to, throughout the whole environment without any further user intervention. This aspect is also mentioned in Humphrey & Thompson (2002).

- **Identity mapping**: Mapping to local credentials shall be implemented in order to achieve single-sign-on. Consequently, the user that has the credentials to use a resource or a service must have a local ID at the sites that need to be accessed, the mapping to be used in this context must be agreed upon between the site administrator and the Grid administrator. This aspect is also mentioned in Humphrey & Thompson (2002).

- **Mutual authentication**: Besides authenticating the user, a server must also do this process to assure that the data and resources received can be trusted and that the submitted data are sent where intended.

- **Certification**: Providing the binding between identity credentials and principals imposes the existence of independent certificate authorities that maintains the heterogeneity of the Grid environment. Certification duration and revocation are also issues that need to be resolved in the dynamic Grid user population.

Authorization is the process of determining whether a specific client may use or access a resource. Authorization is usually the step that follows successful authentication. In the Grid environment, the local access lists and security administrators, that describe the set of principles authored to access a particular object, must be kept separately in each local participating domain that is connected to the network. Specification of which subject that can access the resources and services, and under what certain conditions, must be able to be specified by the local administrators or resource/service owners.

In order to minimize exposure from compromised or misused delegated credentials, local administrators should have control of what controls should be also posed over delegated identities, since along with identity, the set of authorization rights are also delegated. When determining tasks involving an access control decision, there are usually local and global policies and rules to take into consideration.
Especially for the Grid environment, authorization information for all individual users at every site is not appropriate to administer. Instead, letting the administrators at every local site have direct administrative deals with the collaborations they participate in, but not generally with other sites is a more feasible practice for this type of computation environment. An authorization process for a Grid environment is established by imposing agreements among the different local sites rules and the Grid security policy, due to that all the local parties is in control of the resource access. (Angelis, et al., 2004) Furthermore, Angelis, et al. (2004) elicits the security architecture for authorization as the following steps:

- The service to gather additional information associated with the user or the actual session (group membership, role period of validity, etc.)
- The service to gather additional information associated with the protected resource. (e.g., file permissions)
- The checking of any local policy applicable to the request (e.g., a temporarily disabled user).
- The making of an authorization decision based on the identity of the user and the additional information.

Beside the authorization and authentication, Angelis, et al. (2004) also proposes other aspects that should be included in a well-functioning Grid computational system. Network security such as ports configuration of firewalls, and intrusion detection system are aspects that need consideration. In order to preserve the information confidentiality: integrity and trust, encryption, and digital signatures shall be a part of the computational solution. Another key-factor in the security architecture is an effective and cooperative security administration. Furthermore, effective practices to enhance capabilities such as intrusion detection, identification of malicious or accidental security violation attempts are augmented by effective security logging. All these procedural and systematic measures shall be integrated and surround authentication and authorization to form the complete security architecture for the Grid environment. (Angelis, et al., 2004) All these aspects are also mentioned in the articles of Humphrey & Thompson (2002) and Foster, et al. (1998).

In the article “Towards SLA-based Software Licenses and License Management in Grid Computing”, Li, et al. (2004) elicits the challenges and requirements of managing the software license in Grid environments and what a system must be able to handle. The requirements elicited by the authors of the article can be summarized into the following aspects (Li, et al., 2004):

- **Different administrative domains and Virtual Organizations**: In a Grid environment, license management involves several administrative domains and integration of virtual organizations. In these cases, their members need temporary access rights for specific software suites from different domains. This puts a requirement on obtaining temporal licenses needed for the computations and different usage policies for the different users. Usage of e.g., firewalls, Virtual Private Networks (VPNs), and remote usage control should therefore be considered when designing a system.
- **Transparent management**: The licensing process should be transparently managed as a part of the Grid job management.
- **License enforcement**: In a Grid environment the license enforcement is a crucial part for the operability of the network. This puts a stress on remote license enforcement, co-allocation of different resources, and support for workflows. Remote license enforcement is used when the jobs shall be able to be executed remotely, whilst the validity of the licenses has to be guaranteed at the same time. Co-allocation of different resources is when the licenses shall be co-scheduled together with the services and resources utilized. Support for workflows is when different applications may need to be used in different phases of the workflow; particularly this stresses the importance of how to retrieve and reserve the right licenses for the applications in advance.

- **Dynamicy**: The license shall be able to be preempted, suspended, and resumed in a Grid environment.

- **Interoperability**: Web Services or other accepted common standards shall be the foundation of the licenses and license management modules instead of proprietary solutions.

- **Support for virtualization**: Licenses and license management for environments with virtualized resources or based on multi core technology is required.

As the fundamentals of technical properties that help leverage the benefits of Business Agility and the first connection to licensing made, the following sub-chapter will further do a study regarding Grid environments and its business applicability and further strengthen the connection to the topic of licensing.

### 3.2.4. Commercial aspects

“Decision makers increasingly trust technology to solve their business problems.”

(van Oosterhout, 2010)

In the article The Grid Economy, Buyya et al. (2005) elicits several requirements for an effective management of computational resources in a Grid system. The authors stress that Resource management and Scheduling is a crucial part of a successful system. With Resource management and Scheduling, Buyya et al. (2005) mean that the systems for Grid computing need to manage resources and application execution depending on resource consumers’ and owners’ requirements, and they need to continuously adapt to changes in the availability of resources. Mechanisms and tools that allow resource consumers and owners (providers of a service) to express their requirements and facilitate the realization of their goals shall be provided in a Grid environment. A defined Quality of Services requirement, that specifies the resource consumers utility model, demand of resources and preference parameters shall be integrated into the system to facilitate for the consumers usage of the resources. Furthermore, a broker that supports resource discovery and strategies for dynamically scheduling application on distributed resources at runtime depending on e.g., availability, capability, and cost, depending on the business environment for the system. Basically, those resource or service providers need tools and mechanisms that enable protocols for service publication, trading, and accounting, but also price specification and generation scheme to increase utilization of the system. (Buyya et al., 2005)
Furthermore, Buyya et al. (2005) claims that a well-functioning system needs to provide mechanisms and tools to allow end users and providers to express their requirements and facilitate decision-making to cater their objectives. In short, this means that they need means to elicit their objectives and assessments, desired scheduling for resource allocations, and mechanisms to enforce selection and allocation of differential services. Moreover, providers need dynamic adaptation to change the availability of resources and services at runtime. Due to the fact that resource providers and resource consumers have different goals, strategies, and requirements that vary with time: computational Grid systems need to utilize competitive economic models and be capable of adapting when the market circumstance changes. (Buyya et al, 2005) To summarize the findings of Buyya, et al. (2005) an economy-based Grid system need to address these issues:

- An information and market directory for publicizing resources
- Users' Quality of Service requirements-driven brokering/scheduling systems.
- Accounting, billing and Payment Mechanisms that integrates economic models and negotiation protocols

In the article “The BEinGRID Project”, Dimitrakos (2010) presents the background of the BEinGRID project and its implied research results. BeinGrid is an abbreviation for Business Experiments in GRID and was the European Commission’s largest integrated project when conducted. The project consisted of a consortium of 96 partners that represents the leading European organizations in Grid Computing and Service Oriented Infrastructures. To further ensure the report’s business feasibility, several partners from different vertical markets were included in the research comprising: Advanced Manufacturing, Telecommunications, Financial, Retail, Media & Entertainment, Tourism, Health, and Environmental Sciences industries.

The mission of the BEinGRID project was to generate knowledge, technological improvements, business demonstrators, and reference case-studies that facilitate establishment of routes to foster and facilitate the adoption of Grid computing. Furthermore, the project provided guidance in: requirements, design, prototyping, and demonstrating the usage of Grid computing in a commercial environment. The solutions presented in the report meant to be as generic as possible in order to solve common business problems. Those generic solutions aimed to improve commercialization of Grid computing within three different categories: optimized and flexible processes and lower costs by improving resource utilization, collaboration and resource sharing, and new service paradigms centered to “pay-as-you-go”. (Dimitrakos, 2010) Dimitrakos (2010) claims that the focus on collaboration and resource sharing aimed to improve the following aspects the agility of businesses, collaborative processes spanning across enterprise boundaries, shared network-hosted services, and access to heterogeneous geographically distributed data sources.

The project identified requirements that if not managed properly, inhibit widespread commercialization of Grid computing (Dimitrakos, 2010):

- **Virtual Organization Management**: This will ensure capabilities to help businesses establish accountable, secure, and efficient collaborations in a Grid environment through the sharing of
services, resources, and information. A Grid computational system shall be able to secure federation of autonomous administrative domains and, also, the composition of services hosted by different enterprises (such as cloud services). A demand for new participants can appear during the collaboration lifetime, and existing participants may be dropped.

- **Trust and Security**: These aspects are, if not managed properly, a factor that inhibit the commercialization of Grid Computing and Service Oriented Infrastructures. These include solutions for brokering identities and entitlements across different services and platforms, whilst managing access to shared resources, analyzing and reacting to security events. The trust and security capabilities underpin Virtual Organization Management. The security of Business-to-business collaborations needs to be maintained: the businesses participating in collaboration must be able to identify one another, identify messages as coming from other members about the identity and entitlements of a user or other resource.

- **Software License Management**: Is a requirement for being able to adopt utilization, “pay-as-you-go”, or other emerging business models. Software licensing solutions lack feasible solutions in the majority of contemporary Grid and Cloud computing solutions.

- **Data Management**: capabilities enable better storage, access, translation and integration, and sharing of heterogeneous data. A Grid computational system shall include capabilities for aggregating heterogeneous data sources in virtual data-stores and ensuring fine-grained monitoring of usage, performance, and resource utilization.

- **Service Level Agreements**: This will ensure improvements for open standard schemes to specify agreements, to ensure fine-grained monitoring of usage, performance and resource utilization. A Service Level Agreement defines the Quality of Services, the set of quality metrics that have to be achieved during the service provision, of the service offered, and often underpins a formal contract. A Service Level Agreement includes a number of core elements or clauses. Often this also includes rights for the service provider to access customer information for business impact analysis.

- **Grid Portals**: capabilities that enable scalable solutions based on emerging Web Services technologies that provide management for user communities, complex processes and data in computational Grid environments. This will provide the Grid environment with the feature of visibility and manageability of submitting, monitoring and controlling transactions, jobs, and other computational tasks.

In order to shorten lead time for new products to markets time span, thus increasing the Business Agility, IT and communication service providers and their corporate customers have to increasingly interconnect applications and exchange data; hence being the reason for having an interconnected Grid network structure or other connection and exchange alternatives. (Dimitrakos, 2010) The way business interact is therefore evolving to: a work environment that becomes pervasive with a mobile workforce, outsourced Data Centers and in-cloud services, and integrated business process with customers and suppliers across value chains.
3.3. From theory to evaluation model

This section could be seen as a summarization of the findings in section 3.2 that is analyzed and synthesized for elaborating our evaluation model.

In order to develop an evaluation model for analyzing software license management frameworks for computational grid environments, a relationship between Grid computational environments and software licensing has to be further strengthened. The link between a computational grid system and license management is evident from the articles of Cuesta, et al. (2010), Dimitrakos (2010), and Li, et al. (2004).

Cuesta, et al. (2010) opted three different flows for a grid environment: Flows of intangible benefits, tangible goods, and financial transactions. The flows of tangible goods are mainly related to software licenses, and the author stresses the importance of a feasible licensing system for Grid-enabled applications (Cuesta, et al., 2010).

Dimitrakos (2010) elicited, through the collaborative research project BEinGRID, the requirement for software license management in a grid computational system. The author claims that software licenses are necessary for enabling commercial applications from ISVs in Service Oriented Architectures and Cloud Computing environments. (Dimitrakos, 2010).

Li, et al. (2004) enumerates several requirements for software licensing in a Grid computing environment in their article “Towards SLA-based Software Licenses and License Management in Grid Computing”, and opted that software licensing management for grid environments crucial for the commercialization of software from independent software vendors. (Li, et al., 2004)

In order to analyze the different software license management frameworks for Grid environments, a framework consisting of four categories with several internal criteria was established. The theory from section 3.2 was analyzed into different criteria and sub-criteria and were then synthesized into the categories of technology, tract, trade, and transparent. The subsequent part of this section will focus on defining the categories and their respective internal criteria.

3.3.1. Technology

According to the Oxford Dictionary technology is defined as:

“The application of scientific knowledge for practical purposes [and] the branch of knowledge dealing with engineering or applied sciences.” (Oxford University Press, 2013)

In the case of this essay the meaning of the term technology is partially narrowed into the context of this thesis. In this essay we define technology as:

“The branch of knowledge that deals with the creation and use of technical means and their interrelation with software licensing, drawing upon such subjects as Information Technology, IT-architecture, Computer Security, and Computational Grids.”
The below elicited criteria were established through analyzing and then synthesizing the theory in section 3.2 into different criteria and, in some cases, their respective sub-criteria.

The Grid services implemented using Web Services

Foster (2002) opted using standard, open, general-purpose protocols, and interfaces to address the mission for implementing various modules of the computational Grid system. Several authors presented in the previous section adhere to the view of Web Services being a feasible choice for this purpose for the following reasons:

- Boosts Business Agility. (Melarkode, et al., 2004; Weill, et al., 2002; Konsynski & Tiwana, 2004; Boden, 2004; Dimitrakos, 2010; Ross, et al., 2006)
- Augments inter-compatible data architectures. (Weill, et al., 2002; Ross, et al., 2006; Li, et al., 2004; Dimitrakos, 2010) For example through:
  - Standardization. (Weill, et al., 2002; Li, et al., 2004; Boden, 2004; Ross, et al., 2006)
  - Modularization and plug-and-play behavior of software components. (Konsynski & Tiwana, 2004; Boden, 2004)

Thus, the above mentioned benefits of using Web Services induce that the approach towards this criterion is to analyze and discuss comprising aspects of Web Services in the LMFs. Furthermore, if applicable, the above mentioned bullet points will be discussed in relation to each LMF’s usage of Web Services technologies.

Security

Weill, et al. (2002) claimed that security aims to protect the company's IT-assets. A feasible security module in a computational Grid environment, compiled from the different articles in section 3.2, is addressed by the following mechanisms:

- Firewalls (Weill, et al., 2002; Angelis, et al., 2004; Humphrey & Thompson, 2002; Foster, et al., 1998; Li, et al., 2004)
- Security logging (Angelis, et al., 2004; Humphrey & Thompson, 2002; Dimitrakos, 2010)
- Encryption (Weill, et al., 2002; Angelis, et al., 2004; Humphrey & Thompson, 2002)
- Authentication (Weill, et al., 2002; Boden, 2004; Angelis, et al., 2004; Dimitrakos, 2010) Comprising:
  - Delegation of identity (Angelis, et al., 2004; Humphrey & Thompson, 2002; Foster, et al., 1998)
  - Single sign-on (Angelis, et al., 2004; Foster, et al., 1998; Humphrey & Thompson, 2002)
  - Identity mapping (Angelis, et al., 2004; Humphrey & Thompson, 2002; Foster, et al., 1998)
  - Mutual authentication (Angelis, et al., 2004; Foster, et al., 1998)
  - Certification (Angelis, et al., 2004; Foster, et al., 1998)
- Authorization (Weill, et al., 2002; Boden, 2004; Angelis, et al., 2004; Foster, et al., 1998; Dimitrakos, 2010). Comprising:
The service to gather additional information associated with the user or the actual session (group membership, role period of validity, etc.) (Angelis, et al., 2004; Foster, et al., 1998; Dimitrakos, 2010)

The service to gather additional information associated with the protected resource. (e.g., file permissions) (Angelis, et al., 2004; Foster, et al., 1998)

The checking of any local policy applicable to the request (e.g., a temporarily disabled user). (Angelis, et al., 2004; Foster, et al., 1998; Dimitrakos, 2010)

The making of an authorization decision based on the identity of the user and the additional information. (Angelis, et al., 2004; Foster, et al., 1998)

The approach on this criterion will therefore be to analyze a certain LMF on aspects as security. Since an LMF doesn’t specify any particular implementation, we cannot completely judge how secure it will be in its system form. The approach to evaluate this will be to find and discuss certain aspects of security that each LMF demonstrates. Discussion based on findings from the IT-security parts of the theoretical framework, as well as the above mentioned bullet points will be the foundation for discussing the frameworks’ security aspects.

**Local application server**

A managed proxy server, to handle the decisions associated with the autonomous site and linked to a central management service for control purposes, has been part of more successful Grid system implementations than those based on a resilient central server. This solution has shown to provide better control at the point of invocation. (Boden, 2004; Dalheimer & Pfreundt, 2009; Erl, 2005) This server architecture also enhances the network’s self-organizing capabilities; this means that the decision rights shall be given to entities who have the best expertise to make the associated decision; hence having local application servers with the necessary authority. (Konsynski & Tiwana, 2004) Concretely, this criterion will investigate whether:

- A local application server is physically placed for control purposes at every autonomous site of the Grid system. (Boden, 2004; Konsynski & Tiwana, 2004; Dalheimer & Pfreundt, 2009; Erl, 2005)

The approach on this criterion will therefore be to analyze a certain LMF on aspects of having a local application server at every Grid site, in the case of an LMF the local application server could be denoted as local license server.

