

Institute of Architecture of Application Systems
Universität Stuttgart
Universitätsstraße 38
70569 Stuttgart
Germany

Master Thesis Nr. 3349

**Energy-efficient Cloud
Computing Application Solutions
and Architectures**

Abdallah Issa Salama

Course of Study:	Information Technology
Examiner:	Prof. Dr. Frank Leymann
Supervisor:	M.Sc. Wirt.-Inf. Alexander Nowak
Commenced:	05.06.2012
Completed:	05.12.2012
CR-Classification:	D.2.2, D.2.3, D2.11, H.3.4, H.4.2, H.5.2

Abstract

Environmental issues are receiving unprecedented attention from business and governments around the world. As concern for greenhouse, climate change and sustainability continue to grow; businesses are grappling with improving their environmental impacts while remaining profitable. Many businesses have discovered that Green IT initiatives and strategies can reform the organization, comply with laws and regulations, enhance the public appearance of the organization, save energy cost, and improving their environmental impacts.

One of these Green IT initiatives is migrating or building the business applications in the cloud. Cloud computing is a highly scalable and cost-effective infrastructure for running enterprise and web applications. As a result, building enterprise systems on cloud computing platform is increasing significantly today. However, cloud computing is not inherently proposing energy efficiency solutions for these businesses.

In this thesis, a concept has been developed to support organizations choosing suitable energy-efficient cloud architecture while moving their application to the cloud or building new cloud applications. Thus, the concept focuses on how to employ the cloud computing technology as an energy efficient solution from the application perspective. The main idea applied in the concept is identifying architectures for cloud applications depending on the inherent properties of cloud computing such as virtualization and the elasticity that can make them green potential, and identifying correlations between these architectures with already identified business process patterns used in green business process design. Alongside with these correlations, the application has been decomposed into basic technical and business attributes that can describe the application. The relations between these attributes and the cloud architectures have been defined. The relations between the different components the application attributes, application architectures, and the green patterns can lead to not only the energy-efficient cloud architecture for the business application, but also to the architectures that can achieve the organization technical and business requirements. Prototypically, a recommender system has been implemented that supports the identification of suitable energy-efficient cloud application architectures in addition to the cloud migration decision.

Acknowledgment

I would like to thank my advisor M.Sc Alexander Nowak for his valuable guidance and advice throughout working on this thesis. I would like also thank the Institute of Architecture of Application Systems represented by Prof. Dr. Frank Leymann for giving me the opportunity to work on such interesting topic.

Special thanks to the DAAD for their support and guidance during two years of studying. You gave me the chance to be here to study and get to know the people and cultures.

Finally, I would like to thank my family, friends for their motivation and support; I hope that I was up to your expectation.

Contents

1	Introduction	7
1.1	Background	7
1.2	Thesis Motivation	7
1.3	Thesis Objective	8
1.4	Document structure	8
2	Theory and Related Works	10
2.1	Utility Computing	10
2.2	Distributed Computing	10
2.2.1	Clusters	11
2.2.2	Grids	12
2.2.3	Supercomputers	12
2.3	Cloud Computing	12
2.3.1	Cloud Computing Definition	12
2.3.2	Cloud Characteristics	13
2.3.3	Evolution of cloud Computing	15
2.3.4	Drivers to cloud adoption	16
2.3.5	Cloud Services Models	16
2.3.6	Deployment Models	18
2.4	Green Computing	21
2.4.1	Definition	22
2.4.2	Approaches	22
2.4.3	Why Green Computing?	23
2.4.4	Green Computing and Cloud Computing	24
3	Concept Development	30
3.1	The First Step in the Concept: Migration of the Application to the cloud	31
3.1.1	Will all the application move to the cloud?	31
3.1.2	Attributes Map of An application	34
3.1.3	The Contribution of the Application Attributes in the Migration Decision	41
3.1.4	Migration Decision Calculation Process (Migration Algorithm)	45
3.2	The Second Step in the Concept - After the Migration Decision	48
3.2.1	Green Business Process Management	48

3.2.2	Cloud Application Architectural Patterns	50
3.2.4	Pattern Based Application Development	54
3.2.5	The Relations between the Cloud Architectural and Green Business Patterns	57
3.2.6	The Relations between the Application Attributes and the Cloud Patterns	62
3.2.7	From Application Attributes and Green Business Patterns to Energy Efficient Cloud Patterns.....	66
4	Implementation	68
4.1	Client Side	68
4.1.1	Technology Used.....	68
4.1.2	Technical Issues	69
4.2	Server Side	70
4.2.1	Technology Used.....	70
4.2.2	Technical Issues	70
4.3	Database	71
4.3.1	Technology Used.....	71
4.3.2	The Entity Relationship Diagram.....	71
4.3.3	The Tables and the Attributes Description	72
4.4	System Sequence Diagram	74
5	Conclusion and future work	75
5.1	Achieved Results and Discussion	75
5.2	Future work	76
	Bibliography	79

List of Figures

<i>Figure 2.1: Distributed Computing and its Subset</i>	11
<i>Figure 2.2: Characteristics of Cloud Computing</i>	13
<i>Figure 2.3: Cloud Computing Evolution</i>	15
<i>Figure 2.4: Cloud Services Layers</i>	17
<i>Figure 2.5: Green IT dimensions</i>	22
<i>Figure 2.6: Reasons and Benefits for Using Green IT Practices</i>	24
<i>Figure 3.1: Parameters Of Application Dependencies</i>	33
<i>Figure 3.2: Attributes Selection in the Recommender System</i>	46
<i>Figure 3.3: Correlation of Patterns</i>	58
<i>Figure 3.4: Green Patterns Selection in the Recommender System</i>	66
<i>Figure 3.5: Process Flow Diagram of the Recommender System</i>	67
<i>Figure 3.6: Conceptual Model</i>	67
<i>Figure 4.1: ER Diagram of the Recommender System Database</i>	71
<i>Figure 4.2: Junction Table from the Recommender System Database</i>	73
<i>Figure 4.3: Sequence Diagram of the Recommender System</i>	74

List of Tables

<i>Table 1: Comparison of Significant Cloud Datacenters</i>	25
<i>Table 2: Decision Recommendation Table</i>	56
<i>Table 3: Cloud and Green Business Process Patterns Matrix</i>	59
<i>Table 4: Application Attributes and Application Architecture Correlations</i>	65
<i>Table 5: Database tables and attributes Description</i>	73

1 Introduction

1.1 Background

Green IT is becoming an important topic for organizations at the present time. Organizations want to be more sustainable and responsible offering their resources; the drivers behind this trend are related to the customers, governments, and to the organization itself. All of these drivers force the organization to think practically to find applicable solutions that can improve their environmental impacts. One of these possible solutions is the use of cloud computing [1]. The characteristics that cloud computing has as virtualized, elastic, market oriented, and dynamic environment reinforce this perception about the cloud computing. Cloud computing has many advantages on the organization such as reducing the IT investment in the organization and providing a scalable and dynamic infrastructure. However, improving the environmental impacts is not an inherent part of cloud computing [2].

1.2 Thesis Motivation

As a result of the increasing of the pollution rates, global climate change, challenges in ensuring the power resources, and the growth of considerations in corporate social responsibility, the interest in green IT has increased that last couple of years. Organizations are increasingly recognizing the importance of becoming environmentally proactive and are taking the initiative to develop and implement green strategies that protect the environment [3]. The public appearance of the organization, the high cost of the energy, and the governmental regulations to reduce the carbon footprints are the main drivers behind this trend. One of these “possibly green” strategies is the use of cloud computing environment to operate their applications.

In the last few years, computer systems started moving into consolidated, shared machines invisible to users: so called cloud computing systems. The basic idea of cloud computing is to provide computing resources including infrastructures, platforms, and applications as services and make them accessible via internet. Cloud computing enables on demand and accurately allocated computing resources as needed by businesses. Furthermore, cloud computing eliminates the expenses cost companies need to build the IT infrastructures (capital expenditure), and reduces the operating cost of the resources as the billing made for the actual level and usage of these resources based on pay-as-you-use purchase model. Consequently, building enterprise systems on cloud computing platform is becoming increasingly popular today [4]. The cloud computing system significantly reduces the cost of developing, deploying, and maintenance of enterprise systems.

From the previous introduction, cloud computing and green computing (Green IT) are two emergent topics where organizations invest in these days. Since improving the environmental impacts of the organization is not inherent in cloud computing; the question is, how to make

or exploit the cloud computing properties in order to support the organization to improve their environmental impacts when moving or developing their applications to the cloud?

In this thesis a solution for this question has been proposed. The key idea of the proposed solution is that architecting the application using certain identifiable architectural patterns can reduce the power consumed to operate the application in the cloud. These cloud patterns have to be correlated with green business process patterns that the organization can use when it design a certain business process or activity. A concept has been developed to support this idea; a set of cloud architectural patterns have been identified along with green business processes patterns. Moreover, the correlations between the different patterns have been concluded. Additionally, attributes map for an application has been presented along with the relations between these attributes and cloud architectural patterns. The aim of these attributes is to enable the organization from selecting the most relevant attributes that can describe its application and the business and technical requirements. The concept does not only recommend the organization with the most energy-efficient cloud application architectural patterns, but it also ensures that these recommended set of patterns match the organization's technical and business requirements (attributes).

The point that the application basically may not fit the cloud computing environment is not neglected from the suggested concept. A detailed analysis for various classes of applications that can be or cannot be migrated to the cloud has been presented. Furthermore, a concrete analysis for the relation between every application attribute and the cloud environment has been provided and used to develop an algorithm that supports the cloud migration decision.

Prototypically, a web application recommender system has been implemented that supports the identification of suitable energy-efficient cloud application architectures in addition to the cloud migration decision.

1.3 Thesis Objective

The purpose of the Thesis is ethical in essence; the developed concept and the recommender system resulted from the research provide a useful solution for those organizations seek improving their environmental impacts and moving toward Green IT. The developed concept can help organizations to choose the most energy-efficient cloud architectural patterns for their application when they want to migrate or develop them in the cloud. Prior to this main purpose, the developed concept supports the organization to take the proper cloud migration decision.

1.4 Document structure

The thesis is written in five chapters including the introduction. The following is a brief description of these chapters:

1. **Chapter 2:** the theoretical basis of the Cloud Computing and Green Computing is presented along with the discussion of the intersection between the two concepts. Solutions and techniques toward “Green” Cloud Computing systems are provided.
2. **Chapter 3:** the thesis’s concept is explained in details. The key ideas, methodologies, and algorithms of the recommender system are provided and explained deeply.
3. **Chapter 4:** the implementation issues are discussed. Technologies used to implement the recommender system as well as some technical issues are described. The developed web application tiers are explained separately provided with some diagrams.
4. **Chapter 5:** the features, functions, and the results of the developed concept are summarized along with possible future enhancement.

2 Theory and Related Work

The purpose of this chapter is the elaboration and the scientific concepts within the upcoming chapters.

Firstly, it will be important to get a fundamental overview of the utility computing and distributed computing. The explanation of these models gives an entrance to introduce the cloud computing concept. The introduction involves how cloud computing distinguishes from all other concepts mentioned.

Secondly, a detailed information about cloud computing will be shown. That includes the reasons behind moving to the cloud, and the importance of the cloud computing as one of the most emergent topics in IT today. The chapter will present technical information about the cloud, for example, cloud types and services. It gives the information needed to clarify the architecture patterns for the cloud application that will be explained in the following chapters.

Finally, we will cover the second main topic of the thesis which is the green computing. The understanding of this concept facilitates the process of understanding the green business process patterns will be explained in the following chapters as well.

2.1 Utility Computing

Utility Computing is using the computing resources such as infrastructure, storage and core services in the same way of using electricity or water. The user of these resources pays only for what he uses. In this way, the utility can remove the need to purchase, run and maintain hardware, server and application platforms, and to develop core services. Using the computing resources as a metered service became the foundation of the move to on demand computing, SAAS, and cloud computing models that further propagated the idea of computing, application and network as a service [5].

2.2 Distributed Computing

Distributed computing refers to the idea of using distributed systems to process a common goal between multiple several software components that are on a multiple computers but it appears to the users of the system as a single coherent system. The communication between the computers in the distributed systems can be homogeneous or heterogeneous. They can be also connected by local or global area networks [6].

Distributed computing has some advantages that can be summarized as follows [7]:

- **Reliability:** the distributed computing system is more reliable than the single system. If one machine is collapsed, the system can serve as other computers will not be affected.

- Scalability: the distributed computing system can be easily extended by adding more computers.
- Resource Sharing: resources and data are shared in the distributed systems, which is very important to many applications such as banking and reservation systems.
- Performance and Speed: distributed systems have more computing powers than centralized systems, and this necessarily impacts the performance and the speed of system.

Based on the characteristics of the localization or equality; distributed systems are split to different subsets, such as supercomputers, grids, clusters, web 2.0 and clouds [7]. Before going further into these subsets of distributed computing, figure 2.1 [8] is provided to visualize the interconnection between the concepts that will be described in the following sections:

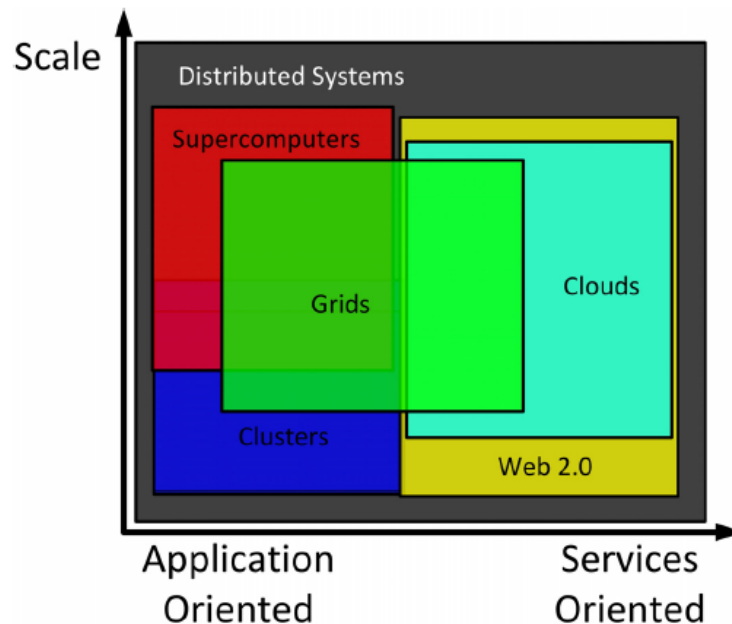


Figure 2.1: Distributed Computing and its Subset

2.2.1 Clusters

A cluster is a parallel and distributed system. It consists of a collection of interconnected standalone computers or UNIX workstations working together as a single integrated highly available computing resource [9]. The main use of cluster computing is the load balancing for the traffic on high-traffic web sites. The server manager receives the web page request, and then it forwards the request to one of the identical web servers to handle the request [10].

2.2.2 Grids

When defining the grid computing it is very important to distinguish it from clusters computing. The clusters are distributed locally and use the same hardware and operating system, while the grids are parallel and distributed systems. Grids consist of group of connected heterogeneous computers and distributed globally. The hardware and operating systems of these machines can also be different [9]. Grids enable sharing, selection and aggregation of resources dynamically at run time depending on the availability, performance, and users' QoS [8].

2.2.3 Supercomputers

Supercomputers can be compared to clusters. It has the same concept. Clusters are locally interconnected with other machines while supercomputers merge machines into one box [8].

2.3 Cloud Computing

With the huge increasing of the speed of the network over the last decades, there is a significant rise in its usage involved millions of Web queries a day. This increasing demand is handled through huge datacenters. Datacenters consist of hundreds or thousands of servers with other infrastructure such as storage and network systems. These huge datacenters are operated by many IT global companies such as IBM, Google, Amazon, and eBay. The commercialization of these developments is defined currently as cloud computing, where computing is delivered as utility on a pay-as-you-go model [11].

2.3.1 Cloud Computing Definition

Cloud computing is a distributed computing paradigm allows to outsource all IT needs (hardware and software) such as processors, storage, platforms, software and services. These resources are accessible and delivered as service over a network (typically the Internet). A clear example of the cloud computing is document-sharing service such as Google Docs.

The literary meaning of cloud computing comes from the terms cloud and computing. "Cloud" in the reality has no shapes and fixed positions, and the cloud server can have any setting and configuration. The location of cloud server can be also anywhere in the world with Internet as fundamental medium. Computing term achieved using set of networked resources which are offered on subscription.

The growing reputation of cloud computing has produced several definitions for the term describing cloud characteristics. The following definitions have been taken from well known scientists and organizations [11]:

- I. [9] defines the cloud computing in terms of its utility to the end user: “A cloud is a type of parallel and distributed computing system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreement established through negotiation between the service provider and consumers”

- II. National Institute of Standards and Technology (NIST) defines the cloud computing as follows: “Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources such as networks, servers storage, applications and services that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and for deployment models.”

2.3.2 Cloud Characteristics

The different definitions of cloud computing shown in the previous section and all other definitions existing in many resources and through the internet, lead back to some main characteristics of cloud computing. These characteristics are the main demands of the industry for future IT.

Figure 2.2 [11] shows the main characteristics: virtualization, service oriented, elasticity, dynamicity, economy of scale, Market-Oriented (Pay as you go) and autonomic.

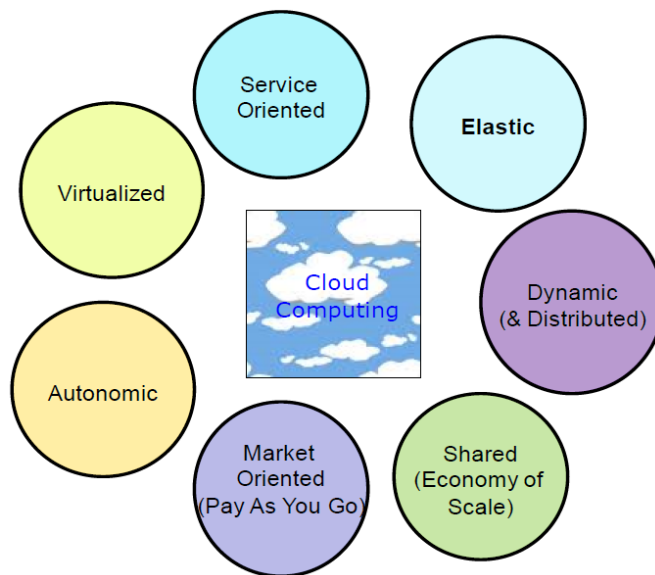


Figure 2.2: Characteristics of Cloud Computing

- Virtualized: Resources in clouds i.e. computing, storage, and networking are virtualized. The virtualization is achieved at virtual machine and platform levels. Applications can run within their operating systems running on the same physical machine. Application can easily be migrated from one physical server to another. In the platform level, every application is mapped to one or more resources offered by different Cloud infrastructure providers [11].
- Service-Oriented: everything in the cloud is a service. Software, platform and infrastructures are offered as services made accessible over the network. To achieve this facility, Cloud is implemented using Service-Oriented Architecture model. SOA is an architectural style for building application, loosely coupled, allowing composition. Cloud can benefit from this paradigm in building any of its services such as infrastructure, so it's called service-oriented infrastructure [12] [13].
- Elastic: Cloud provides the flexibility dynamically provision and vary the resources allocated for its application. The resources (computing, storage, network capacity) increases or decreases at runtime depending on the user QoS requirements [11].
- Dynamic and distributed: to add more performance and reliability to the cloud services delivery; cloud resources are made distributed. The resources are also dynamic and can be changed (increase or decrease) according to the requirements of the user [11].
- Economy of scale: cloud computing is often defined as a technology. However, it is actually a significant shift in the business and economic models for consuming IT that can lead to a significant cost saving. The cost saving can be achieved by the resource pooling; this capability makes the resource shared among the users. Thus, resources can be dynamically allocated according to the applications demand. That is also known as multi-tenant sharing model. Generally, the users neither have any control over the physical resources nor they are aware of the resource location [14].
- Market-Oriented (Pay as you go): this characteristic addresses the utility dimension of cloud computing. Cloud computing is billed much like a utility, so it can be affordable for even the smallest business. The user only pays for the services that are used [11].
- Autonomic: Cloud services are highly reliable. Therefore, clouds show autonomic behavior by managing themselves in case of failures or the performance deprivation [11].

