Tahereh Nodehi
Licenciado em Engenharia Computadores

A new MDA-SOA based Framework for Intercloud Interoperability

issertação para obtenção do Grau de Doutor em
Engenharia Electrotécnica e de Computadores

Orientador: Prof Doutor Ricardo Luís Rosa Jardim Gonçalves, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Constituição do Júri:
Presidente:

- Doutor Virgílio António da Cruz Machado, Professor Catedrático da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Arguente(s):
Vogais:

- Doutor Ricardo Jorge Silvério de Magalhães Machado, Professor Catedrático da Escola de Engenharia da Universidade do Minho;
- Doutor Ricardo Luís Rosa Jardim Gonçalves, Professor Associado, com Agregação da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa;
- Doutor João Pedro Mendonça Assunção da Silva, Professor Auxiliar da Escola de Engenharia da Universidade do Minho;
- Doutora Teresa Cristina de Freitas Gonçalves, Professora Auxiliar da Universidade de Évora;
- Doutor José António Barata de Oliveira, Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa;
- Doutor António Carlos Bárbara Grilo, Professor Auxiliar Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa;
- Doutor Carlos Eduardo Dias Coutinho, Especialista ao abrigo do nº 6 do artº. 34º. Do Decreto-Lei nº 115/2013, de 07 de Agosto (Caixa Mágica Software).

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Dezembro de 2014

Printable organic and inorganic materials for flexible electrochemical devices

Acknowledgements

I would like to express my gratitude to all of those that directly or indirectly contributed and supported my work.

First of all I would like to acknowledge the expert guidance I received from my supervisor, Professor Ricardo Luís Rosa Jardim Gonçalves. I am very grateful to him for the time and effort he has invested in helping me to achieve my goals. He has been constantly encouraging me to progress this research.

Throughout the achievement of this project I had support from a number of people, to whom I am deeply grateful. I would like to thank Professor Antonio Grilo for his support and help. Also I would like to acknowledge Sudeep Ghimire for his technical support.

For this dissertation I would like to thank my CAT committee members: Dr. Joao Mendonca da Silva and Dr. José Barata for their time, interest, and helpful comments.

I would also like to thank Professor Virgilio Cruz-Machado for supporting my research and providing me with a comfortable office space where I could carry out my research activities.

I sincerely acknowledge the financial support from the grant of the Portuguese Foundation of Science and Technology (Referencia da bolsa: SFRH SFRH / BD / 33965 / 2009) and EC 7th Framework Programme under grant agreement n° FITMAN 604674 (http://www.fitman-fi.eu).

I would like to thank my dear parents for their love and support over the years. Finally, I would specially like to thank my wonderful husband Aneesh Zutshi who has provided me with immeasurable emotional and practical support over the past four years.

Resumo

Cloud computing tem sido um dos temas mais importantes na área das TI, que visa assegurar serviços escaláveis e confiáveis on-demand através da Internet. O alargamento do âmbito da aplicação dos serviços cloud exige cooperação entre clouds de diferentes fornecedores que possuem funcionalidades heterogéneas. Esta colaboração entre diferentes fornecedores cloud pode providenciar uma melhor Qualidade de Serviço (QoS) ao preço mais baixo. No entanto, os sistemas cloud atuais foram desenvolvidos sem preocupações de interoperabilidade cloud, e na verdade não suportam a interoperabilidade entre fornecedores cloud. Este trabalho de doutoramento tem como objetivo de investigação resolver problemas de interoperabilidade entre fornecedores de cloud computing.

Esta tese propõe uma framework abrangente que suporta a interoperabilidade inter-cloud num ambiente de computação cloud heterogéneo, com o objetivo de alocar a carga de trabalho para as clouds mais eficientes, disponíveis em tempo de execução.

Através da análise de diferentes metodologias que foram aplicadas para resolver vários cenários problemáticos relacionados com a interoperabilidade, é- sugerido explorar Model Driven Architecture (MDA) e Service Oriented Architecture (SOA) como abordagens adequadas para a framework inter-cloud proposta. Uma vez que a distribuição das operações em ambiente baseado em cloud é um problema de tempo polinomial não determinístico (NP-complete), um job scheduler baseado num Genetic Algorithm (GA) é proposto como parte da framework de interoperabilidade, oferecendo a migração da carga de trabalho com o melhor desempenho ao menor custo. Uma abordagem Agent Based Simulation (ABS)é proposta para modelar o ambiente inter-cloud, com três tipos de agentes: Cloud Subscriber, Cloud Provider e Job. O modelo ABS é proposto para avaliar a framework inter-cloud.

Abstract

Cloud computing has been one of the most important topics in Information Technology which aims to assure scalable and reliable on-demand services over the Internet. The expansion of the application scope of cloud services would require cooperation between clouds from different providers that have heterogeneous functionalities. This collaboration between different cloud vendors can provide better Quality of Services (QoS) at the lower price. However, current cloud systems have been developed without concerns of seamless cloud interconnection, and actually they do not support intercloud interoperability to enable collaboration between cloud service providers. Hence, the PhD work is motivated to address interoperability issue between cloud providers as a challenging research objective.

This thesis proposes a new framework which supports inter-cloud interoperability in a heterogeneous computing resource cloud environment with the goal of dispatching the workload to the most effective clouds available at runtime.

Analysing different methodologies that have been applied to resolve various problem scenarios related to interoperability lead us to exploit Model Driven Architecture (MDA) and Service Oriented Architecture (SOA) methods as appropriate approaches for our inter-cloud framework. Moreover, since distributing the operations in a cloud-based environment is a nondeterministic polynomial time (NP-complete) problem, a Genetic Algorithm (GA) based job scheduler proposed as a part of interoperability framework, offering workload migration with the best performance at the least cost. A new Agent Based Simulation (ABS) approach is proposed to model the inter-cloud environment with three types of agents: Cloud Subscriber agent, Cloud Provider agent, and Job agent. The ABS model is proposed to evaluate the proposed framework.

Keywords: *Cloud Computing, Intercloud, Cloud Interoperability, Service level Agreement, Model Driven Architecture, Service Oriented Architecture, Genetic Algorithm, Job Scheduler, Agent Based Simulation Model*

Contents

List of Figures

List of tables

List of symbols, acronyms and abbreviations

ABS: Agent Based Simulation

API: Application Program Interface

ATHENA: Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications Integrated Project

ATL: ATLAS Transformation Language

B2B: Business-to-Business

CaaS: Communication as a Service

CCFM: Cross-Cloud Federation Manager

CIM: Computation Independent Model

Cloud.SDC: the Software Deployment and Configuration

Cloud.SM SM: Service Management

CP: Computing Resource Cloud Provider

CPU: Central Processing Unit

CRM: Customer Relationship Management

Cross-Cloud Federation Manager CCFM

CS: Computing Resource Cloud Subscriber

CWM: Common Warehouse Metamodel

DaaS: Storage as a Service

DCC: Dynamic Cloud Col-laboration

DCRM: Data Center Resource Maangement

DDOS: Distributed Denial-of-Service attack

DSL: Domain Specific Language

EBNF: Extended Backus–Naur Form

EC2: Elastic Compute Cloud

EMF: Eclipse Modeling Framework

ERP: Enterprise Resource Planning

ESB: Enterprise Service Bus

ETL: Epsilon Transformation Language

ETSI: European Telecommunications Standards Institute

EU: European Union

FCM: Federated Cloud Management

FI PPP: Future Internet Public Private Partnership

FITMAN: Future Internet Technologies for MANufacturing industries

FI-WARE: FUTURE INTERNET Core Platform

GA: Genetic Algorithm

GCM: Grid Component Model

GE: Generic Enabler

GICTF: Global Inter-Cloud Technology Forum

GReAT: Graph Rewrite And Transformation language

GRIS: Group from Research in Interoperability of Systems

HTTP: Hypertext Transfer Protocol

HUTN: Human-Usable Textual Notation

I2ND: Interface to Networks and Devices

IaaS: Infrastructure as a Service

IaaS CPA: IaaS Cloud Providers Agent

IaaS CSA: IaaS Cloud Subscriber Agent

ICIF: Inter-Cloud Interoperability Framework

INTEROP NoE: Interoperability Research for Networked Enterprises Applications and Software Network of Excellence

IoT: Internet of Things

ISP: Internet Service Provider

IT: Information Technology

JET: Java Emitter Templates

JTL: Janus Transformation Language

LAN: Local Area Network

M2M: Model-to-Model

MDA: Model Driven Architecture

MDD: Model-Driven Development

MDI: Model Driven Interoperability

MOF: Meta Object Facility

Mof2T: MOF Model to Text Transformation Language

MOLA: MOdel transformation Language

NGN: Next Generation Network

NIST: National Institute of Standards and Technology

NP-complete: Nondeterministic Polynomial time

OASIS: Advancement of Structured Information Standards

OGF: Existing Open Grid Forum

OMG: Object Management Group

PaaS: Platform as a Service

PIM: Platform Independent Model

PSM: Platform Specific Model

QoS: Quality of Services

QVT: Query/View/Transformation

RDF: Resource Description Framework

SaaS: Software as a Service

SAML: Security Assertion Markup Language

SIIF: Intercloud Interoperability and Federation

SiTra: Simple Transformer

SLA: Service Level Agreement

SLO: Service Level Objective

SME: Small and Medium Enterprise

SMTP: Simple Mail Transfer Protocol

SOA: Service Oriented Architecture

SOAP: Simple Object Access Protocol

SOC: Service Oriented Computing

SPEM: Software Process Engineering Metamodel

STEEP: social-technological-economical-environmental-political

TCP/IP: Transmission Control Protocol/Internet Protocol

UDDI: Universal Description Discovery and Integration

UML: Unified Modeling Language

UML: Unified Modeling Language

VB: Visitor Based

VIATRA: VIsual Automated model TRAnsformations

VM: Virtual Machine

W3C: World Wide Web Consortium

WAN: Wide Area Network

WGWSOA: Web-based Groupware Service-Oriented Architecture

WSDL: Web Services Description Language

XACML: eXtensible Access Control Markup Language

XMI: XML Metadata Interchange

XML: Extensible Markup Language

XMPP: Messaging and Presence Protocol

XSLT: Extensible Stylesheet Language Transformations

xws4j: XMPP Web Services for Java

j_i: job number i

R set of requirements

t_i serving time of job number i

cp_i computing power requirement of job number i

b_i bandwidth requirement of job number i

m_i memory requirement of job number i

d_i maximum possible waiting time of job number i

p_i number of related pricing policy of job number i

φ_0: Intercloud-Interface Module

φ_1: Job-Selection Module

φ_2: Model-Manager Module

φ_3: QoS-SLAs-Repository Module

φ4: Process-Executor Module

φ_5: Resource-Search-Discovery Module

φ_6: Resource-Selection Module

φ_7: Transformation-Engine Module

φ_8: Semantic Module

φ_9: GA based Outsourcing-Job-Scheduler Module

φ_{10}: Job-Results Module

x: number of jobs for each step of GA based solution

n: iteration number

f: fitness function

ph_k: performance history variable

$cost_{ik}$: The cost of computing resource offering from Cloud Provider CP_k for the requirement of job j_i

Chapter 1

Introduction

1 Introduction

Cloud computing as a recent computation paradigm has been developing very quickly. A cloud can offer flexible and cost-effective on-demand services ranging from software to platform or infrastructure services over the internet. The expansion of the application scope of cloud services would require collaboration between different providers that have various functionalities [1]. This cooperation between the heterogeneous cloud vendors can provide better Quality of Services (QoS) (eg. scalability and reliability, service availability and performance), avoidance of vendor lock-in, and reduced service production costs. It also can promote inter-cloud resource sharing and can provide cloud users the ability of using combined services from different service providers. The required seamless interworking mechanism between clouds is called "Inter-cloud Interoperability".

Most of the current cloud environments do not support inter-cloud interoperability and more research work is required to provide sufficient functions to enable global seamless collaboration between cloud services. Hence, inter-cloud interoperability is chosen as a general topic for this thesis. This PhD research work proposes an Inter-Cloud Interoperability Framework (ICIF) that supports interoperability between a Computing Resource Cloud Subscriber (CS) and available Computing Resource Cloud Providers (CPs).

To comprehend the appropriate concepts and approaches for our intercloud interoperability framework, the current state of the art in cloud computing and inter-cloud environment, as well as different approaches to relevant application development are studies. As results, Model Driven Architecture (MDA) and Service Oriented Architecture (SOA) are identified as two appropriate approaches for implementing the model in the framework.

The ICIF focuses on dynamic dispatching of the operations to the most appropriate available CPs based on the job requirements. The Job-Selection module of ICIF integrates the Job-Scheduler Generic Enabler (GE) from FI-WARE Platform (FUTURE INTERNET Core Platform) [2] to select the job operations waiting to receive required resources. This work developed a Genetic Algorithm (GA) based job scheduler for ICIF to select the most effective CPs and uses MDA approach to map the job model between CPs accordingly, and dispatches the job to the selected CP.

For validation process, an Agent Based Simulation (ABS) approach is developed to evaluate the proposed ICIF that simulate an extendable Inter-Cloud environment using the ICIF. The proposed ABS model includes three types of agents for CS, CP, and Job. The results are discussed later.

This chapter includes six sections. The section 1.1 states the general research topic. Section 1.2 describes the research questions. Section 1.3 discusses current challenges in the area of Cloud Computing and continues with explaining “Intercloud Interoperability” issue. Section 1.4 states the four propositions that have been considered in the PhD research work. Section 1.5 describes the scientific methodology that is adopted in this thesis. Section 1.6 specifies the structure of the thesis.

1.1 Research Topic

The topic chosen and reflected in the dissertation title is **“***Intercloud Interoperability between Computing Resource Cloud Providers”* which is a sub-topic of the wider domain: “Cloud Computing”.

1.2 Research Questions

Considering the chosen topic the main research question that defines the scope of this research work proposes one possible solution for Intercloud Interoperability issue.

Q1: How to enhance the existing cloud computing architecture and analyse the relevant requirements to propose a novel InterCloud Interoperability framework, addressing cooperation between Computing Resource Cloud Providers (CP) to support interoperability between IaaS Cloud Providers to deliver services with better performance at the least cost.

To have a successful integration, the main question leads to following questions:

1. What is the scenario for intercloud interoperability considered in the research work?
2. What are the relevant concepts and appropriate requirements to propose ICIF?
3. Which software development approaches should be combined to develop the ICIF. .And how they can enhance interoperability in Cloud environment?
4. What is the research method for validation process?
5. What will be the impact of this solution in Cloud Computing systems?

1.3 Current Challenges in Cloud Computing

Currently, Cloud Computing is a new adopted concept in Internet Technology (IT) areas. Although Cloud Computing shared services have been increasingly used by diverse users, the research on Cloud Computing is still at an early stage. There are many existing cloud challenges that have not been fully addressed, as well as new emerging issues introduced by enterprise applications. These issues can be obstacle to the growth of Cloud Computing and capturing the organizations for outsourcing applications with sensitive information. There are several articles and research work to identify the obstacles in Cloud Computing, such as [3], [4], [5], [6], [7]. In order to specify essential research directions in progress and adoption of Cloud Computing, Table 1-1 presents current challenges and obstacles in Cloud Computing area.

Table 1-1 Existing challenges in Cloud Computing area.

	Challenge	Description
1	**Standards in Cloud**	Existing standards are not comprehensive for Cloud Computing and there is lack of focus within the cloud standards' development process [8], [9], [10]. According to NIST [11], developing standardization in the cloud computing, organizations and working groups should majorly focus on following aspects: • Vendor lock-in • Limitations of developing for proprietary models and Application Program Interface (APIs) • Lack of cloud integration and interoperability • Proprietary integration with internal data centers
2	**Vendor Lock-In**	Vendor lock-in is a condition in which a client using a product or service cannot freely transfer to a competitor's product or service. It is the result of current poor portability, restricted interoperability between clouds and the lack of standardized APIs.
3	**Interoperability**	Interoperability is concerned with the ability of systems to inter-operate. Cloud costumers should be able to migrate in and out of the cloud and switch between cloud providers based on their needs, without a lock-in period. Furthermore, cloud providers should be able to interoperate among themselves to find an alternative cloud provider to give better services. New standards and interfaces should be defined to enable portability and flexibility of virtualized applications. Feldhaus [12] summarized the current challenges in Cloud Interoperability as follow: • Several different Cloud Standards from different parties are existing • Several Open Grid Forum standards currently not or only partly ready for the cloud • A consistent Open Grid Forum Cloud Portfolio is needed • Strategies for combining different Cloud Standards / APIs are needed • Existing implementations of Cloud APIs need to get interoperable • Combined Interoperability Verification Suites need to be developed • People need to be brought together to talk about issues in specifications and implementations

	Challenge	Description
4	**Intercloud**	Intercloud should be able to provide interoperability between various cloud computing instantiations.
5	**Portability between clouds**	It is equal to “Vendor Lock-in” problem and it can happen in two levels [13]: 1. Service portability 2. Data portability
6	**Service Availability**	Service outages become a major concern in the Cloud Computing, since it is essential for customers to access their information and services in the cloud at any time [14]. The Berkeley View of Cloud Computing [15] suggested using "Multiple Cloud Providers" and "Elasticity to prevent Distributed Denial-of-service attack (DDOS)" as opportunities to improve cloud service availability.
7	**Automated service provisioning**	Regarding to providing on-demand services, cloud provider should be able to allocate and de-allocate resources to the costumer while minimizing the cloud operational cost. There is not yet an efficient and reliable way for a service provider to achieve this objective. Specifically, determining a method for mapping service level objectives (SLOs), such as QoS requirements, to low-level resource requirement, like Central Processing Unit (CPU) and memory requirements, is not straightforward.
8	**Data Integration and Synchronization**	Data should be integrate and synchronize to provide an accurate data for the costumers. However there should be more study on novel methods because of distributed and shared nature of Cloud Computing.
9	**Data location**	The geographic location of the data in the Cloud is important for several reasons: • Legal issues: There might be fundamental differences between policies in various countries and a customer could be involved in illegal practices without even noticing. Hence, cloud providers need to understand the regulatory requirements for each country and know the location of data for both individual customers’ data and corporate information. • Natural risk factors: A cloud provider has to reduce the risk of locating a datacenter in a geographic location by duplicating a secondary datacenter in a less risky location. • Performance: The location of a datacenter can have a significant impact on the performance of applications delivered out of a cloud computing environment.
10	**Data segregation**	Due to multi-tenant usage mode of the Cloud, user isolation is a challenge for co-locate different customers’ virtual machines in the same server or data on the same hard disks.

	Challenge	Description
11	**Data Transfer Bottlenecks**	Since bandwidth cost might be expensive for the businesses, many of them are looking for a cost reduction before switching to the cloud. Furthermore, the bandwidth and latency of the networks can be a bottleneck for the data-intensive applications. Regarding to the network connection there can be two points of failure: the connection of the customer organization and the connection of the provider. Berkeley View of Cloud Computing [15] suggested using "FedExing Disks", "Data Backup/Archival", and "Higher Bandwidth Switches" as opportunities to improve cutwork connection quality in the Cloud Computing.
12	**Energy management**	Modify the design of data centers and underlying infrastructures to improve energy efficiency is another major issue in cloud computing. Cut down energy cost in data centers is not the only objective, but also the cloud provider should meet government regulations and environmental standards.
13	**Insiders' privilege abuse**	The threat of malicious insiders with a privileged role (e.g. an administrator) on any outsourced organization is considerable. Abuse by insiders can have affect and damage many customer's operations, such as brand, finance, productivity. Hence, it is essential for the cloud consumers to know what the providers are doing to identify and protect against the malicious insider threat. Cloud Security Alliance [16] suggested following methods to avoid abusing by malicious insiders: Enforce strict supply chain management and conduct a comprehensive supplier assessment. Specify human resource requirements as part of legal contracts. Require transparency into overall information security and management practices, as well as compliance reporting. Determine security breach notification processes.
14	**Monitoring**	Underlying resource monitoring and evaluation is an essential task in the Cloud computing system [17]. Since cloud computing is more complicated than other networks, it needs more study on resource monitoring to improve network analysis, management, fault detecting and recovery, load balancing, and event predicting, in Cloud computing.
15	**Optimization of Resource Scheduling**	Resource scheduling based on Service Level Agreement (SLA) in cloud computing is NP-hard problem. There is still no efficient method to solve it [18].