**Loose coupling and modularization**

Konsynski & Tiwana (2004) claim that loose coupling enhances the flexibility of the components in the implementation but requires modularization (which gives a plug-and-play ability of the components). Loose coupling and modularity concepts bring together robustness to the architecture and facilitate easy modifications and scalability. (Konsynski & Tiwana, 2004; Boden, 2004; Ross, et al., 2006)
The approach on this criterion will be to discuss the LMFs in terms of loose coupling and modularization. This includes discussing flexibility and interdependence of different components of the LMFs.

3.3.2. Tract
According to the Oxford Dictionary tract is defined as:

“An expanse or area of land, water, etc.; region; stretch.” (Oxford University Press, 2013)

In the case of this essay the meaning of the term tract is partially narrowed into the context of this thesis. In this essay we define tract as:

“The definite region or area where a resource, computation, or license is mobilized and executed.”

The below elicited criteria were established through analyzing and then synthesizing the theory in section 3.2 into different criteria and, in some cases, their respective sub-criteria.

Virtualization and integration of heterogeneous autonomous resources

Treadwell (2007) states that a Grid is concerned with the integration, virtualization, and management of services and resources in a distributed, heterogeneous environment that supports organizations (collections of users and resources) across traditional administrative and organizational domains (real organizations). Moreover, a system that supports these aspects also fulfills the following points:

- Network linking all points within an enterprise and/or outside enterprise (Weill, et al., 2002; Li, et al., 2004; Dimitrakos, 2010)
- Use shared services, resources, or have common applications that run independently and not under centralized control (Weill, et al., 2002; Dimitrakos, 2010; Foster, 2002)
- Providing services/resources such as servers and large-scale processing (Weill, et al., 2002; Boden, 2004, Li, et al., 2004; Dimitrakos, 2010)
- Integration of Virtual Organizations (Li, et al., 2004; Dimitrakos, 2010; Weill, et al., 2002; Foster, et al., 2001)
- Support for virtualization (Li, et al., 2004; Foster, et al., 2001)

The approach on this criterion will therefore be to analyze if a certain LMF has the ability of virtualization and integration of heterogeneous autonomous resources. The bullet points and the text will be a foundation for the discussion regarding each LMF’s ability to fulfill the requirements of a Grid system. The approach to evaluate this will be to find and discuss certain aspects of each LMF that demonstrate these features.

Flexible and dynamic resource allocation and management

Grid computing allows the pooling of resources across multiple systems and their allocation on demand to provide the quality of service typically associated with a single large mainframe or supercomputer. This requires a flexible resource allocation and management system. (Boden, 2004; Buyya, et al., 2005;
Foster & Kesselman, 1998; Weill, et al., 2002) This includes according to Dalheimer and Pfreundt (2009) and Li, et al. (2008) that an LMF can offer licenses on demand.

The behavior of such a system is the following:

- Having a resource access framework which applications can call for allocation of resources. (Boden, 2004)
- Resources are used by the requesting application and then subsequently released. (Boden, 2004)
- Optimize the use of spare resources. (Boden, 2004)
- Co-allocation of different resources (Li, et al., 2004; Foster, et al., 2001)
- Dynamicity of licenses; suspending, preempt, and resume the license usage. (Li, et al., 2004)
- Support for workflows; different applications may be used in different phases and be automatically scheduled, and this might consequently induce the need to retrieve and reserve the right licenses in advance so that the resource could be addressed. (Li, et al, 2004; Foster, et al., 2001)

The approach on this criterion will be to analyze and discuss various aspects of flexible and dynamic resource allocation and management, including the support of on-demand licenses. The bullet points mentioned above provide some guidance for the analysis of the LMFs. The approach to evaluate this will be to find, compare, and discuss certain aspects of each LMF that demonstrate flexible and dynamic resource allocation and management.

**Mobility of licenses**

Li, et al. (2008) emphasizes that the licenses must hold regardless of where the computations actually are performed to enhance a mobile and agile enterprise. Moreover, Stanoevska-Slabeva, et al, (2010) opted that support for mobility in an IT-infrastructure has the key role of increasing the agility of a company. Mobility in a Grid environment can be augmented by:

- Remote license enforcement; where the jobs can be executed remotely while the validity of the licenses has to be guaranteed at the same time. (Li, et al., 2004)
- The license can be moved from one computing entity to another. (Li, et al, 2004)

The approach on this criterion will therefore be to analyze if a certain LMF has the ability of providing mobility of licenses. The above-mentioned bullet points will be used as a guidance for the discussion. The approach to evaluate this will be to find and discuss certain aspects of each LMF that demonstrate these features.
3.3.3. Trade

According to the Oxford Dictionary trade is defined as:

“The action of buying and selling goods and services [involving] the people engaged in a particular area of business.” (Oxford University Press, 2013)

In the case of this essay the meaning of the term trade is partially narrowed into the context of this thesis. In this essay we define trade as:

“The commercial aspects of the act or process of buying, selling, or exchanging commodities according to predefined regulation, affecting the trading partners within a computational Grid environment.”

The below elicited criteria were established through analyzing and then synthesizing the theory in section 3.2 into different criteria and, in some cases, their respective sub-criteria.

Regulation of resources based on company policies

In an economy-based Grid computing environment, resource management systems need to provide mechanisms and tools that allow resource consumers (end users) and providers (resource owners) to express their requirements and facilitate the realization of their goals. (Next generation GRIDs Expert Group 2006; Buyya, et al., 2005) This includes systems that enables and facilitates the following aspects:

- Enabling resource consumers’ and owners’ requirements and realization of their goals. (Buyya, et al., 2005; Dimitrakos, 2010; Li, et al. 2004; Weill, et al. 2002)
- Utility models for how consumers demand resources and their preferred parameters. (Buyya, et al., 2005; Dimitrakos, 2010; Li, et al. (2004)

The approach on this criterion will therefore be to analyze a certain LMF’s ability to facilitate economic-based requirements and realization of goals based with the two above mentioned bullet points as guidance for the discussion. The approach to evaluate this will be to find and discuss certain aspects of each LMF that demonstrate these features.

Billing and payment

Dimitrakos, et al. (2010) says that there are business models, which are essential in a business context and shall be supported. These shall be implemented in a billing and payment module to facilitate the economic aspects. (Foster & Kesselman, 1998; Li, et al., 2004; Buyya, et al., 2004) Some points for evaluation of a feasible billing and payment module will be considered:

- An information and market directory for publicizing resources and their values. (Buyya, et al., 2005; Dimitrakos, 2010)
- Accounting, billing and payment mechanisms that integrates economic models and negotiation protocols. (Buyya, et al., 2005; Boden, 2004; Dimitrakos, 2010)
The approach on this criterion will therefore be to analyze a certain LMF’s ability to facilitate billing and payment features. The above mentioned bullet points will be used as a guidance for the discussion. The approach to evaluate this will be to find and discuss certain aspects of each LMF that demonstrate billing and payment features.

3.3.4. Transparent
According to the Oxford Dictionary transparent is defined as:

“Easy to perceive or detect.” (Oxford University Press, 2013)

In the case of this essay the meaning of the term transparent is partially narrowed into the context of this thesis. In this essay we define transparent as:

“Mechanisms that makes relevant information regarding computational data or resources easy to perceive or detect.”

The below elicited criteria were established through analyzing and then synthesizing the theory in section 3.2 into different criteria and, in some cases, their respective sub-criteria.

Provision of information about resources and their availability

One of the key features that are provided by Grid computing is the provision of information about resources and their availability. (Boden, 2004; IBM, 2004; Li, et al., 2008)

There shall, according to Buyya et al. (2005) and Humphrey and Thompson (2002), be information regarding the Grid resources. Humphrey and Thompson (2002) emphasizes that there shall be an information service that allows potential users to locate resources and query them about access and availability.

The approach on this criterion will be to analyze and discuss various aspects of information provisioning regarding its resources and availability. This includes discussing the extent in which an information service is used. The approach to evaluate this will be to find, compare, and discuss certain aspects of each LMF that demonstrate these features.

Transparent licensing mechanism

Li, et al. (2004) clarifies that licenses should be transparently handled as part of the Grid job management. A better visibility of the license usage provides the resource owner and consumers with valuable information of data on customers, products, processes, performance, and capabilities of the system (Weill, et al., 2002). Increased visibility is enhancing coordination and trust among the partners involved in the system (Konsynski & Tiwana, 2004).

Visibility and manageability of submitting, monitoring and controlling transactions, jobs and other computational tasks shall be integrated, which requires a transparent licensing mechanism. This because a licensing mechanism that is transparent for the user, which will reduce the risk of causing frustration for the user. (Dalheimer & Pfreundt, 2009; Li, et al., 2008; Dimitrakos, 2010; Melarkode, et al., 2004)
This criterion will be evaluated from the aspect of expressed mechanisms that provides transparency. Transparency capabilities will be discussed among the different LMFs.

3.4. The Evaluation Model
This section will summarize the synthesized categories and their internal criteria from section 3.3 into a picture that aims to visualize the Evaluation Model that is going to be used for the LMF study.

The categories and their internal criteria will be the foundation for analyzing different LMFs. During the construction of the Evaluation Model, we noticed clear dependencies among two different categories. The Trade category is reliant on the Transparent category since Trade aspects need to be Transparent for management purposes, thus a rightwards arrow will denote this dependency. Another, and the most obvious dependency, is the other categories relying on the Technology category since the other categories depend on technology to be implemented (and is also limited by the choice of technology and its architecture). This dependency will be visualized by arrows pointing towards the Technology category from the other categories. The Evaluation Model can be visualized as:

![Figur 3-1: The Evaluation Model](image-url)
4. Our interpretation of Ericsson’s criteria in relation to the Evaluation Model

In this chapter our interpretation of Ericsson’s criteria in relation to the Evaluation Model are described and the information needed was gathered through interviews with Ericsson employees and the internal document “Focal Point Report”. The questions asked to the employees can be found in appendix 2.

Our interpretation of Ericsson’s criteria in relation to the Evaluation Model is visualized in the figure below. The different criteria of this figure are then explained separately. The sub-criteria are not described since the information that was more detailed than the criteria was too implementation specific and hence irrelevant for an LMF.

Figure 4-1: Our interpretation of Ericsson’s criteria in relation to the Evaluation Model
Technical

In this section Ericsson’s technical criteria are described.

The Grid services implemented using Web Services

The requirements Ericsson have regarding Web Services are regarding standardization and security. (Process Architect ELIS & BNET, 2013)

Standardized protocols are supposed to be used in order to enable interaction between multiple devices even though these devices work in different ways. (Process Architect ELIS & BNET, 2013)

The security aspects imply the usage of cryptography standards in case the Web Service contains sensitive information. (Process Architect ELIS & BNET, 2013)

Security

Product Manager License Management (2013) writes in Focal Point Report that unlawful creation, duplication, and tampering of licenses shall be prevented, in order to satisfy the security requirements. The Director Software and Product Packaging BNET (2013) emphasizes that unauthorized usage of Ericsson’s products must be prevented, and also that there cannot be any open ports in the technical implementation (requirement of firewall).

Ericsson has standardization requirements regarding security and uses for instance the cryptography standard X.509. (Process Architect ELIS & BNET, 2013)

Local application server

Ericsson does not use local application servers at the moment, and there is no criterion that says that they should use it. There is on the other hand no criterion that forbids the use of local application servers. (Process Architect ELIS & BNET, 2013) This means that there will be no Ericsson criteria regarding the local application server.

Loose coupling and modularization

The Price Manager BNET (2013) and the Process Architect ELIS & BNET (2013) explains that loose coupling and modularization is important for an LMS at Ericsson.

The reason for this is that loose coupling and modularization increases the flexibility of the system (Process Architect, 2013).

The use of modular solutions with standardized components, built-in-intelligence, and infinite configurability improves the operational flexibility and responsiveness. (Ericsson’s Strategic Direction, 2013)
**Tract**

In this section Ericsson’s criteria regarding the tract of resources are presented.

**Virtualization and integration of heterogeneous autonomous resources**

The Price Manager BNET (2013) clarifies that it must be possible for the customers to buy virtualized services, such as clouding, in case they do not wish to purchase the physical hardware. This is hence a sub-criterion of Ericsson.

The reason for the importance of clouding is that it enables flexibility, efficiency and innovation while reducing the investment risk of the customer. Virtualization strengthens Ericsson’s cloud approach, so both virtualization and clouding are important for the strategy of Ericsson. (Ericsson D, 2013)

**Flexible and dynamic resource allocation and management**

Strategic Supply Management BNET (2013) explains the usage of execution layers which helps in the resource allocation process. This implies that computations or other resource allocations can be performed either locally or somewhere else in the Grid depending on what services and products the customer has bought. (Strategic Supply Management BNET, 2013)

**Mobility of licenses**

It shall, according to Focal Point Report, be possible to move licenses in between hardware devices. This includes reallocation due to faults in a product, but also reallocation within the networks of the customers. (Product Manager License Management, BNET, 2013)

**Trade**

In this section the commercial criteria of Ericsson are presented.

**Regulation of resources based on company policies**

It must be possible to express rules regarding the handling of customers’ capacity exceeding. Furthermore, it shall be possible to create both perpetual and time limited license keys with a certain time to validity. (Product Manager License Management, BNET, 2013)

Ericsson’s LMS shall be flexible and be able to handle a great number of different products that can be packaged in different ways (Director Software and Product Packaging 2013; Process Architect ELIS & BNET, 2013 A). The demand of the customers can differ which implies specialized solutions and the LMS must be flexible in order to handle this. (Process Architect ELIS & BNET, 2013 A)
**Billing and payment**

It must for commercial reasons be possible to use different pricing strategies, which for instance includes monitoring and estimated usage. Unless the license is perpetual measuring algorithms shall be used and they must correspond to the pricing model. (Director Software and Product Packaging BNET, 2013; Process Architect ELIS & BNET, 2013 A)

**Transparent**

In this section Ericsson’s criteria regarding transparency were presented.

**Provision of information about resources and their availability**

The customers of Ericsson are interested in information regarding their usage, and it is important that they only retrieve the information that is relevant for them. The reason is that redundant information can be confusing for the customer and also make Ericsson look like they cannot handle their business. (Process Architect ELIS & BNET, 2013 A)

**Transparent licensing mechanism**

The licensing mechanism of Ericsson shall be transparent in order to control the system and the customers’ usage of the products. This implies using counters logs for measuring the usage and use alarms to notify the customer when it is necessary. (Product Manager License Management, BNET, 2013; Process Architect ELIS & BNET, 2013 A)

**4.1. Evaluation Model comparison**

By comparing Ericsson’s criteria with the Evaluation Model, it was possible to determine that all criteria from the Evaluation Model could also be found Ericsson’s criteria except for the criterion for local application servers. There was on the other hand no criterion from Ericsson that could not be found in the Evaluation Model.

There was more explicit information regarding the criteria in the Evaluation Model and the criteria often consisted of sub-criteria. There was, however, one sub-criterion, clouding, (under Virtualization and integration of heterogeneous autonomous resources), in our interpretation of Ericsson’s criteria in relation to the Evaluation Model that could not be found in the Evaluation Model.

Due to their similarity the Evaluation Model seems to be feasible for the particular case study and will be used to analyze the LMFs in the next chapter. In the end of chapter 5 Spectrum of License Management Frameworks this comparison will be used in order to analyze the LMFs for the particular case study.
5. A Spectrum of License Management Frameworks

In this section, all the LMFs that could be found in the empirical data collection will be presented; hence there have been no specific selection process applied.

5.1.1. SmartLM

SmartLM is both the name of the European research project and the LMF. The eight originators of SmartLM have various experiences in: Supercomputing, Service Oriented Architecture, Grid, Distributed and Cloud environments, business consulting, virtualization, software licensing, and project management. (Cacciari, et al., 2011) The project was performed as collaboration with 30 software vendors and buyers (end users) to ensure the business feasibility (The 451 group, 2009).

The SmartLM LMF has identified three different actors: the User (and her home organization, which is the licensee), the Independent Software Vendor (ISV), which is the licensor, and the Application Service Provider (ASP), which provides the hardware and hosts the application. The ASP could be the same actor as the licensor, licensee, external service provider, or a combination of these depending on the business context. The LMF enables the actors to make use of economies of scale and SmartLM features; the different actors can introduce new payment models such as pay-per-use that the Grid environment make possible. (Cacciari, et al., 2011)

The authors of the SmartLM LMF have described the relationship between the different actors (denoted the SmartLM ecosystem by the originators) in the picture below:

![Figure 5-1: The SmartLM ecosystem. (Cacciari, et al., 2011)](image-url)
SmartLM was developed in order to reflect the changing paradigms in Information Technology; with a transition from non-dynamic infrastructures of huge computing centers to agile, service oriented architectures. To find a solution to manage software licensing in Grid environments is the main purpose of the SmartLM project. Providing platform independent access to licenses just like other virtualized resources is the core of the SmartLM architecture, and this is done by treating and implementing software licenses as Web Service resources. Licenses are managed through a license service integrated and implemented as a variety of specialized modules (license, accounting and billing, orchestration service, etc.) and realized by software tokens. There is a local application server placed at the user’s home organization for the authorization for using a license. The authorization takes into account availability of the requested license and features, policies defined (by the copyright owner of the software being used), and policies defined by the local administrator; thus providing the capabilities to adapt to specific needs. A binding agreement between user and license service is achieved by Service Level Agreements for all local authorizations for license usage. The outcome of an authorization is digitally signed software token that contains all license-related information (local-user id etc.) plus a number of certificates. This token is then deployed in the environment where the license-protected application is going to be executed, where the token can be used on-demand or also be reserved in advance for usage at a later point in time. (Cacciari, et al., 2011) One important design goal is the adaptability of the SmartLM solution to the needs of different customers; hence the License Service providing the core functionality and the Accounting and Billing Service were designed and implemented as a loosely coupled, modular system (SmartLM consortium, 2010).
Figure 5.2: The License Service architecture follows a layered architecture comprising 6 layers: Co-allocation, Authentication, Administration, Management, Business, and Persistency. (Cacciari, et al., 2011)

According to the picture above the component fulfills the following tasks (SmartLM consortium, 2010):

- In the License Administration Service a set of operations can be done, which enhances the capability to manage the licenses (add or make reservation of licenses). It processes the requests and delegates these decisions to other components using a Web Service-compatible output.
- The License Information Service is a point of aggregation to collect pieces of information from different sources, which are the other components of the License Service. It has been implemented as a Web Service.
- The License Management Service is used as a central service for license administration, license storage and scheduling as well as token processing and usage record creation. It is divided into several components, each responsible for a specific part of the functionality. Since all components, which have a direct interaction with an actor (e.g. an administrator or another service) they are therefore implemented using Web Services technology.
- Policy Management Service is the module that administers the local policies set for the autonomous Grid site.
- The Usage Record Consumer integrates a Web Service that ensures SOAP interactions; this module is responsible for documenting the license usage.
The Service Level Agreement and Negotiation service provides license mechanisms based on Web Service agreement and negotiation.