2.3.3 Evolution of cloud Computing

The growth of cloud computing area started alongside with the internet. In the beginnings of the internet, ISPs offered some services and those services were not cheap and limited. The area of interests of the internet was very small and focused on the scientist and organization not on the normal customers. With the time, the internet became more popular, and then the ISPs started to make more attractive offers for the customers. Some new services have been added such as emails or access to servers in their data centers (Webhosting) [15].

At this time, the services provided were more related to computing infrastructure services. Thus, the next evolution step added more to the existing ones. The Internet service providers started the transformation from pure computing infrastructure to application service providers (ASPs). The result of this process is hosting some applications consumed by the customers. These applications including the infrastructure are owned and managed by the application service providers. The main difference to what known today as SAAS (Software as a service) is that the ASPs provide the service to customer as a single tenant and that includes the infrastructure, while SAAS provides a shared multi-tenant environment [15].

Following this sequence of events, cloud computing appeared and as mentioned before in this chapter there is no clear definition for cloud computing and still in the transformation process. Cloud computing integrate some old concepts mentioned above in this section and their steps in evolution of ISPs and service providers with support of new technologies such as virtualization, Service Oriented Architecture (SOA) and other characteristics to achieve the main demands of the industry. So, cloud computing is not exactly new technology, but it is a new architectural concept merges old and new concepts, and it has been supported by newly evolved strong technologies. Figure 2.3 [15] shows the different evolution stages which lead to cloud computing.

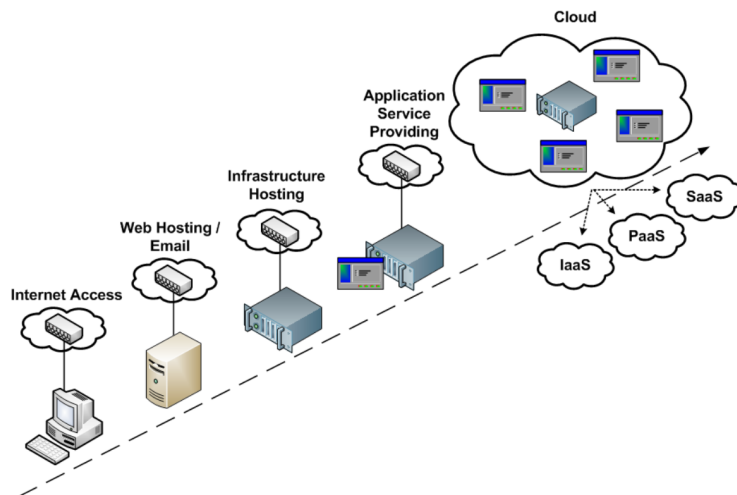


Figure 2.3: Cloud Computing Evolution

2.3.4 Drivers to cloud adoption

Economies of scale: Determining the new IT infrastructure is a complicated process and it is always based on a project which evaluates in its sizing phase the requirements of the hardware or software to accomplish business requirements. Estimating the actual percentage of the performance is difficult as well. Thus, Underestimations and overestimations are a daily occurrence. Cloud computing enables on demand and accurately allocated computing resources as needed by businesses. The results are shorter evaluation projects and a better risk management, because lots of unknown factors are eliminated by default [15].

Small initial investment and low ongoing cost: Cloud computing eliminates the expenses cost companies need to build the IT infrastructures. As everything is hosted to the cloud, the only expenditure required is for accessing cloud services such as networks and client systems. In addition, operating costs are reduced since the billing made for the actual level and usage of the resources on a pay for a use basis. Another important point for expenditure saving is the reduction of maintenance cost. Since cloud computing uses less physical resources, there is less hardware to maintain and power. Through cloud provider, companies can decrease the number or the working hours of server, storage, and virtualization experts; cloud providers are the responsible [15].

Open Standards: Most cloud providers expose well-documented Application Programming Interfaces, but every provider has its own unique implementation and thus it lacks to interoperability. Unless a critical mass of cloud consumers and the market's demand for modularity and standardized cloud interfaces is reached, vendor lock-in is a threat to cloud computing. Nevertheless, these standards will come, e.g. VMware as the biggest virtualization software vendor, announced "Project Redwood" as an interface for interoperability between clouds [15].

Sustainability: This driver is an immense driver for companies of all sizes that benefits from cloud. Moving the IT infrastructure, applications, and services to the cloud allows the enterprises to focus more on their core business and change it quickly in response to changing market demands. Cloud providers are responsible of security, risks and disasters management. In addition, the cloud service provider removes any technological bottlenecks. Sustainability is realized, because both cloud providers and enterprises concentrate on their own core business, and they are not aware about the other task [15].

2.3.5 Cloud Services Models

Cloud computing providers offer their services according to three fundamental models. It is mainly composed of these three models (layers) which form the computing stack of the cloud system. Each of these layers offers different services to

the end user. Figure 2.4 [11] describes the offered services by these three layers, Infrastructure as a service, Platform as a *services* and software as a service.

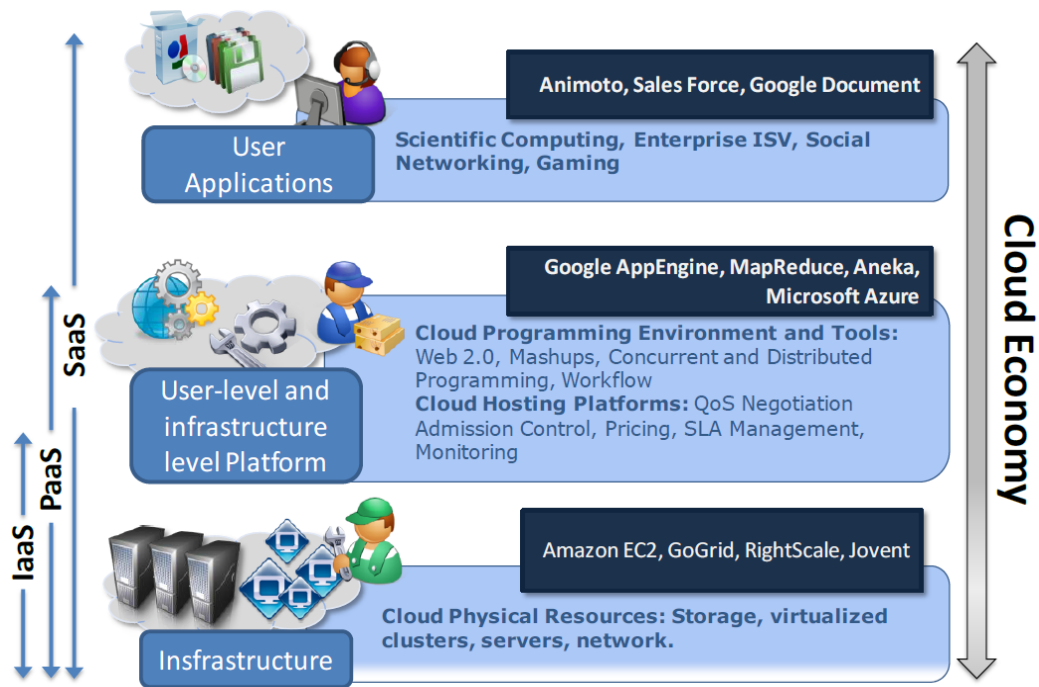


Figure 2.4: Cloud Services Layers

Infrastructure as a service (IAAS): The lowest layer in the computing stack, it consists of virtual or physical machines, storage and clusters. IAAS layer can also contain database management systems and other storage services. The infrastructure is managed by an upper management layer that ensures application isolation, runtime environment customization, accounting and quality of services. A virtualization tool, such as hypervisors sit in this layer to manage the resource pooling, and to partition the physical infrastructure into virtual machines [11].

According to [16], statistically proven figures show that 80% of the computing power is not efficiently used, and about 65% of the storage of servers. Therefore, the move to IAAS becomes a need to the companies that want to reduce the investment in server or network infrastructure. Companies purchase these resources on a rental basis. The providers are aware of the servers, storage, and network settings, while the client has virtual instance of that [9] [16].

Amazon Web Services is an example of IAAS. The infrastructure is billed on a pay per use model. The customers can get servers, storage without caring about co-location, rental or datacenters [17].

Platform as a service (PAAS): Platform as a service provides users with complete development environment to support the development lifecycle from design,

implementation, debugging, testing, deployment, operation and support of rich internet application (RIA) and online services. With PAAS an entire software environment can run at the service provider while not worrying about the technology underneath it. Google AppEngine, Aneka and Microsoft Azure are some of most prominent examples [9] [16].

The main advantage of PAAS is the cost saving in all the development cycle steps. PAAS providers allow cloud users to focus on the application development by reduction the cost resulted from the upgrading of the underlying platform.

Software as a service (SAAS): Software-as-a-service is the top most layer of the cloud computing architecture. SAAS is defined as a software delivery model in which the software (application) is hosted on the cloud with its associated data. It provides on-demand access to applications typically using a thin client or web browser. SAAS became a common delivery model for many business applications in different enterprises from small to large business, including customer relationship management (CRM), enterprise resource planning ERP, and human resource management (HRM) and several other business applications. SAAS providers constitute other layers (services) of cloud computing, and thus, it provides the services required to maintain and store the customer data in addition to configure the applications according to the customer's requirements. SAAS results in a considerable reduction in the cost of buying new software and infrastructure. The customers do not need to keep or maintain any infrastructure or install the applications on every machine. They just need high network speed that enables them to access their applications normally by web browsers [11].

The key feature of SAAS is the multi-tenancy architecture. This architecture distinguishes SAAS from other packaged software solutions. With this model, a single version of the application, with a single configuration for tenants (customers) is used. That allows the companies to outsource the effort of maintaining and upgrading applications, managing large hardware infrastructure, and optimizing resources by sharing the cost among the large user base. Hence, SAAS is interesting for these companies who access software shared between multiple users and configured to the companies' specific needs [11].

2.3.6 Deployment Models

Cloud computing is a paradigm of offering different kind of services in on-demand delivery model. According to the previously mentioned services and cloud computing characteristics, cloud computing deployment models are classified to work in harmony with other elements into four types: Public cloud, Private Cloud, Community Cloud, and Hybrid Cloud.

Public Cloud: public cloud is one of the deployment models, in which a service provider makes the services (application, storage, Infrastructure) available to anyone (General Public) on Internet. Public cloud services can be free or purchased on a pay-per-usage model.

The main benefits of using public cloud services are:

- 1- The overall cost of the system installation is covered by the provider, and so it provides an easy and inexpensive set-up.
- 2- Scalability to meet the needs. To support thousands of public domain users, public cloud providers build quite large datacenters consisting of thousands of servers with high speed network.
- 3- Pay for what you use, no resource wasting.

Examples of public clouds: Amazon Elastic Compute Cloud (EC2) provides infrastructure as a service, Google AppEngine provides an application development platform as a service, and Salesforce.com offers software as a service. Public clouds provide a very good option to handle peak loads on the local infrastructure and for effective capacity planning. Thus, it offers very good solutions to the customers having small enterprises or with infrequent infrastructure usage [11].

Quality of service and security are two main issues have to be ensured in the management process in public clouds. Consequently, a considerable part of the software infrastructure is dedicated to monitor and manage cloud resources, log the history of the cloud usage for each customer, and bill the used services according to the users' contracts [11].

Private Cloud: Although public clouds provide very good solutions that reduce the cost of installation and resources management, in addition to providing a scalable system that can handle the peak loads on the local infrastructure, there are still some cases where company may want to maintain their own specialized clouds. For instance, the health care industry maintains confidential medical data which cannot be stored in public clouds. Thus, private clouds are deployed and operated solely for a single organization to provide IT services to its internal users.

The main benefits of private clouds are [17]:

- 1- Offering greater control over the infrastructure. Due to the fact that the infrastructure is on-site, the organization is involved in the process of maintaining and monitoring its data.
- 2- Improving security by restriction the access to one organization.

- 3- Reducing the time needed to deliver what users demand. The private cloud is deployed inside the firewall on an organization's intranet that means the transfer rates are significantly increased.

Hybrid Cloud: Private clouds provide suitable solutions for organizations seek more control on their data and greater security. Moreover, private clouds increase the performance of accessing the applications data. However, the cost and maintenance are high disadvantages for private clouds. The higher cost resulted from that in the private clouds the organization need hardware and maintenance personnel, in addition to investment in new hardware or using the existing infrastructure in the company. Being some parts of the infrastructure deployed on-site of the organization, that raises up the maintenance problems and risk management [17].

According to the discussion above, hybrid clouds come to compromise a solution between the public and private cloud. In hybrid clouds, organizations host the non-critical and infrequent used data to public clouds, while keep the important information and services on their sites (Private Clouds) [11].

Community Cloud: Community cloud is a deployment model in which infrastructure is shared between several organizations they have the same computing concerns such as security, compliance, performance requirement and jurisdiction. The organizations with the community clouds can take the advantages of a public cloud such as multi-tenancy and pay-as-you-use billing model. Moreover, some benefits related the privacy and security associated to a private cloud can be gained. The cloud may be managed by organizations or a third party and may be existing on-premise or off-premise [18].

2.3.7 Cloud Services Type

According to the cloud service models (IAAS, PAAS, and SAAS) have been described in section 2.3.6, a cloud offers different types of services. These types describe the nature and the behavior of these services. However, the service models depict the style in which resources are provided by the cloud. Three classes of these services will be explained in the next subsections: cloud compute service, cloud storage service, and cloud communication service [19].

Cloud Compute Services: Cloud compute services are responsible of executing the workload. They are mainly having computing functionalities. These services are offered depending on the cloud service models.

There are three main computing services: Elastic Infrastructure, Low Availability Computing Node, and High Availability Computing Node. Elastic Infrastructure is a computing service that handles the issue of dynamic provisioning and de-provisioning

of resources, and thus it provide a solution for the allocation of resources dynamically and on demand. Low availability computing nodes provide a low cost solution for services their availability is not very critical. According to [20], Low available compute nodes usually constitute a public cloud; the relations with other patterns will be explained in the next chapter. On the contrast, the high availability compute nodes used when computing environment shall be provided for service whose availability is critical.

Cloud Storage Services: Cloud storage services are used to store, maintain, and manage customers' data, and make it accessible over Internet. The services differ due to the assured service levels and consistency behavior.

As in the computing services case, these services also depending on the service models shown in the previous sections. For example, public cloud storage is suitable for unstructured data that is not frequently used or subject to a constant change. Public cloud storage service stores the data on multiple inexpensive storage nodes for redundancy and it is accessed by internet protocols. If the organization wants more control over its data, or the data is actively used, then the private cloud is more appropriate. Data in the private cloud is stored in dedicated data center that ensure security and performance.

Cloud Communication Services: Cloud communication services are used to transfer the data between various applications and their components hosted in the cloud. They are also used to exchange the information with resources that located outside the cloud such as traditional data centers.

The communication services provide solutions for challenges related mainly to the process of exchange messages between different applications and components. For example, Message-Oriented Middleware (MOM) handles the problems that come up from the heterogeneity of the applications in the distributed environment. The applications normally use different languages, data format, and technology platforms. Hence, Message-Oriented Middleware connects applications through an intermediary that hides the complexity of addressing and availability of communication partners as well as supports transformation of different message formats.

2.4 Green Computing

The aim of this research is to help the organizations who want to move their applications to the cloud to find the most energy efficient architecture for the applications and their data contributes in reduction the footprint and carbon emissions which indeed resulted in reduction of the pollution and making the environment greener. This section will discuss the green computing topic which is the second main relevant theoretical topic related to the research after the cloud computing. Moreover, a deep discussion will be shown about

the green computing aspects in cloud computing, and whether the cloud computing itself is a green idea or not.

2.4.1 Definition

Green computing [21] is also known as green IT, refers to environmentally sustainable computing or IT whose goal mainly to maximize the energy efficiency and reduce the use of hazardous material. Green computing focuses on reducing resource consumption and disposing the electronic waste in a responsible way. To achieve these purposes, green computing is applied in all aspects related to the computer systems such as implementing of energy-efficient central processing units (CPUs). According to [21], green computing is the study and practice of designing, manufacturing, using and disposing of computers, servers, and related peripherals efficiently and effectively with minimal impact on the environment.

2.4.2 Approaches

Based on the definition of green computing in the previous subsection, Green Computing can be addressed in four dimensions: Green use, Green disposal, Green design, and Green manufacturing as shown in Figure 2.5 [21].

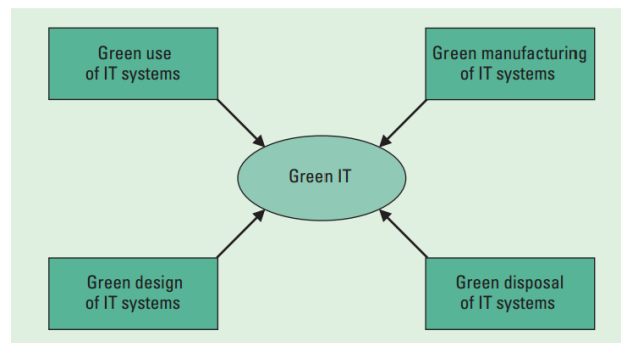


Figure 2.5: Green IT dimensions

Modern IT systems compose of a complicated mix of people, network, hardware, software. Thus, green computing initiative has to cover all of these areas. In the next sub-sections, some of these approaches will be discussed. The following represents environmentally sound practices applied in some dimensions shown in Figure 2.5. The following are some approaches applied in green design and manufacturing of IT systems.

Data Centers Design: The accelerated increasing in the numbers of Internet and Web-based applications leads to the fast growth in the data centers. In the last decade, the number of the server computers installed in data centers increase to 30 million. Moreover, the electricity consumed by these new servers is more than the old ones.

Beside the environmental impact of expanding the data centers, there is a huge increase in the cost of operating these data centers as the cost of energy is growing continuously. Furthermore, the availability of the electrical energy is a critical problem for many companies. Therefore, the social, financial, and the practical constraints force businesses and IT departments to reduce the energy consumption of these data centers [21].