	Challenge	Description
16	**Performance Unpredictability**	Generally, customers always expect to receive the same performance by paying the same money. However since the performance of cloud services depend on various factors that mostly are out of users' control, the performance might be varied. For instance, cloud providers try to increase the utilization level of the infrastructure through multi-tenancy and sometimes one user's activity might affect another user's application performance. Or the data access latency can depend on the datacenter' location and some other network performance parameters. Therefore, customers may have some troubles created by the variance in performance. Berkeley View of Cloud Computing [15] suggested using "Improved Virtual Machine Support", "Flash Memory", and "Gang Schedule Virtual Machine s" as the opportunities to improve performance steadiness in the Cloud Computing.
17	**Recovery and back-up**	Cloud providers should define a solution to back up the data after a data loss event.
18	**Reliability**	Service outages create another fundamental issue in cloud computing, that is occasional lack of desired reliability.
19	**Scaling resources**	A key advantage of cloud computing is the ability to scale up or down resources when it is required [19]. Moreover, this is usually addressed as elastic scale. If this feature is not appropriately implemented or the acceptable response time is not agreed upon beforehand, it can lead to service failures. Current service level agreement (SLA) specifies quality of service needs, however it is not in terms of response time in response to workload variations.
20	**Security**	One of the major challenges in cloud computing is how to address the security and privacy concerns of businesses considering adopting it [20]. Additionally, it is essential for the cloud users to understand the security problems associated with the usage, management, orchestration and monitoring of cloud services. Kevin Hamlen and colleagues [21] from University of Texas at Dallas classified and explained the Security Issues for Cloud Computing in 5 major areas: 1. storage security 2. middleware security 3. data security 4. network security 5. application security

	Challenge	Description
21	**Bugs in Large Distributed Systems**	Berkeley View of Cloud Computing [15] suggested using "Invent Debugger" that relies on Distributed Virtual Machines as an opportunity to reduce bugs in the Cloud Computing.
22	**Criminal abuse**	Already all new information technologies try to improve their security and avoid account, service, traffic hijacking and other criminals [22]. Cloud Computing providers are also being intensely targeted with attackers.
23	**Data Confidentiality and Auditability**	Berkeley View of Cloud Computing [15] suggested using "Deploy Encryption", " Virtual Local Area Networks ", "Firewalls"; and "Geographical Data Storage" as the opportunities to improve data confidentiality and auditability issues in the cloud computing.
24	**Server consolidation**	Server consolidation can be a profitable approach to optimize resource usage while minimizing energy consumption in a cloud computing environment. The problem of maximally consolidating servers in cloud computing is an NP-hard optimization problem [3].
25	**Service Delivery Billing**	Due to the on-demand nature of the services, cost assessment of services provided in the Cloud is complicated.
26	**Application Customization**	Often, the concern of services provided in the public cloud is majority of the users, and they usually address only general solutions and don't admit much personalization. Hence, finding appropriative applications is more difficult compared to the in-house software market where most requirements can be solved.
27	**Shared Technology Vulnerabilities**	Recently, some attacks have been discovered that targeted the shared technology inside Cloud Computing, such as RAMs, Memories, Caches, GPUs. The attackers attempt to have significant impact on the operations of other cloud customers, and illegally access to their data.
28	**Software Licensing**	Berkeley View of Cloud Computing [15] suggested using "Pay-for-use licenses", and "Bulk u se sales" as opportunities to improve Software Licensing issue in the cloud computing.
29	**Traffic management and analysis**	There are still several challenges on measurement and analysis methods of data center traffic in Internet Service Providers' (ISPs) networks and enterprises. Additionally the extension of current traffic measurement and analysis methods for the data centers in the cloud leads to more complexities,

	Challenge	Description
		such as: The density of connections is higher than in ISPs or enterprise networks, which makes the worst-case scenario for existing methods. Most of the current methods can calculate traffic among a few hundred end hosts, and they cannot be used for several thousand servers of cloud computing. Existing methods often presume some default flow patterns in Internet and enterprises networks, but current solution for data centers, such as MapReduce jobs, considerably modify the traffic pattern.
30	**Virtualization**	Virtual machine migration in the cloud can balance traffic across the data center and enables robust and responsive provisioning in data centers. However, using virtualization methods complicates security for both customers and service providers. In virtualization, virtual machines (or collections of them) should be protected instead of a physical server or collection of servers that an application runs on. Additionally, dynamic virtualization is in a vague state and most data centers are still supporting only static virtualization.

As detailed in Table 1-1, there are several particular challenges, such as “vendor lock-in”, “standards” and “interoperability”, which are connected together and improving one of them can have effect on the others. Additionally these challenges should be addressed specifically according to the concept of Cloud Computing. The main goal of this thesis is presenting a solution for Intercloud Interoperability. Sections 2.2 of chapter 2 will describe the concept of Intercloud Interoperability and current state of the art for Intercloud Interoperability.

1.3.1 Challenge of Intercloud Interoperability

To understand “Intercloud Interoperability”, first, the concept of “*interoperability*” should be defined. Enterprise applications and software systems need to be interoperable in order to reduce scaling/producing cost within the development of the components. For a long time, there have been studies to find better solutions to set up system interoperability. In the area of interoperability the challenge is enabling separate entities, systems or artifacts to cooperate effectively together (inter-operate). IEEE Glossary defines interoperability as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” [23]. J. O’Brien and G. Marakas define interoperability as “Being able to accomplish end-user applications using different types of computer systems, operating systems, and application software, interconnected by different types of local and wide area networks" [24]. There are several challenges when heterogeneous systems are required to support interoperability. The first challenge is managing the differences among systems for instance two systems do not use the same language and often they do not share the same syntax. The lack of agreement on the common standards, and the deficiency of appropriate mechanisms and tools are also other issues regarding to providing the interoperability.

The Intercloud concept is based on the fact that each single cloud has limited computing resources in a restricted geographic area. Cloud costumers should be able to migrate in and out of the cloud and switch between providers based on their needs, without a lock-in period. Furthermore, cloud providers should be able to interoperate among themselves to find an alternative cloud provider to give better services. The aim of Intercloud is providing interoperability between various cloud computing instantiations where each cloud would use computing resources of other clouds. The present Intercloud network merely connects different cloud systems and still has major interoperability issue.

Next chapter of current document reviews different approaches that have been applied to resolve various scenarios of interoperability and shows a solution based on Model Driven approach and Service Oriented systems can be used for a Intercloud Interoperability solution.

1.4 Propositions

The principal basis of this research work is proposing an applicable solution for interoperability issue between a cloud subscriber and computing resource cloud providers. This research work has been developed based on the following four hypotheses:

1. **MDA and SOA approaches can clarify semantic interoperability conflicts between CS and CPs.**

   Through a comprehensive literature review of different methodologies that have been applied to resolve various scenarios of interoperability, we conclude Model Driven approach and Service Oriented systems can be appropriate approaches to support Intercloud Interoperability. Considering a MDA-SOA based layer as a top layer of ICIF architecture can acts as the arbiter layer between the other layers. Additionally, this layer can make use of GE integration layer to select job operation waiting for resource allocation.

2. **Genetic Algorithm based solution can be useful for job scheduling.**

   Based on literature, it is proven that an optimization solution for resource scheduling based on Service Level Agreement (SLA) in cloud computing is an NP-hard problem [18][3]. Additionally, Genetic Algorithm is known as an appropriate method for proposing job-scheduler in a distributed environment. Hence, the thesis proposes a GA based job-scheduler that dispatch operation to the available Cloud Providers through ICIF.

3. **Agent Based Simulation approach can be used to model the interactions between cloud subscriber and computing resource cloud providers and outsourcing the operations to them.**

   Agent Based Modeling is an effective way to model systems that contain a large number of interacting "Agents". It is especially effective where rules for interactions between individual agents are well defined and through these interactions, the overall macro phenomena can be observed. This is very similar to an intercloud environment where cloud subscriber and computing resource cloud providers have SLA based agreement and are interacting agents. Moreover the users of Cloud Subscriber requests Computing Resources which can be shown as job agents with number of requirements. Job agents can be outsourced to available cloud through ICIF.

4. **Such simulation models can be used to predict the appropriate factors for GA based job-scheduler and evaluation of ICIF.**

   Simulation model can be set up and initialised with actual number of available Cloud Providers and their properties and different parameters for the GA based job-scheduler. Running the simulation model with different factors for job-scheduler can predict the appropriate values for mutation-rate, crossover-rate and other required factors of job-scheduler to provide a more effective solution. Moreover, through such simulation model, ICIF can be evaluated for variety of intercloud environments.

1.5 Methodology

"Science" is the systematic act of collecting knowledge about the universe, organizing and condensing that knowledge into testable explanations and theories [25][26]. "Science" is used in a broad concept interpreting reliable knowledge about a topic, such as physics, linguistics or political science. Hence, its definition is neither simple nor apparent, but it can be agreed that logic and mathematics are fundamental components of all branch of science [27].

Regarding to the topics addressed in PhD work, Cloud Computing can be defined as computing resources (Infrastructure, Platform, and Application) that are delivered on-demand as a service over a network (typically the Internet). Within the category of computer sciences, computer networks can be defined and analyzed through mathematical equations [28][29][30]. Regarding to the Cloud Computing concept, different methods and theories exist that can scientifically be described. For instance, it is proven that resource scheduling based on Service Level Agreement (SLA) in cloud computing is an NP-hard problem [18]. Also it is confirmed that the problem of maximally consolidating servers in cloud computing can be solved with an NP-hard optimization solution [3]. Hence, the thesis proposes a GA based job-scheduler that dispatch operation to the available Cloud Providers through ICIF.

The ICIF is based on SOA and MDA approaches. SOA methodology is an accepted and generic theory that can be used to solve different types of problems. SOA creates independent units of logic, known as services, with sufficient amount of commonality and standardization that are not isolated from each other [31]. MDA approach [32][33] is a software development method launched by the Object Management Group (OMG) to provides a set of guidelines to structure open, vendor-independent interoperability specifications which are expressed as models.

1.5.1 The Scientific Method

Merriam-Webster dictionary [34] specified that the scientific method includes "the principles and procedures for the systematic pursuit of knowledge involving the recognition and formulation of a problem, the collection of data through observation and experiment, and the formulation and testing of hypotheses". The method should be able to minimize the influence of the researchers' bias on the outcome of an experiment, i.e. personal preferences, common sense assumptions, concealing of data not supporting the hypothesis, etc. [35].

There are several variants of the scientific method; however, the process of investigation is often referred in many textbooks and science courses as a linear set of steps through which a scientist moves from observation through experimentation and to a conclusion. This classic representation can have a number of problems because processes can be iterative, or in some cases can even be skipped. It is not always required to start with a question, and sometimes does not even involve experiments. Instead, the scientific method is a more dynamic and robust process [36][37].

Some scientific investigations achieve results leading in directions not originally anticipated, or even in multiple directions [38]. Therefore, the logic of science is recursive/iterative and also theory-contaminated, i.e., hypotheses have its origins in the existing knowledge of the researcher, which is never universal and can change after experimentation, thus leading to new hypothesis [27].

1.5.1.1 *Adopted Research Method*

Ultimately, the choice of which research method (instantiation of the scientific method) to use is individual and depends on the scientist and the nature of the question addressed. To open to any potential result, the research method adopted for this Ph.D. work is based on an 8 step method that, as suggested in [27] considers the influence of the researcher's background knowledge in the scientific process, and envisages recursive iteration through different steps depending on the results obtained in the hypothesis testing (see Figure 1-1). The adopted method is described in more detail as follows:

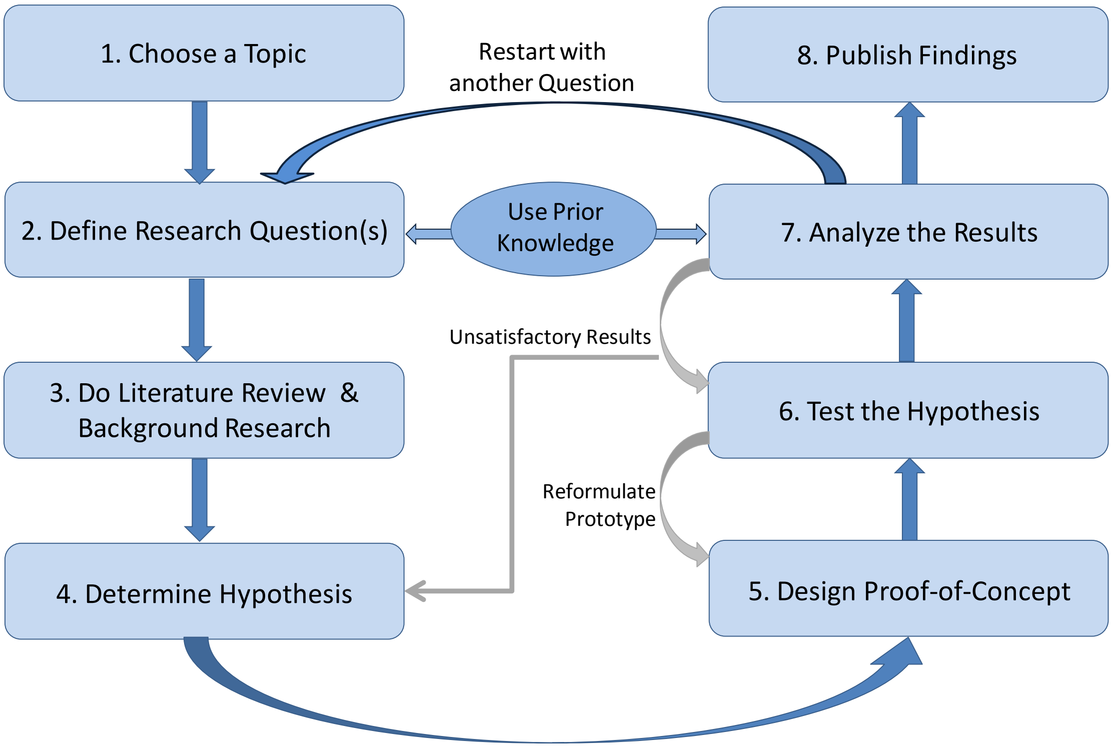


Figure 1-1 Adopted Research Method.

1. **"Choose a Topic"**: The first step towards a successful scientific research consists in choosing a meaningful topic. In fact this can be seen as a preliminary step towards the real method because there is no point in conducting research in areas where the researcher has no interest. Chapter 1 reports on the results of this step.

2. **"Define Research Question(s)"**: This is one of the most important steps of the full method, since it scopes the entire work, and will never be revisited in the same research loop until a conclusion is achieved based on the analysed results [39]. The research

question may be complemented with secondary questions to narrow the focus of the study, but all must be capable of being confirmed or denied, i.e. answered. This way, statements defined under this step should always be clear and interrogative.

Depending on pre-existing knowledge the researcher tends to avoid questions guessed of not leading to concrete answers, thus prior knowledge influences the formulation of the research questions. Chapter 1 reports on the results of this step.

3. **"Do Literature Review & Background Research"**: Through the background research, any studies elaborated for the PhD thesis will have a solid basis on the work of peers. During this stage the researcher will do literature review and join discussion groups to verify if the work has been done previously, to see if there are similar approaches to build upon, and to mark the differences to what will be done [40]. Chapter 2 reports on this step.

4. **"Determine Hypothesis"**: A scientific hypothesis uses the background research to state an educated guess regarding the variables involved [40]. It should be stated in a declarative format, which brings clarity, specificity and focus to a research problem. In critical thinking, as in science, the hypothesis or proposed answer to the research question must be testable, otherwise it is of no use for further investigation. In fact, and as illustrated by Figure 1-1, the hypothesis can be revisited and reformulated in case unsatisfactory results are achieved during the more advanced stages of the scientific method.

   The section chapter 1 reports on the results of this step.

5. **"Design Proof-of-Concept"**: The proof-of-concept is frequently related to engineering research and the development of a prototype. It is the evidence that demonstrates that an idea is feasible. This way, and because many times the complete validation of the hypothesis in a real world environment involves resources (both time and money) that few have access to, this step relates directly to the design of an experiment in a controlled environment.

   However, it is necessary that the proof-of-concept is not only associated to a prototype but also to the thesis validation method. This step also includes planning in detail the validation phase, namely the definition of a scenario and/or test cases. This step will be described in detail at the dissertation document.

6. **"Test the Hypothesis"**: This is the step where the testing of the hypothesis will actually be done. It includes the implementation of the prototype, collection of data, and execution

of tests according to the pre-defined validation method. Considerations regarding the implementations and ultimately the hypothesis will be drawn. At this stage, the researcher may find evidence that the hypothesis needs to be redefined, thus it will need to jump back to step 4, or might need to propose adaptations to the prototype design (previous step).

This stage is reported in chapter 4 this document.

7. **"Analyse the Results"**: The factual results of the testing are to be verified under this step by means of quantitative and qualitative analysis. During this step it is important to have a critical spirit and promote discussion regarding literature and research questions [40]. However, if the conclusion is that the hypothesis failed the tests, it must be rejected and either abandoned or modified. A modified hypothesis must be tested again, and if that would be the cased during this PhD, it will be necessary to return to step 4.

   This stage of the scientific method is reported in chapter 5 of this document.

8. **"Publish Findings"**: If the hypothesis passes the further tests, it is considered to be a corroborated hypothesis, and can be published. It is "mandatory" to publish final findings and provide peers from the scientific community the chance to verify, comment, and use the developed work. Nonetheless, and despite appearing only as the final step of the adopted research method, intermediate findings can also be published.

During the PhD work, it is envisaged to follow the regular flow of the adopted research method going from step 1 to 8. As explained before, intermediate findings will target publications at recognised conferences and journals, and backward loops will apply if required.

1.6 Structure of the Thesis

The PhD thesis includes six chapters:

1. Introduction,
2. Cloud Computing and Intercloud Interoperability
3. MDA and SOA
4. The InterCloud Interoperability Framework (ICIF)
5. A New Agent Based Simulation Model and the Validation Process
6. Discussion and Final Consideration

The Introduction as the first chapter begins by giving an overall view on the background of the research work in the thesis. Also, a summary of the current challenges in Cloud

Computing, including 30 top issues (lack of Standards in Cloud, Vendor Lock-In issue, Interoperability and portability, Data Transfer Bottlenecks, energy management and etc.), has been included, enabling to extract some real challenges that the Cloud Computing is facing. Subsequently, the research topic and research questions are presented. Based on the evidence, hypotheses have been proposed. Furthermore, the research method that has been adopted during this PhD work is described.

The second chapter includes the literature review on the areas targeted in the PhD. This chapter starts by discussing about cloud computing. It defines cloud computing, gives the fundamental characteristics of cloud compared to other computing paradigm, and describes service/delivery classification and deployment models of cloud environment. Moreover, Intercloud Interoperability definition and latest research work on it are addressed.

The third chapter describes two application development approaches that are exploited in the solution proposed by PhD work. The section 3.1, discusses about Model Driven Architecture (MDA) Approach, its principal models, models transformation details, modeling standards, Model Driven Interoperability (MDI), and its advantages (providing Interoperability and Portability). Then Service Oriented Architecture (SOA) and all essential relevant concepts on that including its definition and benefits are described. Rest of this chapter summarizes latest state of the arts on related concepts to the MDA and SOA approaches.

The fourth chapter proposes a novel interoperability framework based on MDA and SOA approaches which support intercloud interoperability in a heterogeneous computing resource cloud environment with the goal of dispatching the workload on the most effective clouds available at runtime. This chapter includes six main sub-sections: first, considering literature review chapter, required concepts for Intercloud Interoperability Framework are discussed. Second, a generic architecture for InterCloud Interoperability Framework is proposed. Third, the detailed model for Computing Resource ICIF is proposed. Fourth, a new Genetic Algorithm based job scheduler is proposed as a part of interoperability framework offering workload migration with the best performance at the least cost. Fifth, a short introduction to FUTURE INTERNET Core Platform (FI-WARE) cloud is proposed that will be integrated in the proposed solution. Finally, section six of this chapter summarises the work has been done in the fourth chapter.

Fifth chapter proposes an ABS model to evaluate the proposed ICIF that simulates an extendable InterCloud environment using the ICIF. The proposed ABS model includes three types of agents representing: (1) Cloud Subscriber (CS) that has limited number of resources, (2) Cloud Providers that cooperate with CS to provide better QoS services for the users of CS, and (3) Jobs that are applied through customers of Cloud Subscriber. The results are discussed later.

Final chapter is “Discussion and Final Consideration”. This chapter summarises the PhD thesis work, providing an overview of “the problem and motivation”, “How thesis deals with the problem and the contribution”, and “The considerations to develop the proposed solution”. Moreover a brief analysis of the contribution of this research work is presented. Finally, areas for further development and research are discussed.

Chapter 2

Cloud Computing and Intercloud Interoperability

2 Cloud Computing and Intercloud Interoperability

Today, cloud computing is a new promising paradigm rapidly developing in the technology industry. The popularity of cloud services has increased its presence across various domains worldwide. Cloud services have the potential to engage in the growth of organisations mainly for Small and Medium Enterprises (SMEs) that contribute significantly to the global development by providing employment and value added services. There are a number of reasons why cloud services are universally used among multiple sectors recently. The main advantages are: the reduction of spending on technology infrastructure and software applications, the ability to pay based on the usage, the capability to increase in flexibility and scalability, and the simplification of personnel training requirement.

The cloud computing idea is based on a number of previous well researched concepts and technologies such as distributed and grid computing, and virtualisation. The great novelty of cloud computing lies in the approach through which it provides services to users. The National Institute of Standards and Technology (NIST) proposed a widely accepted definition of cloud computing that is "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction" [41].