The Storage Service is responsible for storing XML data permanently in files or databases; hence based on Web Services.

The Orchestrator components are also implementing Web Service Agreement and Negotiation and use it to accomplish user driven co-allocation of licenses and compute resources.

Figure 5-3: Purchase and checkout diagram where a job is submitted to the ASP’s site (Cacciari, et al., 2011)

According to Cacciari, et al. (2011), the SmartLM project in brief aims to overcome the limitations described by:

- Decoupling the authorization for using a license from the authorization for executing a license-protected application
- Creating a secure software license token that may be deployed to the environment where it is needed for authorizing the application execution.
- Providing defined Quality of Service in terms of a Service Level Agreement as negotiated between user and license service regarding, e.g., availability of a certain license and corresponding features at a given time and for certain computing resources
- Implementing sophisticated security mechanisms to secure the token and to allow the API to validate a token prior to forwarding the content to the application for authorization.

"Technical

The Grid Services implemented using Web Services

The core of the SmartLM architecture is to treat software licenses as Web Service resources and thereby provide platform independent access like other virtualized resources. Using open standards as far as possible instead of proprietary protocols is considered crucial for the interoperability. SmartLM license usage is governed by Service Level Agreements based on the Web Service agreement specification (SmartLM consortium, 2010).

Security

Securities affect all the architectural layers in SmartLM and provide security logging for all the different modules, the authors of the framework claims that the security mechanisms renders illegal use almost impossible since all unauthenticated or unauthorized calls will be rejected by incoming security handlers and firewalls. Cacciari, et al. (2011) claims that the SmartLM would be useless without a sufficient level of security which is regarded from two aspects: services and licenses. The license security implies that unauthorized usage is prevented by signing and transporting information in a X.509 certificate. The entire information will be signed, so it is not possible to copy the license to another license server without also copying the server’s identity. Optionally, to prevent lodging of the license files on other servers: certain properties of the license server’s hardware can be included in the license server’s identity. (Cacciari, et al., 2011)

All authorization for the use of a license is done locally at the home organization of a user, and sometimes depending on the case routed to a central management service for control purposes, taking into account policies of the independent software vendor, site specific policies defined locally or user-specific attributes (that may be elicited by a specific virtual organization). It successfully decouples the authorization for using a license from the authorization for executing a license-protected application. Signed and encrypted terms of a license are scheduled to the (remote) execution environment. The SmartLM architecture also has a sophisticated security mechanism to secure the token and to allow the application interface to validate a token prior to forwarding the content to the application for authorization. The authentication process always relies on the common Unicore Virtual Organization System (UVOS) service, allowing storing user attributes and validating the user credentials. (SmartLM consortium, 2010)
Local application server

The SmartLM LMF states that the authorization for using a license is done locally at the license service of the user’s home organization and sometimes, depending on the use case, linked to central management service for control purposes. (Cacciari, et al., 2011)

Loose coupling and modularization

Licenses are managed through a license services and realized as software tokens and sent between the modular services. Examples of such services provided by the license management framework are: License, Accounting and Billing, Orchestration Service, etc. This divides the LMF into modules that can be separated and replaced, which can be visualized in the License Service architecture. Furthermore, the License Service providing the core functionality and the Accounting and Billing service were designed and implemented as a loosely coupled, modular system. (SmartLM consortium, 2010)

Tract

Virtualization and integration of heterogeneous autonomous resources

The main concept with SmartLM is, according to Cacciari, et al. (2011), to change the way licensing works and use flexible and non-hardware based licensing solutions that better fit into Grid and Virtual environments, with a possibility of commercial software licenses in Clouds. The integration of autonomous resources is done locally at the license service of the user’s home organization. The license mechanism is designed for Grid environments and works with loosely coupled systems, realizing software licenses as Web Service resources renders licenses adaptable and mobile. The software token
is deployed in the environment where it is going to be executed, an integration of different autonomous resources. All the services are controlled by every site and not centrally. (SmartLM consortium, 2010)

Flexible and dynamic resource allocation and management

The SmartLM LMF supports three different actors in its license management system: the user (and her home organization, which is the licensee), the independent software vendor (ISV), which is the licensor, and the application service provider (ASP), which provides the hardware and hosts the application. The ASP and ISV could be the same actor.

Furthermore, the Orchestrator provides automated mechanisms to support the user in co-allocating computational resources and the licenses both in time and space, allowing to use remote Grid resources for the execution of an application which needs a software license at run time, while being sure that the license required is available at the remote site when the application will start-up and during execution. The software token is created by the License Management Service and transferred to the Orchestrator, which in turn sends the token together with the user’s job to the reserved (external) computational resource (wherever it is going to be computed). (SmartLM consortium, 2010) The SmartLM architecture also offers the on-demand business concept to be employed in the licensing environment. (Cacciari, et al., 2011)

A built-in license scheduler, in the SmartLM architecture, allows licenses to be reserved in advance. With advance reservation a license can be guaranteed to be available at a later point in time. (Cacciari, et al., 2011)

Mobility of licenses

The makers state that the LMF brings along a model that makes licenses mobile objects. These licenses are managed through a license service implemented as a bag of specialized services implemented as Web Service resources, and realized as software tokens, which deliver the required flexibility and mobility.

Decoupling authorization for license usage from authorization for application execution finally allows executing the license protected application without requiring a permanent bidirectional network connection to the license server at runtime controlling the authorization for application execution, hence increasing the mobility since the solution can be used in different computational environments.

Trade

Regulation of resources based on company policies

The SmartLM project’s idea for successful licensing is to achieve software licensing based on business objectives that balance customer needs and vendor business models. The authorization for usage of a specific software component takes into account availability of the requested license and features, policies defined by the ISV (On the detail level of for example: constraints regarding the nationality of a user for certain applications from ISVs with a headquarters in the US) and policies defined by the local
administrator (e.g. usage times for production and test). All local authorizations for license usage are expressed and guaranteed by Service Level Agreements between the user and license service.

On an architectural level, the Policy Management Service, within License Service architecture Management Layer, is the central entity for evaluating policies that define the access to and the usage rights of a certain license or license feature for a user or a group of users. The policies can include such defined by the ISV, policies defined locally by the site owning the license, and policies determined from the membership of the user in a Virtual Organization.

Furthermore the SmartLM LMF has an External Billing Service (EBS) interface and Budget Information Services (provides to the License Service the consumed budget for a specific end user) that makes the system able to obligate the user’s requirements.

**Billing and payment (SmartLM consortium, 2010)**

The Accounting and Billing Service module is a service integrated within the license service manager through Web Services. It can be divided and split into two main levels: an external layer that confers the service with the required communication within the License Service and the end users, and an internal layer that allows the service to treat, send, and filter information regarding the external layer requirements.

![Figure 5-5: Accounting and Billing Service architecture in terms of external and internal components (SmartLM consortium, 2010)](image-url)
The Authentication Layer is composed by three different modules, two external to the Accounting and Billing Service (the License Service and the External Billing Service (EBS)), and one internal, the Portal. Users are authenticated through the Portal, and are later granted different features depending on the user’s role. The portal also provides the user with accounting and billing license usage information.

The Persistence Layer includes only one module, the Database that stores all the Usage Records, that are retrieved from the Storage Service inside the License Service and that contain information about licenses features usage.

The responsibility of the Business Layer is to provide the License Service with the required licenses prices for the execution of SmartLM licensed applications. The Business Layer in turn consists of the Budget Information Service and the Rule Engine. The Budget Information Service provides to the License Service the consumed budget for a specific end user. This information will be used by the License Service to allow or reject the use of the requested license. The Rule Engine is used to map the different

---

**Figure 5-6: The layered architecture of the Accounting and Billing Service. Where the colored components symbols the different layers (Authentication, Business, Management, Persistence layers) (SmartLM consortium, 2010)**
processes for the different licenses’ features with the different users, and set up license resource pricing models. Both the administrator and the ISVs have access to this tool through the Portal and its connection to the Rule Engine Interface.

The Management Layer comprises five modules: The Usage Record Consumer, the Price Information Service, the Usage Record Processing Logic, the Database interface, and the External Billing Service or Portal interface. The main responsibility of this layer is to coordinate the different components inside the Accounting and Billing Service.

The Administration Layer hosts only one module, the Rule Engine Interface and it is responsible to configure the Rule engine components.

The Budget Information Web Service Interface is the interface between the license service and the Accounting and Billing service. The purpose of this interface is to provide the available budget of a user group (project/department/company etc. are managed as user groups) to the license service. It is implemented as a Web Service based solution.

**Transparent**

**Provision of information about resources and their availability**

SmartLM claims that its Quality of Service in terms of a Service Level Agreement as negotiated between user and license service regarding, e.g., availability of a certain license and corresponding features at a given time and for certain computing resources also can be provided and transparent for the different user groups. At a technical level the authorization takes into account availability of the requested license and features. Licenses can also be reserved for usage at a later point in time. Budget Information Services provides to the License Service the consumed budget for a specific end user. The License information service is a point of aggregation to collect pieces of information from different sources that can be viewed and shared for different authorized user groups. (Cacciari, et al., 2011)

**Transparent licensing mechanism**

The Accounting and Billing Service is realized as an additional feature that allows ISV, ASP or end users to have global control and overview of their license usage. This brings the possibility of detailed statistics based on license resource usage that is provided to the final user as graphs, tables, or printable files. Furthermore, the Accounting and Billing Service can enable the end-user to have a full cost control. (SmartLM consortium, 2010)

The license information service within the license service architecture allows both users and administrators to retrieve information on installed ISV licenses, licenses available, reserved or in use. All the information necessary for the end-user can be accessed through the Web-GUI. (SmartLM consortium, 2010)

For interaction with an administrator the Admin Service is used. On the one hand this service provides a set of operations, which allow the administrator to manage the set of handled licenses, for instance add
new licenses or remove existing licenses. The Usage Record Consumer shows the amount of spent resources from the user’s point of view and, hence, increases the transparency of the system. (SmartLM consortium, 2010)

5.1.2. GenLM (Dalheimer & Pfreundt, 2009)

GenLM is a license management framework developed for Grid and Cloud Computing Environments by the computer scientists Mathias Dalheimer and Franz-Josef Pfreundt in 2009, they are both active within the research fields of supercomputing and distributed systems.

GenLM allows an Independent Software Vendor to manage their license usage in a Grid environment, and the main purpose was to help Independent Software Vendors, with a secure and robust solution, to expand into the Grid and Cloud computing environments. The authors claim that the contemporary software licensing models have been developed without the consideration for computing environments that are geographically dispersed. This requires a solution that lets the licenses be mobile; this means that they can be used on any resource the user has access to. This renders the ability that a user can buy a per-job license and run the job on any suitable resource.

The main idea behind the license management framework is to attach the license not to a node or a person, but to issue licenses for the input datasets. This way, a user can buy a per-job license and run the job on any suitable resource. The LMF assumes that there are three stakeholders involved in Grid computing. The users (that buys rights to use software, licensee), the resource providers (providing the capability to process data), and the Independent Service Vendor (that owns the immaterial right to the software, licensor). The resource provider could be the same actor as the licensor, licensee, external service provider, or a combination of these depending on the business context. GenLM participated in a partner project with different Independent Software Vendors in order to obligate to the legal and business structures that exists on the market place.

![Diagram of GenLM computing entities](image)

**Figure 5-7:** An overview of the GenLM computing entities. The figure shows the logical message exchange between user, ISV and resource provider. Please note that the message queue and firewalls are not shown

The basic computing entity architectural overview is shown in the picture above. The sites involved in this diagram are the User Site, The Independent Software Vendor Site, and the Resource Provider Site. The User Site contains a local GenLM client server that is responsible for acquiring a license token. The GenLM server on the Independent Software Vendor’s Site has the task of issuing licenses on demand; it also verifies if a license is valid for the pending job. It later creates the license token, with the User’s identity, for the input data and sends it back to the GenLM client at the User’s site so that he later can...
transfer it to the right (remote) Resource Provider Site. In this stage the (remote) Resource Provider Site has all the data it need to check the validity of the software license and use the input data for execution.

The authors elicited several requirements and solutions they want to fulfill in the article, those are visible in the cited list below:

- Support for existing license agreements: in practice it is almost impossible to change legal contracts during their lifetime. A licensing framework must support existing contracts.
- On Demand licenses: in addition to the classical “CPU per Year” licenses, it should also be possible for an ISV to offer on demand licenses when desired. In combination with the first requirement this would allow complex licensing terms: for example, a user must purchase a base license which is valid for a year. Additional software modules could then be licensed on demand on top of the existing base license.
- Mobility of licenses: as outlined above, grid providers and cloud computing data centers are usually not involved in the license agreement. Technically, the license should be valid regardless of the execution location of the computations.

Beside these requirements we also need to fulfill certain nonfunctional requirements:

- The framework must be easy to integrate with existing software packages. Ideally, the existing license management routines will be replaced or enhanced by the new framework. The framework must also be portable across different operating systems.
- Support for both grid and cloud computing environments must be built in. This requires special care regarding network friendliness and security.
- The system must be highly available. If users can't request licenses when they demand them, this might have serious implications on the ISV.
- The system must be secure. Licenses must not be granted without a license contract. Additionally, it must be difficult for malicious crackers to break the license checks in the software itself.

Technical

The Grid Services implemented using Web Services

For communicating between the different modules or the entities in the GenLM architecture, the authors use Apache ActiveMQ. ActiveMQ supports a wide range of language clients and protocols comprising Web Services (The Apache Software Foundation, 2013).
Security

The authors elicit that one important aspect of the LMF is that the system must be secure. They will fulfill this goal by ensuring that licenses must not be granted without a license contract, and therefore prevent malicious crackers to break the license checks in the software itself. The GenLM framework is designed to work behind firewalls. The whole GenLM tool suite is built around strong cryptographic primitives. The software token is generated by the GenLM client at the User Site and then signed by the GenLM server at the Independent Software Vendor. The GenLM client starts by computing a cryptographic hash for all the input files that will be used in the computation at the resource provider’s site. The framework use SHA-256 cryptographic hash algorithm for the encryption, which according to the American government agency National Institute of Standards and Technologies (2013) is secure and reliable. For signing purposes the certificate standard X.509, which according to Zissis & Lekkas (2013) is secure and trustworthy, is utilized for signing and later sent to the GenLM server at the Independent Service Vendor’s site. The GenLM server then extracts the license term and the user’s identity, which identity can be checked against a customer database in all the steps if necessary. If the user’s request is granted the server uses its own certificate to sign the user’s token and also encrypt the data. This token is then sent back to the GenLM client at the user’s site. The GenLM client, if the request was successfully signed and accepted, transfers the software token together with the input data, everything encrypted, to the (remote) computing site, where a job is queued. Before the job starts, the license token is then verified and inspected at the resource provider’s site. The public key of the license server is verified. If the signature is correct, the hash tags in the input files are compared with the hashes in the token. If everything is done, successfully, the computation at the resource provider’s site will start.

Local application server

The framework states that a GenLM client machine is placed at the User’s site and that this approach is more suitable for distributed environments.

Loose coupling and modularization

The authors of the LMF state that they use a loosely coupled approach for the overall architecture.

Tract

Virtualization and integration of heterogeneous autonomous resources

The authors of the GenLM LMF states that the framework is made for providing a solution to the license dilemma in Grid, virtualized and Cloud computing environments, this is also formulated as a strict requirement by the authors. The data and license can be moved to an arbitrary execution location with no dependency on the environment. They stress the importance that this framework integrates three possible stakeholders in the licensing process: The users (that buys rights to use software, licensee), the resource providers (providing the capability to process data), and the Independent Service Vendor (that owns the immaterial right to the software, licensor). The resource provider could be the same actor as
the licensor, licensee, external service provider, or a combination of these depending on the business context.

Flexible and dynamic resource allocation and management

The GenLM LMF states that the input data and license for the computation will work with an arbitrary execution location, with no dependency on the environment. This is, though, also stated as a requirement.

Mobility of licenses

The authors of the GenLM framework states the mobility of licenses as an important aspect, due to that acquisition of additional licenses for a short period of overload will increase in the future, while the permanent licenses will satisfy the average license usage. Furthermore, a requirement of the GenLM LMF elicited by the authors is that it should support on-demand licenses, in addition to the standard “CPU-per-year” licenses. The GenLM server, placed at the Independent Service Vendor’s site is responsible for issuing licenses on demand.