To improve the data centers efficiency; there are some approaches such as using new energy-efficient equipments, investment in energy management software, enhancing the cooling techniques, on-site electrical generation and recycling of waste heat, and adopting environmentally friendly designs for data centers and new measures to limit data centers' consumption [21].

Software Optimization by Algorithmic Optimization: The efficiency of the algorithms has a great impact on the resources required to compute computer programs. For example, by changing the search algorithm from linear search to a binary, hashed, or indexed search can reduce resource usage for a given task. According to a study has been made by a physicist at Harvard, Google search emits seven grams of carbon dioxide (CO₂) [22]. However, Google opposes this figure, arguing instead that a typical search produces 0.2 grams of carbon dioxide (CO₂) [23].

Virtualization: Virtualization is the key strategy to decrease the power consumption of data centers. The virtualization technology built based on the idea that one physical server hosts multiple virtual servers. Virtualization technology simplifies the data centers, makes them using less electricity, and reduces their energy demands by hosting multiple virtual machines on a small number of more powerful servers. Many enterprises are using virtualization to curb the runaway energy consumption of data centers.

Power Management: There are some power management systems that can allow an operating system to control the power consumption of the underlying hardware. Such systems can automatically switch off components such as hard drives and monitors after set periods of idleness. Moreover, a system may hibernate by turning off most components including the central processing unit (CPU) and the system memory (RAM). The Advanced Configuration and Power Interface (ACPI) and the earlier Intel-Microsoft Advanced Power Management are example of such power management standards [22].

2.4.3 Why Green Computing?

Although adopting green IT practices has great impacts on the environment, it also offers different businesses and individuals financial and other benefits. One example of these benefits that when IT operations achieve better energy efficiency through

green initiatives, and the financial benefits results from reducing the cost of the electrical energy used especially when the issues of the price and availability of electrical energy become main critical challenges for many companies. According to a survey by Sun Microsystems Australia, involving 1,500 responses from 758 large and small organizations said that reducing power consumption and lowering cost are the major reasons for using green IT practices, and then a lower environmental impact came after. Figure 2.6 [22] shows the reasons and benefits of the green computing in percentages according to this survey.

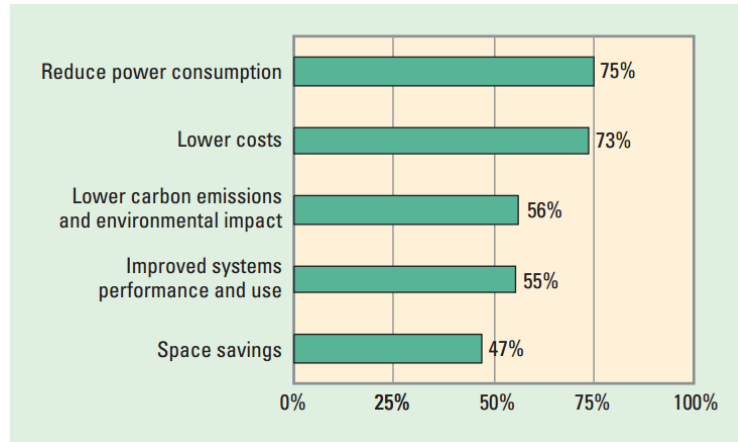


Figure 2.6: Reasons and Benefits for Using Green IT Practices.

2.4.4 Green Computing and Cloud Computing

Green computing and cloud computing have been explained in the previous sections separately. These two concepts are the backbone of the research; finding the concept that can help the organization to find the most-energy efficient “greenest” architecture for the application in the cloud opens the door for the questions about the relation between the green and cloud computing, and whether the cloud computing is a green alternative or not. In this section, detailed discussion will be introduced in order to give a clue about the bond between two concepts, and to give an introduction to the concept that will be explained in the next chapters.

Is cloud computing green?

Some researches demonstrate that cloud computing can make traditional datacenters more energy efficient by using technologies such as resource virtualization and workload consolidation. The traditional datacenters running web applications are often provisioned to handle periodic peak loads, which can result in low resource utilization and consumption of energy. Cloud datacenters, on the other hand, can decrease the energy consumed through the server consolidation technology, whereby different workloads can share the same physical host using virtualization and unused servers can be switched off. A recent

research by Accenture-a leading technology, consulting and outsourcing company- and WSP Environment and Energy-a global consultancy dedicated to environmental sustainability issues- shows that moving business applications to cloud can reduce carbon footprint of organizations. The study found that, for large deployments, cloud solutions can reduce carbon emissions by more than 30 percent. The benefits are even more for small businesses; energy use and carbon footprint can be reduced by more than 90 percent.

Contrary to the above opinion, some researches, for example Greenpeace [24], observe that cloud computing may make the problem of carbon emissions worse. The reason given is that the demand for computing resources will increase significantly in the next few years. The cloud providers do not cope with this increasing demand in computing resources in term of carbon footprints. In the best case some cloud providers will mitigate the carbon emissions. The reason given is that Cloud providers are more interested in electricity cost reduction rather than carbon emission. The data collected by [24] is presented in Table 1 below. Obviously, none of the cloud datacenter in the table can be called as green.

Cloud datacenters	Location	Estimated power usage effectiveness	% of Dirty energy generation	% of renewable electricity
Google	Lenior	1.21	50.5% Coal, 38.7% Nuclear	3.8%
Apple	Apple, NC		50.5% Coal, 38.7% Nuclear	3.8%
Microsoft	Chicago, IL	1.22	72.8% Coal, 22.3% Nuclear	1.1%
Yahoo	La Vista, NE	1.16	73.1% Coal, 14.6% Nuclear	7%

Table1. Comparison of Significant Cloud Datacenters

Key factors of Clouds enabling Green computing

Even though there is a great concern that Cloud computing is not a green alternative, and it can increase energy consumption by the datacenters, the Cloud computing has a green lining. Cloud providers employ several technologies that can help reduce the energy consumed and the carbon emission. For instance, multi-tenancy technology can reduce the IT infrastructure, and thus, lead to higher energy efficient infrastructure. The key driver technologies for energy efficiency in Clouds are “Virtualization” and “Elasticity”. Virtualization as described in

section 2.4.2, presents a logical grouping or subset of the computing resources and server consolidation, companies can save more space, management, and energy, while Elasticity as described in 2.3.2 allows a flexible and dynamic provisioning of the resources. Thus, companies avoid the over or under estimation in the resources needed.

Accenture study [11] defines four key factors that enable the cloud to lower energy usage and carbon emissions. According to the report, companies can reduce the carbon emissions by 30% to 90% depending on their sizes.

Dynamic Provisioning: Provisioning means resource providing and it can be done by several ways. When a cloud provider accepts a request from a customer, the cloud provider generates the appropriate number of the virtual machines and allocates resources to support them. With dynamic provisioning, the provider allocates more resources as they are needed and de-allocates them when they are not needed. The reason behind that is to ensure a certain level for the quality of services to the customers and to handle the peak load. The dynamicity comes from the live migration of virtual machines, when an application needs more resources the virtual machines in Cloud infrastructure can be moved to another host. On the other hand, the applications require less resources are consolidated on the same server. The applications and resources are monitored by the cloud provider, and the resources are allocated on demand. As a result, datacenters maintain the active servers depending to current demand, which results in reduction of the energy consumption [11].

Multi-tenancy: With multi-tenancy, flattening relative peak loads done by serving large numbers of organizations and users on same infrastructure and software; which can minimize the need of extra infrastructure. Clearly, this approach is more energy efficient than installing multiple copies of software on different machines. Multi-tenancy makes the fluctuation on demand much smaller and thus results in better prediction in the infrastructure needed, and results in greater energy serving [11].

Server Utilization: With server utilization, servers operate at higher utilization rates. Generally, on-premise (on-location) infrastructure runs with very low utilization sometimes below than 5 percent of average utilization. By employing the virtualization technology, the number of active servers is dramatically decreased by hosting and executing multiple applications on the same server in isolation. This results in utilization level to 70 percent, and consequently reduces the power consumption [11].

Datacenter Efficiency: The power efficiency of datacenters has great impact on the total energy usage of Cloud computing. Cloud provider can improve the power usage effectiveness (PUE) of their datacenters by employing several technologies

and concept. There are some approaches can significantly improve the PUE in data centers such as using water or air cooling, design the servers in the form of modular containers, and using the power management systems. Moreover, services in cloud computing can migrate between datacenters which have better PUE values exploiting the high speed network, virtualized services, and monitoring and accounting of data centers [11].

Towards Green Cloud computing

To achieve the energy efficiency in Cloud computing, some technologies and concepts have to be applied in the overall cloud system. Recall the cloud computing definition: cloud computing is sharing the resources (hardware and software) stored with data in datacenters and accessed by the network normally via Internet. Clearly, the cloud system composes of application, cloud stack, datacenters, and network. The following subsections show some procedures and approaches can be applied for each element.

Applications: As many companies move their applications and software to the cloud to reduce the IT cost and to increase the scalability, it becomes very important to study the energy efficiency at application level itself. That includes the architecture, development, compilation, deployment of the application. In this thesis, we focus on the energy efficiency in the cloud computing from the application perspective.

The enterprise and commercial application are mostly designed for PC environment, and in general the energy efficiency factor is not taken in consideration when designing and implementing such applications. Some research such as [25] shows that the implementation of even a small task and the system upon which it performed can influence considerably then energy efficiency when it is executed on heterogeneous devices. This requires the research and analysis of trade-off between performance and energy consumption when running of software on multiple platforms and hardware. Furthermore, the energy consumption at the compiler level and code level should be considered by software developers in the design and implementation of applications using a variety of energy-efficient techniques and architectures proposed later in the research.

Cloud Software Stack: Virtualization and Provisioning: In the cloud stack, many techniques are applied at the IAAS provider level in order to schedule and manage the resources, and thus reduce the amount of active resources. The consolidation of virtual machines, virtual machine migration, scheduling heat management and temperature-aware allocation are examples of such techniques that lead to low power consumption. Virtualization is the key technology in these mentioned techniques as it has several features such as live migration of resources and server consolidation. Consolidation helps in managing the trade-off between

performance, resource utilization, and energy consumption [26]. In the same way, VM migration [27] allows dynamic resource management while facilitating faults management and lower maintenance cost. Furthermore, the improvement in virtualization technology has led to significant reduction in VM overhead which increases the energy efficiency of Cloud infrastructure.

Currently, there are several project and research that works on the management of the resources in the cloud computing environment. [28] suggests a power efficient technique that formulates the management problem into an optimization model aiming to reduce the power consumption in the cloud. Another solution proposed by [29] to fix the issues of over-provisioning of resources and under utilization of servers. There are several other research work which focus on minimizing the over provisioning using consolidation of virtualized server [30]. Most of these works use monitoring and estimation of resource utilization by applications based on the arrival rate of requests. On the other hand, due to multiple levels of abstractions, it is hard to maintain deployment data of each virtual machine within a Cloud datacenter. Thus, various indirect load estimation techniques are used for consolidation of virtual machines.

Datacenter level: Cooling, Hardware, Network, and Storage: The power efficiency of datacenters has very important impact on the total energy consumed by the cloud computing. In addition to the high expenditure of energy and cost savings all of these reasons push the cloud providers to adopt practices to reduce the energy consumption for their datacenters. Several practices have been presented to improve the efficiency of the datacenters starting from each device and ending with the processor level.

First level is the efficient construction and the location of the datacenter. The smart construction of data center starts from the thinking in the energy resources, using natural and renewable resources such as wind and sun to generate the electricity is a green option. The data center's location is another point. Currently the geographical features such as climate, fiber-optic connectivity and access to a plentiful supply of affordable energy are taken into consideration when choosing the location of the datacenter.

The cooling system in datacenters is one of the major concerns when talking about energy consumption in datacenters. There are two ways used in datacenter cooling systems: one is by water and the other is by air. In both approaches, it is important to cool the hot equipment rather than all the room area. Thus, newer energy efficient cooling systems are proposed based on liquid cooling, nano fluid-cooling systems, and in-server, in-rack, and in-row cooling by companies such as SprayCool [31].

The servers and processors are another level where the power efficiency addressed. Several enhancements have been made by some vendors such as AMD and Intel on the servers and processors to consume less energy and to be more efficient with a good performance per watt. Examples on these enhancements are that the new architecture of servers enables slowing down CPU clock, or powering off parts of the chips in the idle case. Multi-core processors system improves the computing power and saves more energy; the virtualization technologies play here the same important role as described in the previous sections. In a similar way, storage consolidation helps to reduce the energy requirements of IT systems by building efficient storage network that consolidates all storage.

Monitoring/Metering: In order to measure the energy consumed by devices, how it can be decreased, and to improve the datacenter's efficiency; the Green Grid presents two metrics known as the power usage effectiveness (PUE) and datacenter Infrastructure Efficiency (DciE) [32].

$$\mathbf{PUE} = \text{Total Facility Power} / \text{IT Equipment Power}$$

$$\mathbf{DciE} = 1 / \mathbf{PUE} = \text{IT Equipment Power} / \text{Total Facility Power} \times 100\%$$

These two metrics (PUE, DciE) are most common metrics presented to compare the efficiency of datacenters. In the equations above, the Total Facility Power is the power measured for the datacenter only. The IT Equipment is defined as the total power consumed in the processing, management, storing and routing of data within the datacenter.

3 Concept Development

In the previous chapter, the theoretical background and baselines of the thesis have been introduced, and some solutions and approaches that followed toward saving energy in cloud computing system in general have been discussed. The purpose of this chapter is to explain theoretically the concept that has been developed in this research, and thus, it gives the reader a conceptual understanding for the recommender system. The concept has been developed consists of two steps: in the first step, the system recommends the user with the proper migration decision for his/her application. The second step proposes the highly recommended green cloud architectures that are suitable for his/her applications based on some given data (Application Attributes). Thus, this chapter is divided into two main sections: Migration of the Application to the Cloud, and After the Migration decision.

Firstly, it is necessary to give a fundamental knowledge about the cloud migration especially that it will be the first step in the concept. Information will be given about the cloud migration process that can help to take the migration decision. Furthermore, a deep discussion about the applications that fit the cloud and those does not fit the cloud by analyzing the applications to a group of attributes that can enable the organization to analyze their applications according to these attributes and select which attributes are more relevant to their businesses. The relations between these attributes and the cloud environment will be shown, and the algorithm that has been developed to get the value on which that an organization can decide if it is better to migrate its business's application will be explained.

Secondly, the second step of the concept will be presented. The second step of the concept depends on the green process business patterns [33]. These patterns will be defined and explained. The results the user will get from this step are energy cloud patterns, and so we will introduce the most important cloud patterns. Since the concept is built mainly over the relations among the all patterns (Business Process patterns and Cloud patterns) [19] [33], we will present the relations between the green business process patterns and cloud application patterns [33]. Furthermore, connecting these relations with the application attributes from the first step makes the concept stronger. In this case, we not only recommend the user with "greenest" cloud patterns, but we also suggest the user with the "greenest patterns" that achieve his/her application requirement or quality attributes. To achieve this, we provide a concrete explanation about the relation can be discovered between every application attribute and the cloud environment.

3.1 The First Step: Migration of the Application to the cloud

Today enterprises have many applications that are implemented on-site existing infrastructure with known scalability, availability and resiliency and other issues. The enterprises tend to move these applications to the cloud although moving their applications will not fix all the issues. Enterprises can get financial and operational benefits when moving an application from legacy server to a cloud.

Application migration is the process of transitioning all or parts of an enterprise data, applications and services from on-site premises and redeploying them on a new platform and infrastructure in the cloud. The migration process includes the staging of the new environment before the actual cutover for the application and requires coordination between the IT teams in the time of migration.

3.1.1 Will all the application move to the cloud?

The answer is no. Some applications can be considered as good candidates to move fully or partially to the cloud platform, while other applications do not fit the cloud. To identify which applications can move to the cloud, applications are decomposed to several attributes from technical and business factors. The cloud is also decomposed into attributes to help make decision to run the application in the cloud or not. In the next section, these attributes will be described in details, and their contribution in the migration decision. Moreover, we will show how we employ these findings in the algorithm that is responsible to recommend the user the appropriate migration decision which is the first step in the recommender system.

The following is two lists of applications or characteristics of the applications that can be moved and cannot be moved to the cloud.

Cloud computing is a Fit

Database Intensive Applications: It fits the cloud as the cloud computing and database are compatible. However, the cloud is more suitable for storing large volumes of unstructured data. On the other hand, when the data is accessed frequently in huge number of transactions in applications such as banking systems, then it can struggle the cloud. Such applications are governed by hardware limitations inside all IT environments, and thus there are no or little benefits from sending them to the cloud [34].

Network Intensive Applications: It is a good fit, but it is necessary to have an access to the fastest and highest quality network services in the cloud. Attributes such as network capacity, latency, redundancy and routing flexibility have to be taken into account when searching for the proper cloud provider [34].

Disaster Recovery Applications: Cloud computing provides data recovery platform. Data can be stored safely and accessed anytime regardless the kind of disasters such as earthquakes, floods, or any kind of calamity attacks the business site. Businesses are advised to store regularly copies of their data in at least one remote location. Cloud is not only stores replicas of data, it also eliminates the need to find and pay for a storage location, furthermore, it removes the cost and problem of physically transporting backup media to the remote site. The great advantage for using the cloud too is that the cloud backups the data continuously not periodically. Thus, there is no chance to loss any critical data [34].

Virtualized Applications: It is excellent fit. The cloud is completely compatible with such applications. The best way to ensure the cloud success is to optimize the applications in terms of CPU, storage, interface, and network performance [34].

Low Risk Applications: such as Email, collaboration tools, Human Resource (HR), Customer Relationship Management (CRM), and conferencing software. Such applications can be worthy to move to the cloud. For example, CRM has maintenance overhead that company does not need. Like email, the configuration and setup of video conferencing software is complex and require experts for some days, by moving them to cloud, we can save the cost and move the staff to work on more important issues [35].

When the application architecture is cloud friendly: any application on an x86 platform will operate well in the cloud regardless the operating system. The application has to be re-architected in case that is on a platform other than x86 platforms before moving to the cloud. Moreover, if the application is a web-based or client server, then it is more cloud friendly [36].

Loosely-Coupled Applications: The more independent the nature of application, the easier it becomes to move to the cloud. The idea is that if the applications are tightly coupled, they are difficult to decouple, and thus they cannot work independently on a remote platform. Applications do not exist in isolation. According to Figure 3.1 [37], applications depend on other run-time environments within the Enterprise, require access to data a third party services, and proper network configuration. While some of challenges can be managed quite easily, dependencies that limit the application's performance and include third party providers are a lot more difficult to address. An application with dependency on external web-based services would be limiting its flexibility to scale-out, based on the performance the external service can provide. As a result, loosely coupled applications are more suitable for the cloud [37].