There are many research challenges in the area of the cloud computing (discussed in previous chapter). These issues can be obstacles to the growth of cloud computing for outsourcing applications from heterogeneous organizations. Present state of the art shows intercloud interoperability challenge is a key to the growth of cloud computing. The intercloud concept is based on the fact that each single cloud service provider has limited number of computing resources. However, most of current cloud systems are developed without interoperability concerns and available standards in cloud environment do not support intercloud interoperability and maytake years to fully develop. Thus, more research work is required to provide sufficient functionality to enable global seamless collaboration in the cloud environment. Hence, during the Ph.D. work we aim to propose a novel framework to improve Intercloud Interoperability.

This chapter discusses on the most important concepts related to the cloud computing that can be beneficial in the process of developing our solution for intercloud interoperability problem. Additionally, it studies the state of the art in the area of cloud computing and intercloud interoperability.

2.1 *Cloud Computing*

Cloud computing is a recent computation paradigm which provides on-demand services ranging from software to platform or infrastructure services over the internet. Following subsections will explain Cloud Computing, including its definition and essential concepts, architectural designs, characteristics. The latest research work on Cloud and Intercloud Interoperability are discussed in section 3.3.

2.1.1 Definition of Cloud Computing

The concept of "Cloud" is not a new one and it has been used in several fields such as Automated Teller Machine's networks in 1990s. The term of "Cloud" is used to describe the networks that incorporate various technologies, without the user knowing it. In 1997, as the first academic definition, Chellapa clarified cloud computing as "a computing paradigm where the boundaries of computing will be determined rationale rather than technical" [42].

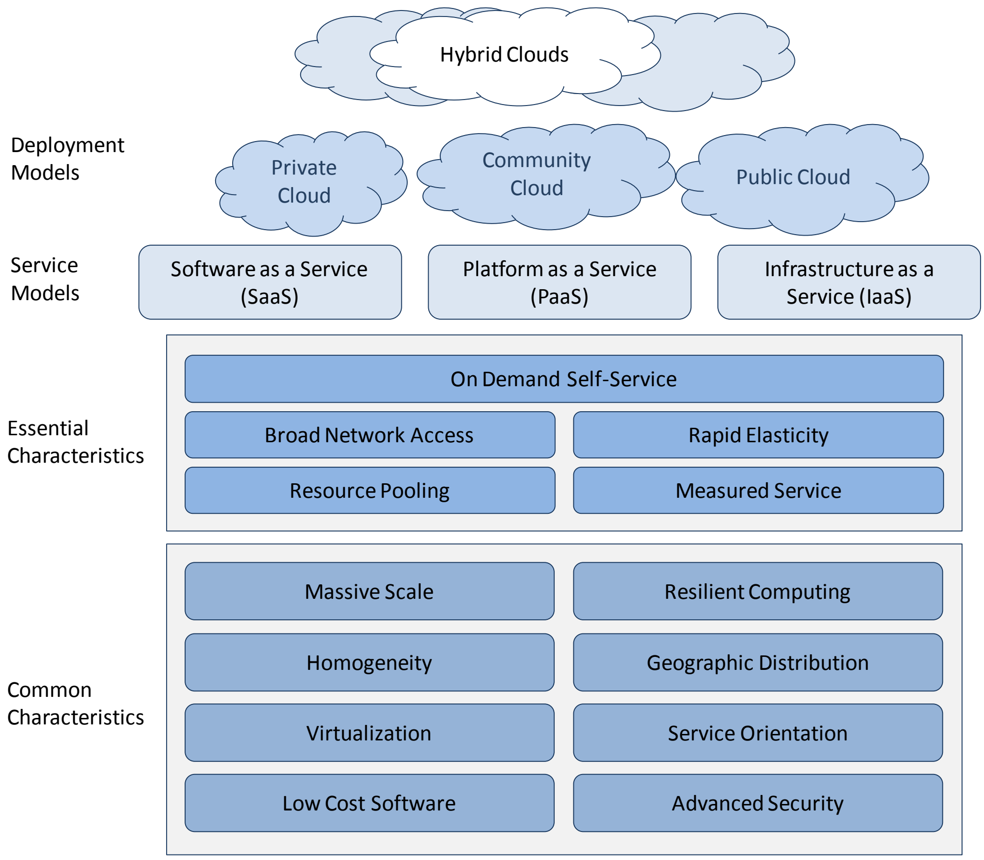


Figure 2-1 Cloud Computing Definition [43].

The National Institute of Standards and Technology (NIST) proposed a cloud computing definition as follows: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., 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 four deployment models" [11][41]. Figure 2-1 shows the framework introduced by NIST to define cloud computing [43].

According to the different perspectives of various corporations such as; academicians, architects, consumers, developers, engineers and managers, there are several definitions for cloud computing [44]. Table 2-1 provides some available cloud definitions.

Table 2-1 Cloud Computing definitions.

Reference	Author(s)	Year	Definition/Excerpt
[42]	Chellapa	1997	"a computing paradigm where the boundaries of computing will be determined rationale rather than technical"
[41]	NIST	2009	"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., 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 four deployment models"
[45]	Bernstein et al.	2009	"Cloud Computing is a datacenter which: Implements a pool of computing resources and services which are shared amongst subscribers. Charges for resources and services using an "as used" metered and/or capacity based model. Are usually geographically distributed, in a manner which is transparent to the subscriber (unless they explicitly ask for visibility of that). Are automated in that the provisioning and configuration (and de-configuration and unprovisioning) of resources and services occur on the "self service", usually programmatic request of the subscriber, occur in an automated way with no human operator assistance, and are delivered in one or two orders of seconds.

Reference	Author(s)	Year	Definition/Excerpt
			Resources and services are delivered virtually, that is, although they may appear to be physical (servers, disks, network segments, etc) they are actually virtual implementations of those on an underlying physical infrastructure which the subscriber never sees. The physical infrastructure changes rarely. The virtually delivered resources and services are changing constantly.” Resources and services may be of a physical metaphor (servers, disks, network segments, etc) or they may be of an abstract metaphor (blob storage functions, message queue functions, email functions, multicast functions, etc). These may be intermixed.
[46]	Gartner	2009	“A style of computing where scalable and elastic IT-related capabilities are provided as-a-service using Internet technologies to multiple external customers”
[15]	University of Berkeley	2009	“Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services”
[47]	Berger	2008	“... the key thing we want to virtualise or hide from the user is complexity. ...with cloud computing our expectation is that all that software will be virtualised or hidden from us and taken care of by systems and /or professionals that are somewhere else – out there in the cloud”.
[48]	Buyya et al.	2008	“A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers”
[48]	Buyya et al.	2008	“a type of parallel and distributed system consisting of collection of interconnected and virtualised computers that are dynamically provisioned and present on or more unified computing resource based on service-level agreements established through negotiation between

Reference	Author(s)	Year	Definition/Excerpt
			service provider and customer".
[49]	Catteddu & Hogben	2009	"on-demand service model for IT provision, often based on virtualisation and distributed computing technologies"
[47]	Cohen	2008	"for me the simplest explanation for cloud computing is describing it as, 'internet centric software'. This new cloud computing software model is a shift from traditional single tenant approach to software development to that of scalable, multi-tenant, multi- platform, multi-network, and global".
[47]	Doerksen	2008	"cloud computing is... the user friendly version of grid computing".
[47]	Edwards	2008	"...what is possible when you leverage web scale infrastructure (application and physical) in an on-demand way. ...anything as a service... all terms that couldn't get it done. Call it 'cloud' and everyone goes bonkers".
[47]	Eicken	2008	"... outsourced, pay-as-you-go, on-demand, somewhere in the internet".
[50]	Forrester Research, Inc.	2008	"A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end-customer applications and billed by consumption"
[51]	Gartner, Inc.	2008	"A style of computing where massively scalable IT-enabled capabilities are de- livered as a service to external customers using Internet technologies."
[47]	Gaw	2008	"refers to the bigger picture...basically the broad concept of using the internet to allow people to access technology enabled services".
[47]	Gourlay	2008	"cloud will be the next transformation over the next several years, building off of the software models that virtualisation enabled"
[47]	Haff	2008	"...there are really only three types of services that are cloud based: SaaS, PaaS, and Cloud Computing Platforms".

Reference	Author(s)	Year	Definition/Excerpt
[47]	Harting	2008	"cloud computing overlaps some of the concepts of distributed, grid and utility computing, however it does have its own meaning if contextually used correctly. Cloud computing really id accessing resources and services needed to perform functions with dynamically changing needs".
[47]	Kaplan	2008	"a broad array of web-based services aimed at allowing users to obtain a wide range of functional capabilities on a 'pay-as-you-go' basis that previously required tremendous hardware/software investment and professional skills to acquire".
[47]	Kepes	2008	"put cloud computing is the infrastructural paradigm shift that enables the ascension of SaaS".
[47]	Klems	2008	"you can scale your infrastructure on demand within minutes or even seconds, instead of days or weeks, thereby avoiding under-utilisation(idle servers) and over utilisation (blue screen)of in-house resources".
[52]	LizheWang & Laszewski	2008	"a set of network enabled services, providing scalable, QoS guaranteed, normally personalised, inexpensive computing platforms on demand, which could be accessed in a simple and pervasive way"
[47]	Martin	2008	"cloud computing really comes into focus only when you think about what IT always needs: a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software"
[47]	Pritzker	2008	"cloud tend to be priced like utilities... i think is a trend not a requirement". 2008
[47]	Ricadela	2008	"... cloud computing projects are more powerful and crash proof than Grid systems developed even in recent years"
[47]	Sheedan	2008	"... 'cloud pyramid' to help differentiate the various cloud offerings out there... top: SaaS; middle: PaaS; bottom: IaaS".

Reference	Author(s)	Year	Definition/Excerpt
[47]	Sheynkman	2008	“the ‘cloud’ model initially focused on making hardware layer consumable as on- demand compute and storage capacity. ... to harness the power of the cloud, complete application infrastructure needs to be easily configured, deployed, dynamically scaled and managed in these virtualised hardware environments”.
[47]	Sultan	2008	“... in a fully implemented Data center 3.0 environment, you can decide if an app is run locally (cook at home), in someone else’s data center (take-out) and you can change your mind on the fly in case you are short on data center resources (pantry is empty) or you having environmental/facilities issues (too hot to cook)”.
[53]	Vaquero et al.	2009	“cloud are a large pool of easily usable and accessible virtualised resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust a variable load (scale), allowing also for an optimum resource utilisation. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the infrastructure provider by means of customised SLAs”

2.1.2 Characteristics

The NIST identified a number of characteristics associated with Cloud Computing to distinguish Cloud from other computing paradigms [11][41][43]. As shown in Figure 2-1 NIST classified five essential characteristics for cloud computing. In addition to essential characteristics, NIST specified eight common characteristics for Cloud Computing listed as follow:

- Massive Scale
- Homogeneity
- Virtualization
- Low Cost Software
- Resilient Computing
- Geographic Distribution
- Service Orientation
- Advanced Security

There are some other research works that identified other characteristics for Cloud Computing:

- Maintenance and upgrading [4].
- Multi-tenancy [3].
- Self-organizing [3].
- Streamlining the Data Center [54].
- Improving Business Processes [54].

The essential characteristics determined by the NIST are described below:

2.1.2.1 *On Demand Self-Service*

A cloud computing vendor has to provide computing resources automatically according to the customer requirements. Ideally, computing resources should be available whenever client needs. Hence, providing on-demand computing resources enables the customer to eliminate dispensable upfront investment in purchasing and installing the resources.

2.1.2.2 *Broad Network Access*

A cloud corporation should be able to provide its available services for any heterogeneous client platforms (e.g., Smart-phones, laptops, and tablets), regardless of specifications, from any Internet connected location.

2.1.2.3 *Shared Resource Pooling*

The cloud computing vendor provides a pool of computing resources to serve multiple consumers using a multi-tenant model, with various physical and virtual resources. The allocation and reallocation of resources is dynamic and in accordance with consumer demand. Examples of resources include storage, memory, processing, network bandwidth, and physical and virtual machines.

2.1.2.4 *Rapid Elasticity*

A cloud provider should be able to rapidly and elastically include or exclude computing resources according to the client's changing needs. To the consumer, the feasible capabilities for cloud network provisioning should appear to be infinite and can be purchased in any quantity at any time. In reality the cloud provider does not have unlimited resources, hence the cloud provider has to arrange appropriate resources to assure fulfilling the current requirements of the clients based on the service level agreements with the costumers - otherwise it may be specified that cloud provider has to pay a penalty for not meeting the correspondent service level agreement.

2.1.2.5 *Measured Service*

In a cloud network, there should be an appropriate mechanisms to automatically monitor, control, and report the utilizing the computing resources to provide transparency between the provider and each individual consumer of the services. Therefore, cloud computing services exploit a metering application which controls, monitors and optimises resource consumption. As a result the customer pays only for the time of utilizing processors or storage. Actually, the cloud computing is using the idea of utility computing, considering the computing resources are being provided on-demand, similar to supplying electricity, water, or gas by a utility company.

2.1.3 Service/Delivery Classification

This section defines more technical aspects of Cloud Computing. According to the NIST [41] definition, cloud computing specifies three delivery models to provide various services such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [44]. Each level of service models adds additional functionality and provides required services for different kind of users from network architectures to end users (shown in Figure 2-2).

Paul Wallis explains Cloud Pyramid shown in Figure 2-2 with more details [47]:

"I would like to propose a 'Cloud Pyramid' to help differentiate the various Cloud offerings out there. Users are truly restricted to only what the application is and can do. Some of the notable companies here are the public email providers (Gmail, Hotmail, Quicken Online, etc.). Almost any Software as a Service (SaaS) provider can be lumped into this group. As you move further down the pyramid, you gain increased flexibility and control but you are still fairly restricted to what you can and cannot do. Within this Category things get more complicated to achieve. Products and companies like Google App Engine, Heroku, Mosso, Engine Yard, Joyent or force.com (SalesForce platform) fall into this segment. At the bottom of the pyramid are the infrastructure providers like Amazon's EC2, GoGrid, RightScale and Linode. Companies providing infrastructure enable Cloud Platforms and Cloud Applications. Most companies within this segment operate their own infrastructure, allowing them to provide more features, services and control than others within the pyramid."

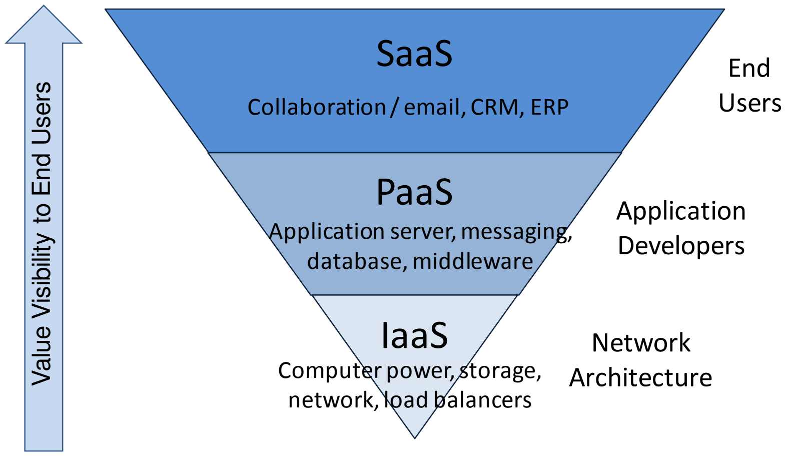


Figure 2-2 Cloud Computing Pyramid.

Vaquero and colleagues introduced a flowchart (shown in Figure 2-3) to illustrate the different actors and service delivery layers in Cloud Computing.

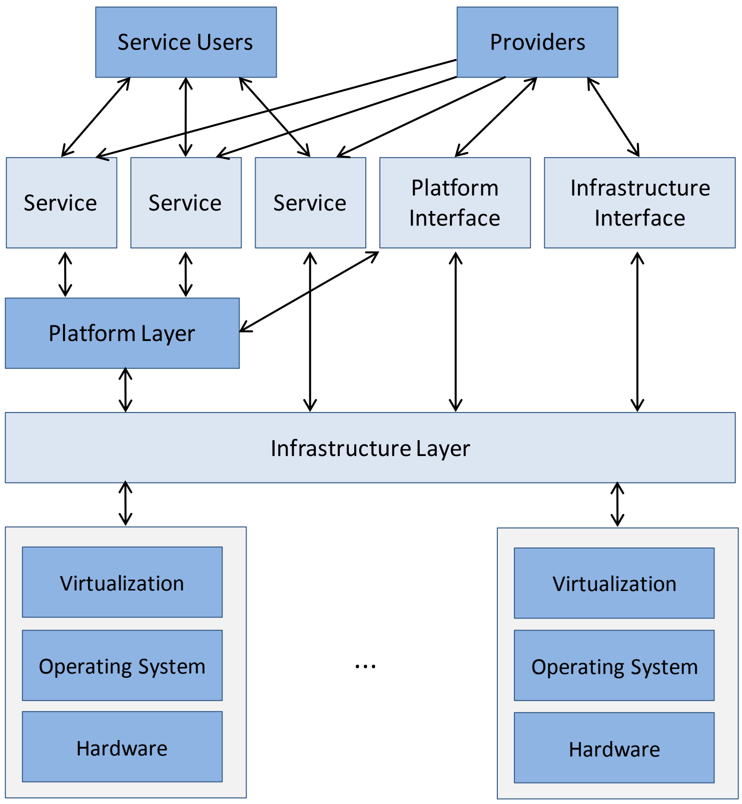


Figure 2-3 Overview of actors and layers in Cloud Computing [53].

The following subsections describe the architecture of cloud to support these thee level of service model and then detail each of these service models.

2.1.3.1 *Cloud Computing Architecture*

Zhang and colleagues [3] proposed a four layered architecture covering the three level of service model in cloud computing. As shown in Figure 2-4, the architecture includes the hardware/datacenter layer, the infrastructure layer, the platform layer and the application layer.

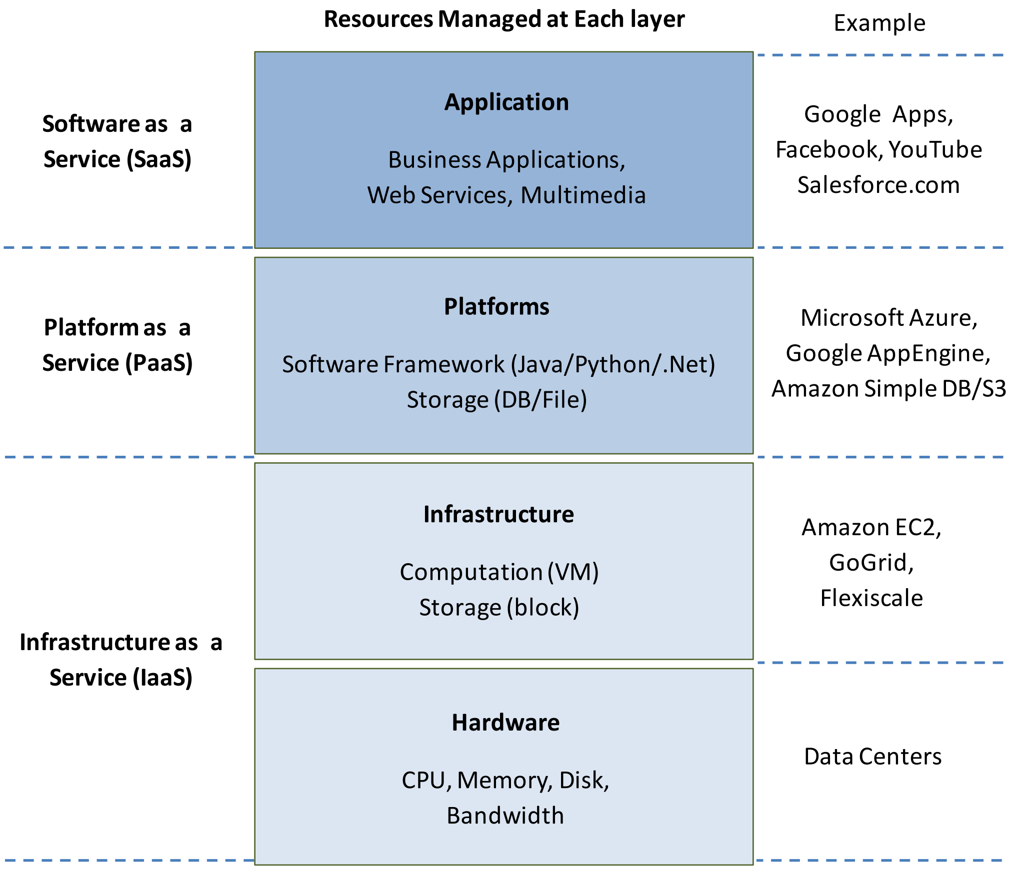


Figure 2-4 Cloud computing architecture [3].

- The *hardware layer* is in charge of the physical resources available in the cloud, such as physical servers, routers, power and cooling systems. The hardware layer is normally implemented in the datacenters.
- The *infrastructure layer*, known as the *virtualization layer* is a crucial part of cloud computing. Its main responsibility is providing a pool of storage and computing resources by logical partitioning of the physical resources using virtualization technologies like Xen [55], KVM (Kernel-based Virtual Machine) [56] and VMware [57].
- The *platform layer* is made up of operating systems and application frameworks to optimize running applications in Virtual Machine (VM) containers.