Trade

Regulation of resources based on company policies

One requirement that adheres to this topic is the “Support for existing license agreements”. The authors of the GenLM framework elicits that their frame must hold for existing contracts. There is also a stated support for integration with different software packages, depending on the vendor’s preferences. The license terms are in GenLM, Independent Software Vendor specific and would typically contain information such as the number of requested cores, the software modules to be used for the job etc. All the license terms are stored in a Policy plugin that is integrated in the GenLM server, residing at the Independent Software Vendors site. The purpose of the plugin is to obligate the business model chosen by the Independent Service Vendor.

Billing and payment

The GenLM LMF has no stated Accounting and Billing module. However, the authors state that all the necessary steps in order to be able to bill the customer will be made in the policy plugin, which might involve putting a billing record in a database.
Transparent

Provision of information about resources and their availability

No information regarding providing the user with information about resources and their availability was stated in the GenLM framework.

Transparent licensing mechanism

No mechanism about enhancing the transparent licensing mechanism was found in the GenLM LMF, even though the authors state that:

“Users expect the licensing mechanism to be transparent. Problems with the licensing mechanism will yield frustration on the user’s side and must be avoided.”

5.1.3. Service Oriented License Providing (SOLP) (Katsaros, et al., 2009)

The Service Oriented License Providing (where SOLP is our own abbreviation since the LMF does not have a name of its own) was created by four scientists that are active in the fields of Computer Science, Service Engineering, Cloud Computing, and Information Management. The LMF implies a solution for managing and providing software licenses to resolve the restrictions and to confront the weaknesses of contemporary LMS. They therefore present a Service Oriented Architecture approach based on the concept of License as a Service (LaaS), which enables a user to buy a license on-demand and pay per unit of use (pay-as-you-go etc.). The SOLP solution provides a solution consisting of independent components or services, resulting in a flexible but also secure and stable system. The authors opt that the SOLP architecture will introduce new roles and business models as well as leverage new technologies and trends into the licensing sector.

The authors have identified three different actors that play a key role in the architecture:

- Independent Software Vendor that play a main role in the provision of licenses to all interested parties. This role may or may not be the implementer of the software, depending on the business context.
- The Resource Provider is, in the context of Grid environments, responsible for the provisioning of computing and/or data resources for the end users. This could be at the Independent Software Vendor or the End User, depending on the business situation.
- The End Users are the consumers in this LMF, because they are the ones that are using the resources and applications.

The solution provided in the SOLP system has two simple Use Cases, as exemplified in the picture below. A user or an application can whenever request a license, while the License Provider provides the License required for the operation.
On a more overview level the figure below shows the operational scenario for a proposed architecture comprising four components as well as a Web Service Interface that manages the communication between the different modules. On the Consumer side of the Unified Modeling Language (UML) diagram, there exists an Application/Resource entity (could be a Grid resource or any other license consumer) integrated with a Service Client that controls and implement the external communication. The Service Clients listen to a port for incoming messages that triggers an event from a request. When a request is received, the Service client sends the package along with the user’s credentials to the Provider.
The License Provider contains two Web Service components and a License Server that could operate with any contemporary LMS such as FlexLM. The License Service is triggered when a License Request arrives, the License Service exchanges messages with the Billing and Accounting Service in order to verify a specific user and check that the entity is valid for acquiring the specific license. When all the mandatory authentication, authorization and billing steps has been accomplished, the License Server returns a response back to the License Service and the Application on the user’s side. The Billing and Accounting service integrates the concept of Service Level Agreements in its modules that is valid between the Consumer and License Provider. The Service Level Agreement (SLA) specifies the license usage, cost, and Quality of Service parameters, but also other constraints applicable for the business context.

The steps in the diagram below are clearly depicted in the article and states:

“

- **Steps 1 & 2**: The whole process is triggered by the Application User who sets up an account on the Billing / Accounting Service of a License Provider and negotiates the respective Service Level Agreement.
- **Step 3**: The User opens the Application
- **Step 4**: The Application requests for a license by sending low level TCP [, a Web Service protocol,] packages to the predefined Service Client.
- **Step 5**: The Service Client captures that request and invokes the License Service using the credentials of the User. We can note that the Service Client operates like a proxy between the Consumer and the Provider upgrading the low level TCP request of the Application to secure Web Service calls.
- **Steps 6, 7 & 8**: The License Service validates the request against the User’s Account and Service Level Agreement, retrieves the license from the License Server and bills the User Account respectively.
- **Step 9**: The License Service after having received the low level packages from the License Server sends back a respective SOAP [, a Web Service Protocol,] message back to the Service Client
- **Step 10**: The Service Client unfolds the soap message and pushes the appropriate package to the application’s socket.

“
When designing the architecture of the system, the general principles for Service Oriented Architectures were used, thus:

- Reuse, granularity, modularity, compose ability, and componentization
- Standards compliance (both common and industry specific)
- Services identification and categorization, delivery, monitoring, and tracking.

This ensures, according to the authors, a compatible system that can be integrated with contemporary LMSs.

**Technical**

**The Grid Services implemented using Web Services**

The authors of the report state that the SOLP LMF follows the Service Oriented Architecture guidelines for development and design principles. The second bullet point (among the SOA guidelines) stresses the importance of standards compliance (both common and industry specific). This is obvious in the framework since all the communication between the components implies the use of Web Services for communication.
Security

When the user initiates a license request the Service client sends the user’s credentials to the Provider. The Billing and Accounting Service handles the Authentication and authorization of the user.

Local application server

There is no specified Local application server within this framework.

Loose coupling and modularization

The authors of the LMF state that the SOLP LMF follows the Service Oriented Architecture guidelines for development and design principles. The first bullet point (among the SOA guidelines) stresses the importance of reuse, granularity, modularity, compose ability, and componentization. Usage of this can be seen in the Service Oriented License Providing Component Diagram, where the different modules are divided and inter-entity loosely coupled. The authors also state in the article that the LMF supports modularity and componentization.

Tract

Virtualization and integration of heterogeneous autonomous resources

The authors state that SOLP shall work with Grid and Cloud computing environments, which shall, per definition, include virtualization and integration of heterogeneous autonomous resources. Furthermore, the authors have identified three different actors that play a key role in the architecture:

- Independent Software Vendor that play a main role in the provision of licenses to all interested parties. This role may or may not be the implementer of the software, depending on the business context.
- The Resource Provider is, in the context of Grid environments, responsible for the provisioning of computing and/or data resources for the end users. This could be at the Independent Software Vendor or the End User, depending on the business situation.
- The End Users are the consumers in this LMF, because they are the ones that are using the resources and applications.

Flexible and dynamic resource allocation and management

The SOLP authors opt that the framework shall work wherever the computing and processing takes place. Also in the descriptions about the different users, they include a Resource Provider that shall be active and a part of the Grid environment, thus providing a flexible and dynamic resource allocation and management.
Mobility of licenses

No statements about the mobility of licenses are implied in the article.

Trade

Regulation of resources based on company policies

The SOLP framework allows integration with existing LMF such as FlexLM, if required by the company. Furthermore, on the non-End User site(s) there exist an integration of Service Level Agreements in the Billing and Accounting service module. This provides the ability to specify parameters regarding the license usage, cost, and quality of service parameters as well as other constraints.

Billing and payment

There exists a billing and accounting module in the LMF.

Transparent

Provision of information about resources and their availability

The authors of the LMF state that the SOLP LMF follows the Service Oriented Architecture guidelines for development and design principles. The third bullet point (of the SOA guidelines) stresses the importance of services identification and categorization, delivery, monitoring, and tracking. No interface regarding provision of information about resources and their availability is stated in the LMF.

Transparent licensing mechanism

The authors of the LMF state that the SOLP LMF follows the Service Oriented Architecture guidelines for development and design principles. The third bullet point stresses the importance of services identification and categorization, delivery, monitoring, and tracking. According to the authors, the license service allows the monitoring and control of license usage on the non-end user site(s).

5.1.4. On-demand software license provisioning in Grid and Cloud computing (OSGC)

The license management framework OSGC (our own abbreviation since the framework has no name of its own) was created by a research team, consisting of four scientists with expertise within grid computing, cloud computing, computer security, cryptography, high performance computing, and web applications, in the year of 2010 and further developed in 2013. (Raekow, et al., 2013)

The purpose of the LMF was to solve the major obstacles for the commercial adoption of Grid or cloud infrastructures. They present a complete license management framework that enables a pay-per-use license management, which can be deployed together with an on-demand computing scenario. The LMF enables authenticated access to a remote license server, and can therefore be deployed in any distributed environment. OSGC, according to the authors, allows an easy transition from current software license business models, due to its support for existing software license management tools.
Moreover the provided solution satisfies the following requirements (Raekow, et al., 2013):

- authenticated access
- detailed accounting and billing
- easy access and job submission
- full cost control by the user
- license monitoring

**Technical**

The Grid Services implemented using Web Services

A Web Service is used in OSGC for parsing information, from a database table from the Accounting and Billing component (which is more thoroughly described in section Regulation of resources based on company policies), which gives the users control of their costs (Raekow, et al., 2010).

The License Monitoring Component (which is more thoroughly described in section Flexible and dynamic resource allocation and management) consist of a Web Service which is called the License Monitor Web Service. Its task is to look up the license server and license server type which is available in an XML file that is stored in a MySQL database, and it thereafter contacts the license server in an appropriate way. (Raekow, 2013)

**Security**

In the current OSGC solution the user authentications implies using an SSL/TLS channel between the client and the server (Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are encryption protocols). This channel is a firewall transversal and can in combination with an internal proxy enable security restrictions (EPRI 2004, 2004). Furthermore, due to the use of SSL the client and server connection can be mutually authenticated. (Raekow, et al., 2013)

Raekow, et al. (2013) suggests a more secure way for future versions, which implies a certificate-based authentication mechanism that for instance supports single sign-on. In this solution the users have a certificate and a private key, which can be stored in a secure environment. The authentication process implies a challenge which is computed by the server and transmitted to the client who computes a response based on the challenge and the private key. (Raekow, et al., 2013)

In OSGC authorization depends on whether the license servers are at secure locations. Customers can set up a license server from which multiple clients access the software. The customer can then gain more rights in an unauthorized way by tampering the license server. This can however be prevented by using a TPM-enhanced server platform (TPM is an international cryptographic standard according to ISO (2009)) and a trusted operating system. (Raekow, et al., 2013)
Local application server

Local license services are included in the architecture of OSGC and are present on the customers’ sites. License Monitor Components are used to provide information on how many licenses that are available at a site before submitting a job. This implies involving license servers and the choice of server is done based on configurable policies, priorities and limits. (Raekow, et al., 2013)

Loose coupling and modularization

The information that was found regarding loose coupling and modularization was the following about modularization:

“The license management architecture components (such as LM-job submission interface, LM-authorization web service, LM-proxy, resource management, LM-accounting) provide a complete license management in distributed environments [...].” (Raekow, et al., 2013)

![Diagram of core components of the license management framework](image)

Figure 5-11: This is an overview of the core components of the license management framework. (Raekow, et al., 2010)

Tract

Virtualization and integration of heterogeneous autonomous resources

OSGC is in a distributed environment and supports both Grid and Cloud infrastructure (Raekow, et al., 2010).

The OSGC architecture implies servers at the customer sites that are integrated in the Grid and Cloud can grant usage for the software products of the provider. (Raekow, 2010) This implies support for real organization, which according to Li, et al (2012) is a prerequisite for virtualization and integration of heterogeneous resources.

There are Web Services that request license servers in the Grid, and local license servers can be used for executing jobs. (Raekow, 2012) This implies that resources, services, and applications, can run independently and are not under centralized control.

There is no explicit information regarding the support for virtualization in OSGC, but it supports clouding whose main technology, according to Hamadaqa and Tahvildari (2012), is virtualization.
Flexible and dynamic resource allocation and management

It is, according to Raekow, et al. (2010), important that the supervised scheduling of licenses is efficient, and that the user receives information of the number of licenses that are available on a particular site.

This is done by the License Monitoring Component which consists of aLicense Monitor Web Service and a job scheduler that have been integrated with a license check. When the job scheduler is ready to run a job, which requires licenses, the License Monitor Web Service receives a request which implies that it has to check the responsible license server for available licenses. The procedure can not start until the licenses are available which implies that the job can, in case the licenses are available, start and otherwise it must wait. (Raekow, et al., 2010)

There is no true co-scheduling because this cannot be supported by the client and server license management, but the resources and licenses can be re-scheduled by letting the local resource scheduler poll for licenses until the license server is able to meet the license requirements. There are, however, scheduling strategies which, according to Raekow, et al. (2010), are near-optimal. (Raekow, et al., 2010)

Mobility of licenses

Jobs are submitted to a Grid site through a portal. That can be done from any computer at any place, or even from mobile phones. This implies that mobile and wireless technologies are supported, which is important for mobility of license according to Melarkode, et al. (2004), and jobs can be executed remotely, which emphasizes the mobility of licenses according to Li, et al. (2004).

Trade

Regulation of resources based on company policies

Service Level Agreements exist in OSGC, according to Raekow, et al. (2010), which emphasizes the possibility for regulation of resources based on company policies. The possibilities for using this is not described in the OSGC articles, except for the configuration of policies and priorities for the job scheduler in the Licensing Monitoring Component, which is described in section Flexible and dynamic resource allocation and management.

Billing and payment

The architecture of OSGC enables pay-per-use, according to Raekow, et al. (2013), which for Li, et al. (2004) shall be included in a billing and payment module.

The accounting and billing component of the OSGC gives the users control of their cost due to the license usage. Different license models and different charges might occur depending on the purpose of the software use, and this can be handled by different license accounts. (Raekow, et al., 2010)
The accounting and billing component contains the following database tables (Raekow, et al., 2010):

- Billing account table: Contains user information such as password, username, and contact e-mail.
- License Account Table: Contains license accounts for all users.
- Prices Table: Contains prices for all the licenses.
- Accounting Records Table: Contains information that is needed to compile and complete the accounting record.

**Transparent**

**Provision of information about resources and their availability**

The License Management accounting module is used to monitor the status of licenses (if they are available) and can upon request send this information to services like license schedulers, license brokers or an SLA monitor entity. (Raekow, et al., 2010)

**Transparent licensing mechanism**

The customers can be given an overview of all their license accounts, and their total cost which based on their license usage. (Raekow, et al., 2010)

The user can, as written in the previous section (Provision of information about resources and their availability) receive information regarding how many licenses that are available on a site before submitting a particular job. Both those factors contribute to a transparent licensing mechanism.

**5.1.5. Floating License Sharing System**

The license management framework Floating License Sharing System (FLSS) was introduced in the article Floating License Sharing System in Grid Environment by Feng et al. (2006), and more technical details were described in the articles Research on Software Manager and Sharing System in Grid by Dong et al. (2006) and A Constellation Model for Grid Resource Management by Wang, et al. (2005). The articles were written by three research teams, with expertise within computer science, and the authors were from the School of Electronics and Information Engineering at Xi’an Jiaotong University in China.

FLSS was elaborated as an alternative to the contemporary License Management Systems, which were based on a client and server solution. These systems were suitable in small and medium scale environments (for example within an enterprise) but when more licenses had to be managed a grid system was more suitable. (Dong et al., 2006)

The architecture of FLSS consisted of resource sharing systems which were interconnected via Internet, where every local resource sharing system consists of an Information Service Center, a License Manager, software license resources, and hardware resources. (Feng et al., 2006)
FLSS uses the concept of floating licenses in which the authorization is handled differently. The environment in which the software (that is protected by the license) is executed, requests a number of licenses from a license server. This grants authorization for that number of executions. This solution implies controlling software execution by dynamically tracing the users instead of the copies of software. (Dong et al., 2006)

**Technology**

**The Grid services implemented using Web Services**

The FLSS includes Web Services that use the advantage from both the traditional method and a Web-based interface. (Dong, et al., 2006)

The traditional method implies usage of an X Window (X Window is a way to remotely use an application with a GUI interface) which affects the security (this is explained in the subsequent section, Security), the user-friendliness, and cause poor load balance. In X Window, details regarding resource selection are not hidden from the users which cause the interface to be user-unfriendly. (Dong, et al., 2006)

The poor load balance was due to the fact that the user chose execution environment resource which implied that the most powerful gets taken. Most grid-systems are therefore using Web-based interfaces, where this reimplementation of GUI applications implies a lot of work, and is not adapted to software resources increment in grid. (Dong, et al., 2006)

The FLSS uses benefits from both these methods where the selection of resources is hidden from the user, the current GUI applications are maintained, the interface is user-friendly, and the security aspect is handled.
Security

X window is an established way in which users remotely uses an application that has a GUI interface. This implies logging in to the server using their account and password, and this not safe due to the fact that the user gets the account and password of the server. (Dong, et al., 2006) The GUI used in FLSS is Web-based and avoids the security problem since this logging in is not required in this solution. (Dong, et al., 2006)

Cacciari, et al. (2011), which are the authors of SmartLM LMF, criticizes the security of FLSS since it, due to its focus on maximizing the license and resource usage in the Grid, requires open firewall ports at runtime.