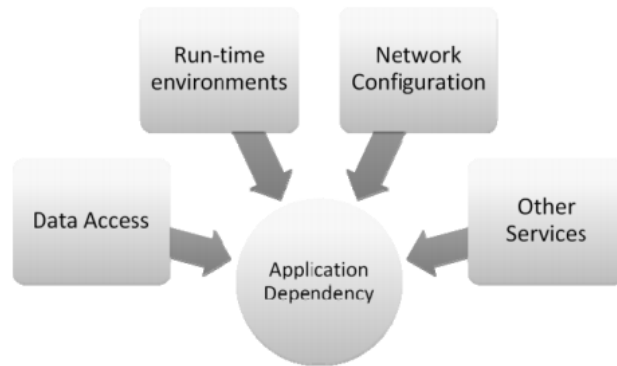


Figure 3.1: Parameters of Application Dependencies

Cloud Computing is not a Fit

Practically, the opposite characteristics for the applications or the classifications of applications mentioned in the previous section are not fit for cloud computing. Here more detailed information will be added about the applications and the characteristics of the applications which are not recommended to move to the cloud.

Highly Secured Applications: when a high level of security is required, or the application has very sensitive data, then moving application to the cloud is not highly recommended. When company moves its IT service on the Cloud, they are no longer in control of the location of data. Thus, the organization has to be careful when investigating the security level/controls that the service provider ensures [37].

When the control is very critical on the business: some components in the applications are very critical to the business. If such components will not be moved to 100% reliable cloud platform, then it could be endangering [38].

When the application is legacy: new applications are much easier to move to cloud computing than old applications [38].

When the application demand is steady and does not fluctuate: depending on the price of the service provider. If the servers on-site are underutilized, then it is better to keep things the way they are. In case that the organization wants to plan for disasters or other unexpected events, then it can be possible to move to the cloud [39].

When the application's licenses are only tied to physical core: many software licenses are not made compatible to work with virtual machines. For example, Oracle licenses can only work on a fixed physical core, whereas virtualization technology was developed to separate the physical layer from the software layer. If Oracle is installed on a virtual machine, the virtual machine will be assigned to certain physical

core(s) at one point in time and some other core(s) at some other point in time, which will violate Oracle licensing rules [39].

When the application needs a native interface: If the application requires leveraging native APIs (such as Win32) and browsers are not an option then cloud computing may not be a good fit [39].

3.1.2 Attributes Map of An application

In the last section, we have described cloud migration concept, and shown some general guidelines and characteristics for the applications that can be migrated to the cloud and for those applications which are not cloud fit. In this section we will show practically the first step in the concept which starts with decomposing the application into several attributes such as data access, security, reliability and interoperability [5]. When the user wants to move his/her application to the cloud; instead of the selection of the application according to its functionality or category, user can select some attributes that describe his/her application. Every application is designed to accomplish some purposes, whether an order management system, flight reservation system, CRM application, or something else. To implement the function of the application, certain attributes need to be presented. For example, with an order management system, transaction and locking support might be critical to the application. This would mean that cloud storage might not be suitable as a data store for such a purpose. Determining the key attributes of any application or subsystem of a larger application is a key step in determining whether or not an application is suitable for the Cloud. Thus, the user can pick up which attributes are more important for his business, and then the recommender system can return the most suitable architectures in more precise way. For example, some businesses are aware of security attributes more than reliability attributes. Hence, the system provides the facility to prioritize these attributes. In the following subsections we will add more details about these attributes, their contribution in the migration decision, and algorithm implemented to get the decision of the migration depending on the user selection for the attributes and the priority (level of importance) of the selected attributes.

➤ Data

Most of the enterprise applications contain data layer. Application's data is often stored in persistent storage in different kind such as database. The ways of storing and accessing data are different from application to application. The following shows some low-level attributes for "Data" high-level attribute.

Offline: generally, offline means the capability to connect to the network but it is not necessary that currently there is a connection to the network. Offline application's data is the storage of the application's data outside the network in daily use such as

backup tapes, and it is accessed through the off-line storage system (caching in the client-side), not the network.

Online: it is opposite to offline data concept. The application accesses the data directly from the server via network.

Structured: data is organized in an identifiable way, and it can be accurately identified. The database is the most common example of structured data where data resides in fixed fields within a record or a file. Even though that XML files are not fixed in location such that are not stored based on column and rows methodology like traditional databases, but they are considered structured data because the information is tagged and can be accurately identified.

Unstructured: data has no identifiable structure, and it does not reside in fixed locations. The term refers to free-form data, which is ubiquitous. Images, videos, email, documents and text are the most common examples of unstructured data. This is in contrast to a database, which is a common example of structured data.

Indexed: to increase the speed of data retrieval operations from a certain dataset such as database table, indexing concept is used such that indices can be created using one or more columns of a database table; providing a way to perform rapid random lookups and efficient access of ordered records without need to search every records sequentially.

Searchable: It refers to datasets that are searchable by typing in a query. The term looks redundant as all structured data such as traditional databases are searchable. Furthermore, indexing indeed improves the searching capability in the database.

Transaction Management: transaction comprises a logical unit of work that contains one or more SQL statement performed within database management system (DBMS) against a database. Transaction has to be performed in a reliable and independent way to achieve the ACID conditions (Atomicity, Consistency, Isolation, and Durability) providing all-or-nothing proposition such that the transactions have to be completely performed or rolled back without any impacts on data. Theoretically, a database management system ensures the ACID properties for each database transaction.

Persistence: the data that is not likely to be modified and it is accessed infrequently. Persistent data is retained after the execution of the program which created it.

State full/ Stateless: generally, state full and stateless are properties that describe whether a program designed to remember sequence of preceding events have been made by user's transactions, another program, a device or other outside components. State full data keeps a track of the changes have been performed on the data for

further use, for example by storing a log file for every data element. In the stateless case, there is no record saved to show the state of the data element and there is no way to see the preceding changes on data.

➤ **Security**

Security in cloud computing is one of the main issues that concern the organizations when they think to move their applications to the cloud. The following are some of the security features.

Emergency hotfix: it is a piece of code used to fix some urgent bugs in an application or a product. Hotfixes are sometimes packaged in one package called a service pack [41].

Trust relationship with platform: trusted platform/infrastructure is a concept related more to the hardware protection. To achieve the trusted platform, the trusted computing group (TCG) has developed the trusted platform module (TPM). TPM is a specialized chip used for the purpose of hardware authentication. It authenticates the computer rather than the user by storing information related to the host system, such as private encryption keys, certificates and passwords. As a result of deploying TPM, the risk of hacking computer data by physical theft or an attack by external threats will be reduced [42].

Application security model: application security is the use of hardware, software, and procedural methods to protect the applications from the external threats. The security procedures became a part in the application development as the enterprise applications are widely accessible via internet, and thus are more vulnerable to various kinds of threats. Application security can be improved by profiling the security issues for each application, identifying and prioritizing threats, and documenting adverse events and the actions taken in each case. That is known as threat modeling. The key in threat modeling is to know where the most work should be performed in order to keep the application secure [42].

Malicious code: the term refers to any unwanted code in any part of the software that is injected to do undesired effects, security breaches such as gathering some sensitive information or damage of the system. Malware is the general terms used to describe various forms of intrusive software, and it includes viruses, Trojan horses, spyware, and other malicious code [43].

Access controls: The term refers to security mechanisms that control who can access specific resources in a computer system. These resources can be system components, files, objects or other forms of data or functionalities.

Remote Access: is the ability to connect to a computer, or network, and access data-processing system from a remote distance, such as a virtual private network (VPN)

Cryptography: is a security technique to secure data storing and transmission by transforming the information (encrypt) the data into unreadable format called cipher text. The parties who have the secret key who can decrypt the encoded information and get the original data [44].

Information Auditing: it is the process of checking of the information system in an organization. It involves the technical, physical, and the administrative security levels to ensure that the correct and the updated processes and infrastructure are being applied, and to detect any rectify, duplication, and leakage of information. An audit includes a series of tests to ensure that information security meets the organization's requirements. The aims of auditing are to improve accuracy, security, and timeliness of the recorded information [45].

Authentication/Authorization model: authentication provides a way to identify the user wants to access an application or a computer resource. Following the authentication, a user has to have an authorization for doing a certain task; the system administrator defines the user that are allowed to access the system/application and what privileges of use such as access to file directories, specified application's components, hours of access some resources etc.

➤ **Maintainability**

When talking about an enterprise application being accessed by thousands or hundreds of thousands users, then the maintainability becomes a critical attribute. Maintainability is the ability of the system to undergo changes with a degree of ease. These changes could impact components, services, features, and interfaces when adding or changing the functionality, fixing bugs, isolate and correct defects, and meeting new business requirements. In some cases, maintainability involves a system of continuous improvement; using the maintenance experience in order to improve reliability of system. The following are some low-level attributes belong to maintainability [46].

Application standards: are the principles and guidelines that the organization adheres to when they develop an application to guide the development of the code. These principles (standards) are not only applied at the quality assurance gate, but also at the start of code design. Some examples of application standards are: applications should be integrated using standard interoperability methods and protocols, application should be designed with security in mind, applications should be accessible via Application Programming Interfaces, application should be flexible in that changes can easily be made, application should be scalable etc [47].

Configuration management: is the process of recording and updating of information that describe an enterprise's hardware and software. The information includes the software packages versions and updates have installed and the network addresses of hardware devices. Configuration management has an advantage in the ability to review the system changes and to make sure that these changes have no impact on any other components or systems. In software development process, configuration management is used to help the developer to keep track of the source code, documentation bugs, and the change logs have been made on the code [48].

Operational management: is a management field that concerned with the development and implementation of efficient business process used to produce services. It involves the responsibility of ensuring that business operations are efficient in term of resources used.

Code complexity and volume: code complexity is a measurement for how the program is complex. It measures the number of linearly independent paths through a program's source code. Code volume is a code size (Lines of Code); code size and complexity are important because it indicates the size of computing resources we need to execute the program, and how easily the system can be maintained or modified.

➤ **Affordability**

Resource cost: the cost of the elements that used to perform the activities in the overall system. In IT systems, it includes the cost of the equipments and hardware, infrastructure, the salaries of technical and developers (human resources), and software license costs.

Software enhancements cost: it is the cost that organization expends after it commences its application to the market. The need to retain the organization competitive edge impose the company to add new features and functionalities, improve the performance, fix bugs, or migrate the application to new platforms.

Licensing: it is the allowance to use the software, platform, and other services used in the application, and normally it is not free. Licensing has different kinds. Some are depending on the number of the machines that will run the software whereas others are based on the number of users that will use the software [49].

Development cost: the cost of create a certain application starting from the decision taken to develop a certain system, and ending with the release of the formal system or production.

➤ Scalability

The application is scalable when it is always ready to be enlarged to handle the increasing in load without impact the performance. For the organization, scalability is one of the main reasons to move their application to the cloud. The following is some parameters of scalability quality attribute [46].

Replication: Many applications require a high scalability and availability; replications can be used as a solution in both data and computing. By adding more replicas for data and computing, the reliability, fault-tolerance, and accessing will be improved. An example of replication approach is adding geographically distributed servers such that each server supports read requests from closer regions, and thus increase both the scalability and availability of the application [50].

Caching: it is a temporary storage of web content, such as HTML pages, images, files and web object that previously requested from the origin server. Caching increase the scalability of the system by offloading the origin or web or application server and decreasing the time required to deliver the content to a client.

Pooling: it is a resource management term refers to the grouping computing resources such as networks, servers, storage, applications and services to serve multiple users, maximizing the advantages and minimizing risk to the users. [33]

Scale out: it is called also horizontal scaling. When scaling horizontally, more machines are added to the system that runs the application. An example of scaling out is the scaling out from one web server to three [46].

Scale up: it is called vertical scaling. When scaling vertically, more resources such as CPU, memory, and disk are added to a single system [46].

Hardware load balancing: Load balancing can be provided through specialized devices, such as a multilayer switch that can route packets within a cluster.

➤ Reliability

The application is reliable when it is able to continue working in the expected way over time. Reliability is measured as the probability that the application will not fail and it will accomplish its function under unforeseen conditions and for a specified time interval. Reliability is one of the common quality attributes for any application, and thus, a reliable application is a necessity for organizations who want to move their applications to the cloud. The following is some low-level attributes related to reliability high level attribute [46].

System performance: is measured by the amount of work that performed by the system compared to the time and resources used. Depending on the application itself, system performance may include one or more of the following: short response time for a piece of work, high throughput, optimal usage of the resources, and high availability of the application.

Fault tolerance: is the ability of the system to continue working in case if some components of the system failed. In the fault-tolerant systems, when a component fails, a backup component can immediately replace it without loss of service.

Startup and automatic recovery: startup recovery is the ability of system to detect certain problems that may prevent the system from starting correctly and fix them, while the automatic recovery is the process of detection the system failures automatically by certain tools such as watchdog timer, and attempt recovery [51].

➤ **Interoperability**

Interoperability is the ability of the application to work successfully with other applications by communicating and exchanging data with other applications that can be developed by external parties using different platforms. The key considerations for interoperability are communication protocols, interfaces, and data formats. The following is sub-attributes belong to Interoperability [46].

Architecture compatibility: some patterns are correlated with each other. The compatible architectures are easy to interoperate. In the next section we will show some of compatible architecture patterns.

Ease integration (APIs): an application programming interface (API) is a set of programming specifications and instructions that help to access the application. The API is considered as a software-to-software interface, it is used in order to facilitate the interaction between the applications.

➤ **Availability**

Availability is the proportion of time that the system is working and functional. Availability is measured by the percentage of the system in the downtime over a predefined period of time. The following are some attributes that affect system's availability [46].

Uptime requirements: refers to the total amount of time that the system is available for end-use applications. The value is stated as a percent of total scheduled working hours.

Technology/Configuration/Implementation to support availability: some technologies, configurations, and implementations can be used to increase the availability. For example, adding monitoring tool to any system can improve the availability by detection systems error, adding more infrastructures in case of component/hardware failure.

3.1.3 The Contribution of the Application Attributes in the Migration Decision

In this section we will list the attributes that we were able to find a reasonable relation between them and the cloud computing. In the previous section we gave a brief description for every application attribute used to describe an application in the system. As mentioned previously, the contribution of each attribute in the cloud migration decision comes from whether the attribute is supported by the cloud or not. Thus, the following shows some of these attributes and their relations to cloud computing based on some findings extracted mainly from the advantages and disadvantages of cloud, or in others word, what the cloud likes and what does not like.

Online Data Access: this attribute affect the decision when the company wants to use the cloud storage services. If the application requires online access for data, then cloud would be an excellent choice as the company does not need to keep any data on-site, and thus it does not need to move huge bulk of data to their local servers [5].

Offline Data Access: on the contrast to online access data, when an organization requires pure offline access of data, then it is better to keep data locally rather than put it on the cloud [5].

Structured Data: cloud supports structured data by services that provide database-like functionality. However, when the application is a database intensive with huge amount of transactions, this can cause some reliability issues. Structured data requires relational database such as SQL databases; SQL databases are difficult to scale, meaning that they are not natively suited to a cloud environment [52].

Unstructured Data: it is more supported by cloud environment. An example of such data is a file system that stores media files. Unstructured data is non-relational or NoSQL database; it is built to service heavy read and write operations and it is able to scale up and down easily, and thus it is more natively suited to run on the cloud [52].

Indexed/Searchable data: these are more related to relational or SQL databases, which for the previously mentioned interpretation is not supported natively by the cloud.

Transaction Management: the application contains transactions processing is not suitable to run on the cloud for some reasons: the latency issue associated with running transaction processing applications on the cloud, and the biggest reason

which resulted from the distributing nature of the cloud infrastructure in addition to the extra layers of software that virtualization adds make the ability to discover, manage, and govern any new transaction more complex.

Emergency Hotfix: due to the distributing nature of cloud infrastructure, and resources sharing between several clients, the issue of fixing any trouble could urgently happen in the application or making any update to the existing software can be more difficult.

Trusted Relationship with Platform: this attribute is also known as Hardware Authentication. Several cloud providers ensure physical infrastructure secured including servers, routers, and storage devices. Safeguards include the adequate control and monitoring of physical access using biometric access control measures and closed circuit television (CCTV) monitoring and by using trusted platform module (TPM) to authenticate the hardware as explained in the previous section [53].

Application Security Model: organizations want to maintain the level of security that they have when they migrate their applications to the cloud; the challenge is to virtualize as many of protective devices and applications as are available on on-premises servers, such as load balancers, intrusion prevention appliances, and firewalls [54].

Access Control: it is one of the security risks in cloud computing. Many cloud providers apply access as an all-or-nothing proposition “Fine-Grained Access Controls”, meaning that once a user authenticates to the cloud, s/he has the freedom to do a lot of unintentional damage. For example, starting and stopping a virtual server [55]. Thus, a robust access control may not be supported by all cloud providers.

Cryptography: it is necessary to secure the data that will be moved to the cloud. If the data is very sensitive so it is better to encrypt it before moving to the cloud or ensure that the cloud encrypt the data with frequently changing keys that managed also by the customer. Due to this analysis, cryptography attribute is not very supported by the cloud.

Authorization/Authentication: the enterprise authentication and authorization framework does not naturally extend into the cloud [56]. The multi-tenancy architecture of the cloud force different mechanisms for authentication and authorization; cloud application services multiple tenants using one server and one application instance. Thus, there will be two kind of identification (authentication): tenant and user identification. Tenant can be identified by the domain name of the email address for example, while the user identification can be done by using special user database for every tenant (company).

Configuration Management: there are many configuration management challenges introduced when moving the application to a cloud computing environment, [57] puts them in two categories: abstraction and dynamic change. The abstracted design of the cloud (cloud stack) and the dynamicity in cloud environment (dynamic provisioning and de-provisioning) creating challenges to maintain an effective configuration management. Thus, configuration management may not be supported by solid solutions from all cloud providers.

Operational Management: operations management faces some challenges when moving the application to the cloud. The challenges come from the virtualized and dynamic nature of the cloud infrastructures [58]. Thus, operational management may not be supported by solutions from all cloud providers.

Application Code Complexity and Volume: according to [59], moving large-scaled business applications can be scary and need a lot of research. If the application is huge, then it will be subtle to maintainability problems related to operational and configuration management. Thus, the application has high volume and complex code is not recommended to move to the cloud.

Resource Cost: due to cloud computing characteristic: virtualization, multi-tenancy, and economies of scale. The resources cost are significantly reduced and, only the operational cost which is paid.

Development/ Software Enhancement Cost: the pay-as-you-go payment models that most of cloud providers offer, making the using the cloud platforms that also include development environments as services requires less cost. Accordingly, these attributes are positively matched to the cloud.

Licensing: it is negatively matched to cloud computing. When the application's licenses are only tied to physical core; software licenses will not be compatible to work with virtual machines. Thus, licensing can be one of the challenges when moving the application to the cloud.

Replication: cloud is tightly related to replication technology. Replication can be performed for data or computing nodes. When the main replica crashes, one of the other replicas compensates the failed one instantaneously. Moreover, these replicas can be used in the load distribution so the application can handle the increasing demands.

Caching: cloud supports caching by adding another storage layer "distributed data cache" makes data access quick.

Pooling: Cloud is a shared pool of computing resources such as networks, servers, storage, applications, and services. Pooling is strongly correlated to the cloud environment.

Software/Hardware Load Balancing: replication and loads migration between different nodes in addition to load balancers applied in many clouds can help to cope with this issue. Thus, cloud in general provides good solutions for business that seeks high scalability for their applications.