- The *application layer* includes the cloud applications that can trigger the auto-scaling feature to achieve better performance, availability and lower operating cost.

2.1.3.2 ***IaaS (Infrastructure as a Service)***

Cloud Infrastructure as a Service (IaaS) is the lowest layer where infrastructure providers deliver fundamental computing resources such as CPU power, memory and storage, virtual machine, network capabilities et cetera [41]. IaaS vendors help customers to reduce infrastructure investment cost and increase efficiencies of modernizing and developing IT capabilities. The IaaS vendors provide a scalable, secure, and accessible infrastructure over the Internet [58].

The cloud consumer can manage the allocated cloud infrastructures to develop, deploy and run applications. The applications may include operating systems as well as other applications. In this case, the user does not have control over the underlying cloud management infrastructure but may control the deployed applications and operating systems, storage and selected network components [41].

Amazon's EC2 [59], Windows Azure Virtual Machines [60], and Rackspace Cloud [61] are some popular available IaaS.

2.1.3.3 ***PaaS (Platform as a Service)***

A cloud Platform as a Service (PaaS) vendor provides infrastructure as well as a number of supported programming languages and tools to develop applications [44][41]. The consumer does not administer the underlying cloud infrastructure containing servers, operating systems, network, or storage, but is able to manage deployed applications and perhaps application hosting environment configurations. Fundamentally, PaaS provides a high level of abstraction to allow developers to focus on building higher level applications. Software developers can provide a custom developed application without bothering customers with managing and maintaining the infrastructure. Folch [62] defined three characteristic points in PaaS:

- Services for deployment, testing and maintenance of applications
- Multi-user architecture, specifically, scalability.
- Collaborative tools.

Delgado [4] proposed a server stack comparison between the managing capabilities of an IaaS or PaaS user as well as a private on-premises server user (Figure 2-5).

Google Compute Engine [63], AWS Elastic Beanstalk [64] and Microsoft Azure are popular PaaS examples.

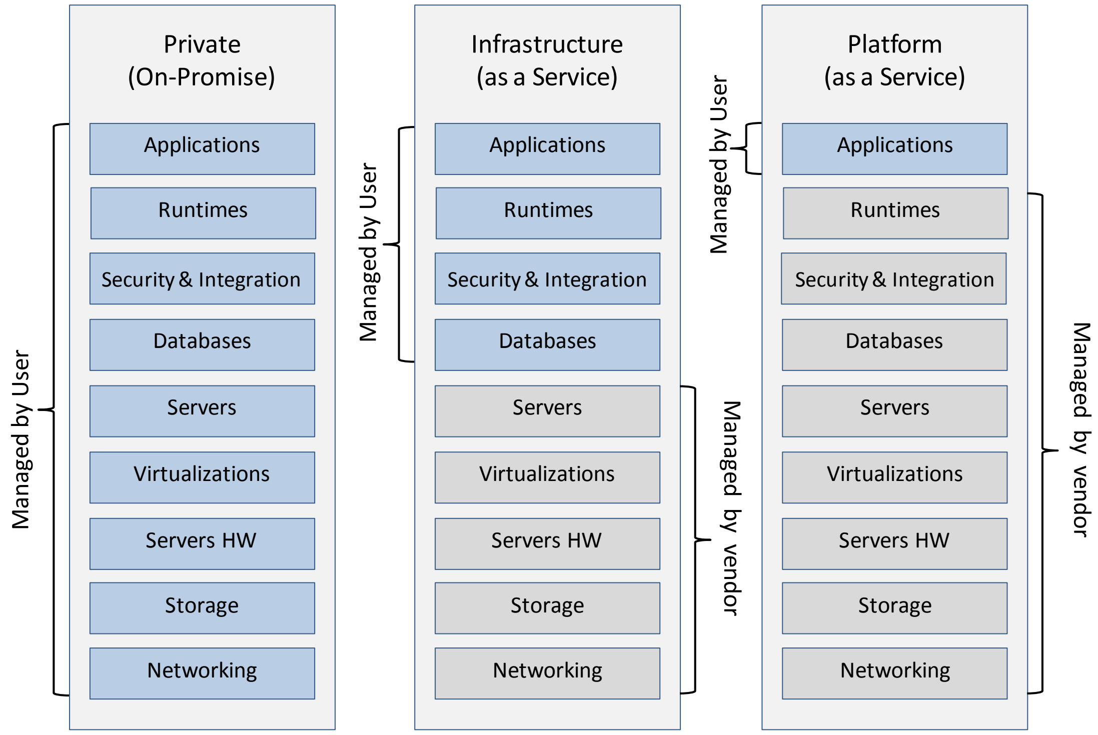


Figure 2-5 Server stack comparison between on-premise infrastructure, IaaS, and PaaS [4].

2.1.3.4 *SaaS (Software as a Service)*

Software as a Service (SaaS) is a cloud computing layer where users access applications running on a cloud infrastructure and offered on a platform on-demand [44][41]. The applications are available over Internet. Usually the users are able to run these applications using a client interface, like a web-browser. Practically, all of the underlying implementation and deployment is abstracted from the SaaS clients and only a specific set of configuration controls are accessible. Furthermore, the relevant data of SaaS applications is transparently placed in the cloud infrastructure.

Google Apps [65], Salesforce [66], SuccessFactors [66] are popular SaaS examples.

2.1.4 Deployment models

There are four generic types for cloud computing infrastructure deployment: public cloud, private cloud, community cloud and hybrid cloud [67][44][41][43][49]. The architecture, the datacenter's location, and the requirements of cloud customers determine different deployment strategies [4]. These various deployment models, shown in Figure 2-6, are explained in the following subsections.

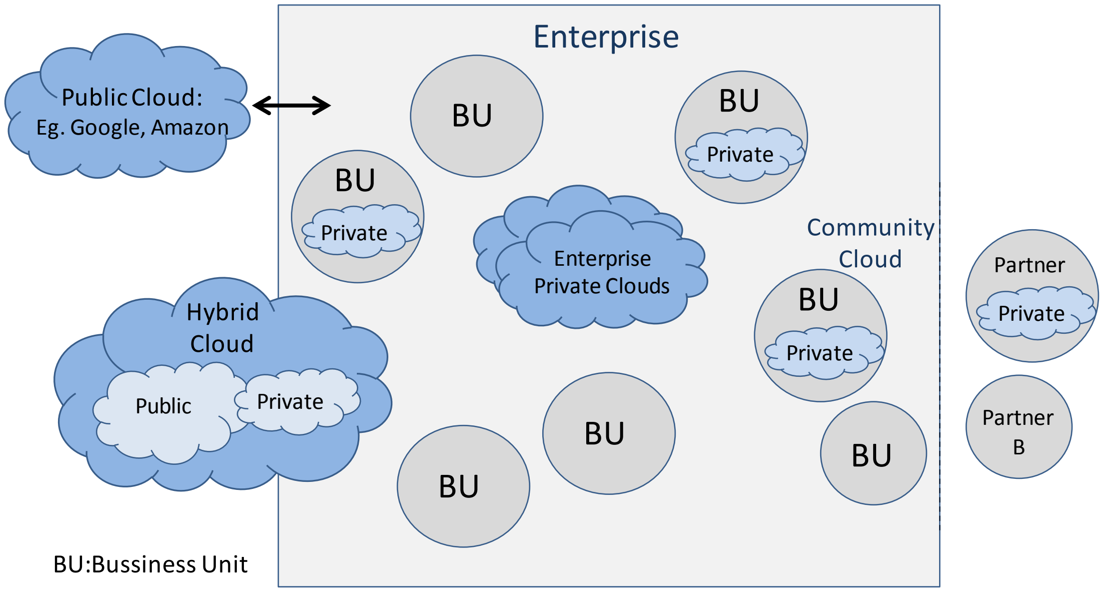


Figure 2-6 deployment models.

2.1.4.1 *Public clouds*

Public cloud is the most prominent form of current Cloud deployment models. According to the NIST definition [67][44][41][43], public cloud services are accessible publicly over the internet and a public cloud provider is in charge of management, maintenance and expansion of the shared infrastructure. Public cloud services may be free or served as a pay-per-usage model. User's data is not visible for the other public cloud customers and there is an access control mechanism for the users. Furthermore, Public clouds deploy solutions are delivered in an elastic, cost effective approach. Figure 2-7 shows the structure of a public cloud.

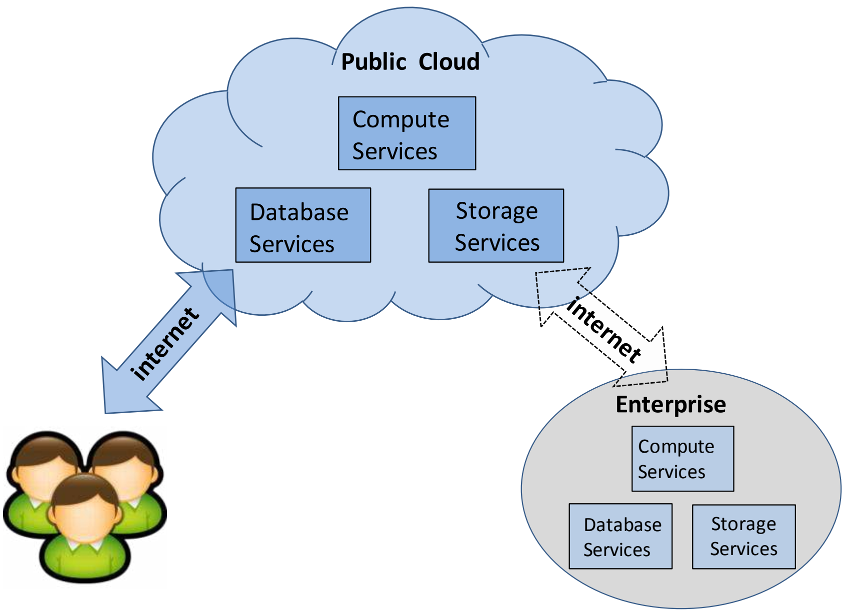


Figure 2-7 Public Cloud [67].

2.1.4.2 ***Private clouds***

Private cloud [67][44][41][43] is another deployment model for delivering cloud computing services within the Internet for private use (private networks). In this model the computing resources are operated only for a company and they are not available for unknown third parties. A private cloud customer can be the cloud owner, however, there can be a third party to install, administer, and maintain the cloud also. The cloud resources might be located within the customer's organisation premises or situated in a collocation facility as an off-site location.

Additionally, there is another term called *Virtual Private Cloud* as an alternative to a private cloud [4] where allocated physical resources are located within a public cloud. However, the allocated servers are not accessible by the other cloud customers. Figure 2-8 shows the structure of a private cloud.

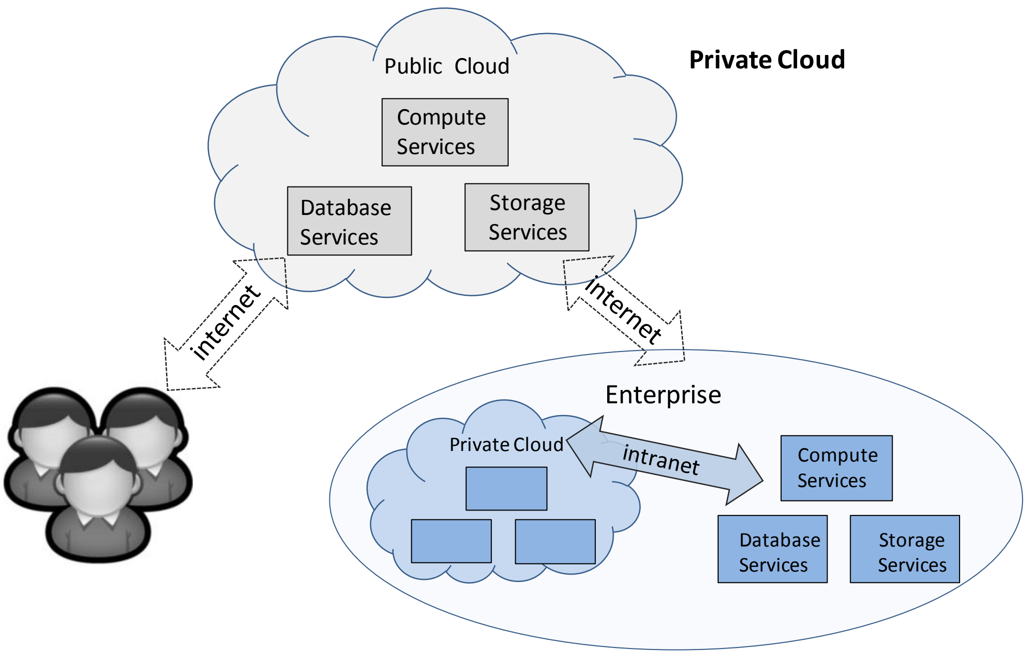


Figure 2-8 Private Cloud [67].

Dillon and colleagues [5] listed several justifications to establish private cloud within an organization instead of using public cloud:

1. The utilization of existing in-house resources increases efficiently.
2. There is lesser security concerns and legal issues to process, manage and control data within a private cloud [41].
3. Network bandwidth limitations and data transfer costs are insignificant compare to a Public Cloud.
4. The organizations can have better control over mission-critical activities behind their firewalls.

5. Universities, research centers and institutions can have their own private cloud for research and education purposes.

2.1.4.3 *Hybrid clouds*

Hybrid cloud is a model of deployment which combines of two or more clouds for example the private and public clouds that interoperate [67]. Organizations use the hybrid cloud model in order to outsource non-business-critical information and processing to the public cloud, and have the business-critical services and data locally in their control. In this model the joined clouds preserve their identities, however, they are bound together "by standardised or proprietary technology" [44]. standardization and cloud interoperability are the crucial concern in the hybrid cloud [5]. Figure 2-9 shows the hybrid cloud structure.

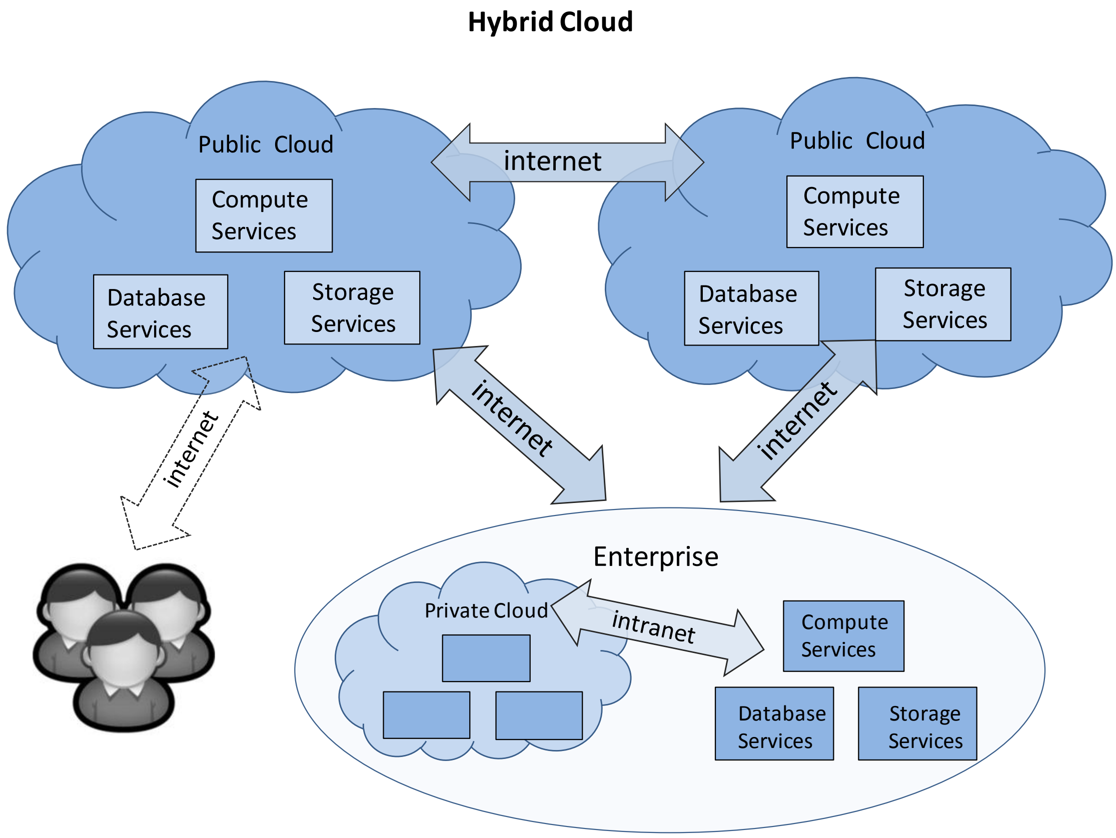


Figure 2-9 Hybrid Cloud [67].

2.1.4.4 *Community clouds*

Community cloud is the fourth deployment model for providing cloud services. The requirements for a community cloud are a subset of the requirements for a hybrid cloud [67]. In this model the cloud infrastructure is jointly constructed and shared among a group of organisations that have a common purpose such as policy considerations or certain security requirements [67][44][41][43]. Community cloud might be managed and controlled by the organisation or a third party vendor and it also might be located on-premises or off-premises. The communication between the community and the community cloud is done through an intranet. Figure 2-10 shows the community cloud.

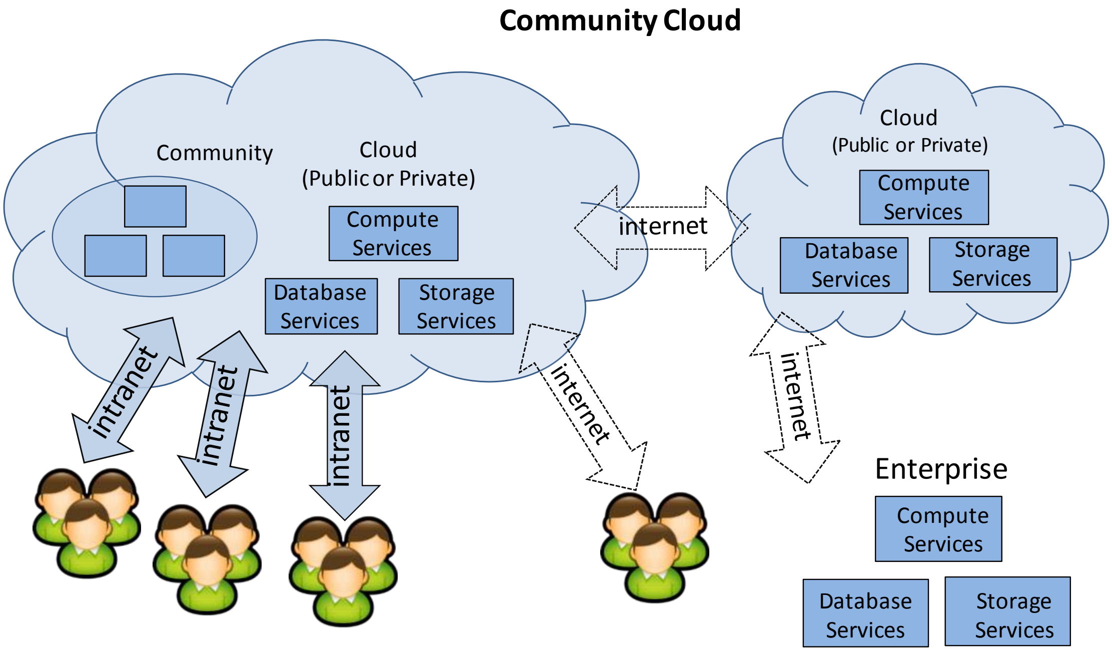


Figure 2-10 Community Cloud [67].

2.1.5 Current state of the art in Cloud Computing

Referring to the progress of cloud computing in the recent time, the IT industry has moved into a wide array of cloud platforms. All concepts regarding to definition and characteristics of Cloud Computing, service/delivery classification, deployment models, limitations and challenges in Cloud Computing are discussed in previous subsections. Table 2-2 summarizes current research works on Cloud Computing.

Table 2-2 Current state-of-the-art for cloud computing.

	Author (s)	Year	Title	What had been done?
[68]	Caryer et al.	2009	Grid/cloud computing interoperability, standardization and the Next Generation Network (NGN).	This paper discusses the relationship between grid and cloud computing, identifies gaps and overlaps in existing standards and identifies how grid and cloud technology could be exploited to improve the efficiency of NGN resources and to offer new "data" services to consumers.