Local application server

FLSS uses local license servers that are placed on the different sites and interconnected to each other over internet or other networks. When a user commits a license usage request through a portal, an Information Service Center is used to verify whether the license can be granted by the local server. In case the local license server can be used it reserves the corresponding number of licenses for the user. In case that is not possible, the local Grid is searched for a suitable license server. When an execution environment has been selected the reserved license server verifies the authorization. (Dong, et al., 2006)

Loose coupling and modularization

The system is divided into several components which can be seen in the figure below from the article Research on Software License Manager and Sharing System in Grid by Feng, et al. (2006). The peer-to-peer overlay network is, according to Wang, et al. (2005) loosely coupled.
The information of the resources can be updated from each place without affecting the overall system, and all Information Service Centers and execution environments can join and leave the system as a whole without affecting its structure. (Dong, et al., 2006)

Due to the design of the Web Services the GUI applications there is no need for them to be reprogrammed when the system is introduced. (Dong, et al., 2006)

**Tract**

**Virtualization and integration of heterogeneous autonomous resources**

The services in FLSS can easily be accessed by the use of standardized interfaces,

“Which make the heterogeneous resources easy to integrate”. (Wang, et al., 2005)

Dong, et al. (2006) describe that the license authorization is not under centralized control but done by a local license server or, if needed, somewhere else in the grid.
Flexible and dynamic resource allocation and management

The Quality of Service of the FLSS is good according to Dong, et al. (2006). Feng, et al. (2006) emphasizes that the objective of the sharing system is to give Quality of Service to its users.

In the FLSS solution, when no application software is available in the hardware environment, automatic and dynamic deployment can be executed. (Dong, et al., 2006)

The FLSS solution implies use of license scheduling policies. The licensing occurs when the users’ request immediately can be satisfied by more than one resource. The policies that are supported are the least policy and the most policy. (Feng, et al., 2006)

The least policy implies that if n licenses are needed when n jobs are supposed to run in parallel then the resource with the least number of licenses will be chosen in order to avoid fragments of license resources. (Feng, et al., 2006)

The most policy implies that only one license is needed regardless of the number of parallel jobs and in order to avoid license resource starvation the resource with the largest number of available resources is chosen. (Feng, et al., 2006)

Licenses can be reserved if the licensing request cannot be satisfied at once. The license execution order regards three policies, where policy one implies that the wait time for job with high priority is minimized. Policy two means that the wait time for jobs with medium priority should be short, and policy three implies that the reservation for jobs with low priority should not affect the resource’s availability. (Feng, et al., 2006)

The scheduling flow of the resources is handled by the Resource Management Service and Scheduling Service module. In case a task is ready for execution in advance the License Service is notified which checks if there is a satisfying license. If there is then it is executed, otherwise the Local manager is notified to perform other tasks. (Feng, et al., 2006) The actual application programs are created by a Job Supervisor Service which also monitors the execution environment. (Feng, et al., 2006)

Mobility of licenses

The mobility of licenses is eased because of the architecture of the FLSS. This is due to the peer-to-peer architecture over internet that FLSS implies (Feng, et al. 2006).

Furthermore, the jobs can be granted either by a local server nearby or remotely in the Grid (Feng, et al. 2006).
Trade

Regulation of resources based on company policies

The FLSS solution regarding the acceptance and reservation of licenses are presented in the section Flexible and dynamic resource allocation and management.

In FLSS rules can be negotiated and determined with a Service level agreement. These rules can differ between different sites of the system, but if two sites have much in common they can cooperate to perform jobs. (Wang, et al., 2005)

Billing and payment

No billing and payment component was presented in the articles by Feng, et al. (2006), Wang, et al. (2005), and Dong, et al. (2006).

Transparent

Provision of information about resources and their availability

Information regarding the organization and management of the resources, and the execution environments are, according to Feng, et al. (2006), stored in the Information Service Center. This information is used when the Information Service Center estimates whether a license request can be satisfied or if the Grid is required (Dong, et al., 2006).

A License Information Management handles information for different types of licenses. It uses a database which stores all necessary information about the license such as usage information, maximum license number, and available license number. The License Information Management offers real-time license information and improves the gathering process of license information. (Feng, et al., 2006)

Transparent licensing mechanism

The ability of the system to use experiences emphasizes its transparency, according to Dong, et al. (2006) and the information available through the License Information Management (which is more thoroughly described in section Provision of information about resources and their availability). The module Job Supervisor Service increases the FLSS transparency since it gathers information of application programs and execution programs. (Feng, et al., 2006)

Moreover, transparent bindings are established between the license server and the execution to which the license server delegates authorization. (Dong, et al., 2006)
6. Analysis

This chapter provides the foundation for the conclusions in the next chapter. In this chapter the LMFs will be merged and compared with the theoretical framework. On an overview perspective, the analysis is performed by first analyzing the individual LMFs. Next, an analysis of all the LMFs combined will be presented in a table. The chapter is finally concluded with a summarizing analysis on the entire chapter.

6.1. SmartLM

Technology

The Grid services implemented using Web Services

The authors state that the core of the SmartLM architecture is to treat software licenses as Web Service resources and thereby provide platform independent access like other virtualized resources and using open standards as far as possible instead of proprietary protocols is considered crucial for the interoperability. This is also evident in the architecture since all the enumerated modules clearly state examples of Web Service implementations.

Security

There is a detailed part in the LMF description that provides the reader with a lot of details regarding several security aspects in the LMF. The security descriptions are more on a framework level than on an implementation level, which probably will facilitate the process regarding implementing the LMF in a real situation.

<table>
<thead>
<tr>
<th>Firewalls</th>
<th>The SmartLM architecture allows license management behind firewalls.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security logging</td>
<td>Security affects all layers in the SmartLM architecture and provides logging at all layers.</td>
</tr>
<tr>
<td>Encryption</td>
<td>All license tokens are encrypted to prevent unauthorized access when traveling towards the computational resource. The encryption mechanism used is not stated.</td>
</tr>
<tr>
<td>Authentication</td>
<td>The description regarding the authentication processes is rigorous and fulfills our framework.</td>
</tr>
<tr>
<td>- Delegation of identity</td>
<td>The Policy Decision Point in the application verifies the correct path of delegations (ISV to License Server to User) to ensure that the license usage is obeyed for the execution.</td>
</tr>
<tr>
<td>- Single sign-on</td>
<td>The license token is only signed once and then forwarded to the (remote) environment where it is executed.</td>
</tr>
<tr>
<td>- Identity mapping</td>
<td>The software token is signed with a local-id that has the credentials for execution in the (remote) environment. Therefore the local-id is mapped in the (remote) environment.</td>
</tr>
<tr>
<td>- Mutual authentication</td>
<td>The local server and (remote) destination servers’ identity is included in the software token. This ensures that the incoming license is intended.</td>
</tr>
<tr>
<td>- Certification</td>
<td>The SmartLM framework uses X.509 certificates for the certification process, which according to Zissis &amp; Lekkas (2013) is secure and trustworthy.</td>
</tr>
</tbody>
</table>

**Authorization**

- The service to gather additional information associated with the user or the actual session
  
  The signed software token contains information regarding the user’s identity and validity of the license.

- The service to gather additional information associated with the protected resource
  
  The authorization process is based on Service Level Agreements, which in the SmartLM case specifies permissions for execution.

- The checking of any local policy applicable to the request
  
  The site administrator is able to define local policies, which for instance could disable certain users (the signed token includes information regarding these aspects).

- The making of an authorization decision based on the identity of the user and the additional information
  
  The Service Level Agreement has specific rights and grants for certain groups or individuals, which is authorized at the (remote) computational environment.
Local application server

The SmartLM LMF states that the authorization for using a license is done locally at the license service of the user’s home organization, and therefore they use a local application server at every user site for autonomous decision making.

Loose coupling and modularization

The architecture of the SmartLM LMF is loosely coupled and modularized, which can be seen in the architectural overview pictures in the License Service and Accounting and Billing Service architecture. An example that enhances this is that the different component parts that for instance comprises the Accounting and Billing Service is separately divided into their component parts with communication links between them. The strong utilization of Web Services within the framework also augments these aspects, since Boden (2004) stresses that Web Services boosts loose coupling.

Tract

Virtualization and integration of heterogeneous autonomous resources

The LMF is made to change the way licensing works and use flexible and non-hardware based licensing solutions that better fit into grid and virtual environments, with a possibility of commercial software licenses in Clouds. The integration of autonomous resources is done locally at the license service of the user’s home organization. The license mechanism is designed for grid environments and works with loosely coupled systems, realizing software licenses as Web Service resources that render licenses as adaptable and mobile. The software token is deployed in the environment where it is going to be executed, an integration of different autonomous resources. All the services are controlled by every site and not centrally.

<table>
<thead>
<tr>
<th>Network linking all points within an enterprise and/or outside enterprise</th>
<th>The LMF integrate three different actors: the User, Resource provider, and the Independent Software Vendor. Since the Resource provider can be the Independent Software Vendor, the user, or an external part the LMF therefore integrates points within and outside the enterprise. The LMF has also a stated requirement for support in Grid and Cloud environments, which augments this side.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use shared services, resources, or have common applications that run independently and not under centralized control</td>
<td>The local application server makes local decisions at the different sites and not under centralized control. The Grid environment prototype shares services and resources in the LMF, which SmartLM provides several examples of.</td>
</tr>
<tr>
<td>Providing services/resources such as servers and large-scale processing</td>
<td>Since the integration of a Resource Provider, the LMF is capable of providing such services as servers and large-scale processing.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Integration of Virtual Organizations</td>
<td>The LMF has a stated support for virtual organizations, where every site is autonomous with its own policy decision making unit to make the site unique. The policies can be defined by the ISV, by the site owning the license, and determined from the membership of the user in a Virtual Organization.</td>
</tr>
<tr>
<td>Support for virtualization</td>
<td>The SmartLM LMF has a stated support for grid and virtual environments, so this aspect should be supported. Another support for this is the shared resources within the Grid environment where the user can execute several services and resources without minding the implementation details.</td>
</tr>
</tbody>
</table>

**Flexible and dynamic resource allocation and management**

The SmartLM LMF support three different actors in its license management system: the user (and her home organization, which is the licensee), the independent software vendor (ISV), which is the licensor, and the application service provider (ASP), which provides the hardware and hosts the application. The ASP and ISV could be the same actor. This is one condition that makes the flexible and dynamic resource allocation and management feasible since there are several actors involved in the network where the locations for the computations can differ.

Furthermore, the Orchestrator provides automated mechanisms to support the user in co-allocating computational resources and the licenses both in time and space, allowing to use remote Grid resources for the execution of an application which needs a software license at run time, while being sure that the license required is available at the remote site when the application will start-up and during execution.
| Having a resource access framework which applications can call for allocation of resources | The local application server and orchestration unit provides the capability for the user to access and flexibly and dynamically allocate and manage resources. |
| Resources are used by the requesting application and then subsequently released | A built license-scheduler provides the feature of allocating resources and subsequently releases them. |
| Optimize the use of spare resources | Not mentioned, but the orchestration unit provides the capabilities of optimizing the use of spare resources since it also has the capability of co-allocating computational resources. |
| Co-allocation of different resources | The orchestrator unit provides this capability of co-allocating computational resources. |
| Dynamicity of licenses; suspending, preempt, and resume the license usage | The Quality of Services and Service Level Agreement implementations in the SmartLM framework provides the capabilities to “e.g., availability of a certain license and corresponding features at a given time and for certain computing resources” (Cacciari, et al., 2012) This and the Policy decision making unit should make it possible to integrate those aspects in the LMF. |
| Support for workflows | The orchestration unit provides the capability to retrieve and reserve the right licenses for the applications in advance and other functionalities associated with workflows. |

**Mobility of licenses**

The authors of the SmartLM LMF claim that the LMF deliver the required flexibility and mobility in the licensing process through the software tokens. Decoupling authorization for license usage from authorization for application execution allows executing the license protected application, without requiring a permanent bidirectional network connection to the license server at runtime controlling the authorization for application execution. This increases the mobility since the solution can be used in different computational environments. This will therefore ensure that the licenses can be used on different machines and environments and will therefore fulfill the mobility requirements.
Remote license enforcement

The local application server and the other mechanisms involved in the SmartLM ecology provide the capability to execute licenses remotely, hence fulfilling this criterion. The main principle of the LMF is that licenses become mobile objects, which may travel to the environment where they authorize the execution of a license protected application.

The license can be moved from one computing entity to another

The authors of the framework state that licenses can be moved between computing entities, but do not explain any details. Probably it will be handled in the local application server.

Trade

Regulation of resources based on company policies

The SmartLM project’s idea for successful licensing is to achieve software licensing based on business objectives that balance customer needs and vendor business models. The authorization for usage of a specific software component takes into account availability of the requested license and features, policies defined by the ISV and policies defined by the local administrator. All local authorizations for license usage are expressed and guaranteed by Service Level Agreements between the user and license service.

On an architectural level, the Policy Management Service, within the License Service architectures Management Layer, is the central entity for evaluating policies that define the access to and the usage rights of a certain license or license feature for a user or a group of users. The policies can be defined by the ISV, defined locally by the site owning the license, and determined from the membership of the user in a Virtual Organization.

Furthermore the SmartLM LMF has an EBS interface and Budget Information Services (provides to the License Service the consumed budget for a specific end user) that makes the system able to obligate the user’s requirements.
Enabling resource consumers’ and owners’ requirements and realization of their goals

Both parties’ requirements and realization of goals are fulfilled. Because the LMF states that the three actors involved in the licensing process can express their requirements. The different components augment the ability regarding realizing different goals (e.g., Policy Management Service).

Utility models for how consumers demand resources and their preferred parameters

The Electronic Billing System and Budget Information Services provides the user with options to set his or hers requirements, which will facilitate the usage.

Billing and payment

The SmartLM LMF integrates an Accounting and Billing module that integrates the various aspects required from a commercial perspective.

| Grids enable payment models such as “pay-as-you-go” and other on demand models | The SmartLM LMF enables models such as pay-per-use. |
| Models for establishing the value of resources | The Accounting and Billing Service has a price information module that should be able to display this information. |
| Resource pricing schemes and publishing mechanisms | Not mentioned, but should be possible to integrate in the user interfaces and be available. |
| Economic models and negotiation protocols | The SmartLM LMF enables models such as pay-per-use within the Price Information Module where the Economic models are stored. |
| Mediators to act as a regulatory agency for establishing resource value, currency standards, and crisis handling | Not mentioned, but could be integrated in a module within the Accounting and Billing Service. |
Accounting, billing and Payment Mechanisms

Yes, see the Accounting and Billing Service. It has a module called the Electronic Billing Service (EBS) that handles those requests.

Transparent

Provision of information about resources and their availability

The SmartLM LMF provides the user with a License Information Service that is a point of aggregation to collect pieces of information from different sources that can be viewed and shared for different authorized user groups. Thus, providing transparent information regarding provision of information about resources and their availability.

Transparent licensing mechanism

The Accounting and Billing Service is realized as an additional feature that allows ISV, ASP or end users to have global control and overview of their license usage, which brings the possibility of detailed statistics based on license resource usage. This is provided to the final user as graphs, tables, or printable files. Furthermore, the Accounting and Billing Service can enable the end-user to have a full cost control. This brings a good control over the business perspective in the SmartLM LMF.

The license information service within the license service architecture allows both users and administrators to retrieve information on installed ISV licenses, licenses available, reserved or in use. All the information necessary for the end-user can be accessed through the Web-GUI. This provides all the user’s eligible for the information mechanism to enter and have a control over the entities he or she wishes to have.

For interaction with an administrator the Admin Service is used. On the one hand this service provides a set of operations, which allow the administrator to manage the set of handled licenses, for instance add new licenses or remove existing licenses. The Usage Record Consumer shows the amount of spent resources from the user’s point of view and, hence, increases the transparency of the system.

With these three points we can conclude that the transparency within the SmartLM LMF is sufficient both in terms of the end-user and other parties in the Grid environment.

6.1.1. Summary

The SmartLM LMF basically fulfills all of the points mentioned in the Evaluation Model. The architecture is very detailed and provides the user with a lot of aspects that can be directly used for implementation of the LMF. According to the framework, no specific drawbacks or uncertainties can be found. Some aspects, such as which encryption should be used, are not mentioned but this requirement can vary between different implementations.
6.2. GenLM

Technology

The Grid services implemented using Web Services

For communication between the different modules, the GenLM LMF uses Apache ActiveMQ which supports Web Services protocols and other architectures. Nothing about Web Services is stated in plain text, so the utilization of Web Services is therefore unknown but probably used because of the specified use of internet as a medium for transportation of messages.