Scale Out/Up: application migrated to the cloud can benefit from the elastic nature of cloud computing, and therefore, scale-out and scale-up attributes are supported by the cloud with pay-as-you-use purchase model.

System Performance: can be increased when migrate application to the cloud, but it may depend on the architecture of the application on the cloud.

Fault Tolerance: different cloud providers suggest different Fault-Tolerance techniques. Replication and job migration are examples of these techniques that ensure an application fault tolerant. Thus, cloud can be a solution for the application that we want it to be more faults tolerant.

Startup and Automatic Recovery: for the same reasons that support the Fault Tolerance attribute, startup and automatic recovery is supported by the cloud environment.

Architecture Compatibility: the architecture of the application when hosted locally may not apply when moving the application to the cloud. Thus, these changes in the architecture of the applications that are integrated using a specific middleware can affect the integration process when they move to the cloud.

Ease Integration (API): applications integration in cloud environment has some restrictions. With SAAS, latency will be more of an issue, and impoverished APIs may limit integration benefits. Generally, data integration and asynchronous process integration are main restrictions to integrate in the application in SAAS.

Uptime Requirement: many clouds are built using commodity hardware to reduce costs. This results in a reduced availability of individuals cloud resources. Thus, some cloud patterns such as Watchdog are used to cope with failing resources. Therefore, availability can be a challenge in the cloud environment [19].

3.1.4 Migration Decision Calculation Process (Migration Algorithm)

In the preceding section, we have explained the role of each attribute and low level attribute in the migration decision to cloud computing environment, and how it can affect the organizations' decisions when they think to move their applications to the cloud. Every organization has their own combination of these attributes according to how these attributes are more critical for their businesses. For example, some organizations concerns in scalability more than affordability, while others are aware of the security and availability issues. As mentioned before, the first step in the concept is to tell the user (organization) how good or not its decision to move the application to the cloud. Thus, the system takes this preferable combination from different kind of attributes and makes some calculations (Migration algorithm) on them, and returns to the user the score of this combination and the recommended decision that organization should take. The following steps show in details the process to perform these calculations (Migration Algorithm).

1. Every attribute has been given a certain value (weight) represents the contribution of this attribute in cloud computing. In other words, how much the cloud computing supports the attribute. For example, online data access has been given a positive value as it is recommended by the cloud, while offline attribute has been given a negative value as it is not recommended moving the application that accesses its data offline to the cloud. The range of weights varies, for example, scale-out/up attributes are tightly coupled with cloud concept, and so they are given higher value than system performance -for instance- that can depend on some variables such as the application architecture.
2. The weights mentioned in the first point are given for every application attribute. Based on the analysis in (Section 3.1.3). The idea behind these weights is to measure the impact for the attribute in the application migration decision as they will be used in the equation in the point 6; the attribute has a stronger relation with cloud will have higher weight and indeed will be weighted more in the equation. Not all weights are given positive values, some attributes are given negative value and these attributes can decrease the application score in the equation if they have been selected by the user, and consequently, reduce the chance to migrate the application to the cloud
3. The organization can select any number or combination from these attributes even though that some attributes can have conflict contribution in the cloud decision. For instance, user can select both online and offline data access or structured and unstructured data storage which is a conflicted attributes combination.
4. To solve the issue of selecting some conflicts in the selected attributes, we added levels (importance levels) for every selected attribute such as low, high and critical. For example, if the user selects offline and online data attributes, online

attribute can be more important for the application than offline, then the user is able to choose “High” or “Critical” levels for online and low level for offline. In this case and based on the algorithm –we will describe later- the online attribute will have more contribution in the cloud migration decision than offline attribute.

5. In addition to the weights given to the attributes as numerical values, the levels (low, high, critical) are given numerical values as well. Low, high, and critical are assigned to 1, 2, and 3 respectively. Figure 3.2 is a snapshot taken from the recommender system illustrates some attribute and level options selection. Both weights and levels are used to calculate the application score which controls the migration decision. If the user is very concerned of scalability attributes such as scale out, then he can select “scale out” and “scale up” for instances, and also select “Critical” level for them. In this case if “scale out” and “scale up” have 3 as weights, then the total contribution for both attributes in the application score is $3 \times 3 + 3 \times 3$ which equal 18, while if he selects “Low” the contribution of these attributes in the application score will be $3 \times 1 + 3 \times 1$ which equals 6 (much less).

The screenshot shows a web interface for attribute selection. At the top, there is a checked checkbox for 'Authentication' and three radio buttons for 'Low', 'High', and 'Critical'. Below this is a section titled 'Scalability' in bold. Under 'Scalability', there are five rows, each with a checked checkbox and three radio buttons for level selection:

- Replication: Low, High, Critical
- Caching: Low, High, Critical
- Pooling: Low, High, Critical
- Scale out: Low, High, Critical

A 'Submit' button is located at the bottom of the form.

Figure 3.2: Attributes Selection Recommender System

6. From the selected combination from the attributes and their levels, the business logic (migration algorithm) calculates the following:
 - a. The lowest value in the score scale (the lowest application score): it is the lowest score can be obtained from any selected combination the user can select from the available attributes. It is calculated as if the user selects the most negative combination of attributes and levels. To clarify, user only selects the attributes which have negative weights. Moreover, user selects “Critical” as level for every negative attribute. Thus, any application score can be obtained later will not be less than this value. This lowest value represents will be the lowest point in the “ruler” that will measure the application score for the migration decision.
 - b. The highest value in the score scale (the highest application score): it is the highest score can be obtained from any selected combination the user can

select from the available attributes. It is calculated as if the user selects the most positive combination of attributes and levels. To clarify, user only selects the attributes which have positive weights. Moreover, user selects “Critical” as level for every positive attribute. Thus, any application score can be obtained later will not be higher than this value. This lowest value represents will be the highest point in the “ruler” that will measure the application score for the migration decision.

- c. The real score value: it is the real score of the selected combination, and depending on this value the decision is taken to move the application to the cloud or not. The value is calculated according to the following formula:

$$\text{Application Score} = \sum A_w \times L_v$$

Where A_w = the weight of the attribute.
 L_v = the value of the level

7. From the lowest the highest application scores calculated from the previous points, we define the scale “ruler” that can measure the score of the application to move to the cloud or not.
8. To add more flexibility to the score scale, we divided the resulted scale into five periods. Two periods in the negative side, two periods in the positive side, and one in the middle around {0} as follows:
 - The middle period starts with a certain value represent percentage from the lowest score. This percentage has been taken as 10%. The middle point ends with a value equal to 10% from the highest score. The idea behind this small period compared to the other four periods that we want to make a period where the user can decide between move the application to the cloud or not.
 - The remaining scale is divided equally in both sides.
9. When the application’s score comes in the first period that means that the application is not a cloud fit completely, while if the score comes in the last period that means that the application is strongly recommended to move to the cloud. The middle period gives the user the freedom to choose either to move the application or keep the thing as they are. The other two periods: one comes in the negative area and it does not support the migration decision, while the other one located in the positive area and it recommends the user to move the application to the cloud.

3.2 The Second Step in the Concept - After the Migration Decision

In the previous section we have explained the first step in this concept which helps the organizations to take the appropriate decision for migration their applications to the cloud. Recall that the purpose of the concept is to recommend the organizations with the “greenest” cloud application architectures that fit their applications; the second step in the concept depends on the green business process patterns [33]. By explaining these patterns and their relations with the different cloud application architectures, cloud services type, and deployment models, we can draw a strategy that depends on these patterns to lead us to the greenest architectures. Returning back to the first step of the concept, the application’s attributes can interfere in the second step by extracting some relations between these attributes and different cloud patterns, and thus the system can give better architecture solutions by matching the application’s attributes with the cloud patterns together with the green business process patterns.

Before introducing the second step in the concept, it will be better to have a fundamental knowledge about the green business process patterns and cloud computing architectural patterns. Understanding these patterns facilitate the understanding of the relations between them, and thus, getting a clear figure for the backbone of the concept.

3.2.1 Green Business Process Management

Business processes are one of the key possessions of organizations. A business process is defined as an activity or a group of activities to perform a certain organizational purpose and produce specific output. To improve and adapt the business processes continuously, a holistic management approach (Business Process Management) is applied in order to align business processes with the requirements and needs of clients. Business process management (BPM) enables organizations to be more efficient, more effective and more responsive to the changes than other traditional management approaches [33].

Green Business Process Patterns

Organizations nowadays want to be greener and more sustainable. There are several drivers to green computing as mentioned in section 2.4. [33] summarizes these factors to three drivers: (1) the high cost of the energy and the raw materials, (2) governmental regulation to decrease carbon footprints, (3) the public appearance of the organization. Green business process patterns have been identified to present proper solutions for various optimization challenges. To realize those patterns organizations need to introduce or extend their business process management (BPM) architecture to support the identification, determination, and analysis of environmental metrics. There are two different classes of patterns have identified [33]:

Basic patterns extend or change a process by providing explicit sustainable alternative service, products, or resources.

Green Compensation: some processes in the organization are very difficult to change or redesign for some reasons such as the nature of the task itself or other internal limitations. The challenge is to find an appropriate process for compensating the environmental impact without changing the process. To solve this issue, for every unchangeable process is created, a compensation process is created to compensate the environmental impact caused by the original process.

Green Variant: organizations may violate some internal policies when they make their business processes greener, and thus decrease the market share. Organizations want to decrease the environmental impact without changing the functional outcome. To solve this issue, organizations provide another green business process variant that accomplishes the same result and it might be with new different steps, processes, resources, or patterns.

Resource Change: organizations usually select the resources based on their functional capabilities and the cost which are business objectives for the organization, while the ecological objectives are not considered. The need to achieve both business and ecological objectives by decreasing the environmental impact without influencing the other metrics becomes a challenge. The solution is to exchange the resources of certain processes or activities so that the environmental impact of business process will decrease without neglecting the traditional key performance indicators (metrics) of this business process.

Green Feature: changing small feature in the product can have a significant influence on the environmental impact of the product. Analysis of the component or features for the service/product can lead the organization which component or feature can be replaced with greener one with taking the satisfaction of the customer in consideration.

Process-centric patterns achieve a more sustainable process design by directly modifying the structure or the way a process is performed. The following patterns have been identified:

Common Process Improvement: as mentioned previously that the purpose of business process management (BPM) is to improve the business process. Organizations apply different techniques and approaches to manage and optimize their business process. The dimensions of the optimization process are cost, quality, time, and flexibility. When adding the environmental impact dimension, the trade-off between dimensions becomes more complicated. Thus, common process improvement pattern comes to solve this issue by identification of the activities and processes to be optimized, and investigate them against the environmental impact.

Process Automation: reducing the environmental impact of a process or activity by automating it. Automating a process can affect the cost and quality attributes,

therefore the trade-off between cost, expected quality and the environmental impact has to be addressed in details, in addition to ensure the global integration with other processes and activities that depend on the automated process/activity.

Human Process Performance: some business processes produces high carbon emissions especially those activities/processes are enacted by machines that cause environment pollution; replacing these processes/activities with human executed activities can result in greener and more environment friendly business process.

Outsourcing: In the enterprise, the services and products are usually not provided from one organization; services can be provided from more than one organization integrating with each other to achieve a certain goal better than single organization can achieve. To improve the existing choreography, the business partners can be swapped if the service level agreements have been met and ecological impact has been decreased. Outsourcing some activities from their owner to another partners can perform the same activity more efficiently can reduce the carbon footprint, and thus improve the environmental impact.

Insourcing: in contrast to outsourcing; in some cases a centralization of certain activities can affect the environmental impact positively. The challenge comes from the integration between the outsourced processes. The integration achieved in two aspects: the functional layer and the efficiency/performance of the process from a global point of view. Insourcing provides a solution for this issue by moving the processes or services utilized on a third parties and integrate them in-house or in a cloud provider.

3.2.2 Cloud Application Architectural Patterns

As mentioned previously that the output of the recommender system has been developed is to recommend the organization with the most energy efficient architectures the organization should apply when migrate. [19] defines a set of cloud patterns; these patterns used to abstractly describe solutions that can be offered by cloud services. [19] also derives reoccurring architectural patterns on how to design, build and manage application that used cloud services. The abstract architectural patterns can be applicable to challenges that developers of cloud application face today regardless the underlying used technologies. One of these challenges the thesis argues is finding the appropriate energy efficient patterns applicable to the organization application. Thus, these patterns have been investigated to find the green potential. The fact that these patterns basically advertise the individual properties of cloud environment such as virtualization and elasticity makes them energy efficient architectures. Later in this chapter, the result of this investigation and the relation with the green business process patterns [33] will be shown.

The following subsections described these patterns as found in [19]. The architectural patterns are categorized into four categories: Basic, Multi-tenancy, Elasticity, and Availability pattern.

➤ **Basic Patterns**

Basic architectural patterns cover the basic structures of the cloud applications. They have emerged from the areas of grid computing, and service oriented architectures (SOA). They are applicable to distributed systems, and thus, can be used in cloud computing. The basic patterns are composite application, loose coupling, stateless component, and idempotent component.

Composite Application: enterprise applications usually interact with other applications (third parties) to use some services and functionalities. Holistic applications are difficult to interact with other applications. Moreover, they are very hard to extend their functionalities and component, and they are inflexible to any change in their components. To face these challenges, composite application pattern used as a solution to divide the application functionality into multiple independent components with specific functionalities for each component. The pattern simplifies the process of integration and extension of the application.

Loose Coupling: the loose coupling architecture is vital for creating robust systems where the failure of some components do not affect the overall system. In cloud computing environment it is the same; by removing the assumptions between the application's components or between the applications to be integrated, the management processes such as scaling, failure handling, or update management can be simplified. Furthermore, by decoupling components, the process of removal failure, or update of one component has minimal or no impact on other components.

Stateless Component: we have mentioned in chapter 2 an important feature for cloud infrastructure which is the dynamic provisioning. The resources are given and taken from the application dynamically. Thus, if the application shall scale out, component instances are added and removed regularly. The challenge comes when the application stores any internal state in a component instant, and then this data will be lost when this component failed or released. The solution is implementing components rely on external persistent storage and do not contain any internal state.

Idempotent Component: applications or components should handle the duplicate messages even though the messaging middleware can guarantee the reliable messaging and exactly-once delivery. This desired behavior of the messages receiver/handler can be achieved by two approaches. The first approach, messages filter at the receiver can be used to drop the duplicate messages depending on a unique identifier stored with every message received. However, a second alternative

approach is designing operations that have the same result/effect when executed multiple times.

➤ **Multi-tenancy Patterns**

With multi-tenancy, devastation relative peak loads done by serving large numbers of customers on same infrastructure and software by sharing the resources between customers achieve better utilization for the resources. To share higher level application components, multi-tenant patterns are required; the term tenant refers to a party (organization) that uses an application. A tenant can represent multiple users share one copy of the application on the tenant's behalf. For example, a tenant can be a company that registers to an application that is used by its employees.

Single Instance Component: a componentized application is provided without individual configuration for each tenant. Application components are configured equally for all tenants. Accordingly, sharing component instances is possible. This is performed by deploying components with equivalent configuration for all tenants only once and sharing it between tenants.

Single Configurable Instance Component: a componentized application is provided to multiple tenants with individual configuration for each customer/tenant. Customers often need to apply individual configurations to shared components. For example, using individual database schema for shared database between multiple tenants. This is performed by deploying one component that uses individual configurations for each tenant.

Multiple Instance Components: a componentized application is provided to multiple tenants that may configure it and deploy individual instances. When an application shall be instantiated for multiple tenants, some of the application's components cannot be shared due to some law the preventing tenants from sharing resources or handling some critical functions with others. Also, tenants may integrate the application with their individually developed applications. To solve these issues, individual component implementations and configurations are deployed for each tenant.

➤ **Elasticity Patterns**

Dynamic resources allocation is one of the main key features of cloud computing. The resources can be assigned or freed from the application as needed. This includes three aspects: first, applications have to be monitored externally or internally to know if the application is over utilized or underutilized; second, resources provisioning is based on this information; and third, applications have to cope with addition and removal of resources.

Map Reduce: querying data capabilities are limited in many cloud computing offerings in order to improve the performance and scalability. This leads to larger data sets being returned to the applications that need to be processed efficiently. Executing of complex queries is distributed among multiple computing nodes for the consideration of scaling-out of cloud computing.

Elastic Component: the elastic infrastructure of cloud is dynamic, and consequently, there is a necessity to fully benefit from this feature. Application components that are scaled out have to be adjusted automatically depending on changing workload. As mentioned in chapter 2, the pricing model in cloud is pay-as-you-go model, and therefore, the need to such management process from the resources become more significant because it affects directly the running cost of the organization. To achieve this, automatic monitoring is adjusted to monitor the utilization of the computing nodes that host the application components and align their numbers using the provisioning functionality provided by the elastic infrastructure.

Elastic Load Balancer: to benefit from elastic infrastructure regarding adjusting of resources numbers to experienced workload. Requests sent to an application are good indicator of experienced workload, and thus, using the requests number in scaling decision. To achieve this, an elastic load balancer is used to determine the required resources based on the number of requests and provisions the required resources accordingly using the elastic infrastructure's API. Each computing node is determined to handle a certain number of requests, and thus, the number of resources can be computed.

Elastic Queue: is used to adjust the application components that are scaled-out depending on the number and type of messages the asynchronous requests contain. The elastic queue determines the number of the computing nodes to be provisioned. The computing nodes host the application components that handle different message types stored in the elastic queue. To speed up this process, images for application components in the image database in the elastic infrastructure.

➤ **Availability Patterns**

Many clouds are constructed using low-cost hardware (commodity hardware). This leads to a reduced availability of individual cloud resources. Thus, cloud application have to be designed in a way that avoids availability problems of cloud resources, copes with failing resources, and to guarantees the required availability.

Watchdog: Applications need to be highly available even if they are built upon a cloud infrastructure that offers a low availability computing resources. To handle this issue, the application's architecture has to be modified in order to cope with such resources. Thus, more redundant computing nodes doing the same functionality with a

reliable messaging middleware between the nodes are added. Additionally, compute nodes are monitored and replaced with one of added nodes in case of failure.

Update Transition: the hosted applications need to be updated from time to time. The updates include application's components, middleware, operating systems etc. some applications provide very critical business functionalities and those required to be available all the time including the update time. Thus, the transition shall be achieved with minimal transition time. To realize this, additional compute nodes are provisioned containing the new application's component or middleware versions additionally to the old versions. After update completion, the old computing nodes switched off.

3.2.4 Pattern Based Application Development

In this thesis we not only assist the organization selecting the energy-efficient cloud application architectures, but also we help it to choose the most appropriate cloud environments, storage patterns, and communications patterns. In this section, we will introduce a method to obtain a set of applicable patterns from given requirements and express the relations between the patterns in a decision recommendation table. In the concept, this recommendation table is helpful to determine the cloud environments, storage patterns, and communications patterns that correlate with the recommended green application architectures. The decision recommendation table can also help to extract more correlations between the green patterns and cloud application architectures patterns. In Section 3.2.5 we will identify some correlations defined in [33] and from the definitions we extract more relations with other patterns; in the decision recommendation table there are several correlations between different cloud patterns and few of application architecture patterns have strong cohesion relations with some green business process patterns. From both decision tables we can get more relations in a way looks very similar to transitive relationships.