	Author (s)	Year	Title	What had been done?
[69]	Dikaiakos & Katsaros	2009	Cloud computing: Distributed Internet computing for IT and scientific research.	This paper introduced key challenges in Cloud Computing including Cloud Interoperability, Data Management and Security and Privacy in Cloud environment.
[54]	"Open Cloud Manifesto"	2009	Open Cloud Manifesto.	This paper outlined the challenges, goals, and principles facing organizations that want to take advantage of open cloud.
[5]	Dillon et al.	2010	Cloud Computing: Issues and Challenges.	In this paper Service-Oriented Computing and Grid computing and their relationship with Cloud computing are reviewed. Furthermore, the main challenges of the Cloud computing are classified. Finally, the paper focused on the Cloud Interoperability issue.
[9]	Borenstein & Mimecast	2011	Cloud Computing Standards.	This article is concerned mostly with standards for cloud service users and providers. It discussed on two main type of standards for cloud computing: (1) prescriptive standards to state specific aspects of doing something, like Simple Mail Transfer Protocol (SMTP) or Transmission Control Protocol/Internet Protocol (TCP/IP); (2) evaluative standards to provide a uniform concept to assay the quality of doing something, like ISO 9000.
[67]	Computing et al.	2010	Cloud Computing Use Cases A white paper produced by the.	This document attempted to discuss the capabilities and requirements that have to be standardized in a cloud environment to ensure interoperability, ease of integration and portability.
[10]	Jr et al.	2011	The Problem with Cloud-Computing Standardization.	In order to avoid having multiple standards address the same issues while having no standards addressing others, this paper described the standard bodies and the standardization challenges till 2011. Additionally, the problems in standardization process in cloud computing are highlighted.

	Author (s)	Year	Title	What had been done?
[3]	Zhang et al.	2010	Cloud computing: state-of-the-art and research challenges.	This paper presented an overview of cloud computing, also identified key concepts, architectural principles, latest implementation as well as research challenges in Cloud environment.
[70]	Shroff	2010	Enterprise Cloud Computing Technology, Architecture, Applications.	This book explored the technical aspects of cloud computing including: the major cloud platforms, key technologies for building cloud platforms (Web services, AJAX and mashups, Virtualization technology, and Multi-tenant software), and new programming models and development paradigms (Data in the cloud, MapReduce and extensions, and Dev 2.0 platforms)
[71]	Peng et al.	2009	Comparison of Several Cloud Computing Platforms.	This paper compares Abicloud, Eucalyptus, Nimbus and OpenNebula cloud computing platforms. Focused on the aspects such as the architectures, characteristics, application and so on, a detailed comparison has been presented in this paper.
[72]	Fan et al.	2011	Toward Optimal Deployment of Communication-Intensive Cloud Applications.	This paper presented a new clustering-based approach to opt optimal cloud nodes for spreading communication intensive applications to the cloud environment.
[73]	Ahson & Ilyas	2010	Cloud computing and software services: Theory and techniques.	This book discussed on technical information, from fundamental approaches to research grade material including future directions, of cloud computing.

2.1.6 Current alternatives in the cloud computing market

Nowadays cloud environments include hundreds of independent, heterogeneous, private/hybrid clouds, but many business operators have predicted that the process toward interoperable cloud scenarios will begin in the near future. In order to analyzing the actual platform, Table 2-3 introduces a few important cloud computing offers and specify the type of provided services according to the service/delivery model classification presented previously. Appendix A presents more existing cloud service providers.

Table 2-3 Current alternatives in the cloud computing market.

	Company	Service	Company's description
1	**Amazon EC2**	Infrastructure as a Service	Since staking its claim with Amazon Web Services in early 2006, Amazon.com has established itself as a pioneer. Amazon EC2 (Elastic Compute Cloud) users obtain and configure capacity and control computing resources while running them on Amazon's environment. The real draw is the ability to add capacity and scale in seconds, or reduce capacity as needed while customers only pay for what they use. It also is designed for use with other Amazon Web Services.
2	**GoGrid**	Platform as a Service	Do you have only minutes to build an enterprise-grade cloud infrastructure? GoGrid's got you covered. The GoGrid platform lets users deploy Web and database cloud services, mount infinite-volume cloud storage, add load-balancing and create, save and deploy custom cloud server images. GoGrid makes it even easier by tying in API libraries and tools.
3	**Google**	Software as a Service	If there were any doubt that cloud computing -- and Google Apps in particular -- were ready for prime time, it dissipated last year when the Los Angeles city government adopted Google's e-mail and on-demand applications under a $7.25 million contract. L.A. chose Google Apps over Microsoft, which competed for the sale. What's more, in early 2009 the company began offering its Google Apps Premier Edition hosted office productivity software through solution providers for the first time.
4	**Google App Engine**	Platform as a Service	With Google App Engine, users can build, run and maintain their applications on Google's infrastructure with no servers to maintain. Apps can be served from their own domain or a free domain on Google's appspot.com domain. As with most platforms, App Engine is pay to play. It supports several programming languages and costs nothing to get started. Apps have up to 500 MB of storage and enough CPU bandwidth to support an app serving about 5 million page views a month.
5	**IBM**	Infrastructure as a Service (Storage	IBM's Smart Business Storage Cloud is a private cloud service that supports multiple petabytes of data and billions of files. It is based on IBM's blade server and XIV storage technologies.

	Company	Service	Company’s description
		Vendors)	The service lets businesses build an on-site storage cloud managed by IBM, or back up data to one of IBM's own data centers. IBM also plans to build a business-grade public cloud for storage.
6	**IBM**	Software as a Service	Many industry observers have long viewed IBM's Lotus division as one more road-kill victim of the Microsoft juggernaut. But Lotus is meeting with some success with its LotusLive offerings, a collection of on-demand collaboration and communications applications that provide an alternative to on-premise applications such as Microsoft Office and cloud-computing personal productivity tools such as Google Apps.
7	**IBM**	Infrastructure as a Service	When it comes to the cloud, IBM isn't messing around. The proof is in the pudding with its Smart Business Cloud services and solutions. With its combination of services and systems, which comprises public and private clouds and cloud-based versions of some of IBM's most popular applications, IBM is looking to the cloud for everything from analytics and software and services delivery to services such as storage management and cloud-based e-mail, scheduling and contact information.
8	**Microsoft**	Platform as a Service	Windows Azure is Microsoft’s cloud computing platform, available now for free. Set to debut Feb. 1 as a paid service, Azure offers an environment for developers to create cloud apps and services. The platform will also run alongside current Microsoft environments offering an OS as a service in Windows Azure, a relational database in the cloud in Microsoft SQL Azure and the Windows Azure platform AppFabric, which eases connections between cloud and on-premise apps.
9	**Open Nebula**	Infrastructure as a Service	This open-source toolkit fits snuggly into existing data center environments to build any type of cloud deployment. OpenNebula can be used to manage virtual infrastructure in the data center or to manage a private cloud. It also supports hybrid clouds to combine local infrastructure with public cloud infrastructure for hosting environments. Additionally, it

	Company	Service	Company's description
			supports public clouds by offering cloud interfaces to expose its functionality for virtual machine, storage and network management.
10	**RackSpace**	Platform as a Service	With its CloudServers offering, RackSpace delivers servers on-demand via a cloud-driven platform of virtualized servers. Users can add new instances and reduce instances within seconds while paying for what's provisioned. It also offers CloudSites, a fully-managed Web hosting platform that lets the users code it and load it and offers patching and security, monitoring, redundancy, clustering and the power of the cloud. Add to that RackSpace's CloudFiles file storage and hosting in the cloud, and the platform is complete.
11	**Salesforce.com**	Platform as a Service	The cloud computing behemoth is kicking its presence up a notch. Its Force.com development platform lets users log in, build an app and push it out into the cloud. All told, it's supposed to help build and run applications faster at a fraction of the cost of traditional software platforms. The platform includes a database, security, workflow, user interface and other tools to guide the process for building business apps, mobile apps and Web sites.
12	**Salesforce.com**	Software as a Service	What Salesforce.com has done is popularize the concept of cloud computing, turning a vague IT architectural concept into a mainstream computing practice and providing Customer Relationship Management (CRM) SaaS applications that -- for many businesses --were their entre into cloud computing. Salesforce has sought to solidify its position as a SaaS/cloud computing leader with its Force.com platform and infrastructure tools for developing and running cloud computing applications. Yet Salesforce's on-demand CRM sales and customer service applications still account for the bulk of the company's sales.

2.2 Intercloud Interoperability

Currently, cloud computing is an emerging computation paradigm in information technology and networking. Although Cloud Computing shared services has been increasingly utilized by diverse users, the research on Cloud Computing is still at an early stage. There are many existing cloud challenges that have not been fully addressed in addition to the new emerging issues introduced by enterprise applications. These can be an obstacle to the growth of Cloud Computing and hinder its use by organizations for outsourcing applications with sensitive information. One of the existence challenges is the Intercloud Interoperability issue.

Intercloud became popular in early 2009 [74][45][75]. The Intercloud concept is based on the fact that each single cloud has limited computing resources in a restricted geographic area. Intercloud addresses the interoperability between various cloud computing instantiations where each cloud would use computing resources of other clouds. Cloud Computing environments need to be interoperable in order to reduce scaling/producing cost within the development of the components. Cloud costumers should be able to migrate in and out of the cloud and switch between providers based on their needs, without a lock-in which restricts customers from selecting an alternative provider. Furthermore, cloud providers should be able to interoperate among themselves to find an alternative cloud provider to give better services. The present Intercloud network merely connects different cloud systems and each cloud provider has its own way on how cloud applications/customers interact with the cloud. Feldhaus [12] summarized the current challenges in Cloud Interoperability as follow:

- Several different Cloud Standards from different parties are available.
- Existing Open Grid Forum (OGF) standards not or only partly ready for the cloud.
- A consistent OGF Cloud Portfolio is needed.
- Strategies for combining different Cloud Standards / APIs are needed.
- Existing implementations of Cloud APIs need to get interoperable.
- Combined Interoperability Verification Suites need to be developed.
- It is essential to discuss on issues related to specifications and implementation.

Currently different organizations, such as IEEE, are working on developing essential standards and appropriate APIs for Intercloud Interoperability. The future Intercloud network will expand the required functions to prepare collaboration among cloud services. Grozev & Buyya summarized their studies and classified 20 major Intercloud developments including both academic and industry projects [76]. According to their studies, Intercloud is classified as (Figure 2-11):

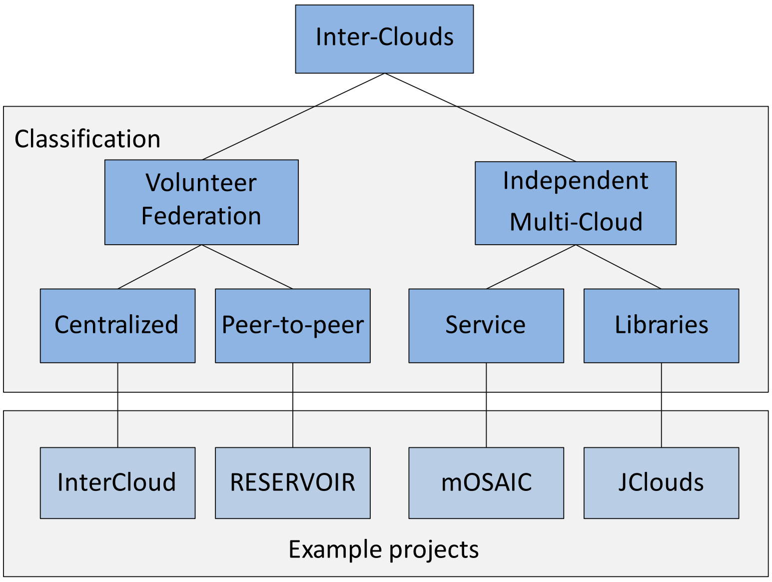


Figure 2-11 Architectural classification of intercloud [76].

- *Volunteer federation*: when there is voluntarily collaboration between cloud providers that is often feasible for governmental clouds or private cloud portfolios.
- *Independent*: when an application or its broker independently from the cloud providers (both governmentally and private clouds) exploit multiple clouds.

Volunteer federation is classified in two architectural categories (Figure 2-12) [76]:

- *Centralised:* there is a central entity in this architecture for intercloud to perform or facilitate resource allocation.
- *Peer-to-Peer*: in this architecture, each cloud cooperates with the others directly.

Furthermore, the *Independent Intercloud* development is classified in two architectural categories (Figure 2-12) [76]:

- *Services:* a service hosted externally or in-house by the users provides the application. Often, broker components are part of this type of services, and an SLA or a set of provisioning rules for application developers are defined by application and the service executes in the background according to the predefined attributes.
- *Libraries:* usually custom application brokers are required to provide and schedule application components directly across clouds. These approaches exploit intercloud libraries which facilitate utilizing multiple clouds uniformly.

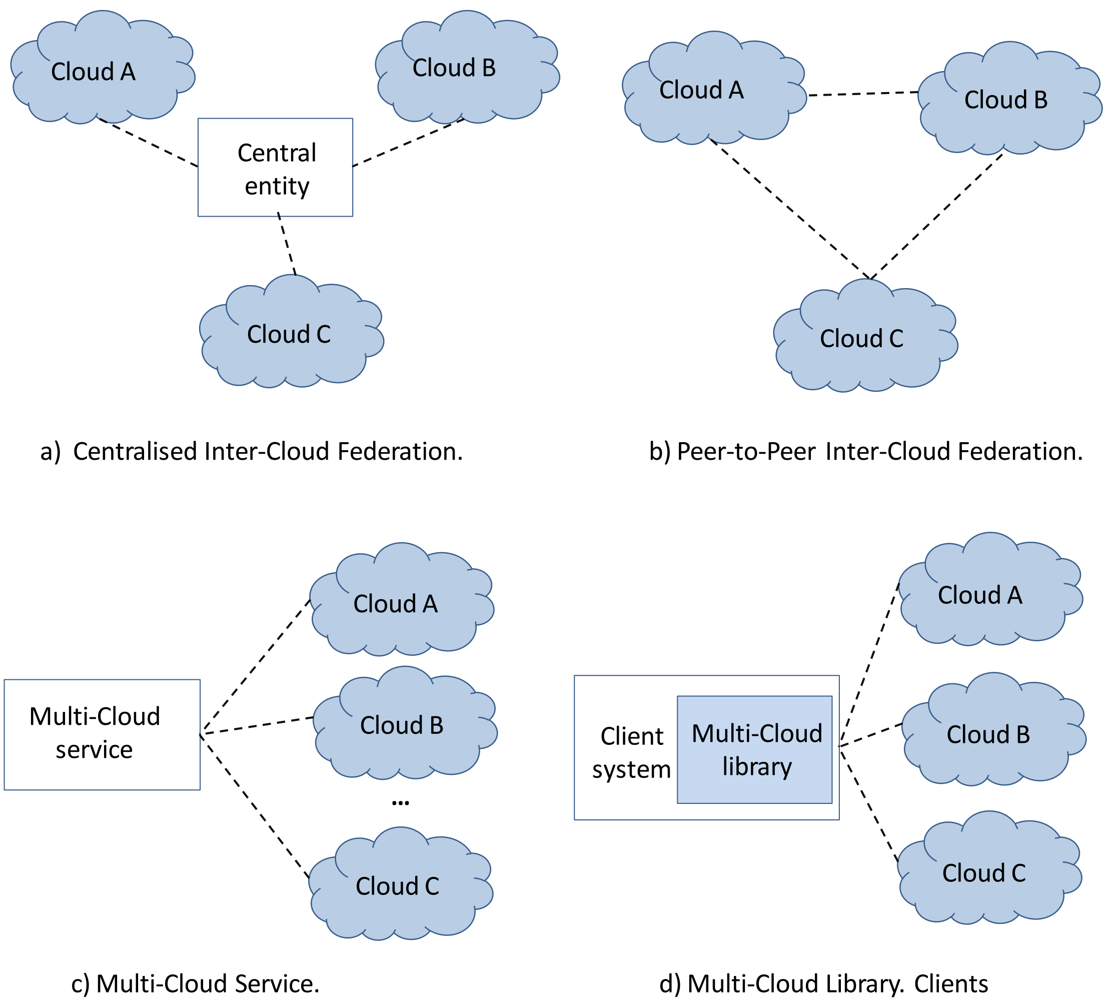


Figure 2-12 Intercloud developments' architectures [76].

Grozev & Buyya [76] summarized their studies and classifications on 20 major Intercloud developments include both academic and industry projects in Table 2-4.

Table 2-4 Summary of Intercloud projects [76].

Project	Type, Organization	Architecture	Brokering Approach
InterCloud	Research project , University of Melbourne	Centralised federation	SLA based and Directly managed
Contrail	Private and public European research organizations Funded by European Union (EU)	Centralised federation and Independent Service	SLA based
Dynamic Cloud Col-laboration (DCC)	Academic research project supported by South Korean research funds.	Centralised federation	SLA based

Federated Cloud Management (FCM)	Academic research project supported by EU funds.	Centralised federation	SLA based
RESERVOIR	Private and public European research organisations Funded by EU	Peer-to-peer federation	SLA based and Trigger-Action
Open Cirrus	Research testbed by academic and industry partners. Partially funded by US NSF.	Peer-to-peer federation	Directly managed
OPTIMIS	Private and public European research organisations Funded by EU	Peer-to-peer federation/ Independent service	SLA based
Arjuna Agility	Commercially owned	Peer-to-peer federation	Trigger-Action
Global InterCloud by Bernstein et al.	Publications are by people from miscellaneous companies -CISCO, Huawei Technologies, EMC Corporation	Peer-to-peer federation	SLA based
mOSAIC	Private and public European research organisations Funded by EU	Independent service	SLA based
STRATOS	York University. Supported by Canada's NSERC funds, Amazon and CA Inc.	Independent service	SLA based
Commercial Cloud Management Systems (RightScale, EnStratus, Scalr, Kaavo)	Commercially owned	Independent service	Trigger-Action
Libraries (JClouds, LibCloud, DeltaCloud, SimpleCloud, Apache Nuvem)	Open source projects	Multi-Cloud libraries	Directly managed

Dillon and colleagues [5] summarized some key intentions to solve the interoperability issue in the Cloud environments:

- *Intermediary Layer*: Providing an intermediary layer between cloud users and cloud computing resources (e.g.VM) may help improving cloud systems' interoperability. For instance, an abstraction layer can be developed at a higher level to provide a single resource usage model, user authentication model and API to support heterogeneous cloud providers.

- *Standard:* Standardization can be a solution to address the interoperability problem. The consensus between existing cloud providers, such as Amazon, Microsoft, or Google, is a big problem that makes standardization process very intricate.
- *Open API:* Open cloud API can define a set of clear and simple web services interfaces, to allow consumers to create and administrate cloud resources, including compute, storage, and networking components in a unified way.
- *SaaS and PaaS Interoperability:* Cloud providers mostly focused on IaaS interoperability problems, and few studies have highlighted interoperability issues in the other service deployment models.

In order to show the distinctive ways of interaction between cloud users and providers, NIST [77] defined following *use cases for Cloud Computing Interoperability:*

- Copy Data Objects Between Cloud-Providers
- Dynamic Operation Dispatch to IaaS Clouds
- Cloud Burst from Data Center to Cloud
- Migrate a Queuing-Based Application
- Migrate (fully-stopped) VMs from One Cloud Provider to Another

Lewis [78] after studying use cases proposed by NIST and OMG, presented four main cloud interoperability use cases that can benefit from current standards:

1. User Authentication: A user who has established an identity with a cloud provider can use the same identity with another cloud provider.
2. Workload Migration: A workload that executes in one cloud provider can be uploaded to another cloud provider.
3. Data Migration: Data resided in one cloud provider can be moved to another one.
4. Workload Management: Custom tools developed for cloud workload management can be used to manage multiple cloud resources from different vendors.

Bernstein and colleagues [45] defined "Intercloud vision" shown in Figure 2-13 to depict that various services from heterogeneous cloud systems are interoperable. Reference topology in Figure 2-14 shows of how clouds interact in an InterMany of standards from current Internet networks are appropriate standards to reuse in Intercloud. Bernstein and colleagues [45] collected protocols, standards, formats, and common mechanisms as a beneficial architecture to implement Intercloud interoperability (Figure 2-15).

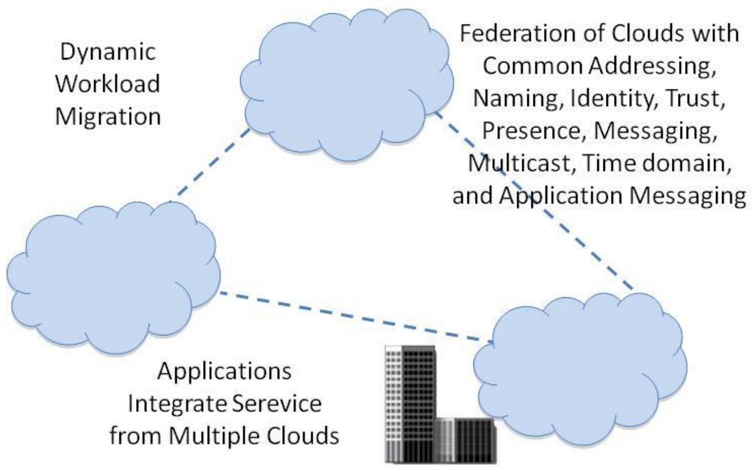


Figure 2-13 The Intercloud Vision [45].