Security

The explanation of the security used in GenLM is rigorous and many aspects are included on a detailed level about techniques and processes, a summary in relation to our framework is stated in the table below:

<table>
<thead>
<tr>
<th>Firewalls</th>
<th>The LMF is designed to be used behind firewalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security logging</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Encryption</td>
<td>Utilization of SHA-256 cryptographic hash algorithm, which is secure and reliable</td>
</tr>
<tr>
<td>Authentication</td>
<td>Seems to be complete in every aspect beside the delegation of identity</td>
</tr>
<tr>
<td>- Delegation of identity</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Single sign-on</td>
<td>The GenLM client puts the user’s identity within the software token and this token is used in the whole chain, thus single sign-on</td>
</tr>
<tr>
<td>- Identity mapping</td>
<td>A customer database can be used at all stages for identity mapping</td>
</tr>
<tr>
<td>- Mutual authentication</td>
<td>In the license token, sent between the different stages in the LMF, the server’s identity is stored so it can be authenticated</td>
</tr>
<tr>
<td>- Certification</td>
<td>Utilization of certificate standard X.509, which is secure and trustworthy</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Authorization</td>
<td>Information regarding the authorization process is missing, most of the information has to be taken out of its context and then applied in this analysis</td>
</tr>
<tr>
<td>- The service to gather additional information associated with the user or the actual session</td>
<td>The user’s identity in the license token can be matched against a user database where additional information can be stored</td>
</tr>
<tr>
<td>- The service to gather additional information associated with the protected resource</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- The checking of any local policy applicable to the request</td>
<td>There is a policy engine on the Independent Service Vendors site that enforces the specified business model. However, any existence of any at the Resource provider’s or User’s site is not mentioned.</td>
</tr>
<tr>
<td>- The making of an authorization decision based on the identity of the user and the additional information</td>
<td>The software token contains information regarding the User’s identity and the input job; this is used to make the authorization decisions.</td>
</tr>
</tbody>
</table>

**Local application server**

There is a local application server placed at the User’s site for this purpose.

**Loose coupling and modularization**

The authors of the LMF state that they use a loosely coupled approach for the overall architecture, but no usage of modularization is claimed. No further details in pictures etc. can confirm these aspects. The Policy plugin manages some parts of the Accounting and Billing process, which could be interpreted as that some parts could be more modularized, but with regard to the vague descriptions of the architectural properties: no real conclusions can be drawn.
**Tract**

**Virtualization and integration of heterogeneous autonomous resources**

The authors of the LMF claim that they integrate actors in a Grid and Cloud environment, the Users, the Resource Providers, and the Independent Service Vendor. The general description of the licensing process is augmenting the virtualization and integration of heterogeneous autonomous resources aspects of the framework.

<table>
<thead>
<tr>
<th>Network linking all points within an enterprise and/or outside enterprise</th>
<th>No information regarding the internal network is mentioned. However, the authors state that the Resource provider could be the User, an external Resource Provider, or the Independent Software Vendor. This means that the frame is able to link all points within an enterprise and/or outside the enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use shared services, resources, or have common applications that run independently and not under centralized control</td>
<td>The integration of the different roles specified in the LMF augments this capability. The local application server make the specific site not under centralized control</td>
</tr>
<tr>
<td>Providing services/resources such as servers and large-scale processing</td>
<td>The Resource Provider role can provide the User with these capabilities such as servers and large-scale processing</td>
</tr>
<tr>
<td>Integration of Virtual Organizations</td>
<td>The authors claim that the framework is solving the issue of integrating virtual environments. The stated support for flexible, secure, coordinated resource sharing confirms these capabilities</td>
</tr>
<tr>
<td>Support for virtualization</td>
<td>The authors claim that GenLM support virtualized environments and virtualization. Any concrete proof is not provided</td>
</tr>
</tbody>
</table>
Flexible and dynamic resource allocation and management

The GenLM framework has the requirement of being able to do computations regardless of the execution location of the computation.

| Having a resource access framework which applications can call for allocation of resources | Not mentioned |
| Resources are used by the requesting application and then subsequently released | Not mentioned |
| Optimize the use of spare resources | Not mentioned |
| Co-allocation of different resources | Not mentioned |
| Dynamicity of licenses; suspending, preempt, and resume the license usage | Not mentioned |
| Support for workflows | Not mentioned |

Mobility of licenses

In text, the authors claim that the license should hold regardless of the execution location of the computations.

| Remote license enforcement | At the Independent Service Vendors site, the GenLM server checks the validity of the pending job. If this check is successful, then a process like the remote license enforcement is conducted that guarantees the validity of the license when the files travel to the Resource Providers site. |
| The license can be moved from one computing entity to another | Not mentioned, only information regarding the pending job and the user’s identity is extracted into the software token at the User’s site. This could possible mean that as long as the information regarding the User’s identity is the same, the computing entity would not matter. |
**Trade**

**Regulation of resources based on company policies**

The GenLM LMF elicited a requirement for “Support for existing license agreements”. There is also a stated support for integration with different software packages, depending on the vendor’s preferences. All the license terms are stored in a Policy plugin that is integrated in the GenLM server, residing at the Independent Software Vendors site. The purpose of the plugin is to obligate the business model chosen by the Independent Service Vendor.

| Enabling resource consumers’ and owners’ requirements and realization of their goals | Only the Independent Service Vendor has the ability to elicit requirements in the LMF. |
| Utility models for how consumers demand resources and their preferred parameters | Not mentioned |

**Billing and payment**

The GenLM LMF has no stated Accounting and Billing module. However, the authors state that all the necessary steps in order to be able to bill the customer will be made in the policy plugin, which might involve putting a billing record in a database. This policy plugin therefore facilitates the Accounting and Billing process.

| Grids enable payment models such as “pay-as-you-go” and other on demand models | The LMF states a support for “pay-per-use” or “pay-as-you-go” payment models. |
| Models for establishing the value of resources | Not mentioned |
| Resource pricing schemes and publishing mechanisms | Not mentioned |
| Economic models and negotiation protocols | Not mentioned, but the economic models such as “pay-per-use” have a stated support |
| Mediators to act as a regulatory agency for establishing resource value, currency standards, and crisis handling | Not mentioned |
| Accounting, billing and Payment Mechanisms | Not included in the LMF, but the Policy plugin facilitate and do parts of this job. |
**Transparent**

**Provision of information about resources and their availability**

Any aspects regarding provision of information about resources and their availability is not stated in the LMF.

**Transparent licensing mechanism**

No mechanism about enhancing transparent licensing mechanism was found in the GenLM LMF, even though the authors states that:

> “Users expect the licensing mechanism to be transparent. Problems with the licensing mechanism will yield frustration on the user’s side and must be avoided.”

(Dalheimer & Pfreundt, 2009)

How the authors provide the Users of the LMF with transparent mechanisms is unknown.

**6.2.1. Summary**

Various aspects in our evaluation model were mentioned in the description of the LMF, but lacks detailed descriptions on how it is implemented and utilized in a real implementation of the system. Statements such as “The license mechanism must be transparent” are frequently used but, thus, not described further. Furthermore, there are vague descriptions regarding the utilization of Web Services, Loose coupling and Modularization, the IT-architectural properties, flexible and dynamic resource allocation and management, etc. This will, thus, make the LMF harder to implement and many of these aspects has to be recreated by an individual that want to use this framework in the enterprise’s business operations.

The LMF stresses the importance and details regarding the security mechanism in the Evaluation Model, which is evident in the analysis where almost every point can be expressed in terms of our framework.

The framework does not integrate a Billing and Payment module, but claims that the policy plugin can handle any of these aspects and put the invoices directly into a database, if desired. This plugin is only available for the Independent Service Vendor and not the other users in the same network. It does not exist a separated Billing and Payment module for managing the invoices, which could be a sign of that the modularization of the IT-architecture could be revised. The lack of details in these aspects inhibits us to draw any strict conclusions regarding this framework.
6.3. Service Oriented License Providing (SOLP)

Technology

The Grid services implemented using Web Services

As mentioned earlier, the SOLP framework is based on the guiding principles of Service Oriented Architectures, the one that adheres to this point the most is Standards compliance (both common and industry specific). This is apparent in the SOLP architecture, due to that all the components is based on common Web Service standards. One missing fact on this point is that there is little information regarding what Web Service components and standards that are used in the article, this implies that the ones that want to use this LMF will have to analyze what component to be used in this context.

Security

The authors of the LMF states that there is an authentication and authorization process included in the LMF, but no more details are presented regarding this. Furthermore, the LMF also uses a certificate when sending information from the Client to the Provider. There is hence no information regarding Firewalls, security logging, encryptions or further details about the authentication, and authorization. When implementing this LMF, the utilizer will therefore have to take those security module decisions.

<table>
<thead>
<tr>
<th>Firewalls</th>
<th>Not mentioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security logging</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Encryption</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Authentication</td>
<td>No details revealed except that there is a authentication process</td>
</tr>
<tr>
<td>- Delegation of identity</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Single sign-on</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Identity mapping</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Mutual authentication</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Certification</td>
<td>Mentioned, but not what kind of technique that is utilized</td>
</tr>
<tr>
<td><strong>Authorization</strong></td>
<td>Aspects regarding the authorization request has more details, except the gathering of additional information from a service with regard to a protected resource</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>- The service to gather additional information associated with the user or the actual session</td>
<td>The Billing and Payment and License Service module verifies if the user is eligible for acquiring the specific license</td>
</tr>
<tr>
<td>- The service to gather additional information associated with the protected resource</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- The checking of any local policy applicable to the request</td>
<td>The License Service checks the local policies applicable by checking the Service Level Agreement</td>
</tr>
<tr>
<td>- The making of an authorization decision based on the identity of the user and the additional information</td>
<td>The License Service validates the request against the User’s Account</td>
</tr>
</tbody>
</table>

**Local application server**

There is no stated Local application server in the LMF. Although, the authors states that the modularity and componentization of the LMF allows the implementation of various models and license scheme, which could imply that a Local application server could be used, with some modifications of the LMF.

**Loose coupling and modularization**

In the Service Oriented License Providing Component Diagram you can see that there are different modules and a level of componentization. The link between the different modules are based on Web Services, which according to Boden (2004) are examples of loose coupling.

**Tract**

**Virtualization and integration of heterogeneous autonomous resources**

The LMF states that it supports three different kind of user’s in its specification: The Independent Software Vendor, Resource Provider, and End Users. This could be seen as an integration of heterogeneous autonomous resources. Furthermore, the LMF also claims that it supports Grid and Cloud environments. Moreover, the authors claim to integrate the Resource provider from a Grid
environmental perspective, which augments the argument that the LMF support virtualization and integration of heterogeneous autonomous resources.

<table>
<thead>
<tr>
<th>Network linking all points within an enterprise and/or outside enterprise</th>
<th>Because of the property that a Resource Provider can be either an independent actor, the User, or the Independent Service vendor means that the LMF is capable of linking all points within and outside an enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use shared services, resources, or have common applications that run independently and not under centralized control</td>
<td>Due to the fact that no local application server is provided and that the License server totally controls license requests, the LMF supports a centralized control of these requests</td>
</tr>
<tr>
<td>Providing services/resources such as servers and large-scale processing</td>
<td>Since the integration of a Resource Provider in the LMF, this actor could therefore provide the service of servers and large-scale processing</td>
</tr>
<tr>
<td>Integration of Virtual Organizations</td>
<td>Not mentioned, but probably due to the fact that they integrate the three main actors in a Grid environment and state the support for Grid environments. This should imply integration of virtual organization.</td>
</tr>
<tr>
<td>Support for virtualization</td>
<td>Not mentioned</td>
</tr>
</tbody>
</table>

**Flexible and dynamic resource allocation and management**

The SOLP authors opt that the framework shall work wherever the computing and processing takes place. Also in the descriptions about the different users, the authors of the LMF include a Resource Provider that shall be active and a part of the Grid environment, thus providing a flexible and dynamic resource allocation and management.

| Having a resource access framework which applications can call for allocation of resources | The Service Client provided at the User’s site is doing the resource allocation and provides the ability to allocation to different sites. |
Resources are used by the requesting application and then subsequently released | Not mentioned
---|---
Optimize the use of spare resources | Not mentioned
Co-allocation of different resources | Not mentioned
Dynamicity of licenses; suspending, preempt, and resume the license usage | Not mentioned
Support for workflows | Not mentioned, but the License Service is responsible for this tasks. No specific example is raised when there are workflows involved in the license process.

Mobility of licenses

No information is given regarding the mobility of licenses in the framework. Thus, it should be able to achieve this through that the service client should be able to handle different user and be able to be installed on different machines. This is an aspect that could be regulated in the Service Level Agreement, but on a technical level it shall be feasible.

Remote license enforcement

<table>
<thead>
<tr>
<th>The License Service provides the information of Remote license enforcement and manages these aspects.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The license can be moved from one computing entity to another</td>
</tr>
<tr>
<td>Not mentioned, even though it should be able to do if you can sign-in on the other entity with the same information. Can, thus, be regulated in the Service Level Agreement between the different parties.</td>
</tr>
</tbody>
</table>

Trade

Regulation of resources based on company policies

The SOLP framework allows integration with existing LMF such as FlexLM, if required by the company. Furthermore, on the non-End User site(s) there exist an integration of Service Level Agreements in the Billing and Accounting service module. This provides the ability to specify parameters regarding the
license usage, cost, and quality of service parameters as well as other constraints. However, the User cannot specify any regulation of resources based on company policies.

<table>
<thead>
<tr>
<th>Enabling resource consumers’ and owners’ requirements and realization of their goals</th>
<th>Only the Owner can specify policies on his side. However, the parties can negotiate the terms in the Service Level Agreements, but the User cannot specify any terms directly in the system.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility models for how consumers demand resources and their preferred parameters</td>
<td>Not mentioned.</td>
</tr>
</tbody>
</table>

**Billing and payment**

There is a Billing and payment module in the LMF, no further details regarding its commercial aspects are provided.

<table>
<thead>
<tr>
<th>Grids enable payment models such as “pay-as-you-go” and other on demand models</th>
<th>The LMF states that it can implement Grid enabled models such as “pay-as-you-go”.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models for establishing the value of resources</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Resource pricing schemes and publishing mechanisms</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Economic models and negotiation protocols</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Mediators to act as a regulatory agency for establishing resource value, currency standards, and crisis handling</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Accounting, billing and Payment Mechanisms</td>
<td>The Billing and payment module is responsible for billing the customer, no further information is provided.</td>
</tr>
</tbody>
</table>
Transparent licensing mechanism

The authors of the report state that the SOLP LMF follows the Service Oriented Architecture guidelines for development and design principles. The third bullet point stresses the importance of services identification and categorization, delivery, monitoring, and tracking. According to the authors, the license service allows the monitoring and control of license usage on the non-end user site(s). However, no way to leverage the transparent license mechanism from the end user’s point of view is, thus, not mentioned and provided.

6.3.1. Summary

Various aspects in our evaluation model were mentioned in the description of the LMF, but lacks detailed descriptions on how it is implemented and utilized in a real implementation of the system. Statements such as “Comprise Billing and Payment module” are frequently used but, thus, not described further. Furthermore, there are vague descriptions regarding the utilization of Web Services, Loose coupling and Modularization, the IT-architectural properties, flexible and dynamic resource allocation and management, etc. This will, thus, make the LMF harder to implement and many of these aspects has to be recreated by an individual that want to use this framework in the enterprise’s business operations.

The framework gives a general overview of the whole process and gives very little in-depth details regarding the mechanisms that manage the different details of the process. An example of this is the Billing and Payment modules that are mainly described as checking if the user is eligible for the request and manages the billing aspects, but no further details are given. Many aspects that the framework should be able to provide, such as mobility of licenses is not described but should be technically feasible according to our analysis. Almost no details are given regarding the security aspects in the framework, and the information that we acquired in this analysis has to be taken out of various steps that vaguely included some security aspects such as authorization and authentication. The LMF does not seem to have any specialization that makes it specifically good in any aspect (e.g., security).
6.4. On-demand software license provisioning in Grid and Cloud computing (OSGC)

Technology

The Grid services implemented using Web Services

Web Services are a part of the OSGC solution and are both used for providing the customer with cost information, by parsing information to the Accounting and Billing component, and check for available licenses for the License Monitoring Component.

Security

Many security aspects have been covered in OSGC. For instance firewalls and a cryptographic standard can be used in the LMF.

<table>
<thead>
<tr>
<th>Firewalls</th>
<th>Firewalls can be used due to the encryption protocol channel between the client and server in the OSGC architecture.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security logging</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Encryption</td>
<td>OSGC uses the international cryptographic standard TPM.</td>
</tr>
<tr>
<td>Authentication</td>
<td>The certificate-based authentication mechanism and the encryption protocol channel enhance the authentication security, but some aspects are however missing.</td>
</tr>
<tr>
<td>- Delegation of identity</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Single sign-on</td>
<td>This is supported in the future version which implies a certificate-based authentication mechanism.</td>
</tr>
<tr>
<td>- Identity mapping</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Mutual authentication</td>
<td>Mutual authentication is supported due to the use of an encryption protocol channel between the client and the server.</td>
</tr>
<tr>
<td>- Certification</td>
<td>The future versions imply a certificate-based authentication mechanism.</td>
</tr>
<tr>
<td>Authorization</td>
<td>It is difficult to discuss the authorization aspects due to lack of information.</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>- The service to gather additional information associated with the user or the actual session</td>
<td>Not mentioned, but the authors state that users can retrieve information regarding their usage which due to the encryption protocol channel between the server and the client should be secure.</td>
</tr>
<tr>
<td>- The service to gather additional information associated with the protected resource</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- The checking of any local policy applicable to the request</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- The making of an authorization decision based on the identity of the user and the additional information</td>
<td>Not mentioned</td>
</tr>
</tbody>
</table>

**Local application server**

There are local license servers on all the customers’ sites in the OSGC architecture which implies decentralized control where these license servers can locally grant usage.

**Loose coupling and modularization**

The OSGC architecture consist of several components with different purposes and it is hard to tell that these components should be divided into smaller components with even more specific purposes. This gives a hint of that OSGC is decently modularized.

The framework does not explicitly tell whether the components of OSGC are loosely coupled. However, in figure 6-1, the components and their relations to each other are shown and Web Services are often used between components, which according to Boden (2004) boosts loose coupling.
Tract

Virtualization and integration of heterogeneous autonomous resources

OSGC seems to have prerequisites for virtualization and integration of autonomous resources, but that is hard to determine due to the available framework information.