Populating a Decision Recommendation Table

In Table 2 we will propose the interrelations between the patterns based on their definitions in the previous sections. The decision recommendation table contains patterns interrelations of three different types [20]:

- 1- The strong cohesion relation (+) states that one pattern is likely to be correlated with the related patterns. For example, in public clouds where resources have low-availability, patterns that improve the availability are highly recommended to use with such as watchdog pattern. Another example, Map Reduce pattern aims to increase the performance of complex queries on large unstructured datasets, and thus, blob and NoSQL storage patterns are highly recommended to use with Map Reduce pattern.

- 2- The exclusion relation (-) states that the two patterns cannot be combined together. When a pattern provides a solution for the same issue that another pattern can provide, then there is no need to combine these two patterns together. For instance, map reduce is used to ensure the exactly-once delivery model by handling the duplicate messages. In the same direction, Reliable Messaging provides the same reliable communication service. Thus, there is exclusion between Map Reduce and Reliable Messaging.
- 3- The underdetermined relation (o) states that there is specific relation between the two patterns, neither strong cohesion nor exclusion. For example, there is no specific connection between Stateless or Idempotent component patterns with any cloud type.

At the beginning, the relations between all patterns initially set to undetermined (o), and then all related patterns are obtained from the description of each pattern. For example, if a pattern uses the description of another pattern in its context, then they are correlated to each other. This is the case of NoSQL storage setting the context of Map Reduce pattern, and Message Oriented Middleware setting the context of Idempotent Component pattern. In the same fashion, a pattern may state that it is cannot combined with another pattern. For instance, in the watchdog's context mentioned that the application is build upon low available resources (High availability with unreliable Compute Nodes), and thus, High-Available Compute Node is not related to Watchdog pattern. The relations found in Table 2 are obtained from the literature [20].

The following table shows some of these relations between the cloud architectural patterns represented by the rows of the table, and cloud types, communication, and storage patterns and represented by the columns in the table.

		+ : strong cohesion - : exclusion o : undetermined												
		Composite Application	Loose Coupling	Stateless Component	Idempotent Component	Map-Reduce	Elastic Component	Elastic Load Balancer	Elastic Queue	Watchdog	Update Transition	Single Instance Component	Single Configurable Instance Component	Multiple Instance Component
Cloud Type	Public Cloud	+	+	o	o	o	+	+	+	+	+	o	o	o
	Private Cloud	+	o	o	o	o	+	+	+	o	+	o	o	o
	Hybrid Cloud	+	+	o	o	o	+	+	+	+	+	+	+	+
	Community Cloud	+	+	o	o	o	+	+	+	o	+	+	+	+
Cloud Storage Patterns	Block Storage	+	+	+	o	o	+	+	+	o	+	o	o	o
	Eventual Consistency	+	+	o	o	o	o	o	o	o	+	o	o	o
	Relational Data Store	+	+	+	o	o	+	+	+	o	o	o	o	o
	Blob Storage	+	o	+	o	+	+	+	+	o	+	o	o	o
	NoSQL Storage	+	+	+	+	+	o	o	o	o	o	o	o	o
Cloud Communication Patterns	Message Oriented Middleware	o	+	+	+	+	o	o	+	o	+	o	o	o
	Reliable Messaging	o	o	o	-	o	-	-	+	o	o	o	o	o
	Exactly-once Delivery	o	o	+	+	o	-	-	+	o	o	o	o	o
	At-least-once Delivery	+	+	+	+	o	-	-	+	o	o	o	o	o

Table 2: Decision Recommendation Table

Identification of Applicable Patterns

The relations between different cloud patterns describing cloud types, cloud service models, cloud communication services, cloud storage services, and cloud architecture patterns help the user to identify the applicable patterns to the application. The user selects patterns that describe the environment in which his application will be developed or deployed. In the recommender system these patterns are the green business process patterns [33]. Based on these selections, cloud application architecture patterns are recommended to the user. The architect then can select the relevant patterns from the recommendation decision table that can be used and omit those cannot be used. The system can do that automatically by recommending the user with the proper cloud type

and the storage patterns for the application. On the other hand, user can perform this process manually, instead of selecting the green business patterns, user can select directly from the decision table the proper patterns he sees and then he follows the interrelations between different patterns to identify the applicable pattern to his application. However, the resulted patterns in this case are not necessary to be the greenest for his application.

As a result of a selection, four different sets of patterns can be discriminated: (i) patterns that are selected from the user, (ii) patterns that are likely to be related to the selected patterns with strong cohesion, (iii) undetermined patterns, (iv) and patterns that cannot be used. The user can make refinement operations iteratively on a set of applicable patterns to find out the most effective patterns for his concrete use case.

For example, a user starts by selecting the public cloud pattern making his decision depending on that the security in his application is not that critical. Further, the user selects NoSQL to be used a storage pattern as his data is not structured. Based on the user selections for the cloud types and cloud storage patterns, cloud architecture patterns are recommended for implementation. The Composite Application pattern is recommended due to the strong cohesion to the public cloud pattern and Map Reduce pattern is recommended because it has a strong cohesion with NoSQL pattern. If the user selects more recommended patterns, he can evaluate the strong relations between these patterns to recommended further patterns. Iterative execution of these steps leads to a user-driven refinement of the set of patterns to be used in a concrete solution.

3.2.5 The Relations between the Cloud Architectural and Green Business Patterns

The correlation of cloud architectural patterns and green business patterns is the key idea of the concept. A correlation is required to guide developers and architects during the identification of how to transform the modifications proposed by the green process patterns to the application's architecture and infrastructure. In this concept we suggest that these correlations can help employing the most-energy efficient architectures to the cloud application the organization wants to migrate. Figure 3.3 illustrates three different layers relevant for adapting applications based on the suggested green and cloud patterns: (1) the business process layer, which is focused on changes to the business process model. (2) The application architecture layers, which is focused on the design and implementation of application components and services by the processes. (3) The infrastructure layer, which deals with the resources used for providing the application runtime.

According to Figure 3.3 [33], different kinds of cloud patterns address different layers. Cloud offering such as communication and storage services has been put within the infrastructure layer because they describe the infrastructure resources assigned for a certain application. Application architecture layer describes the architecture of the application components and thus cloud architectural pattern are assigned to it. In this concept, we concern of cloud architecture patterns (architecture layer) as we discuss the

energy efficiency of these patterns from application perspective. We also recommend the user with other kind of patterns (storage patterns) and cloud types beside the appropriate application architecture patterns. We will use also the cloud offering, cloud service models, and deployment models in order to extract more relations between the cloud application architectural patterns and green business process patterns.

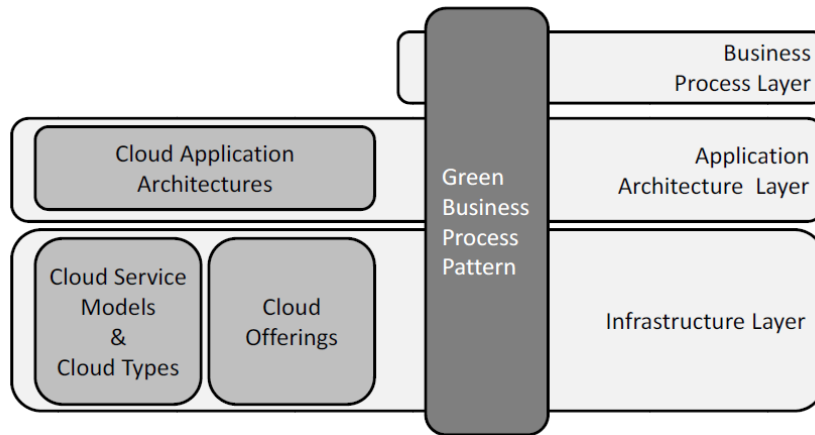


Figure 3.3: Correlation of Patterns

The figure also shows how green process patterns can affect the three layers. For examples, Resource Change pattern can be applied in infrastructure layer by exchanging resources or in the application layer by changing the application’s component. In this section we will provide a complete correlation Table 3 between these green process patterns and cloud application patterns. These correlations can benefit the application developer by helping him to identify the possible implementation solution depending on the cloud pattern, and to identify cloud patterns necessary for adaptation, e.g., when moving components from private to public cloud.

In this concept these correlations are necessary to have a vision for which application architectures can we recommend the organization. The second step of the system allows the user to select green patterns that organization wants to apply to its application’s component, and then according the correlations we can find the “candidate” set of application architectures can be applied for the selected green process pattern(s). We have said “candidate” because this is not the final solution that the user is waiting for. Return back to the first step in the concept and the selected attributes for an application; we will also use these attributes that describe the application in order to provide more “filtered” list of possible application architectural patterns “the greenest patterns” the organization can apply to its application in the cloud. To complete the second step in the concept we have to describe the relations or some relations between different application attributes mentioned in Section 3.1 and the application architecture patterns described in Section 3.2.2; later in this chapter we will present these correlations between the application attributes and the cloud architecture patterns.

		Green Process Patterns								
		Green Compensation	Green Variant	Resource Change	Green Feature	Common Process Improvement	Process Automation	Human Process Performance	Outsourcing	Insourcing
Cloud Patterns										
Cloud Types	Public cloud	x		x		x	x		x	
	Private cloud	x		x	x	x	x			x
	Hybrid cloud	x	x	x		x	x		x	
	Community Cloud	x		x		x	x		x	
Elasticity Patterns	Elastic Component	x	x				x		x	x
	Elastic Load Balancer		x				x		x	x
	Elastic Queue		x				x		x	x
	Elastic Infrastructure		x	x			x		x	x
	Map Reduce						x		x	x
Multi-Tenancy	Single Instance		x		x		x		x	x
	Single Configurable Instance		x		x		x		x	x
	Multiple Instance				x		x		x	
Availability/Basic Pattern	Composite Application	x	x		x		x		x	x
	Loosely Coupled	x	x		x		x		x	x
	Stateless Component				x		x		x	x
	Idempotent Component		x		x		x		x	x
	Watchdog	x			x		x			
	Update Transition				x		x			

Table 3: Cloud and Green Business Process Pattern Relations

The correlation matrix in Table 3 describes which cloud pattern supports a green business process pattern; correlations are marked by an “x”. As described in Section 3.2.2, the application patterns can be divided to different categories: cloud types, Elasticity, multi-tenancy, basic, and availability patterns. In the following paragraphs we provide more details of these correlations in Table 3:

- 1) **Green Compensation:** in green compensation, the structure, the resources and the behavior of the process cannot be modified. Therefore, additional services or components are added to compensate the environmental impact of the existing infrastructures and components. Accordingly, Green Compensation can affect the infrastructure layer, and so all cloud types can provide the compensation services. Green compensation also affects the application components by creating new compensation activity whenever a business process that cannot be changed is instantiated. Thus, component application and elastic component patterns would be suitable for this process model. Moreover, loosely coupled pattern can be beneficial here as we want to ensure that there are no impact and ease of integration for the new compensation activity.
- 2) **Green Variant:** it directly affects the application components. Therefore, the elasticity patterns are the keys for energy efficient usage of resources by dynamic provisioning and de-provisioning of application components. Moreover, basic patterns especially loosely coupled and composite application patterns can be used with this pattern as it includes adding new component to the existing components. Watchdog pattern can be used here to monitor the new component. Additionally, [33] suggests the hybrid cloud as a recommended cloud type as new variant usually includes new environment.
- 3) **Resource Change:** it only affects the infrastructure layer; in this pattern the resources that used by the process are replaced with some have less environmental impact such that the application components do not change. Thus, any cloud offering can provide an elastic infrastructure is able to reduce the energy consumption.
- 4) **Green Feature:** it includes changing some features in an application or organization within the organization. Accordingly, this pattern is related to the private cloud. Moreover, a green feature can be achieved by changing the usage of the cloud application; by employing multi-tenancy pattern the usage changing of the application would be easier since all users share one copy of the application. Additionally, all basic patterns are correlated to this pattern. For example, changing a feature involves exchange one component with a greener one, and therefore, applying stateless component is important here in order not to lose any data “can be very critical data” stored in the old component. Idempotent component can be used to handle duplicate request, and then reduce energy consumption resulted from handling the replicated requests. Furthermore, when we replace a component in the application we need to ensure that the application availability will not be affected in the replacing time for the component. If the replaced component provide very critical business function, then the need for update transition pattern is necessary in order to keep the application operating during the transition time. To ensure the availability of the new component, the watchdog pattern can be used in order to monitor the working of the new component.

- 5) **Common Process Improvement:** affects both of processes' structure and infrastructures used to perform them. Any modification in the structure of the processes or the underlying resources requires set-up new cloud environments. Thus, this pattern has no correlation with application patterns as it applies an optimization solutions to the existing activities
- 6) **Process Automation:** in process automation the environmental impact decreased by automating some processes. That includes creating new application components and defining the infrastructures used to perform them. Therefore, all cloud patterns provide appropriate solutions. To clarify, (1) creating the new components requires to set up new cloud environments, (2) elasticity patterns are required to build componentized application such that switching on and off the components on demand, (3) for saving the resources by sharing them, the multi-tenancy patterns are used, (4) basic patterns can be used as well, e.g., loosely coupled is required to ensure that the new automated process will not affect other components that depend on it, stateless component is also necessary to ensure that no data loss from the replaced process, and (5) availability patterns may be used to monitor the availability and to ensure a smooth components transition.
- 7) **Human Process Performance:** in this pattern, human-centric activities are improved. Since the cloud patterns are applicable to only automated business process, human process performance is not correlated to any cloud pattern.
- 8) **Outsourcing:** reducing the environmental impact for an activity can be achieved by outsourcing it to a third party that can perform it more efficiently. Outsourcing may affect all the cloud patterns except the private cloud. Organization uses a private cloud for those components cannot be shared or distributed for several reasons such as security concerns, and therefore, all cloud patterns can enhance the usage of the resources with the impossibility to apply the private cloud.
- 9) **Insourcing:** this pattern is correlated to private cloud pattern as it aims to run the services within the organization property. Advantages can be gained by using some elasticity, multi-tenancy, and other basic patterns. However, multiple instance component patterns may not be used here as some organizations refuse to share some of their application's components.

3.2.6 The Relations Between the Application Attributes and the Cloud Patterns

In the previous section we have explained the correlations between the cloud patterns and green business patterns and we displayed the relations between them in a matrix form Table 3. In this section we will introduce other correlations are also necessary for the concept toward recommending the user with most energy efficient patterns that user can apply. The correlations are between the application attributes that described in Section 3.1.2 and the cloud patterns. In the second step in the model, the user selects among various green patterns and the recommender returns back a set of cloud patterns. That can be beneficial but it does not exactly match what the user wants; in the first step of the concept the user has already decomposed his application in term of attributes, and thus, it will be very important if there are direct relations between these attributes and the cloud patterns, and so recommending the user with more “accurate” or filtered solutions (patterns). To summarize, the set of patterns the organization will get at the end is the intersection between the cloud patterns that correlate the selected green business process patterns from one side, and the cloud patterns correlate the selected attributes from another side.

Composite Application: In this pattern, the application is divided into multiple components, each one with specific functionalities. Composite application pattern facilitate the extension the application, the integration of the applications with other applications, and management processes, because application components can be treated individually. As a result, usage of composite applications will significantly increase the level in which the application can benefit from cloud properties such as elasticity, pay per use, and standardized management. From the last analysis, composite application is correlated with some application attributes: scale-out, software/hardware load balance, resource cost, ease integration, fault tolerance, and operational management.

Loose Coupling: in loose coupling pattern, the assumptions such as time, space, and address between communication components are decreased. Components communicate asynchronously using middleware e.g. Message Oriented Middleware (MOM) making the components’ functionality more robust. Accordingly, loose coupling enhances directly the availability of the application e.g. uptime requirements.

Stateless Component: in stateless component pattern, the application’s components do not keep any internal state. However, components share external storage to store their states. This leads to flexible scaling such as scale-out. Stateless component is correlated with scalability attributes.

Idempotent Component: this pattern in order to enable the component that receives messages to handle the duplicate messages. An idempotent component can be used to

safely use a message oriented middleware that guarantee at-least-once delivery model (reliable messaging functionality). Thus, this pattern would improve the interoperability between the applications and thus it matches the interoperability's attributes.

Map Reduce: map reduce pattern handles complex queries on large data sets by splitting the data sets into many computing nodes (scatter-gather). This pattern is applied when the used storage solution does not support such queries e.g. NoSQL or BLOB storage. Thus, this pattern correlates with unstructured and persistent data attributes. Furthermore, this pattern is negatively matched with availability attributes because it uses multiple low-availability computing nodes whose availability is often limited

Elastic Component: in elastic component pattern, the number of application components is adjusted automatically on demand using the provisioning functionality provided by the elastic infrastructure. This pattern correlates to scalability attributes such as scale-out and it helps in load balancing by add/remove additional computing node. Moreover, elastic component pattern affects the running cost of the application especially when pay-per-use pricing models are available. Thus, elastic component pattern correlates the affordability attributes such as resource cost

Elastic Load Balancer: in this pattern, the application components that are scaled out are adjusted automatically based on the number of requests. The load balancer treats requests by forwarding them to specified know-number computing nodes. Accordingly, elastic load balancer is correlated to elasticity attributes such as software/hardware load balance, scale-out. Additionally, the system performance can be enhanced by the proper distribution for the requests among the nodes.

Elastic Queue: in this pattern, the application components that are scaled out are adjusted based on the number of asynchronous request. Elastic queue can store requests that contain different types of messages and forward them to the appropriate computing node. Thus, as in elastic load balancer, elastic queue is tightly correlated to elasticity and scalability attributes such as software/hardware load balance, scaling out. Furthermore, since elastic queue stores the asynchronous requests, the interoperability between the applications and the availability of the application can be improved. Thus, interoperability and availability attributes are related to elastic queue pattern.

Watchdog: to solve the issue of the low-availability of the most cloud infrastructure used watchdog pattern is applied. The application architecture is modified by adding more redundant nodes that perform the same functionalities in addition to reliable messaging system between them. Accordingly, solving the unreliability in the cloud infrastructure would improve the reliability and the availability of the application. For example, by adding redundant components the application will be more fault-tolerant

in case a component fails. Moreover, using watchdog timer can also increase the capability for the automatic detection/recovery by monitoring the components and immediate compensation from the available correct nodes.

Update Transition: this pattern is used to perform a smooth update for the components in the way that ensures the availability of the system during the update time. Basically, Update Transition is applied to increase the availability of the application, and so the expected results from using update transition is to improve the uptime requirement of the application.

Single Instance Component: the aim of this pattern is to achieve a better utilization for resources by deploying one copy of application components (pool) with single configuration and share them between multiple tenants. Thus, the runtime cost per tenant can be reduced. Accordingly, as only one instance shared between tenants that lead to reduce the cost of running the application; the cost is not only the resource cost, but also involve the licensing cost as some kind of license depends on the number of machines that host the application. Being one instance installed that also facilitate the maintainability of the applications. For example, bugs fixing or redesigning some components (Application Standards) in addition to the configuration management. Some platform services that used for authentication or user rights management implement the single instance pattern, and thus, some security attributes are correlated to single instance pattern such as authorization/authentication model and access control mechanism.