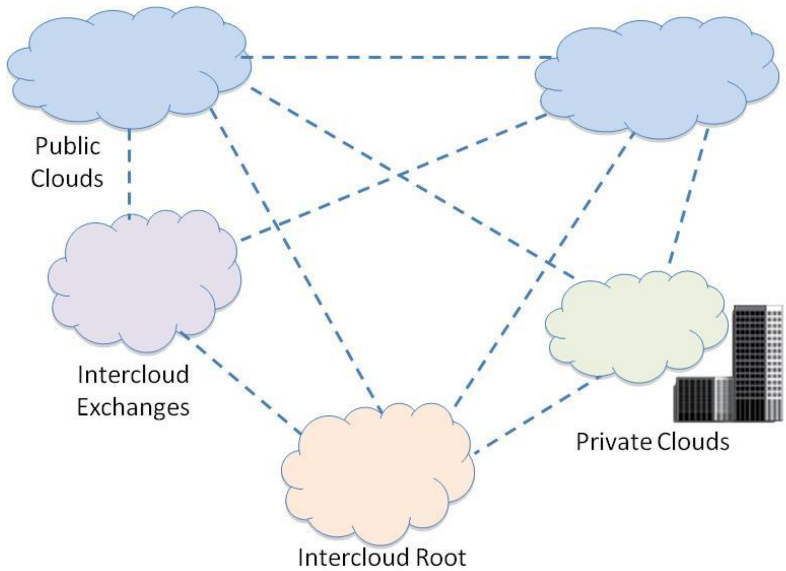


Figure 2-14 Reference Intercloud Topology [79].

Parameswaran and Chaddha [75] explained and examined two approaches in order to provide Intercloud standards and interoperability view:

- Approach 1: Unified Cloud Interface/Cloud Broker and approach.
- Approach 2: Enterprise Cloud Orchestration Platform /Orchestration layer.

Recently, the IEEE P2302 group [80] has been focusing on cloud-to-cloud interface standards for Intercloud Interoperability and Federation. Celesti in 2010 [81] proposed a three-phase (discovery, match-making, and authentication) cross-cloud federation model. This model represents an architectural solution (with some restrictions) to provide interoperability.

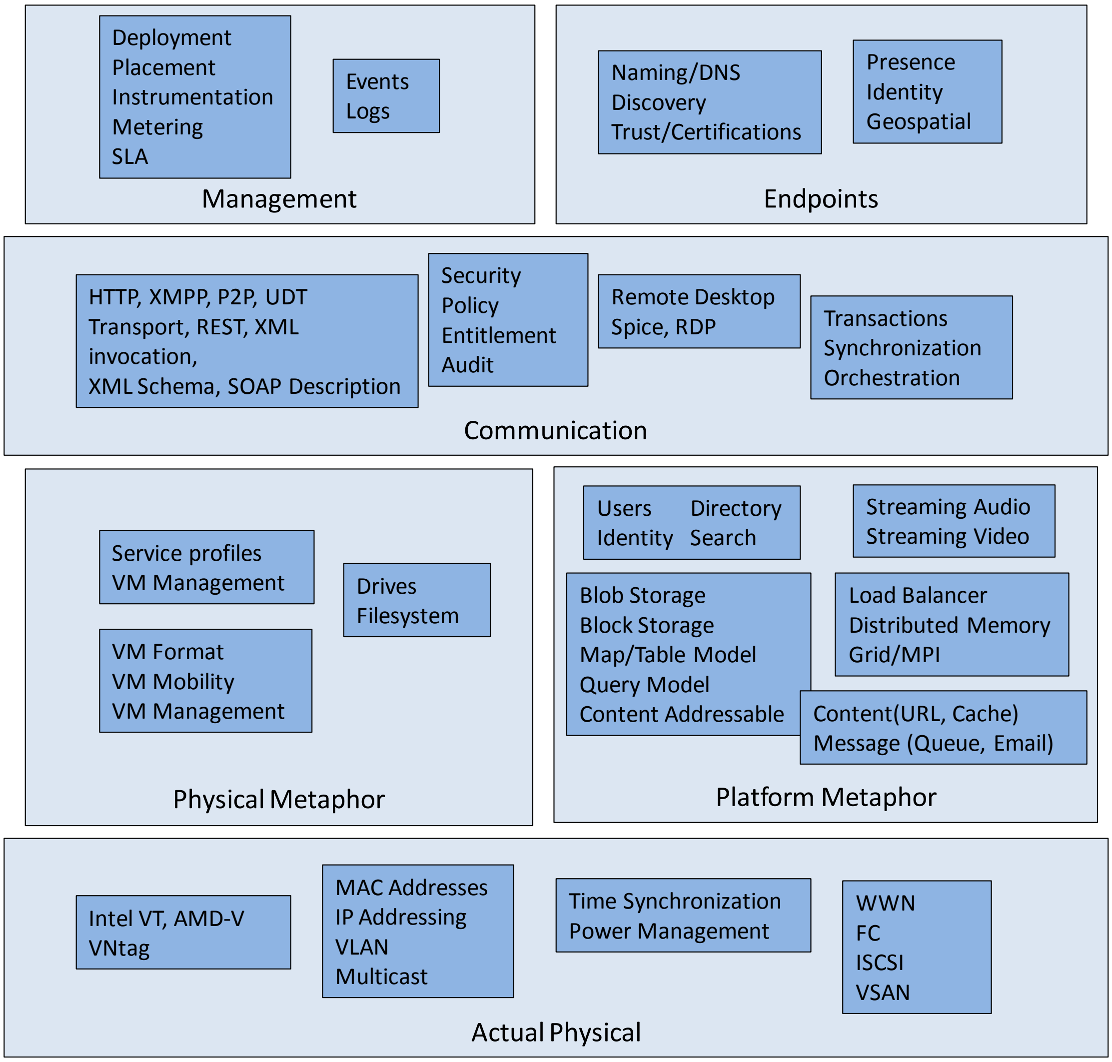


Figure 2-15 An Architecture for Intercloud Standards [45].

In July 2009 in Japan, the Global InterCloud Technology Forum (GICTF) published Intercloud Protocol [82][83][84][85] and Resource Data Model [85] to recognize the operational requirements of Intercloud systems and describe a peer-to-peer intercloud interface. However, it has been claimed in [86] Point to Point protocols are not appropriate for Intercloud Protocols and accordingly many-to-many mechanisms including Messaging and Presence Protocol (XMPP) for transport, and Semantic Web techniques such as Resource Description Framework (RDF) as a way to specify resources have been proposed. Bernstein and colleagues [86] used an XMPP Java API for a Cloud Service. Celesti also selected Extensible Markup Language (XML) based technologies like XMPP to address interoperability issues [81].

Nagireddi and Mishra [87] proposed an ontology based framework for searching services provided by different Cloud Service Providers. Abouzamazem and Ezhilchelvan [88] studied tolerating outages by intercloud replication and proposed an approach to replicate a service on N outage-independent clouds. Pop and colleagues [89] presented a genetic scheduling algorithm for independent tasks in intercloud environments where the selection phase is based on reputation evaluation. Finally, Demchenko and colleagues [90] presented their on-going research on developing a Interoperability Framework to support on-demand provisioning by heterogeneous cloud service providers.

Nevertheless, there is not yet a comprehensive proposal that support the intercloud interoperability concerns. This thesis is proposing a novel solution that can support intercloud interoperability for dynamic operation dispatch to IaaS Cloud Providers (CPs).

The concept and main literature review regarding to Cloud Interoperability and Intercloud are discussed in current section. Following table summarizes some other available research works including these concepts.

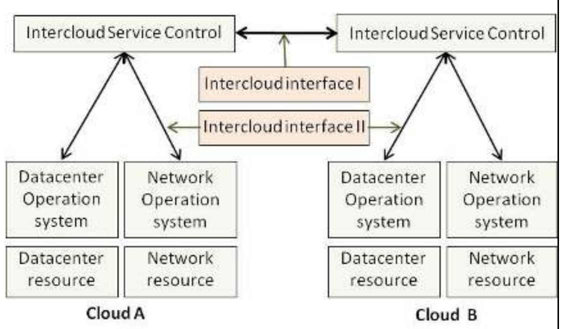
Table 2-5 Current state-of-the-art for Cloud Interoperability and Intercloud.

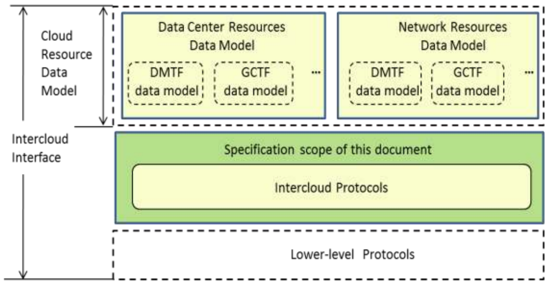
	Author(s)	Year	Title	What had been done?
[75]	Parameswaran & Chaddha	2009	Cloud interoperability and standardization.	This paper presented cloud computing standards and interoperability view. Two interoperability approaches, Unified Cloud Interface/Cloud Broker and Enterprise Cloud Orchestration Platform /Orchestration layer, were reviewed and future of these approaches were discussed. Finally, the emerging scenario and important interoperability factors from several perspective were explained
[8]	Rings et al.	2010	On the Standardization of a Testing Framework for Application Deployment on Grid and Cloud Infrastructures.	The paper presented a testing framework to determine interoperability in grid and cloud computing environment. The framework developed by European Telecommunications Standards Institute (ETSI) test process based on Grid Component Model (GCM) standard.
[91]	Coutinho et al.	2012	Cloud-based negotiation for sustainable enterprise	This paper specified a collaborative framework to improve interoperability between organisations acting in a current industrial market, using a model-driven, cloud-based platform and services.

			interoperability	
[20]	Kretzschmar & Hanigk	2010	Security Management interoperability challenges for Collaborative Clouds.	This paper highlighted Cloud security management issues and interoperability challenges for Collaborative Clouds.
[92]	CloudStandards	2013	Cloud Standards Customer Council	It is an end user advocacy group dedicated to accelerating cloud's successful adoption, and drilling down into the standards, security and interoperability issues surrounding the transition to the cloud.
[45]	Bernstein, Ludvigson et al.	2009	Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability.	The authors considered the use cases for interoperability and identified a set of protocols and formats, collectively “Intercloud Protocols” for enabling the use cases. Additionally the authors discussed that a set of common mechanisms, collectively “Intercloud Root”, are required both inside and among the Clouds. The paper specified the set of these common protocols as “Intercloud Root”.
[79]	Bernstein, Clara et al.	2010	Using Semantic Web Ontology for Intercloud Directories and Exchanges.	The authors claimed in [79] instead of point to point connection, using Intercloud Directories and Exchanges will promote collaboration among heterogeneous cloud providers. In this regard, the authors introduced a mechanism to describe, catalog, and mediate Cloud Computing resources using the Semantic Web Resource Definition Framework (RDF) and a common Ontology of Cloud Computing Resources. The paper concluded that “Intercloud Exchanges” along with “Ontology based Computing Resources Catalog” and “Extensible Messaging and Presence Protocol” (XMPP) protocol are the essential factors to implement “Federated Cloud” environment.

[80]	IEEE	2011Started	IEEE P2302 Working Group (Intercloud).	P2302 is an IEEE working group on Standard for Intercloud Interoperability and Federation (SIIF). This group aim to develop standards to define topology, functions, and governance for Intercloud interoperability and federation.
[81]	Celesti et al.	2010	How to Enhance Cloud Architectures to Enable Cross-Federation.	Celesti and the colleagues, considering architectural limitations for providing Intercloud Interoperability, presented some improvement on federation capabilities. The proposed solution is a module called Cross-Cloud Federation Manager (CCFM) that is compatible with any cloud architectures and is exploited new XML based technologies like XMPP, eXtensible Access Control Markup Language (XACML), and Security Assertion Markup Language (SAML). The module contains three agents (Discovery, Match-making and Authentication) responsible to (1) discover other available clouds; (2) select the appropriate ones between the recognized clouds; and (3) established a trust context with the preferred clouds.
[83]	Global Inter-Cloud Technology Forum (GICTF)	2010	Use Cases and Functional Requirements for InterCloud Computing.	This document represented the essential functionalities for intercloud systems and the specifications for intercloud interfaces. It included discussion on : "quality requirements for services"; "advantages of intercloud computing" (guaranteed end-to-end quality for each service, performance, availability, and service cooperation support); "Functional requirements for intercloud computing" (matching between service quality requirements of consumer and Service Level Agreement (SLA), monitoring (resource, service, and dead/alive), provisioning, resource discovery and securement, resource management, service setup, authentication interworking, network interworking, alternation and retrieval of data for access route from consumer, releasing resources);

				"Functional structure and interfaces of cloud systems in intercloud computing".
[84]	Global Inter-Cloud Technology Forum (GICTF)	2012	Technical Requirements for Supporting the Intercloud Networking.	This paper after clarifying Intercloud concept, specified technical requirements for the Intercloud systems, for instance address management, mobility management, routing management, security, network infrastructure management, cloud VPN management, and system collaboration. Furthermore, the paper indicated the anticipations of the Intercloud network around 2014, and 2016, based on the use cases of the Intercloud as considered by GICTF in 2010.
[85]	Global Inter-Cloud Technology Forum	2012	Intercloud Interface Specification Draft (Cloud Resource Data Model).	In order to define the intercloud interface, GICTF released two documents: (1) Intercloud Protocol [82] and (2) Cloud Resource Data Model [85]. "Cloud Resource Data Model" document includes an example of a cloud network resources data model that contains the internal Local Area Networks (LANs) and external Wide Area Networks (WANs) of a data center within a cloud resource data model.
[93]	Bernstein & Vij	2010	Intercloud directory and exchange protocol detail using XMPP and RDF	This paper investigated the practicability of an XMPP mechanism for many-to-many connectivity instead of point-to-point connectivity for Intercloud. Also it discussed about the solution to describe, catalog and mediate Cloud Computing resources using Semantic Web Ontologies, implemented using RDF methods.
[86]	Bernstein & Vij	2010	Using XMPP as a transport in Intercloud Protocols	Regarding to integrate Intercloud Protocol to solve intercloud interoperability, this paper proposed many-to-many mechanisms including XMPP as an appropriate choice and it is claimed that point-to-point protocols such as Hypertext Transfer Protocol (HTTP) are not competent for one-to-many or many-to-many connectivity. To evaluate suitability of XMPP, following techniques were

				considered: Applying XMPP into an Intercloud Topology Securing the XMPP conversation through TLS Authentication over XMPP through SAML Service Invocation over XMPP through IO Data XEP, XMPP Web Services for Java (xws4j) RDF and SPARQL within XMPP XMPP Java API to a Cloud Service
[94]	Bernstein, Vij et al.	2011	An Intercloud Cloud Computing Economy - Technology, Governance, and Market Blueprints.	This paper surveyed the latest achievements, until 2011, in cloud computing federation, the Intercloud. It perceived that the technical governance of the Internet can be used as a model for the Intercloud, but the operational model of the Internet is outdates and U.S.-centric, and needs to be re-evaluated for the Intercloud.
[82]	Global Inter-Cloud Technology Forum (GICTF)	2012	Intercloud Interface Specification Draft (Intercloud Protocol).	In order to define the intercloud interface, GICTF released two documents: (1) Intercloud Protocol [82] and (2) Cloud Resource Data Model [85]. The Intercloud Protocol document determined the functional necessities of intercloud systems and specified an intercloud interface in particular terms with three reference points: between intercloud service controls of different cloud service providers between intercloud service controls and data center operation systems between the intercloud service controls and the network operation systems.  *Figure 2-16 Cloud resource data model* [82].

				The structure of proposed interface shown in [82] with three main layers: the lower-layer protocols, the intercloud protocol, and the cloud resource data model.  *Figure 2-17 Intercloud interface structure* [82].
[76]	Grozev & Buyya	2012	Inter-Cloud architectures and application brokering: taxonomy and survey	This paper classified and analysed the state of the art in InterCloud developments (Already discussed in this section)

The Intercloud network scenario is still in an early stage. It needs more research work to provide sufficient functions to enable collaboration between cloud services. We are planning to present a framework to develop Intercloud Interoperability using two key technologies, MDA and SOA, described in 3.1 and 3.2 sections.

Chapter 3

MDA and SOA

3 MDA and SOA

As cloud computing is a new paradigm exploited rapidly in various enterprises, in previous chapters, we discussed the concepts, current state of the art and challenges in the area of the cloud computing. Based on our study, intercloud interoperability is a critical research challenge with several use cases. Intercloud Interoperability can enable cloud providers to deliver better quality of services, avoid data lock-in, and reduce scaling/producing costs. Presenting a solution to support intercloud interoperability can have a significant impact on the future of cloud companies and customers. This PhD thesis proposes a novel solution to support intercloud interoperability for dispatching operations to the most effective computing resource cloud providers. To achieve this goal, it is fundamental to identify the most appropriate developing methods.

In order to devise the best approaches for implementation of the framework, current research different application design approaches were studied. Based on our study, applying Service Oriented Architecture (SOA) in developing cloud services can provide the required service models with agility and scalability. Additionally SOA can provide interoperability between applications by put up application systems as group of published services. Furthermore, According to the literature, interoperability between applications and services is inherent to the system design using Model Driven Architecture. In general our studies show, recently, Model Driven approaches from OMG and Service-Oriented base methodology are increasingly exploited to develop different frameworks to solve several issues such as interoperability. Thus, we selected MDA and SOA approaches to develop our intercloud interoperability framework.

In order to have better understanding of MDA and SOA approaches, this chapter describes the capability of MDA and SOA approaches as well as current research work in utilizing these approaches.

3.1 The Model Driven Architecture (MDA) Approach

The Object Management Group (OMG) announced the Model Driven Architecture (MDA) initiative as a software development approach to system-specification and interoperability based on the use of formal models [32]. MDA focuses on the development of models rather than detailed, platform-specific code which can be generated when needed. Instead of requiring developers to define every detail of a system's implementation using a programming language, it lets them model what functionality is needed and what overall architecture the system should have.

The MDA approach gives the facility to understand complex and real-world systems while providing an abstraction of the physical system as shown in Figure 3-1 [95]. This abstract view of the system is represented through the OMG's modeling standards including the Unified Modeling Language (UML) [96], Meta-Object Facility (MOF) [97], Common Warehouse Metamodel (CWM) [98] ,and XML Metadata Interchange (XMI) [99] which facilitates automatic generation of an XML-based document for a model according to its MOF definition.

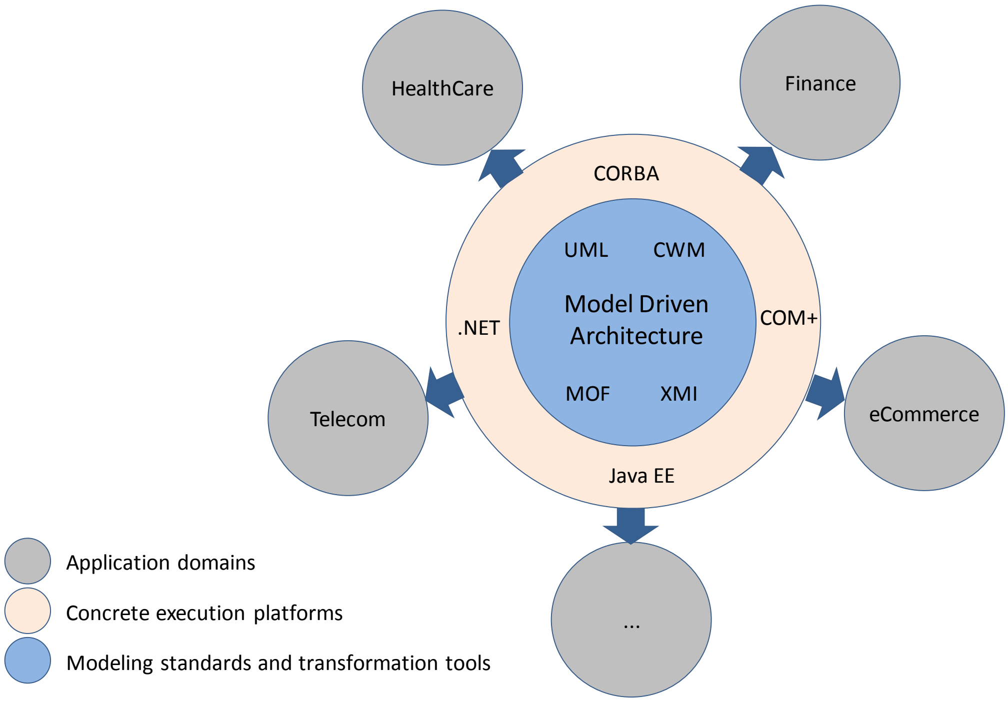


Figure 3-1 The Model Driven Architecture [95].

Following sections describe the MDA's models and modeling standards.

3.1.1 MDA Models

A model is a formal specification of the function, structure and/or behavior of an application or system [100]. Models are an important means for specifying large-scale solutions, and must be expressed by means of well-defined notations [101]. The MDA focuses on building systems with various levels of modeling abstractions. MDA specifies three level of modeling abstractions: Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM) (see Figure 3-2).