<table>
<thead>
<tr>
<th>Network linking all points within an enterprise and/or outside enterprise</th>
<th>Integration of points within an enterprise is required. This is done between the customer and the Independent Software Vendor (in order to for example provide the customer with their usage cost or allow them to use their licenses). OSGC supports Cloud and Grid which also enhances the Network linking.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use shared services, resources, or have common applications that run independently and not under centralized control</td>
<td>Services, resources, and common applications can run independently and are not under centralized control in the OSGC architecture.</td>
</tr>
<tr>
<td>Providing services/resources such as servers and large-scale processing</td>
<td>The system supports the use of local servers and web services. Nothing explicit is said about large-scale processing so it is hard to determine.</td>
</tr>
</tbody>
</table>
### Integration of Virtual Organizations
Not mentioned

### Support for virtualization
Not mentioned, but it is the main technology for clouding which is supported in OSGC.

---

**Flexible and dynamic resource allocation and management**

The use of grid and cloud should imply flexible and dynamic resource allocation and management but further information is needed in order to clarify all the aspects presented in the table below.

<table>
<thead>
<tr>
<th>Having a resource access framework which applications can call for allocation of resources</th>
<th>The job scheduler determines where a job shall be executed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources are used by the requesting application and then subsequently released</td>
<td>Not mentioned, but the jobs can be executed independently at different license servers and it is therefore likely that the resources are subsequently released.</td>
</tr>
<tr>
<td>Optimize the use of spare resources</td>
<td>A strategy is used to optimize the scheduling for the resource usage. It seems however inefficient for the local resource scheduler to pull for licenses until the license server is ready.</td>
</tr>
<tr>
<td>Co-allocation of different resources</td>
<td>Co-scheduling is supported but it requires that the local resource server pulls for licenses.</td>
</tr>
<tr>
<td>Dynamicity of licenses; suspending, preempt, and resume the license usage</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Support for workflows</td>
<td>Not mentioned</td>
</tr>
</tbody>
</table>
Mobility of licenses

The licenses in OSGC are mobile in the aspect of where the customer can be located. It is not explained if the licenses can be moved between computing entities.

<table>
<thead>
<tr>
<th>Remote license enforcement</th>
<th>The computers or mobile devices used for the submitting can be located anywhere.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The license can be moved from one computing entity to another</td>
<td>Not mentioned.</td>
</tr>
</tbody>
</table>
Resource pricing schemes and publishing mechanisms

Prices are stored in the Prices table in the Accounting and Billing component and can handle different pricing models.

Economic models and negotiation protocols

There are models to calculate the cost for the end user. No negotiation protocols are however mentioned.

Mediators to act as a regulatory agency for establishing resource value, currency standards, and crisis handling

Not mentioned

Accounting, billing and Payment Mechanisms

There is a complete Accounting and Billing component in OSGC.

**Transparent**

Provision of information about resources and their availability

There is information regarding the number of licenses that are available for a site’s license server before a job is submitted. This was the information sharing found regarding resources and their availability, but it is possible that more information could be shared through the same interface.

**Transparent licensing mechanism**

There is transparency in the OSGC licensing mechanism. The licensing usage and cost is visible for the user. This should make it easier for both providers and customers to trust the system since the correctness of the cost can be checked.

There is transparency regarding the information of resources and their availability, which should give the provider better control of the system.

**6.4.1. Summary**

The technical aspects of OSGC are regarded where Web Services and local license servers are included in the architecture and most security aspects, including support for firewalls and encryption, are fulfilled. However the aspects of modularization and coupling are not explicitly written in the framework articles but seem to be satisfied considering the illustrations of the architecture.

The tract criteria were partly satisfied but more information would be required for them to be determined.
When it comes to trade it seems satisfying regarding Billing and payment. Even though more information would be useful regarding what can be achieved using the SLA, it is likely that the criterion Regulation of resources based on company policies is satisfied.

OSGC is regarded as transparent considering that it provides the users with information of their usage and cost. Furthermore, there is also information regarding the availability of the resources.

All in all, more information would have been necessary in order to draw strict conclusions regarding the OSGC LMF.

6.5 Floating License Sharing System (FLSS)

Technology

The Grid services implemented using Web Services

Web Services are included in the FLSS architecture which uses the benefits from the traditional method and Web-based interface. This implies that resource selection is hidden from the user, the current interface of the GUI applications do not need to be rewritten in order to work with the Web Service, the interface is user-friendly, and the security aspect is handled. This seems like good use of web services but it would however been useful with more details regarding where and how these Web Services shall be used. Also, it would have been good with clarification regarding in which way the Web Services of FLSS are secure.

Security

There is not much information available regarding the security of FLSS. It is mentioned how the security flaw of X window is avoided but nothing is actually said about how the security is handled. Nothing is said about firewalls or encryption. It does not mean that these security aspects cannot be included in FLSS but it is alarming that they are not mentioned. Furthermore, Cacciari, et al. (2011) criticizes the security of FLSS since it requires open firewall ports at runtime.

<table>
<thead>
<tr>
<th></th>
<th>Not mentioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewalls</td>
<td></td>
</tr>
<tr>
<td>Security logging</td>
<td></td>
</tr>
<tr>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td></td>
</tr>
<tr>
<td>- Delegation of identity</td>
<td></td>
</tr>
<tr>
<td>Service Description</td>
<td>FLSS Status</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>- Single sign-on</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Identity mapping</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Mutual authentication</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- Certification</td>
<td>Not mentioned</td>
</tr>
<tr>
<td><strong>Authorization</strong></td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- The service to gather additional information associated with the user or the actual session</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- The service to gather additional information associated with the protected resource</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- The checking of any local policy applicable to the request</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>- The making of an authorization decision based on the identity of the user and the additional information</td>
<td>Not mentioned</td>
</tr>
</tbody>
</table>

**Local application server**

In the FLSS architecture there are local servers on the different sites. This implies that all control is not centralized, since license verification can be performed locally which is the point with a local application server.

**Loose coupling and modularization**

FLSS consist of various components divided to serve different purposes which enhance the perception that FLSS is modularized.

The peer-to-peer overlay network is loosely coupled. The perception of the system to be loosely coupled is strengthening by the fact that Information Service Centers and execution environments can join and leave the network. Furthermore, the interfaces of the GUI applications do not need to be reprogrammed when integrated with Web Services. The use of Web Services boosts the loose coupling, according to Boden (2004).
The FLSS solution hence seems to be both loosely coupled and modularized.

**Tract**

**Virtualization and integration of heterogeneous autonomous resources**

The heterogeneous resources of FLSS are easy to integrate but nothing explicitly is said about virtualization. This is hard to determine but it is possible that virtualization can be used in FLSS.

<table>
<thead>
<tr>
<th>Network linking all points within an enterprise and/or outside enterprise</th>
<th>Not mentioned, but the vendors and customers must somehow be integrated for the customer to use the protected software.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use shared services, resources, or have common applications that run independently and not under centralized control</td>
<td>Independence of centralized control is obvious due to the use of local license servers. This implies that license request can be managed and the resource control is handled locally.</td>
</tr>
<tr>
<td>Providing services/resources such as servers and large-scale processing</td>
<td>License servers are provided. Large-scale processing is, however, not mentioned.</td>
</tr>
<tr>
<td>Integration of Virtual Organizations</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Support for virtualization</td>
<td>Not mentioned</td>
</tr>
</tbody>
</table>

**Flexible and dynamic resource allocation and management**

There are algorithms for performing jobs and handling queues for this. However, information is missing regarding support for workflows, dynamicity of licenses and support for workflows, but it is not unlikely that these are satisfied.

<table>
<thead>
<tr>
<th>Having a resource access framework which applications can call for allocation of resources</th>
<th>This is performed by the Job Supervisor Service.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources are used by the requesting application and then subsequently released</td>
<td>The scheduling policies include job priorities which affects the request of resources. It is likely that the resources are subsequently released, so more jobs can be executed.</td>
</tr>
</tbody>
</table>
Optimize the use of spare resources

Both the most policy and the least policy are used in order to use the resources in an efficient way. Furthermore, there are also policies for reserving licenses.

Co-allocation of different resources

Sites with similar rules cooperate to perform jobs.

Dynamicity of licenses; suspending, preempt, and resume the license usage

Not mentioned

Support for workflows

The scheduling workflow is handled by the Resource Management Service & Scheduling Service module.

**Mobility of licenses**

There are prerequisites, due to the wireless technologies and remote license enforcement, for mobility of licenses but it is specifically written whether licenses can be moved between computing entities.

Remote license enforcement

Jobs can be remotely granted depending on whether a local server is available or not.

The license can be moved from one computing entity to another

Not mentioned

**Trade**

**Regulation of resources based on company policies**

There is not much information available regarding Regulation of resources on company policies. It is due to the use of SLA these bullets can be negotiated and then determined.

Enabling resource consumers’ and owners’ requirements and realization of their goals

Not mentioned, but it should be possible to negotiate goals using the SLA in FLSS which thereby are established.
Utility models for how consumers demand resources and their preferred parameters | Not mentioned, but this should be possible since SLAs are used in FLSS.
---|---

**Billing and payment**

No billing and payment component was present. It does not imply that it cannot be supported but it is alarming that it is not integrated in the FLSS.

<table>
<thead>
<tr>
<th>Grids enable payment models such as “pay-as-you-go” and other on demand models</th>
<th>Not mentioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models for establishing the value of resources</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Resource pricing schemes and publishing mechanisms</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Economic models and negotiation protocols</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Mediators to act as a regulatory agency for establishing resource value, currency standards, and crisis handling</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>Accounting, billing and Payment Mechanisms</td>
<td>Not mentioned</td>
</tr>
</tbody>
</table>

**Transparent**

**Provision of information about resources and their availability**

Information can be gathered from the Information Service Center when checking whether a local license server can be used or if the Grid has to be used. Furthermore, the License Information Management offers real-time license information and improves the gathering process of license information. All in all there is information about resources and their availability.

**Transparent licensing mechanism**

The licensing mechanism is transparent according to Dong, et al. (2006) and it seems possible regarding the available information that was presented in the previous section. However, it would have been useful with transparency regarding the customer usage and payment which is not described. It is logical that this transparency has not been described since no billing and payment have been presented for FLSS.
6.5.1. Summary

FLSS uses decentralized control whereas local license servers are used. Also the system appears to be modularized and loosely coupled, and includes Web Services, which increases its flexibility.

It is disturbing that the security aspect barely has been covered in the LMF, since it is important to prevent unauthorized usage. Furthermore, a transparent billing and payment model would have been desirable which supports different pricing models. Security aspects and the support for a billing and payment component should be investigated before considering implementing FLSS. Furthermore, it would have been useful to know the support for virtualization.

All in all, more information would have been necessary in order to draw strict conclusions regarding the FLSS.

6.6. The LMF comparison

The LMF comparison have, in order to be more perspicuous, been summarized in table X, and every aspect of each criteria have been shortly commented in this table. These comments categorized the information available for the aspects, which hence, could be everything from fully utilized to not mentioned (specified as NM* in the table). The different comments are, though, explained:

- Fully utilized: The LMF information clarifies that the aspect or criterion is fully satisfied.
- Utilized: The LMF information clarifies that the aspect or criterion is satisfied.
- Not mentioned (NM*): Implies that not enough information have been presented to determine whether the criterion or aspect is satisfied.

There are more varieties of comments but these are more straightforward and do not require any further explanations. It is our perception of the LMF information that determines the categorization and the difference between for example utilized and fully utilized might be subtle but it is nevertheless worth to make this distinguish in order to give a comparable view of the LMFs.

The comparison table followed the structure of the Evaluation Model and was in the left column categorized after the four categories (technology, tract, trade, and transparency). In the second column the criteria of the different categories were written, in the third column the sub-criteria were connected to their corresponding criteria. In case a criterion did not have any sub criteria only a conclusion for the whole criteria was required, but also for criteria with sub-criteria a conclusion was presented.

<table>
<thead>
<tr>
<th>Category</th>
<th>Criterion</th>
<th>Sub-criterion or Conclusion</th>
<th>SmartLM</th>
<th>GenLM</th>
<th>SOLP</th>
<th>OSGC</th>
<th>FLSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>The Grid services implemented using Web Services</td>
<td>Conclusion</td>
<td>Fully utilized</td>
<td>Probably utilized, but some details missing</td>
<td>Utilized, but some details missing</td>
<td>Probably utilized, but some details missing</td>
<td>Utilized, but some details missing</td>
</tr>
<tr>
<td>Security</td>
<td>Conclusion</td>
<td>Firewalls</td>
<td>Security logging</td>
<td>Encryption</td>
<td>Authentication</td>
<td>Authorization</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>-----------</td>
<td>------------------</td>
<td>------------</td>
<td>----------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fully</td>
<td>Fully</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>detailed</td>
<td>utilized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Detailed,</td>
<td>Fully</td>
<td>NM*</td>
<td>Utilized</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>but</td>
<td>utilized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>missing</td>
<td>points</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Almost</td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Firewalls</td>
<td>Fully</td>
<td>Fully</td>
<td>NM*</td>
<td>Utilized</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>utilized</td>
<td>utilized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Almost</td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security logging</td>
<td>Fully</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>utilized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Almost</td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encryption</td>
<td>Fully</td>
<td>Fully</td>
<td>NM*</td>
<td>Fully</td>
<td></td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>utilized</td>
<td>utilized</td>
<td></td>
<td>utilized</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Almost</td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td>Fully</td>
<td>Utilized,</td>
<td>NM*</td>
<td>Utilized,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>utilized</td>
<td>but some</td>
<td></td>
<td>but some</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Details</td>
<td>missing</td>
<td></td>
<td>details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Missing</td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Delegation of</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>identity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Single sign-on</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td>Utilized,</td>
<td></td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>but some</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>missing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Identity</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mapping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Mutual</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td>Utilized</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>authentication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Certification</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized, but</td>
<td>Utilized,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>some details</td>
<td>but some</td>
<td></td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>missing</td>
<td>details</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>missing</td>
<td>missing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authorization</td>
<td>Fully</td>
<td>Utilized,</td>
<td>Utilized, but</td>
<td>Utilized,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Utilized</td>
<td>but some</td>
<td>aspects</td>
<td>but some</td>
<td></td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>but some</td>
<td>aspects</td>
<td>missing</td>
<td>aspects</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NM*</td>
<td>missing</td>
<td></td>
<td>missing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NM*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- The service to gather additional information associated with the user or the actual session</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*, but probably</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- The service to gather additional information associated with the protected resource</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- The checking of any local policy applicable to the request</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- The making of an authorization decision based on the identity of the user and the additional information</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local application server</td>
<td>Conclusion</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Missing</td>
<td>Utilized</td>
<td>Utilized</td>
<td></td>
</tr>
<tr>
<td>Tract</td>
<td>Conclusion</td>
<td>Fully utilized</td>
<td>Utilized, but some details missing</td>
<td>Utilized, but some details missing</td>
<td>Utilized, but some details missing</td>
<td>Utilized, but some details missing</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>----------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------</td>
<td></td>
</tr>
<tr>
<td>Loose coupling and modularization</td>
<td>Conclusion</td>
<td>Fully utilized</td>
<td>Utilized</td>
<td>Fully utilized</td>
<td>Utilized</td>
<td>Fully utilized</td>
<td></td>
</tr>
<tr>
<td>Virtualization and integration of heterogeneous autonomous resources</td>
<td>Conclusion</td>
<td>Fully utilized</td>
<td>Utilized</td>
<td>Fully utilized</td>
<td>Utilized</td>
<td>Fully utilized</td>
<td></td>
</tr>
<tr>
<td>Network linking all points within an enterprise and/or outside enterprise</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td></td>
</tr>
<tr>
<td>Use shared services, resources, or have common applications that run independently and not under centralized control</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td></td>
</tr>
<tr>
<td>Providing services/resources such as servers and large-scale processing</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td></td>
</tr>
<tr>
<td>Integration of Virtual Organizations</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*, but probably</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------</td>
<td>----------</td>
<td>--------------------</td>
<td>-----</td>
<td>-----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for virtualization</td>
<td>Fully utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM, but probably</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flexible and dynamic resource allocation and management</td>
<td>Conclusion</td>
<td>Fully utilized</td>
<td>NM*</td>
<td>Mostly NM*</td>
<td>Partly Utilized</td>
<td>Partly Utilized</td>
<td></td>
</tr>
<tr>
<td>Having a resource access framework which applications can call for allocation of resources</td>
<td>Utilized</td>
<td>NM*</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resources are used by the requesting application and then subsequently released</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*, but probably</td>
<td>Utilized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimize the use of spare resources</td>
<td>NM*, but probably</td>
<td>NM*</td>
<td>NM*</td>
<td>Utilized</td>
<td>Utilized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Co-allocation of different resources</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td>Utilized</td>
<td>Utilized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trade</td>
<td>Conclusion</td>
<td>Supported</td>
<td>Partly utilized</td>
<td>Partly utilized</td>
<td>Partly utilized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamicity of licenses; suspending, preempt, and resume the license usage</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for workflows</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*, but probably</td>
<td>NM*</td>
<td>Utilized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobility of licenses</td>
<td>Conclusion</td>
<td>Supported</td>
<td>Partly utilized</td>
<td>Partly utilized</td>
<td>Partly utilized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote license enforcement</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The license can be moved from one computing entity to another</td>
<td>Utilized</td>
<td>NM*, but probably</td>
<td>NM*, but probably</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regulation of resources based on company policies</td>
<td>Conclusion</td>
<td>Fully utilized</td>
<td>Partly utilized</td>
<td>Partly utilized</td>
<td>NM*</td>
<td>NM*</td>
<td></td>
</tr>
<tr>
<td>Enabling resource consumers’ and owners’ requirements and realization of their goals</td>
<td>Utilized</td>
<td>Partly utilized, only for ISV</td>
<td>Partly utilized, only for the ISV</td>
<td>NM*, but probably</td>
<td>NM*, but probably</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Billing and Payment</td>
<td>Utility models for how consumers demand resources and their preferred parameters</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*, but probably</td>
<td>NM*, but probably</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>----------</td>
<td>-----</td>
<td>-----</td>
<td>-------------------</td>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>Conclusion</td>
<td>Partly utilized</td>
<td>Mostly not utilized</td>
<td>Mostly not utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grids enable payment models such as &quot;pay-as-you-go&quot; and other on demand models</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>Utilized</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Models for establishing the value of resources</td>
<td>Utilized</td>
<td>NM*</td>
<td>NM*</td>
<td>NM*, but probably</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource pricing schemes and publishing mechanisms</td>
<td>NM*, but probably</td>
<td>NM*</td>
<td>NM*</td>
<td>Utilized</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Economic models and negotiation protocols</td>
<td>Utilized</td>
<td>NM*, but probably</td>
<td>NM*</td>
<td>Partly utilized</td>
<td>NM*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mediators to act as a regulatory agency for establishing resource value, currency standards, and crisis handling | NM*, but probably NM* | NM* | NM* | NM* | NM* |
---|---|---|---|---|---|
Accounting, billing and Payment Mechanisms | Utilized | Not utilized | Utilized, but some details missing | Utilized | NM* |