Single Configurable Instance Component: in this pattern, single instance installed for multiple tenants. However, each tenant has its own configuration. As in Single Instance Pattern, Single Configurable Instance Pattern affects the affordability quality attribute such as resource cost and licensing. Conversely, it increases the complexity of single configurable component implementation and can also cause some reduction in system performance in case the configuration requirements of tenants differ greatly. Thus, this pattern affect the configuration management attribute negatively.

Multiple Instance Components: when sharing components is unfeasible between tenants, then individual component implementations and configurations for each tenant are deployed. This pattern may utilize the resources but it is not to level that Single Instance or Single Configurable patterns do. Multiple Instances Pattern can improve the usability of the system.

The following is Table 4 shows the relations between some application's attributes and the cloud architectural patterns. Wherever matching symbol "x", there is a clear relation between the attribute the cloud pattern. We reached to these relations by the previous analysis that mainly depends on the definition and the context for the patterns and the definitions of the attribute. We also get conclude some relations from table (). For example, structured data is related to a relational data store pattern, and relational data

store has strong cohesion relations with basic architectural and elasticity patterns. Thus, we can find out that Structured attribute has relations with these patterns as well.

App Attribute \ App Pattern	Composite Architecture	Loose Coupling	Stateless Component	Idempotent component	Map Reduce	Elastic Component	Elastic Load Balancer	Elastic Queue	Watchdog	Update Transition	Single Instance	Single Configurable Multiple Instance Component	
Online											x	x	x
Offline													
Structure	x	x	x			x	x	x					
Unstructured	x	x	x	x	x								
Transaction Management													
State full													
Stateless			x										
Persistence					x				x	x			
Emergency hotfix	x								x	x	x		
Trust relationship with													
Applications security model											x		
Malicious code													
Access controls											x		
Remote access													
Auditing													
Cryptography													
Authentication/Authorization											x		
Application standards											x		
Configuration management	x										x		
Operational management	x										x		
Resource cost	x					x	x	x			x	x	x
Development cost	x												
Licensing											x	x	
Software enhancement cost	x												
Replication	x								x	x			
Caching	x									x			
Pooling	x										x	x	
Software load balancing						x	x	x	x				
Scale out	x		x			x	x	x	x				
Scale up	x					x	x	x	x				
Hardware Load Balancing						x	x	x	x				
Technology/Configuration		x		x					x	x			
Uptime Requirement		x		x					x	x			
Auto/Startup recovery	x			x					x				
System performance	x	x		x		x	x	x	x	x	x		
Fault tolerance	x	x	x	x					x				
Architecture compatibility	x	x		x				x					
Ease integration (APIs)	x	x		x				x					

Table 4: Application's Attributes and Application Architectures Correlation

3.2.7 From Application Attributes and Green Business Patterns to Energy Efficient Cloud Patterns

In the previous sections, the concept has been explained in two steps. To connect all information mentioned above and give a clue to what the concept (system) does, a use case scenario, a process flow diagram, and conceptual module have been added:

➤ Use Case Scenario

- 1- A user selects some application attributes and prioritizes them.
- 2- The cloud migration decider applies the migration algorithm and returns the migration decision.
- 3- If the migration is not recommended, user can reselect the attributes.
- 4- If not, user selects the green business process patterns that user wants to apply to the process or activity as illustrated in figure 3.4.

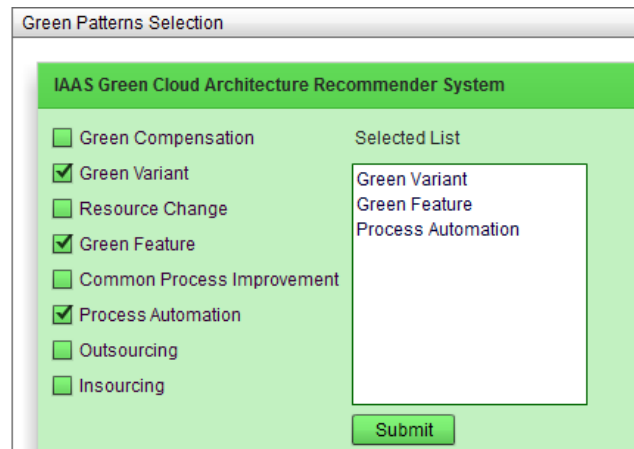


Figure 2.4: Green Process Patterns Selection

- 5- A set of correlated cloud patterns to the selected green patterns are selected based on the correlation matrix Table 3. The set can also be enhanced to match the user's requirements (attributes) by using the correlation matrix Table 4.
- 6- Cloud storage and communication patterns can be returned to the user in addition to the architectural patterns resulted from the previous step. This operation depends on the correlation matrix between different cloud patterns Table 2.

➤ **System Process Flow Diagram**

The following is the process flow diagram of the recommender systems. The diagram shows the border between the two steps explained in the concept as well as the sequence of processes done.

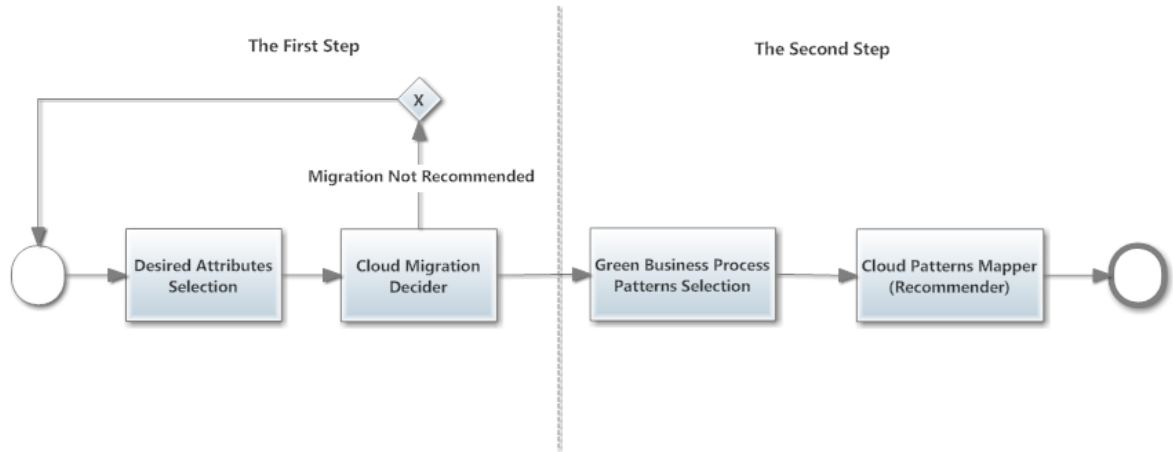


Figure 3.5: Process Flow Diagram of the Recommender System

➤ **System Conceptual Model**

The following diagram shows the relations between applications attributes, cloud application patterns, cloud services, and green business process patterns.

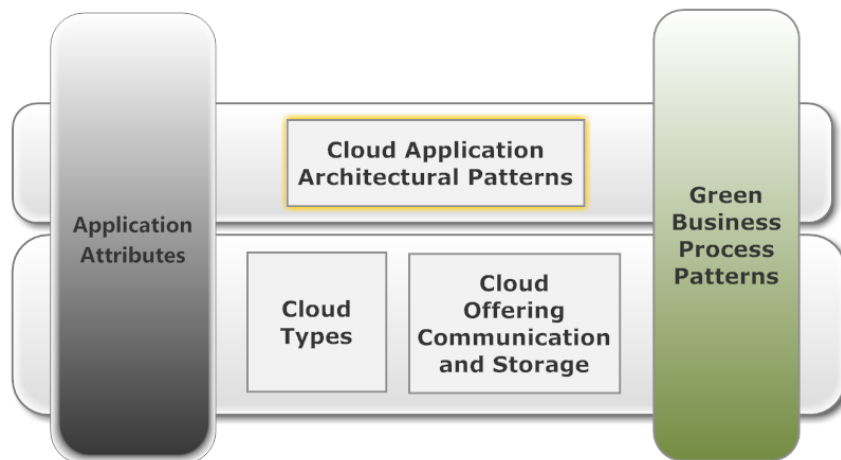


Figure 3.6: Conceptual Model

4 Implementation

The concept introduced in the chapter 3 is translated into a web application recommender system. The recommender system has been developed takes the selected application's attribute from the user as inputs, and returns to the user the recommended energy-efficient application architectures, storage patterns, and cloud types. The same as most of the web-based application, the recommender system is a three-tier application: user interface (UI), server, and database. In this chapter we will introduce in details the architecture of the recommender system, development environment, technology used, and some diagrams.

4.1 Client Side

As known in web applications, a web client application typically has the user interface (UI) of the application and has some form of programmatic control. It runs within the browser, and is responsible to interact with the user actions as shown in the sequence diagram Figure 4.3. Through the Web UI provided in the recommender system, user can easily select the desired application attributes and prioritize them as shown in Figure 3.2 as well as the desired green pattern as illustrated in Figure 3.4. Some UI components have been provided in the Web interface to show the results. In this section, we will introduce the client part of the recommender system such as technologies used and other technical issues.

4.1.1 Technology Used

The user interface in the recommender system has been developed using Flash technology powered by Adobe Flex platform. The reasons behind this selection are several; the main reason is that this technology enables developer to build easily various Internet Applications using very flexible object-oriented scripting language ActionScript 3. With these technologies, it is feasible to build a dynamic user interface, e.g., number of application's attributes can increase or decrease which leads to continuous changes on the UI, and so we need flexible technologies to perform this functionality and make any further enhancement to the system UI not very complex. We will add some information about the technologies that we used in the client side.

Flex Technology: We have developed the client side (Web User Interface) using Flex 4.6 SDK. Adobe defines flex as follows [61]: "Flex is a powerful open source application framework that allows you to easily build mobile applications for iOS, Android, and BlackBerry Tablet OS devices, as well as traditional applications for browser and desktop using the same programming model, tool, and codebase". Flex SDK is the software development kit for development and deployment of Rich Internet Application based on the Adobe Flash platform.

Flex is commonly compared to some technologies such as Ajax, XUL, JavaFX, and HTML5. The typical workflow for Flex is as follows:

- Define an application interface using a set of pre-defined components (form, buttons etc)
- Build a user interface design by arranging the pre-defined component.
- Use the styles to define the visual design.
- Add the dynamic behavior by a scripting language.
- Define and connect to data services or web services as needed.

To write applications in Flex, Flex framework provides two languages: MXML and ActionScript. MXML is an XML-based declarative language used to layout user interface (UI) components. MXML can be used in non visual aspects of an application, such as defining data bindings between user interface components and data sources on the server. ActionScript is used as a scripting language with MXML. It is an object-oriented language and a dialect of ECMAScript (a superset of syntax and semantics of the languages as JavaScript). The compiled file from the source code contains mxml and action script is a SWF file (Shockwave Flash) that can be embedded inside normal HTML pages or JSPs.

4.1.2 Technical Issues

As shown in the sequence diagram Figure 4.3, when user selects desired attributes; the “programmatic control” that is written in ActionScript takes these selected attribute, stores them, and prepare the parameters to send to the server. The parameters include the application attributes and their priorities (low, high, and critical) are sent to the server as pairs from the attribute and its priority. To connect the client side with the server, HTTPService connection component has been used. HTTPService is a defined component in ActionScript 3 used to connect to any servers technology [61]. It handles the request object that will be sent to the server as well as the response object that will be received from the server. HTTPService is defined in Flex code as follows:

```
<s:HTTPService id="httpXmlDataService"
  url="http://localhost:8080/ThesisWork/CloudDecisionRecommender" resultFormat="e4x"
  result="resultHandler(event)" fault="faultHandler(event)"
  useProxy="false" method="POST" />
```

The “url” attribute represents the address where the request will be sent to be processed. In the code snippet shown above, the “url” attribute denotes to CloudDecisionRecommender which is the program that will execute the migration algorithm as shown in the sequence diagram Figure 4.3. The “result” attribute has the name of the function that will handle the response when comes from the server. It is also shown in the code snippet that POST method has been used to fix the issue of query string length when GET method is used. The response return in an XML data format; and the data is parsed in the function defined by “result” attribute. The same sequence is

followed when the user selects the desired Green Process patterns to get the recommended related Architectural patterns.

4.2 Server Side

In the server side, the business logic of the recommender system has been implemented. The business logic (1) exchanges data between the database and the user interface, (2) implements the migration decision algorithm in section 3.1, and (3) retrieves the related patterns from the database, analyzing them, and returns the most relevant cloud patterns back to the client.

4.2.1 Technology used

Java Servlet programming language has been used in order to develop the recommender system server components. Servlet is a Java code that runs in a server application to answer client requests. Servlet uses the standardized JAVA APIs. Additionally, it has its own APIs such as `HttpServletRequest` that used to read client requests and generate the response. Eclipse integrated development environment (IDE) has been used as a development tool for Java Servlet. Eclipse is an open source java application platform has major advantages over other IDEs. It is widely used among most companies nowadays.

Apache Tomcat has been used as a web server where the recommender system server components reside and run. According to [62], Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies. It powers numerous large-scale, mission-critical web applications across and diverse range of organizations.

4.2.2 Technical Issues

Two Servlet programs have been developed to execute the business logic of the recommender system: The first Servlet serves the first step of the concept that explained in section 3.1 which recommends the user with the migration decision. As shown in the sequence diagram Figure 4.3 `CloudDecisionRecommender` Servlet executes the migration algorithm: it reads the client request, parses the parameters which include the attributes and their priorities, builds the queries that will be executed in the database, and finally it uses the weights and level values retrieved from the database to calculate the application score on which the migration decision depends. The second Servlet implements the second step which recommends the user with the energy efficient cloud patterns. As shown in the sequence diagram, `ArchitectureRecommender` Servlet reads the client request, parses the parameters which include the desired Green Business Process patterns, uses the selected attributes from the first Servlet, and builds the query that will retrieve cloud patterns that are relevant to both of green patterns and application's attribute.

4.3 Database

The place where all data includes the attributes, cloud patterns, business process patterns, and their relations used in the recommender system are stored. We preferred to save all the data required in the database even those used in the first step calculation to add more flexibility and extensibility to the system.

4.3.1 Technology used

MySQL Database has been used in the system. MySQL has many advantages mentioned in [63]: it is an ACID compliant database with full commit, rollback, crash recovery, and row level locking capabilities. MySQL also improves the scalability, performance, and delivers the ease of use that has made MySQL a preferable database solution. WampServer has been used as a database server. WampServer is a Windows web development environment. It allows the user to create MySQL database and to use PHPMyAdmin and SQLiteManage to easily manage the database.

4.3.2 The Entity Relationship Diagram

Figure 4.1 is the entity relationships (ER) diagram of the database of the recommender system. It explains the relations between different kinds of patterns used in the system.

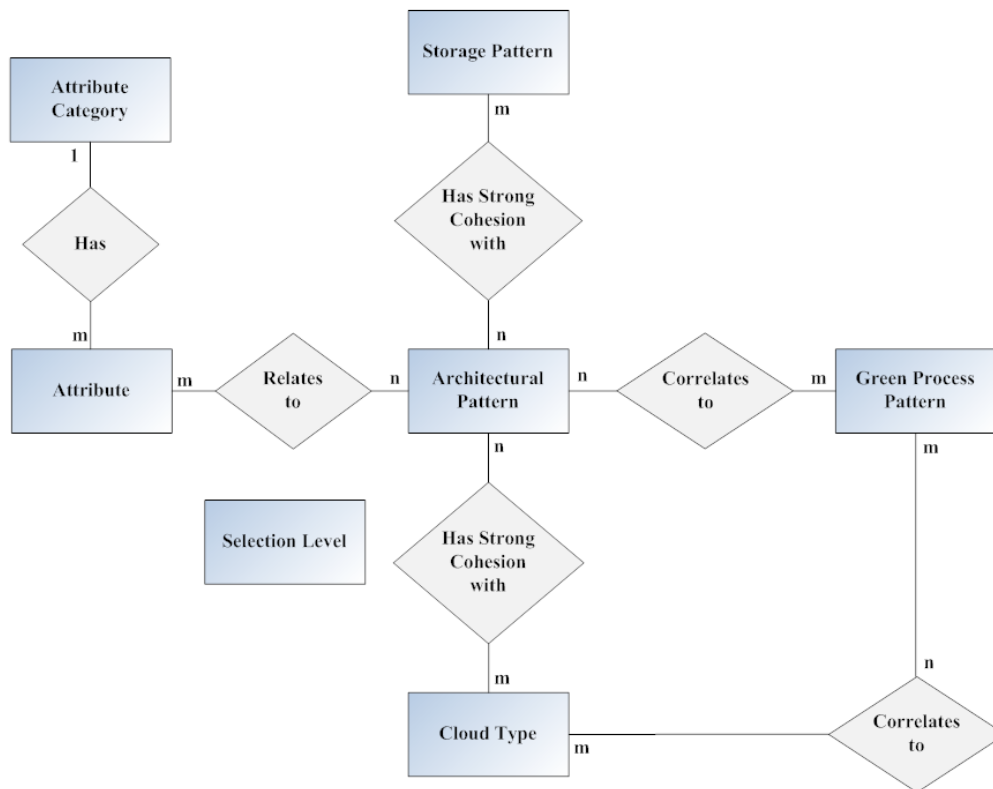


Figure 4.1: ER Diagram of the Recommender System Database

4.3.3 The Tables and the Attributes Description

In the previous subsection, the entity relationship diagram of the system's database has been presented. In this subsection, the description of the database's tables and attributes will be provided. The relations between some tables will be explained as they represent the relation tables shown in Chapter 3. These relations are which the concept primarily depends on. The following is the description of some of recommender system tables:

Table		Table/Attribute Description
ATTR_CATEGORY		Stores the high level application quality attributes information such as Data, Security etc.
Attribute	CAT_ID	Unique incremental number (Primary Key)
	CAT_NAME	String data represents the name of the attribute
ATTRIBUTE		Stores the low level application's attribute for every high quality attribute.
	ATTR_ID	Unique incremental number (Primary Key)
	ATTR_NAME	String data represents the name of the attribute (Display Name)
	ATTR_NAME_ID	Summarized name used as an identifier (parameter names) in the query string that comes from the client request.
	DESCRIPTION	Textual data describes the application's attributes
	CAT_ID	Foreign key to join the attribute with category table
	WEIGHT	Signed value measures the contribution of an attribute in the migration decision.
CLOUD_ARCHITECTURE_PATTERN		Stores the cloud architectural patterns data
	ID	Unique incremental number (Primary Key)
	NAME	Textual data represents the display name of the pattern
	DESCRIPTION	Textual data describes the pattern
CLOUD_STORAGE_PATTERN		Stores the cloud storage patterns data such as NoSQL and Blob storage
	ID	Stores the cloud storage patterns information
	NAME	String data represents the display name of the pattern

	DESCRIPTION	Textual data describes the pattern
GREEN_PATTERN		Stores the green business process patterns data
	ID	Unique incremental number (Primary Key)
	NAME	String data represents the display name of the green pattern
	NAME_ID	Summarized name for the pattern used as identifier (parameter name) in the client request
	DESCRIPTION	Textual data describes the pattern
	CLASS	String data describes the class of the green pattern: basic pattern or process centric
GREEN_ARCH_PATTERN		Junction table used to join the architectures with the green patterns tables such that translate the relations between patterns described in Chapter 3 to database records. Figure 4.2 illustrates one of the junction tables in the recommender system database.
	ID	Primary key of the record (Relation ID)
	GREEN_ID	Foreign key to join the green patterns table
	ARCH_ID	Foreign key to join the architecture table
SELECTOR_OPTION		Stores the priorities of the application's attributes (Low, High, and Critical) and their numerical values that used in the migration algorithm.
	ID	Unique incremental number (Primary Key)
	NAME	The name of the level (Low, High, Critical)
	VALUE	A numerical value for the level.