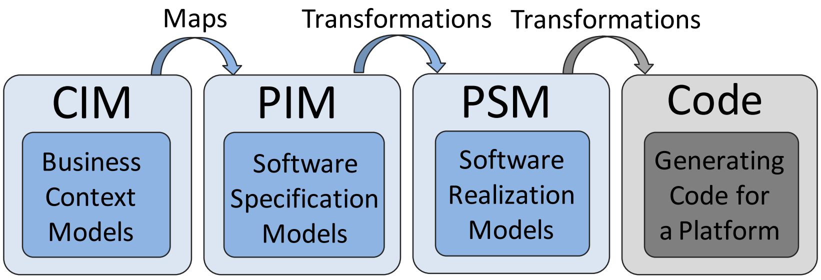


Figure 3-2 Model Driven Architectures levels.

3.1.1.1 ***Computation Independent Model (CIM)***

The Computational Independent Model (CIM) represents what the business actually does or wants to do in future, but hides all information technology related specifications to remain independent of how that system will be implemented. CIM is independent from the use of the system as a computer system, and excludes any implementation details [101]. In other words, this model could be viewed as a contractual element that acts as a reference to check if client requirements have been correctly fulfilled.

As shown in Figure 3-3, the CIM model describes the application's business functionality and behavior through use case and activity diagrams and the actors that interacts with the application. The CIM model does not include information about the final application neither about the programming or platform technologies used to implement this latter. The CIM plays an important role in bridging the gap which typically exists between these domain experts and the information technologists responsible for implementing the system.

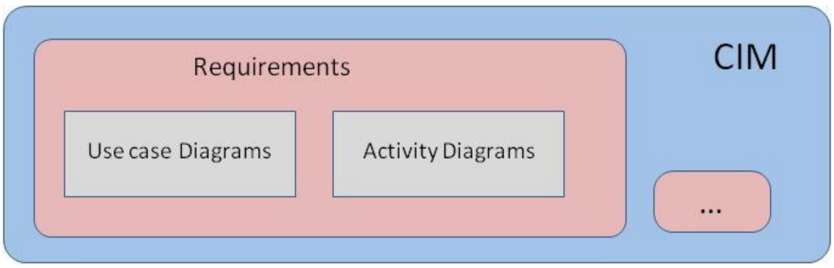


Figure 3-3 Computation Independent Model (CIM) [95].

3.1.1.2 ***Platform Independent Model (PIM)***

Ideally, software application design should be appropriate for all type of execution platforms (different operating systems, hardware, network protocols, programming languages, etc.) To achieve this Platform Independent Model (PIM) has been defined which provides a formal definition of the functionality of software system without addressing any specific operating platform.

A platform-independent modeling language, such as Unified Modeling Language (UML), is used to design PIM model. The PIM model defines data, dependencies and architectural realizations (Figure 3-4). The model elements should provide enough information to make accordant code generation possible in next step.

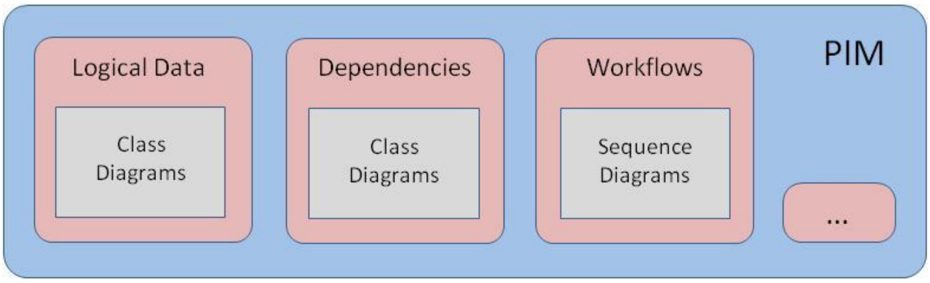


Figure 3-4 The purposes of the PIM model: realizing logical data, establishing dependencies and defining workflows processes [95].

3.1.1.3 *Platform Specific Model (PSM)*

Based on platform independent model, Platform specific model (PSM) provides the details to specify how the system uses a particular type of platform. In other words, PSM intends to ease generating corresponding code from the PIM that fits the operating platform. Figure 3-5 shows some of PSM code models, like interface code, class or schema models [95].

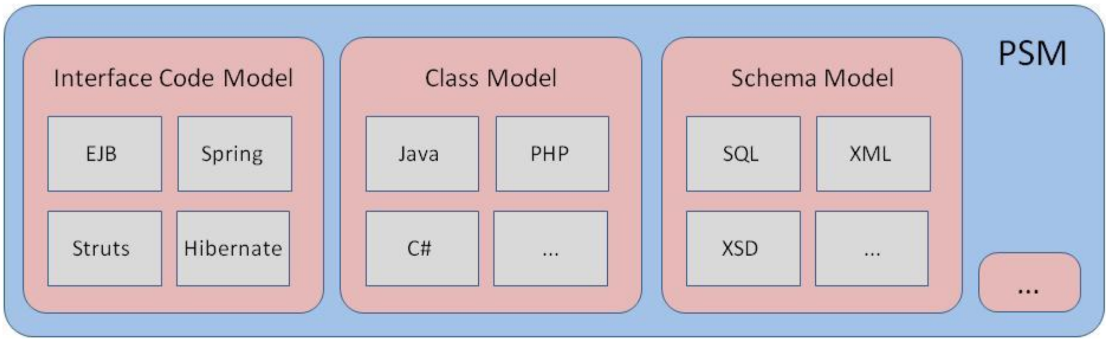


Figure 3-5 The PSM model describes platform and language specific elements.

As mentioned earlier, the PIM describes the system independent of XML, Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), Universal Description Discovery and Integration (UDDI), Java, and other implementation technologies. The model-to-model and model-to-code transformations process would be accomplished using transformation tools that generate XML, WSDL, SOAP, UDDI, and the technology-specific artifacts and finally the implementation code from the design input [102].

3.1.2 Models Transformation

Transformation techniques play a key role in making Model Driven approach successful. The process of automatic generation of one model – the source models – to another model – the target model – from the same system is called "Model Transformation" [32]. A model can be transformed to various models which are functionally same but with different non-functional properties [103]. For instance, one model uses fewer resources and the second is time-efficient. A software developer should be able to specify model transformations that generate a model with desired specifications. Figure 3-6 illustrates the general MDA Structure purposed by Jilani and colleagues in [104], which CIM is mapped on PIM. Then PIM transform to PSM. Finally code is generates from PSM.

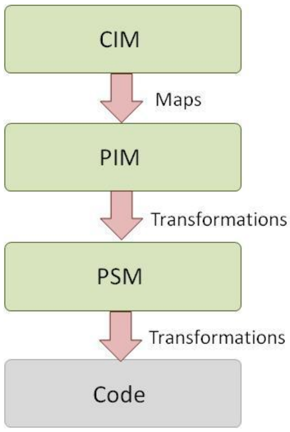


Figure 3-6 Major Structure of MDA [104].

Following sections discuss more about MDA Transformation.

3.1.2.1 *MDA Transformations Classifications*

Transformations can be categorized based on the type of source and destination they operate on. At top level, model transformation approaches can be identified as *model-to-code* transformations or *model-to-model* transformations. Jimenez in [105] classified MDA transformations more detailed:

- **Code to Code:** Here the source and target are textual artifacts. Extensible Stylesheet Language Transformations (XSLT) is an example of code-to-code transformation. XSLT technology transforms XML documents into other XML documents [106].
- **Model to Code:** This kind of transformation can produce source code from models. Converting PSM to code corresponds to the model-to-code transformation. For example the Eclipse Modeling Framework (EMF) generates code based on Java Emitter Templates (JET) [107][108]. Model-to-code transformation can generate textual representations of

models, like the ones specified by the XML Metadata Interchange (XMI) [109] and Human-Usable Textual Notation (HUTN) standards [110]. Following techniques are the main model to code transformation techniques [111]:

1) *VB: Visitor Based Approach:* VB provides a visitor process to traverse the internal representation of a source model and generate code for each model element [111]. Jamda tool is an example of this approach that provides a structure and building blocks and based on the needs of projects generates java code [112].
2) *TB: Template Based Approach*: This approach includes template to generate code. A template is made of rules mapped on source model. Most of available MDA tools provide template-based model-to-code generation, such as JET [107][108], AndroMDA [113], OptimalJ [114] etc.

- **Code to Model:** Code to model transformations generates models from textual representations. A HUTN parser is an example of code-to-model transformation.

- **Model to Model:** Model-to-model transformation automates the refinement process between models. This approach can be categorized into CIM to CIM, CIM to PIM, PIM to PIM, and PIM to PSM. Since transformation between PIMs or CIMs occurs at the same abstraction levels, it is also considered as a *horizontal transformation* [115]. The transformation between PIM and PSM is also called a *vertical transformation as* it moves from a high/low level abstraction to a low/high level [115].

Czarnecki and Helsen classified the model-to-model transformation approaches to following categories [111][116]:

1) ***Direct Manipulation Approaches:*** In this approach, from the beginning, implementation of transformation rules, tracing, scheduling, and other facilities should be mainly in a programming language.
2) **Operational approach.** Operational (or imperative) is similar to direct manipulation approach; however it provides more support for model transformation.
3) **Relational approach.** The main concept of relational approach (as a declarative approach) is mathematical relations. It uses non-executable constraints to determine the relations between source and target model.
4) **Graph Transformation Based Approaches**: Graph is an appropriate tool to represent models, transformations of visual models can be naturally specified by graph transformations. This approach operates on typed, attributed, labeled graphs particularly designed to represent UML-like models.

5) **Structure Driven Approaches:** First, this approach makes the hierarchical structure of the target model, and then it sets the attributes and references in the target. The framework specifies the scheduling and application strategy; users deal with providing the transformation rules.
6) **Hybrid approach.** It combines some techniques from the previous categories
7) **The CWM transformation framework**
8) **Transformation implemented using XSLT**

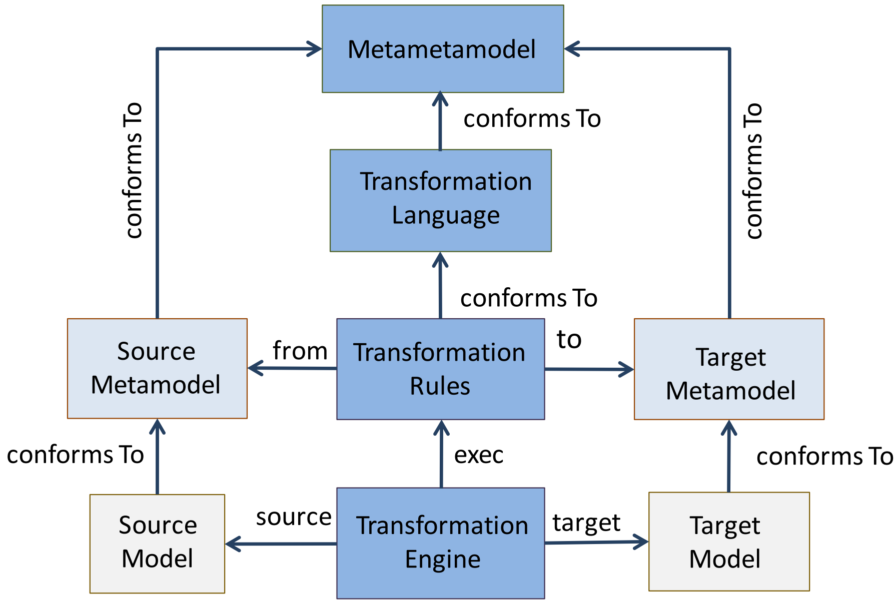


Figure 3-7 Model transformation pattern [103].

3.1.2.2 *Model Transformation Languages*

The conversion from source model to target model is done by the transformation rules. Transformation rules are written using transformation language [117]. There are several Model Transformation Languages to specify model transformations. A transformation language is determined at metamodel level and identifies in what way the specific input metamodel elements are transformed to the output metamodel [103]. Figure 3-7 shows the basic model transformation pattern that a Model Transformation Language is applied at model level to convert source model elements to target model elements. Source model and target model represent the same data with two different ways.

Various transformation languages and tool suites have been developed, although most of them are at experimental stage yet to be applied to industrial practice. Following list shows available transformation languages:

- **QVT** (Query/View/Transformation)**:** QVT is defined by the OMG to describe the requirements of a standard language for the specification of model transformation [118]. QVT standard defines three sublanguages. These transformation sublanguages operate on models in term of MOF 2.0 metamodel. The QVT specification is a mix of declarative/imperative language. The declarative part creates the framework for the dynamic semantics of the imperative part. Figure 3-8 shows the relationships between QVT metamodels. Relations and Core are parts of declarative architecture. Relations includes a user-friendly metamodel and language. Core also includes a more specific metamodel and language. Operational Mappings and Black Box are the imperative approaches

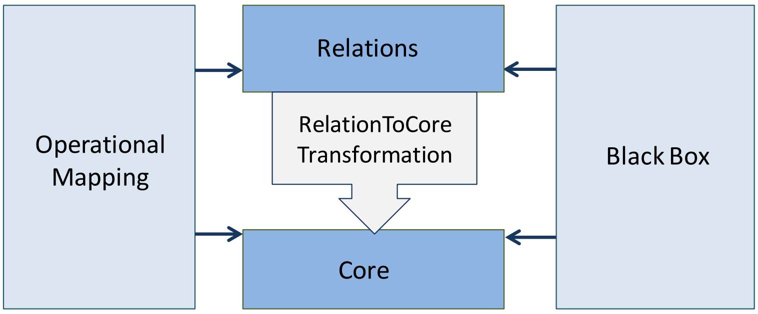


Figure 3-8 Relationships between QVT metamodels [109][119].

- **ATL** (ATLAS Transformation Language)**:** ALT [120] is has a hybrid declarative/imperative nature too. The declarative part correlates with simple model transformations, whereas the imperative part supports high level complex transformation. Changes are not admitted during the execution of a transformation. Input models may be navigated but output models cannot be navigated.

- **Beanbag:** An Operation-based Heterogeneous Synchronization Language [121]

- **ETL** (Epsilon Transformation Language) [122]: ETL is a hybrid, rule-based model-to-model transformation language. This language provides all the standard features of a transformation language and provides enhanced flexibility as it can transform many input to many output models, and can query/navigate/modify both source and target models.

- **GReAT** (Graph Rewrite And Transformation language)**:** GReAT [123][124][125] is a metamodel-based graph transformation language. It designed to deals with the high-level complexity model transformation programs. It uses sequenced graph rewriting rules to transform source model to target model. GReAT includes three parts:

    1. *Pattern specification language* to describe complex patterns conformed to select elements in the current graph.

2. *Graph transformation language* as a rewriting language to use the pattern language. In this part, source model, target model and temporary objects are handled as a single graph based on a unified metamodel. The temporary objects are deleted at the end of the transformation.

3. *Control flow language as* a high-level control flow language to control produced application and to provide the user the ability to manage the complexity of the transformations.

- **JTL** (Janus Transformation Language) [126][127]**:** JTL is a bidirectional model transformation language that support non-bijective transformations and change propagation. This language is designed to propagate changes occurring in a model to one or more related models according to the certain transformation without considering the transformation direction. JTL is embedded in a framework available on the Eclipse platform which aims to facilitate the use of the approach, especially in the definition of model transformations.
- **KerMeta** [128][129][130][131][132]: Kermeta is a modeling and programming language for metamodel engineering which allows describing both the structure and the behavior of models.
- **Lx family** [133]: A set of low-level transformation languages.
- **Model-to-Model** (M2M) [134]: The Eclipse implementation of the OMG QVT standard
- **Mof2T** (MOF Model to Text Transformation Language) [135]**:** Mof2Text is a OMG specification for a model transformation language that aligned with UML, MOF, and OCL. It specifically is designed to support transformations which transform a model into various text artifacts such as code, deployment specifications, reports, documents, etc.
- **MOLA** (MOdel transformation Language) [136][137][138][139][140]: The MOLA is designed through combination of traditional structured programming languages and pattern-based model transformation rules, both in a graphical form.
- **MT:** A transformation language developed at King's College, London [141].
- **SiTra:** (Simple Transformer): A transformation approach using a standard programming language, e.g. Java, C# [142].
- **Stratego/XT:** A language and toolset to develop transformation systems [143][144].
- **Tefkat:** A declarative model transformation language for Model-Driven Development (MDD) and data transformation. [145][105][146]

- **Tom:** A language extension for programming various transformations on tree structures and XML documents [147]
- **UML-RSDS:** Model Transformation and Model-Driven Development Tools [148][149]
- **VIATRA (VIsual Automated model TRAnsformations):** VIATRA [150][151] is an Eclipse-based model transformation framework to support the entire life-cycle for the specification, design, execution, validation and maintenance of transformations within and between various modeling languages and domains.

3.1.2.3 *Metamodel*

A model transformation produces target models from source models. This process requires specific transformation techniques called metamodels. Metamodel defines the abstract syntax of models and interrelationships between model elements. Metamodel specifies the structure of an application to determine models and the model as an instance of metamodel contains specific details. For instance, a metamodel can define the models and relationships of model elements using classes, objects and methods in UML. Then, according to the specific platform, the application derived from model runs in real world.

In this regard, OMG has introduced a 4-layer architecture called the MOF metamodeling stack as shown in Figure 3-9 MOF is a Domain Specific Language (DSL) to specify metamodels. M0 describes the real system. Level M1 is a model to represent the real system which includes the details of application. Level M2 is the metamodel to define boundaries of the model in level M1. Metametammodels are used to define the concept of metamodels. The metamodel in level M2 conforms to the metametamodel in level M3.

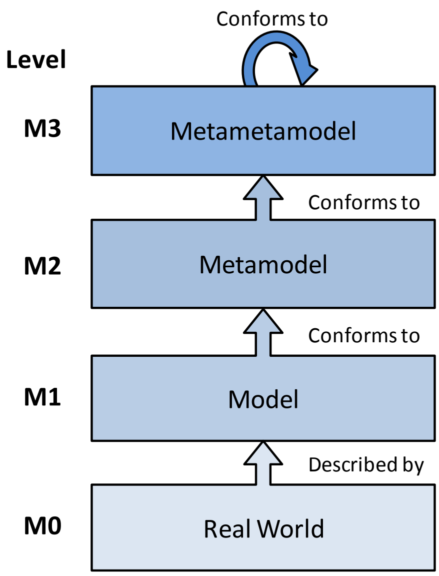


Figure 3-9 The four layer meta-modeling architecture [152].

Figure 3-10 shows two examples for MOF metamodeling 4-layer stack. As shown in Figure 3-10 (a) the real system can be modeled by UML and MOF can define the UML metamodel. Software Process Engineering Metamodel (SPEM) or Common Warehouse Metamodel (CWM) also can be used in level M2 and comply with MOF. Figure 3-10 (b) shows C language is used to implement in EBNF (Extended Backus–Naur Form). The EBNF confines software developer to use the syntax that is defined for C.

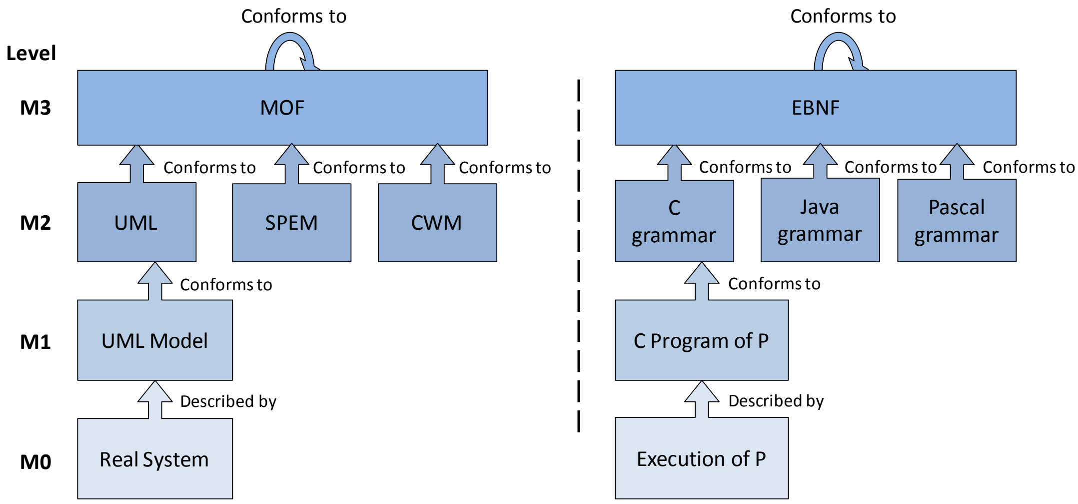


Figure 3-10 Examples for MOF metamodeling stack [153].

3.1.2.4 ***Model transformation taxonomy***

T. Mens and colleagues [154] introduced the taxonomy of model transformation to help developers to choose the most appropriate model transformation approach. Basically there are several essential questions to be answered for addressing a specific problem:

1) **Specify the input and output artifacts of transformation process.**
2) **Specify endogenous versus exogenous transformations.** (Endogenous transformation is a transformation between models created in the same language and exogenous one is a transformation between models created using different languages (e.g., UML for design models, and programming languages for source code models). The syntax and semantics of the modeling language is expressed by a metamodel (e.g., the UML metamodel).)
3) **Specify horizontal versus vertical transformations**.
4) **Specify the possible mechanisms for model transformation:** The major difference between transformation mechanisms is whether they rely on a declarative or an operational (or imperative) approach. The declarative approach seems the most favourable one for transformation. It is formally founded, it is bidirectionality, and it proposes a simpler semantic model to understand and specify model transformations.