<table>
<thead>
<tr>
<th></th>
<th>Provision of information about resources and their availability</th>
<th>Conclusion</th>
<th>Utilized</th>
<th>NM*</th>
<th>Utilized, but some details missing</th>
<th>Utilized</th>
<th>Utilized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transparency</td>
<td>Transparent licensing mechanism</td>
<td>Conclusion</td>
<td>Utilized</td>
<td>NM*, but probably</td>
<td>Utilized, but some details missing</td>
<td>Utilized</td>
<td>Utilized</td>
</tr>
</tbody>
</table>

The LMF that had the best match with the Evaluation Model and fulfilled all the different areas was the SmartLM LMF, which also was the only one that has been trialed in a real business context. The other four LMF had missing information on various aspects; hence leaving the utilizer unaware of how to integrate and solve these problems in a real context. These aspects will be briefly mentioned below for the four out of five LMFs that were not detailed on every aspect:

In the technical area one criterion that is worth mentioning is the security category. Three out of five LMFs almost did not mention any details regarding security aspects, leaving the utilizer oblivious of how to implement this in a real LMS. This could be crucial in a contemporary business environment based on trust and under constant threat from cyber-attacks. In the tract area, the criterion “Flexible and dynamic resource allocation and management”, two out of five LMFs gave almost no details regarding this aspect; thus leaving the utilizer unaware of how to implement these details.
In the trade area one surprising aspect was that four out of five LMFs partly or almost not integrated any aspects that are commercially related. Two out of five LMF partly integrated the enabling of only owners’ requirements and realization of their goals while two others did not mention it at all. In the Billing and Payment criterion, three out of five LMFs gave almost no details regarding this aspect; hence leaving the utilizer without any implementation details. One out of five LMFs did not even mention or integrate any Billing and Payment aspects in the system. This is strange due to that all LMFs claim that they shall work in a real business environment.

On the topic of transparency, four out of five LMFs have little details regarding how the transparency is enhanced for the different users; thus leaving the utilizer without any implementation details. This comparison cannot result in a strict recommendation but it is obvious that there is most information available regarding SmartLM LMF. It has been proven to support most of the aspects of the Evaluation Model but this does not automatically mean that it is better than the other LMFs since they support more aspects even though it is not explicitly written.

Considering that there was information available, for the LMFs, regarding most of the aspects of the Evaluation Model, it increases the generalizability of the Evaluation Model. The LMFs presented in this thesis are supposed to be applicable in different contexts, considering that they have not been designed for a certain company or business area. This does not mean that we can guarantee that the Evaluation Model is applicable for all contexts but it enhances the likelihood to be feasible in other contexts than this particular case study.

### 6.7. The Ericsson analysis

All the criteria that could be found among Ericsson’s criteria have been discussed for the five LMFs. Even though the content of the criteria from Ericsson and the Evaluation Model differed, the discussion of the criteria of the Evaluation Model covers the Ericsson’s criteria, apart from the clouding aspect. The Evaluation Model is, as mentioned earlier, supposed to be general and work for this particular case study. This perception is enhanced since all criteria but one (usage of local application servers) could be found among Ericsson’s criteria. Therefore, we make the assumption that the content of the Evaluation Model is feasible for Ericsson as well.

As mentioned Ericsson emphasizes that clouding shall be supported which is not explicitly specified in the Evaluation Model. The reason for this is that the customer wants to have the possibility not to buy the actual hardware. This is therefore shown in the table below:

<table>
<thead>
<tr>
<th>Aspect</th>
<th>SmartLM</th>
<th>GenLM</th>
<th>SOLP</th>
<th>OSGC</th>
<th>FLSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clouding</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>NM*</td>
</tr>
</tbody>
</table>

* NM = Not mentioned

All LMFs do explicitly support clouding except for FLSS. It is hard to determine whether this can be supported or not, and that would require further investigation, such as further research or an interview.
with the framework authors (in case the authors know if FLSS supports Cloud). This was not further investigated due to the limited time.

The local application server was, as previously mentioned, a criterion in the Evaluation Model but was not present among Ericsson’s criteria.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>SmartLM</th>
<th>GenLM</th>
<th>SOLP</th>
<th>OSGC</th>
<th>FLSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local application server</td>
<td>Fully utilized</td>
<td>Fully utilized</td>
<td>Missing</td>
<td>Fully utilized</td>
<td>Fully utilized</td>
</tr>
</tbody>
</table>

Local application servers are included in the architecture for four out of the five LMFs, and are missing in SOLP. In case it is important for Ericsson not to use local application servers, then SOLP might be the most feasible LMF, considering that the license application servers are fundamental in their respective LMF architecture and removing them would probably affect other aspects negatively as well. The exact consequences of excluding local application servers are, however, not within the scope of this thesis. Furthermore, Process Architect ELIS & BNET (2013) emphasizes that Ericsson has no criterion that forbids local application servers so this is most likely not an issue for Ericsson.

SmartLM seemed to be the most feasible solution, after analyzing the LMFs using the Evaluation Model. After regarding the Cloud aspect this does not change, and it is valid also after regarding the local application server (given that Ericsson does not mind using local application servers).
7. Conclusions and discussions

This chapter summarizes all the findings of this study and validates them against the purpose and research questions. Furthermore, it also discusses any weaknesses and considerations regarding the results. Finally, suggestions on further research are discussed.

There is a myriad of conclusions that can be drawn from the empirical material and analysis conducted, but this chapter will only focus on the most relevant aspects that aim to fulfill the purpose and answer the research questions of this thesis.

**RQ1:** Which aspects have previous authors argued for regarding systems in Grid environments?

The aspects that previous authors argued for, and were mentioned in this essay, were summarized in to an evaluation model and synthesized into categories (Technology, Tract, Trade, Transparency) together with their internal criteria (aspects). The Evaluation Model and eliciting of aspects were a result from the work conducted in section 3.3.

The categories and their internal criteria were the foundation for analyzing the different LMFs. During the construction of the Evaluation Model, we noticed clear dependencies among two different categories. The Trade category is reliant on the Transparent category since Trade aspects need to be Transparent for management purposes, thus a rightwards arrow will denote this dependency. Another, and the most obvious dependency, is the other categories relying on the Technology category since the other categories depend on technology to be implemented (and is also limited by the choice of technology and its architecture). This dependency will be visualized by arrows pointing towards the Technology category from the other categories. The aspects (criteria) can be visualized in the Evaluation Model below:

![Figure 7-1 The Evaluation Model](image-url)
**RQ2:** Which criteria do five licensing experts at Ericsson consider important when evaluating a license management system?

The requirements elicited from the interviews and internal Ericsson documents corresponded with the different dimensions in the Evaluation Model. The only difference was that Ericsson also wanted a support for Cloud computing and did not have any preference for a local application server. These additional requirements can be modified in the Evaluation Model without rewording the fundamental structure and dimensions; thus Cloud computing were integrated in an already existing criterion under the tract category and local application server were removed from the technical category. Thus, one could see Cloud computing as already integrated in the Evaluation Model since it corresponds with the topic of virtualization and integration of heterogeneous autonomous resources under the tract dimension. The conclusion is drawn that our Evaluation Model was feasible for the particular case study and is likely to be that in other cases too due to the nature of generalization.

**RQ3:** Which license management frameworks for Grid environment are mentioned in previous research?

The LMFs found and used in this study were the following: SmartLM, GenLM, Service Oriented License Providing, On-demand software license provisioning in Grid and Cloud computing, and Floating License Sharing System.

**RQ4:** To which extent do license management frameworks for Grid environment, presented in previous research, match the aspects of our evaluation model?

With regard for correspondence with the four areas of technical, tract, trade, and transparent, and each area’s respective criterion. The LMF that had the best match with the Evaluation Model and fulfilled almost all the different areas were the SmartLM LMF, which also was the only one that has been trialed in a real business context. The other four LMFs missed information on various aspects; hence leaving the user unaware of how to integrate and solve these problems in a real context. No LMF was considered to be out of the scope for comparison with the Evaluation Model, most of the analysis that were labeled as not mentioned was due to that an aspect (e.g., security) was not mentioned at all in the LMF. All these observations could therefore be synthesized in to the conclusion that our evaluation model had a good match with the different LMF used in this study and should, thus, increase the likelihood of having a good match with other LMFs due to the nature of generalization.

**RQ5:** Which license management frameworks for Grid environments seem to be the most feasible for the particular case study?

For the case of Ericsson, the results of the comparison between the different LMFs can be generalized to their context. SmartLM is the most detailed LMF and could be therefore be utilized for Ericsson’s business context, while the others lack in details and need additional attention for making them feasible in a business context. The additional requirements elicited by Ericsson was the required support for Cloud computing and no preference for local application server.
Regarding support for cloud computing: SmartLM, GenLM, SOLP, and OSGC claim to support this concept, while FLSS does not state any support.

On the topic of local application server SOLP is the only LMF that does not utilize this aspect.

Depending on the importance of lack of local application server in Ericsson’s requirements, the most feasible choice according to our evaluation model would therefore be the SmartLM (if local application server is feasible for their business context) and SOLP (if Ericsson rejects this requirement).

Thus, no strict recommendations can be made for Ericsson due to the property of LMFs being evaluated, many aspects are not mentioned in the LMFs and therefore it is hard to say what would be best as an LMS. All the LMFs studied in this research could all be adjusted to fit the needs of Ericsson, but the SmartLM LMF gives the most comprehensive details that could probably speed up the implementation of a new LMS. The nature of the SmartLM LMF to have been trialed in a real business context strengthens its ability to work well for the business cases (including Ericsson), due to the nature of generalization.

Other research implications synthesized

One conclusion drawn, regarding the developed evaluation model, is that it is a seemingly good start for making informed decisions when choosing or designing an agile system for Grid environments. This conclusion is drawn by synthesizing the following three observations:

First, the case study, where Ericsson elicited requirements for an LMS and was used to underpin the Evaluation Model, morphed well into the Evaluation Model; hence making other case studies more likely to morph well into our evaluation model too due to the nature of generalization.

Second, in most cases various aspects of the LMFs were mentioned in the different dimensions in the Evaluation Model. Regarding aspects that were labeled, but not mentioned in the Evaluation Model, it was mainly due to that it was not specified in the LMF, but not impossible that it could be integrated in an LMS or extended into the LMF.

Third, all of the dimensions in the Evaluation Model were mentioned in systems that have proven business feasibility. The SmartLM LMF and Ericsson LMS are two systems that have been trialed in real business environments and are feasible solutions; hence making it more likely that other systems that correspond well with our evaluation model to also have a good business feasibility due to the nature of generalization.

The property of some LMF having dimensions labeled as not mentioned when the analysis was conducted could be either a strength or weakness depending on the utilizer’s context. Not mentioned could be seen as a freedom of choosing various aspects without being influenced by the LMF; which, thus, could result in a better solution for the particular business case. One should not consider the extreme case where all aspects are not mentioned since that would not even imply an LMF. Conversely, aspects that are not mentioned could also increase the time from transforming an LMF to an LMS. Adding additional aspects to an LMF that was not mentioned (e.g., local application server), could imply
major changes in the fundamental structure or making the realization impossible of the LMF, affecting more parts and, thus, increasing the time for realizing an LMF. Regarding the not mentioned aspects in the analysis enhancing the freedom of choice, one could argue that the property of loose coupling and modularization brings the same benefits as having aspects as not mentioned. This due to that Konsynski & Tiwana (2004) argued that the practice of loose coupling and modularization makes the components semiautonomous and augments the plug-and-play ability; hence bringing the same benefits of freedom as not mentioned and provides the utilizer with an implementation alternative. Using this logic could increase the argument for recommending choosing the SmartLM LMF, since it is the most comprehensively documented LMF and has a high utilization of loose coupling and modularization as according to our analysis.

**Implications for further research on the same topic**

For further research on the same topic we would like to see:

1) An LMF or an information system built on the principles elaborated in the Evaluation Model and trialed in a business environment to strengthen its generalizability of applicability in a real business context.

2) Additional case studies based on our Evaluation Model would have the same effect as the previous suggestion.

3) Further considerations on new criteria or sub-criteria that would further improve the agility of systems in a Grid environment could augment the evaluation of LMFs or facilitating choosing or designing agile systems for Grid environments.

4) More research regarding the business applicability of some criteria would be interesting due to that many of the criteria grasped in the theoretical framework were taken from research that mainly had a technical standpoints for their justifications. This could affect if they are going to be included in our evaluation model. One aspect that was made visible during the conducted interviews was the fear from customers reacting on having a local application server on their site. One interviewee argued that some companies might want to force Ericsson to pay for the local application server, since the licensing process is mainly their concern and interest. This would affect the profitability for Ericsson since the local application server, if paid by Ericsson, would imply an extra cost. These aspects of business applicability were, thus, out of the scope of this thesis.
8. Bibliography


9. Interviews

Process Architect ELIS & BNET, 2013-09-02 – 2013-12-10

Strategic Supply Manager, 2013-09-11 and 2013-09-17

SW Supply Preparation Manager, 2013-10-07

Product Manager License Management BNET, 2013-11-15

Price Manager BNET, 2013-11-15

Director of Software and Product Packaging, 2013-11-15

Strategic Supply Manager BNET

Strategic Supply Management BNET
10. Appendix 1 – Terminology

In Appendix 1 we will define some important concepts that are essential for the reader to understand.

10.1. Web Service
The World Wide Web Consortium (2004), which is an international community where member organizations, a full-time staff, and the public work together to develop Web standards, defines a Web Service as:

“[...] a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web Service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”

10.2. Virtual Organization
Open Grid Forum (2007) defines a Virtual organization as:

“A virtual organization [(VO)] comprises a set of individuals and/or institutions having direct access to computers, software, data, and other resources for collaborative problem-solving or other purposes. VOIs are a concept that supplies a context for operation of a Grid that can be used to associate users, their requests, and a set of resources. The sharing of resources in a VO is necessarily highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs.”

10.3. Quality of Service (QoS)
Open Grid Forum (2007) defines Quality of Service as:

“A measure of the level of service attained, such as security, network, bandwidth, average response time or service availability.”
11. Appendix 2 - interview questions

In Appendix 2 the interview questions, that turned out to have an impact on the outcome of this thesis, are shown in chronological order and are categorized after their respective meeting.

- Which requirements does Ericsson have for regarding their licensing?
- Which of these criteria are satisfied for the current license management solution?

- Which overall criteria does Ericsson have for a license management system?
- Which employees of Ericsson could you recommend in order for us to get a broader perspective of the licensing criteria of Ericsson?

2013-10-14 Product Manager License Management BNET
- Can you please give a short summary of your major tasks and responsibility at Ericsson?
- In which way do you work with the current license management system?
- Which benefits/drawbacks have you observed with the current license management system?
- Which is the requirement of the license management system?
- Which of these requirements does the license management system actually handle?
- What is the long term goal with the license management system?
- Which recommendations would you give us if we would introduce a new license management system at Ericsson?

2013-11-15 Price Manager BNET
- Which are your criteria for a license management system?
- Which of these criteria do you satisfy?
- Which of these criteria do you not satisfy, and why?
- How should a license management system strengthen the strategic alignment?
- How do the current license management systems strengthen the strategic alignment?

2013-11-15 Director of Software and Product Packaging
- Which are your criteria for a license management system?
- Which of these criteria do you satisfy?
- Which of these criteria do you not satisfy, and why?
- How should a license management system strengthen the strategic alignment?
- How does the current license management system strengthen the strategic alignment?
- Which are your criteria for a license management system?
- Which of these criteria do you satisfy?
- Which of these criteria do you not satisfy, and why?
- How should a license management system strengthen the strategic alignment?
- How does the current license management system strengthen the strategic alignment?