Table 5: Database tables and attributes Description

	ID	ARCH_ID	GREEN_PATTERN_ID
<input type="checkbox"/>	5	11	2
<input type="checkbox"/>	6	12	2
<input type="checkbox"/>	7	15	3
<input type="checkbox"/>	8	11	4
<input type="checkbox"/>	9	12	4
<input type="checkbox"/>	10	6	6
<input type="checkbox"/>	11	7	6

Figure 4.2: Junction Table from the Recommender System Database

4.4 Recommender System Sequence Diagram

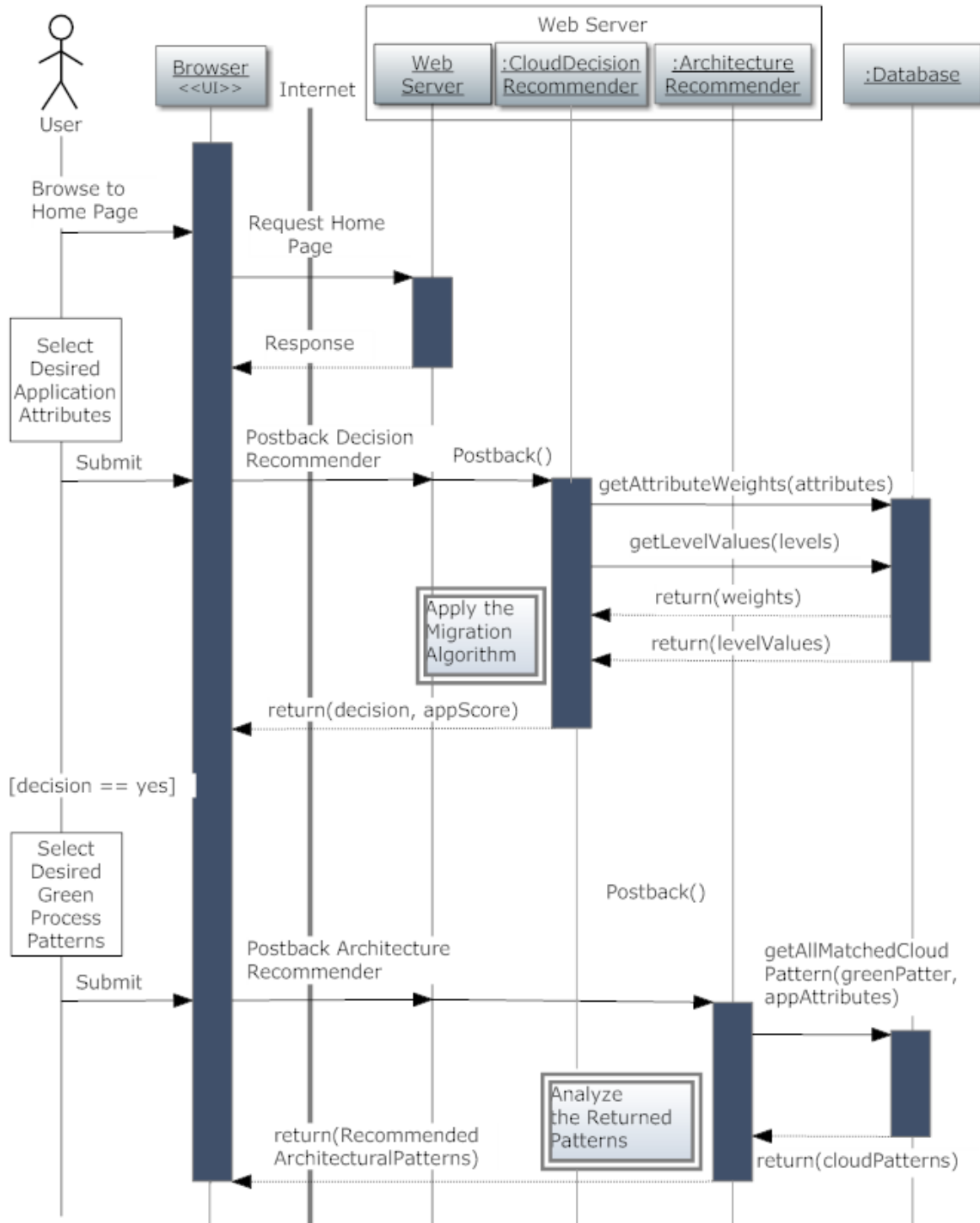


Figure 4.3: Sequence Diagram of the Recommender System

5 Conclusion and Future Work

5.1 Achieved Results and Discussion

The interest in green computing and cloud computing has increased significantly in the last few years. Organization recognized the importance of reducing their carbon emissions and improving their environmental impacts; the high energy cost and the public appearance of the organization are becoming very critical drivers for the moving to the green IT especially in a world complains the climate change, pollution problems, and difficulties in providing the energy resources. Cloud computing is considered one of technologies that organization can employ to achieve their new trends. However, cloud computing does not achieve this trend directly; several contradictory researches argue the energy efficiency of the cloud computing. Cloud computing is a complete system consists of some elements. To achieve the energy efficiency in cloud computing, individual approaches and techniques have to be performed in each element of the cloud computing system including application, network infrastructure, hardware components, and datacenters. In this research, we focused on the energy efficiency of cloud computing from the application perspective exploiting the inherent properties of cloud computing such as the elasticity, dynamicity and economies of scale to identify energy-efficient cloud computing solutions and application architectures. For example, multi-tenancy architecture exploits the property of resources sharing to reduce the IT infrastructure in the tenant by installing one instance of the application used by all users, and thus, reducing the power used to run the application if it is installed on individual machines.

In this thesis, a concept supports the organization choosing the suitable energy efficient cloud application architectures has been developed. Prototypically, a web application recommender system has been implemented. The concept has been presented in two steps: the first step is the cloud migration decision. In the first step the organization can use the concept in order to decide if it is better to move or develop its application to the cloud or not. We took into consideration that some applications do not fit the cloud and they are better to keep them on-premise infrastructure. The application attributes and their relations to the cloud environment have been concluded and gathered in a matrix form. These relations are used to calculate the score of the application that decides if the application is recommended to move to the cloud or reside in the local infrastructure. The algorithm depends on the weights of these attributes (how much these quality technical and business attributes can be achieved by the cloud environment) and the priorities (the importance of the attribute) to calculate the appropriate migration decision based on the selections of the organization. The second step of the concept depends on identifications of energy efficient cloud architectural patterns and green business patterns, and then finding the correlation between these patterns. The relations are concluded from the definition, context, and the usage of these patterns; when the organization applies one of the green process patterns in the design of a business process or activity, the concept facilitates identifying the most energy efficient cloud patterns that suitable to architecture the application to accomplish this business process or activity. In this step also we took into account the application attributes that describe the technical and the business and technical requirements of the application. A deep analysis has been made to find the

relations between these attributes and the various cloud patterns. Consequently, by these two steps, the system ensures the organization not only selects the most energy-efficient application architectures, but also the architectures that fulfill the organizations needs.

5.2Future work

The future work or enhancements on the system can be performed by two ways: internally and externally with respect to the system/concept. Internal enhancement can be accomplished by identifying more energy efficient cloud patterns, adding more attributes, and finding more correlations between the different patterns along with application attributes, this results in proposing more accurate solutions to the organizations. External work or enhancement which is out of the scope of this research answers the possibly suggested question by the organizations: where should we migrate or build our application? Which cloud providers can be suitable for certain architecture, and which provider is the most environment-friendly? These questions can be answered by adding a third step. The purpose of this step is to recommend the organization with the suitable cloud providers that fit their applications, support the energy efficient patterns, and run the application with minimal consumed energy.

Bibliography

- [1] Leymann, F. (2009): Cloud Computing: The Next Revolution in IT. In: Photogrammetric Week.
- [2] Nowak, A.; Leymann, F.; Schumm, D.; Wetzstein, B. (2011): An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering. In: Proc. of the ICTGLOW 2011.
- [3] Ambtman, E.L. (2011): Thesis: Green IT Auditing. VrijeUniversiteit. Netherland.
- [4] Fujiwara, I. (2011): Study on Combinatorial Auction Mechanism for Resource Allocation in Cloud Computing Environment. A dissertation submitted to the Department of Informatics, School of Multidisciplinary Sciences. The Graduate University for Advanced Studies: (SOKENDAI). Doctor of philosophy. Published in March, 2012.
- [5] Mapping Applications to the Cloud: <http://msdn.microsoft.com/en-us/library/dd430340.aspx>
- [6] TXSeries for Multiplatforms: <http://publib.boulder.ibm.com/infocenter/txformp/v6r0m0/index.jsp>
- [7] Distributed Computing-Advantages and Disadvantages: <http://www.platformusers.net/distributed-computing/distributed-computing-advantages-and-disadvantages.html>
- [8] Foster, I., Zhao, Y., Raicu, I., Lu, S. (2008): Cloud Computing and Grid Computing 360-Degree Compared. In Grid Computing Environments Workshop, (16 November 2008), pp. 1-10.
- [9] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I. (2009): Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 25(6), pp. 599-616.
- [10] What is cluster computing? - Definition from WhatIs.com: <http://searchdatacenter.techtarget.com/definition/cluster-computing>
- [11] Garg, S. K. and Buyya, R. (2011): Green Cloud computing and Environmental Sustainability. Cloud computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering. University of Melbourne. Australia.
- [12] The cloud-SOA connection | Cloud Computing: <http://www.infoworld.com/d/cloud-computing/cloud-soa-connection-724>.
- [13] IBM - How SOA can ease your move to cloud computing: http://www-01.ibm.com/software/solutions/soa/newsletter/nov09/article_soaandcloud.html.

- [14] The Economic Benefit of Cloud Computing – Forbes:
<http://www.forbes.com/sites/kevinjackson/2011/09/17/the-economic-benefit-of-cloud-computing/>.
- [15] Aumueller, D.C. (2010): IT-Compliance Analysis for Cloud Computing. A thesis submitted for graduation to the academic degree Master of Science. University of Applied Sciences Darmstadt: Faculty of Computer Science. Germany.
- [16] Armbrust , M., Fox , A., Griffith , R., Joseph , A.D., Katz , R.H., Konwinski , A., Lee , G., Patterson , D.,A.,Rabkin , A.,Zaharia, M. (2009): Above the Clouds: A Berkeley View of CloudComputing. ElectricalEngineering and Computer Sciences.University of California at Berkeley. Technical Report No. UCB/EECS, 2009.
- [17] Cloud Computing: Private Vs. Public <http://blog.nskinc.com/IT-Services-Boston/bid/26417/Cloud-Computing-101-Private-Vs-Public>.
- [18] What is community cloud? - Definition from WhatIs.com:
<http://searchcloudstorage.techtarget.com/definition/community-cloud>.
- [19] Fehling, C., Leymann, F., Mietzner, R., Schupeck, W. (2011): A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-based Application Architectures. Universität Stuttgart: FakultätInformatik, Elektrotechnik und Informationstechnik: TechnischerBericht Nr. 2011/05. Germany
- [20] Fehling, C., Leymann, F., Retter, R., Schumm, D., Schupeck, W. (2011): An Architectural Pattern Language of Cloud-based Applications. In: Proceedings of the 18th Conference on Pattern Languages of Programs,PLoP.
- [21] Murugesan, S. (2008): Harnessing Green IT: Principles and Practices. IT Professional 10(1): pp. 24-33.
- [22] Research Reveals Environmental Impact of Google Searches | Fox News:
<http://www.foxnews.com/story/0,2933,479127,00.html>.
- [23] Official Google Blog: Powering a Google search:
<http://googleblog.blogspot.de/2009/01/powering-google-search.html#!/2009/01/powering-google-search.html>.
- [24] Greenpeace International. 2010. Make IT Green <http://www.greenpeace.org/international/en/publications/reports/make-it-green-Cloudcomputing/>
- [25] Mayo, R. N. and Ranganathan P., 2005. Energy consumption in mobile devices: Why future systems need requirements-aware energy scale-down. Proceedings of 3rd International Workshop on Power-Aware Computer Systems, San Diego, CA, USA.
- [26] Srikantaiah, S., Kansal, A., and Zhao, F. 2008. Energy aware consolidation for Cloud computing. Proceedings of HotPower '08 Workshop on Power Aware computing and Systems, San Diego, CA, USA.

- [27] Kephart, J. O., Chan, H., Das, R., Levine, D. W., Tesauro, G., Rawson, F., and Lefurgy, C. 2007. Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs. Proceedings of 4th International Conference on Autonomic Computing, Florida, USA.
- [28] Abdelsalam, H., Maly, K., Mukkamala, R., Zubair, M., and Kaminsky, D. 2009. Towards energy efficient change management in a Cloud computing environment, Proceedings of 3rd International Conference on Autonomous Infrastructure, Management and Security, The Netherlands.
- [29] Ranganathan, P., Leech, P., Irwin, D., and Chase, J. 2006. Ensemble level power management for dense blade servers, SIGARCH Computer Architecture News, 34(2):66–77.
- [30] Chase, J.S., Anderson, D.C., Thakar, P.N., Vahdat, A.M., and Doyle, R.P. 2001. Managing energy and server resources in hosting centers. Proceedings of 18th ACM Symposium on Operating Systems Principles (SOSP '01), Banff, Canada.
- [31] Woods, A. 2010. Cooling the data center. Communications of the ACM, 53(4):36-42.
- [32] Rawson, A., Pflueger, J., and Cader, T., 2008. Green Grid Data Center Power Efficiency Metrics. Consortium Green Grid.
- [33] Nowak, Alexander; Binz, Tobias; Fehling, Christoph; Kopp, Oliver; Leymann, Frank; Wagner, Sebastian: Pattern-driven Green Adaptation of Process-based Applications and their Runtime Infrastructure. In: Computing, Springer Wien, 2012
- [34] Cloud Computing: What Does and Doesn't Fit « IO Blog:
<http://www.iodatacenters.com/blog/cloud-computing-what-does-and-doesn%E2%80%99t-fit/>
- [35] Cloud Computing Is More Likely for Low-Risk Systems: Report:
<http://www.eweek.com/c/a/IT-Management/Cloud-Computing-Is-More-Likely-for-LowRisk-Systems-Report-625051/?kc=rss>.
- [36] Part 2 – Your application would be a GREAT FIT in the Cloud | Gravitant Blog:
<http://blog.gravitant.com/2011/11/02/application-fits-in-the-cloud/>.
- [37] Planning the Migration of Enterprise Applications to the Cloud: A Guide to Your Migration Options: Private and Public Clouds, Application Evaluation Criteria, and Application Migration Best Practices. (2010). Cisco System, Inc. USA.
- [38] When cloud computing is a fit (and when it's not) | ITworld:
<http://www.itworld.com/saas/84131/when-cloud-computing-fit-and-when-its-not>.
- [39] Part 1 – Your application may NOT YET be ready for the Public Commodity Cloud if | Gravitant Blog: <http://blog.gravitant.com/2011/09/13/application-feasibility-in-public-cloud/>.

- [40] Transaction Management:
http://docs.oracle.com/cd/B19306_01/server.102/b14220/transact.htm.
- [41] What is hotfix? - Definition from WhatIs.com:
<http://searchwindowserver.techtarget.com/definition/hotfix>.
- [42] What is trusted platform module (TPM)? - Definition from WhatIs.com:
<http://whatis.techtarget.com/definition/trusted-platform-module-TPM>.
- [43] What is malicious code? | Webopedia:
http://www.webopedia.com/TERM/M/malicious_code.html.
- [44] What is Cryptography? | Webopedia:
<http://www.webopedia.com/TERM/C/cryptography.html>
- [45] What is Information Security Audit?
<http://www.techopedia.com/definition/10236/information-security-audit>
- [46] Quality Attributes: <http://msdn.microsoft.com/en-us/library/ee658094.aspx>.
- [47] Application Standards Examples - Enterprise Architecture: Different Views:
<http://itknowledgeexchange.techtarget.com/enterprise-architecture/application-standards-examples/>.
- [48] What is configuration management (CM):
<http://searchciomidmarket.techtarget.com/definition/configuration-management>.
- [49] What is software licensing:
http://www.webopedia.com/TERM/S/software_licensing.html.
- [50] Improving Both Availability and Scalability: [http://msdn.microsoft.com/en-us/library/ms151244\(v=sql.105\).aspx](http://msdn.microsoft.com/en-us/library/ms151244(v=sql.105).aspx)
- [51] Startup Repair - Windows 7 Forums: <http://www.sevenforums.com/tutorials/681-startup-repair.html>.
- [52] Are databases in the cloud really all that different? | Software, Interrupted - CNET News: http://news.cnet.com/8301-13846_3-20022794-62.html
- [53] IBM. (2011): Security and high availability in cloud computing environments. IBM Global Technology Services: Technical White Paper. IBM Corporation. USA.
- [54] Application Protection: How Secure Is The Cloud:
<http://www.tomshardware.com/reviews/cloud-computing-security,2829-5.html>.
- [55] Fine-Grained Access Controls: How Secure Is The Cloud:
<http://www.tomshardware.com/reviews/cloud-computing-security,2829-3.html>.
- [56] Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., Molina, J. (2009). Controlling data in the cloud: outsourcing computation without outsourcing control. Proceedings of the 2009 ACM workshop on Cloud computing security. Chicago, Illinois, USA, ACM: pp. 85-90.

[57] Hurley, J. Cloudy With a Chance of Configuration Management: http://www.cloud-practice.de/sites/default/files/downloads/cloudy_with_a_chance_of_configuration_management.pdf

[58] Operations Management for Cloud Infrastructures: 5 Strategies to Consider | Virtual-Strategy Magazine: <http://www.virtual-strategy.com/2010/07/09/operations-management-cloud-infrastructures-5-strategies-consider>

[59] Cloud Computing: Size Matters:
http://www.wwpi.com/index.php?option=com_content&view=article&id=15034:cloud-computing-size-matters&catid=317:ctr-exclusives&Itemid=2701734.

[60] Nowak, A.; Leymann, F.; Schleicher, D.; Schumm, D.; Wagner, S. (2011): Green Business Process Patterns. In: Proceedings of the 18th Conference on Pattern Languages of Programs (PLoP 2011), Portland, USA, pp. 21.-23. Vol10.

[61] Free, open-source framework | Adobe Flex:
<http://www.adobe.com/products/flex.html>.

[62] Apache Tomcat: <http://tomcat.apache.org/>

[63] MySQL :: MySQL Database 5.5:
<http://www.mysql.com/products/enterprise/database/>