Whereas an operational approach is better when transformations required to incrementally update a model.

3.1.3 Modeling Standards

OMG introduced four core interoperability standards to support MDA: UML, CWM, MOF and XMI. The main goal of these slanders is utilizing metamodels to specify models for providing a common understanding between all parts of system and simplify communication among the models [155].

3.1.3.1 *Unified Modeling Language (UML)*

The Unified Modeling Language (UML) [96] is a visual modeling language used for analysis, design, implementation, modifying, and documenting software-intensive systems. UML includes a set of graphic notation techniques to represent models of applications. These graphical representations include Activity diagram, Class diagram, Component diagram, Composite structure diagram, Deployment diagram, Object diagram, Package diagram, Profile diagram, and Use Case diagram to describe and model the software system.

3.1.3.2 *Common Warehouse Metamodel (CWM)*

Common Warehouse Metamodel (CWM) [98] defines a metamodel to specify metadata in data mining and warehousing. Also it represents interfaces to enable interchange warehouse and business intelligence metadata between tools, platforms and metadata of warehouse in distributed heterogeneous environments. CWM is expressed in the UML and it is compatible with XMI, and MOF standards.

3.1.3.3 *Meta Object Facility (MOF)*

OMG introduced the Meta-Object Facility (MOF) [152] based on UML class modeling capabilities. MOF provides a common, abstract language to model metamodels. It is a common model to specify CWM and UML metamodels. It also makes different metamodels from various domains to be used in an interoperable manner. If models are MOF-based, they can be exported from one software system and imported to another one. These models also can be converted into dissimilar formats and transformed and used to produce the code.

3.1.3.4 *XML Metadata Interchange (XMI)*

The XML Metadata Interchange (XMI) [99] is another standard from OMG to support requirements of MDA. It enables interchange metadata information between UML and MOF based models and metamodels via Extensible Markup Language (XML).

3.1.4 Model Driven Interoperability (MDI)

The concept of "interoperability" is specified in previous chapters. The MDD methodology provides an approach to solve interoperability issues. This technique is called Model Driven Interoperability (MDI) which is based on MDA approach. MDI provides an abstract and technical support to create interoperable enterprises using ontologies and semantic annotations. In 2004, MDI was introduced in two research projects by the European Commission:

INTEROP NoE (Interoperability Research for Networked Enterprises Applications and Software Network of Excellence, FP6-IST 508011) [156].

ATHENA IP (Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications Integrated Project) (FP6-IST-507849)[157][158]

Both of the projects defined a methodological framework and the application of MDI on concrete case.

The **MDI Reference Model** shown in Figure 3-11 introduces different conceptual levels and possible model transformations between them [159]. These abstract levels are based on three levels of MDA approach (CIM, PIM, and PSM) in order to reducing the gap between enterprise models and code level during model transformations. Furthermore, the CIM level is divided into two sub-levels with the purpose of reducing the gap between the CIM and PIM levels.

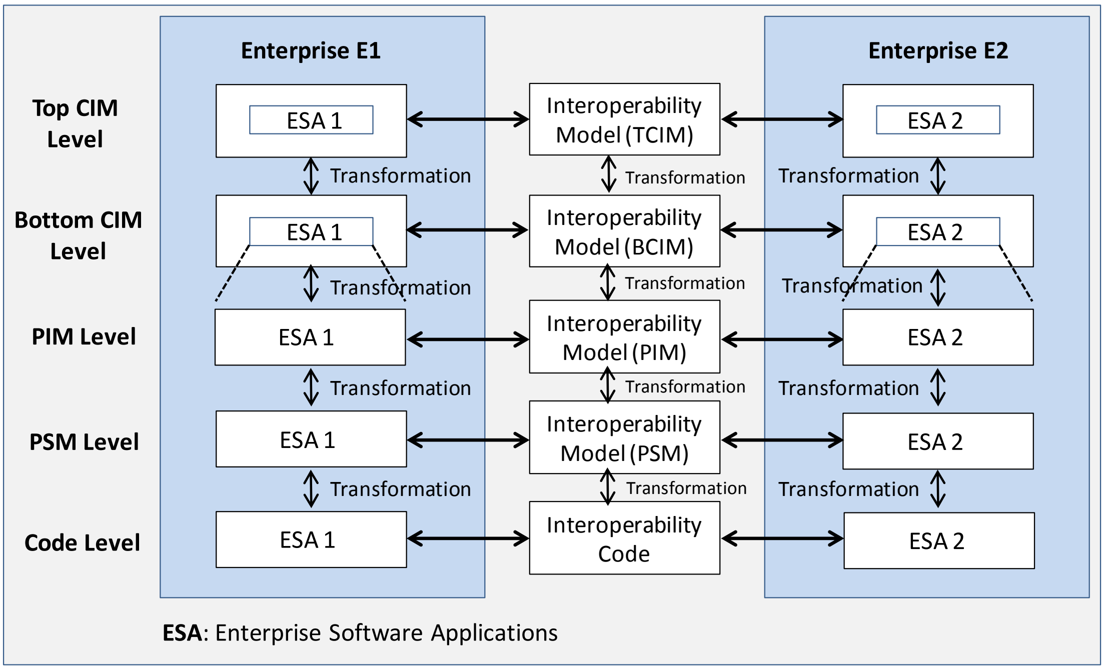


Figure 3-11 Reference Model for MDI [160].

As shown in Figure 3-12 the **ATHENA MDI Framework** [161] describes four categories of system aspects for interoperability reference architecture. This framework defines four levels:

- Interoperability at the enterprise/business level: Interoperability in this level should support cooperation between heterogeneous organizations and enterprises with different working rules, cultures and commercial approaches.
- Interoperability at the processes level: Interoperability in this level provides the ability to get various processes work together.
- Interoperability at the services level: Interoperability in this level is related to identifying, creating and executing independent applications and exchange services.
- Interoperability at the information/data level: Interoperability in this level is concerned with organizing, interchanging and processing of different documents, messages or structures by different collaborating entities.

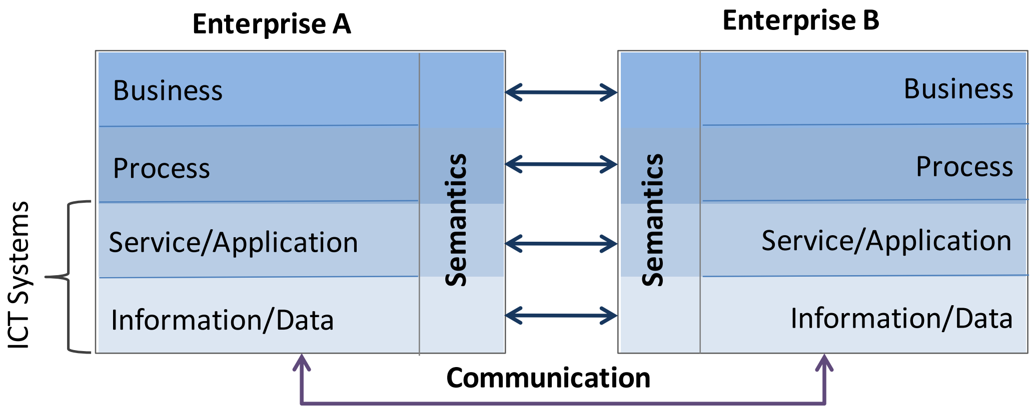


Figure 3-12 Interoperability on all layers of an enterprise [161].

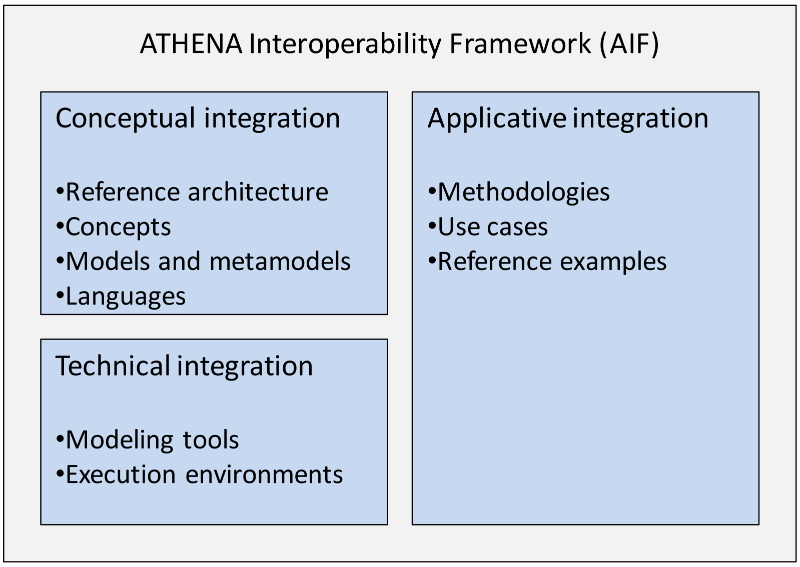


Figure 3-13 ATHENA Interoperability Framework [162].

The **MDI Framework** [159] from ATHENA (shown in Figure 3-13) demonstrates how MDD should be applied to achieve interoperability. The framework includes three main integration areas:

1. *Conceptual integration:* Concepts, metamodels, languages and model relationships are addressed in conceptual integration. Conceptual integration supports organizing various aspects of software model interoperability.
2. *Technical integration:* It focuses on the software development and execution environments. Software model development tools and software model execution platform are provided by technical integration.
3. *Applicative integration:* Methodologies, standards and domain models are addressed in applicative integration. Fundamentally, the guidelines, principles and patterns used to achieve interoperability are provided by applicative integration.

3.1.5 Advantages of MDA approach

The MDA approach promises a number of benefits including automatic code generation, platform independence architecture, improved portability, increased productivity, wider interoperability, and reusability. In following sections the most important ones are described

3.1.5.1 ***Interoperability***

The interoperability between applications and services is inherent to the system design using MDA approach. MDA defines services, facilities, and applications through platform-independent model (PIM). Transforming the PIM to the PSM and then generating the code is based on the links provided between models. These links are specified by the metamodels' mappings which allow platform specific and independent implementations to interoperate. Interoperability between two applications is provided by the mappings via the relevant metamodels of models.

3.1.5.2 ***Portability***

MDA approach enhanced portability of applications due to platform independent models (PIMs). Using MDA, software system is developed through models independently from platform, and then Platform Specific Model is produced for different platforms.

3.2 The Service Oriented Architecture (SOA) Approach

Service Oriented Architecture (SOA) is a new architectural style to develop applications through services.

3.2.1 Definition

A Service-Oriented architecture is a collection of independent services which communicate with each other. The communication can include a simple data passing or two or more services coordinating the same activity. The Figure 3-14 depicts a basic Service-Oriented architecture that a service consumer is sending a message to a service provider to request a service and the service provider replies through a response message. The connection for exchanging request and subsequent response messages are specified in an understandable way to both the service consumer and provider.

SOA is a new paradigm for solution architects to facilitate developing new value-added solutions by incorporating different solution artifacts such as business processes, services, packaged applications, and manageable attributes all over their lifecycle [163]. Organization for the Advancement of Structured Information Standards (OASIS) presented following definition for SOA [164]:

"A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations".

Additionally, International Business Machines Corporation or IBM collected the following definitions of SOA for different purposes (shown in Table 3-1) [165] .

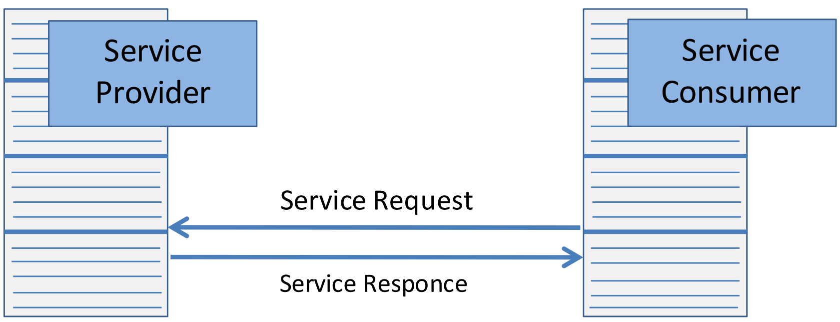


Figure 3-14 A basic Service-Oriented architecture [166].

Table 3-1 Definitions of SOA [165].

From the point of view of:	SOA is
Business executive and business analyst	A set of services that constitutes IT assets and can be used for developing solutions and representing them to customers and partners
Enterprise architect	A set of architectural principles and patterns that include the general characteristics of the solutions: encapsulation, modularity, loose coupling, reusability, composability, and etc.
Project manager	A development method that addresses massive parallel development
Tester or quality assurance engineer	An approach to modularize, and consequently simplify, overall system testing.
Software developer	A programming model integrated with standards, tools, and technologies, such as Web services

3.2.2 SOA Entities

SOA defines an interaction model between three main functional units, shown in Figure 3-15, in which the service consumer identifies adequate service via communication with the service provider through searching registry [167]. Practically, SOA contains six entities in its conceptual model, described as follow [167]:

- *Service Consumer:* It is the entity that requests a service to execute a demanded function. If consumer knows the location of the service, it can communicate directly with the service provider, otherwise, it can detect the service location through the registry.
- *Service Provider:* It is an addressable entity of network that receives and executes the requests of consumers. It can provide the determined service description and the implement the service.
- *Service Registry:* It is a directory for available services which can be exploited through network. Service Registry should be able to publish and save service descriptions from providers and deliver the descriptions to the interested service consumers.
- *Service Contract:* It is a description that explicitly defines how the service consumer and provider should communicate. It includes information about the format of request-response message, the conditions in which the service should be executed, and quality aspects of the service.

- *Service Proxy:* It is an optional entity that facilitates the interaction between service provider and consumer through providing an API created in the local language of the consumer.

- *Service Lease:* It specifies and maintains the relationships between service consumer and provider. It defines the executive well-defined binding timeframes for the services that is managed by registry. It provides loose coupling between service provider and consumer as well as maintenance of state information for the service.

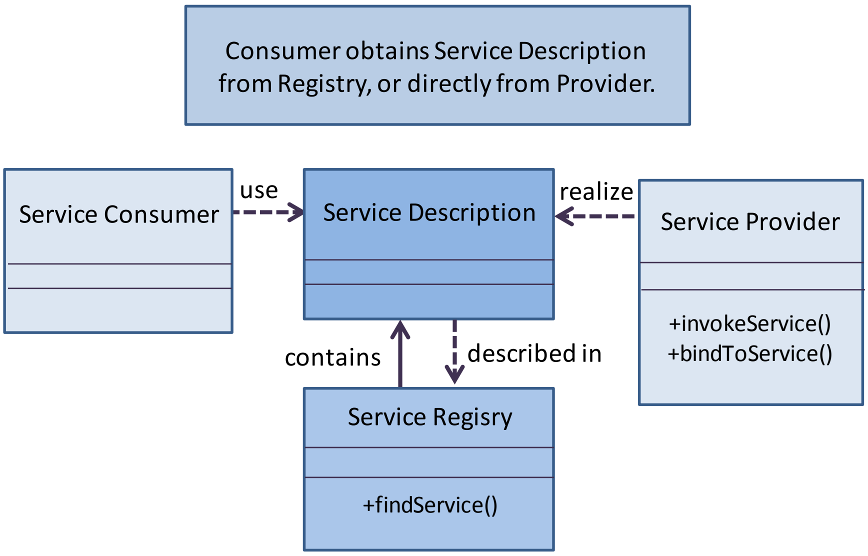


Figure 3-15 Service Oriented Architecture Conceptual Model [167].

3.2.3 An architectural template for a SOA

Arsanjani presented a Layered Architecture for SOA in which services are layered on top of components that are responsible to provide certain functionalities and maintain quality aspects of the service [168] (shown in Figure 3-16). Each layer has specific architectural characteristics described below:

L1) *Operational Systems Layer:* This layer includes the existing applications such as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) packaged applications, Object Oriented systems, and Business Intelligence applications. These applications can provide the background for the services with proprietary structures, databases and other system resource access.

L2) *Enterprise Components Layer:* This layer consists of the components specialized to provide certain functions and requirements for the services. Enterprise Components exploits the functionality of interfaces to specify service realization at runtime. This layer uses container-based technologies such as application servers to implement the components, workload management, high-availability, and load balancing.

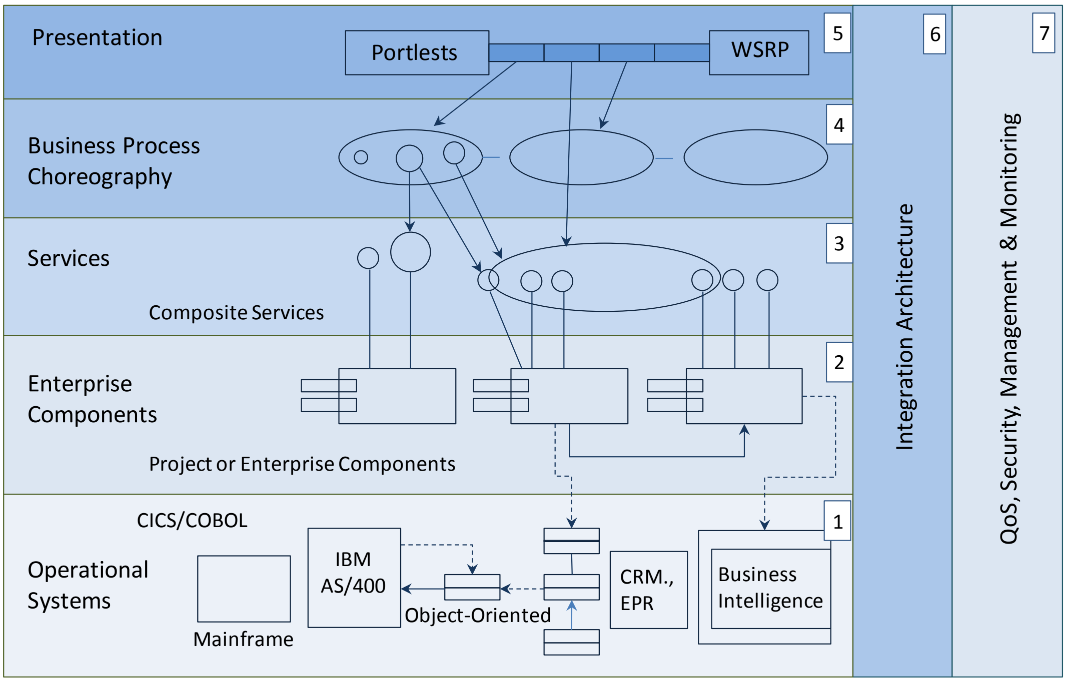


Figure 3-16 The layers of a SOA [168].

L3) *Services Layer:* This layer consists of the available services defined within the SOA. The services are functions that can be detected and invoked across the network using well-defined interfaces. In this layer, the interfaces are exported out as service descriptions where they are exposed for use.

L4) *Business Process Composition Layer:* Design and compositions of services presented in third layer are defined in this layer. Services are combined into a flow through orchestration and operate as a single application.

L5) *Access or Presentation Layer:* This layer provides user interfaces for services and composite applications. This layer is not a direct concern for SOA.

L6) *Integration Layer:* Integration layer, that is often called the Enterprise Service Bus (ESB), is a key enabler for SOA to provide the integration of services through the reliable set of capabilities, such as intelligent routing, protocol mediation, and other transformation mechanisms. Web Services Description Language (WSDL) defines a binding to indicate the location where a service is provided. An ESB, on the other hand, provides a location-independent mechanism for integration.

L7) *Quality of Service (QoS) Layer:* This layer provides the capabilities required to monitor, manage, and maintain QoS such as security, performance, availability, loose coupling, and increased virtualization.

3.2.4 Web Services and SOA

A "service" is defined as a function that is well-defined, self-contained, and independent of the state of other services [166] and a "Web Service" is described as a software system designed to provide interoperable machine-to-machine interaction over a network [169]. A web service includes an interface specified in a machine process-able format. Additionally, a service intended to be an independent building block to represent an application environment. A service also involves a number of unique characteristics to be able to participate as part of a Service-Oriented architecture [170]. One of these characteristics is complete isolation from other services. In other words, each service is individually in charge of its own domain that is restricted to a particular business function (or a group of related functions). This design approach creates independent units of business functionality loosely bound together by a common agreement for a standard interaction framework. Additionally, the programming logic is encapsulated in a way that it is not required to comply with any platform or technology set.

Other systems are able to communicate with the Web Service according to its description using Simple Object Access Protocol (SOAP) messages, often conveyed through HTTP with an XML sequence connected to the other web-related standards. Figure 3-17 illustrates the roles of various components in Web Services Architecture. Architectural model of Web Services is based on a layered family of technologies. Each layer is interrelated with all the others, and supports a level of abstraction and functionality to develop Web Service based applications.

The components of Web Services Architecture, shown in Figure 3-17, are the basic and core standards for Web Services and Service-Oriented architecture. The Web Service Definition Language (WSDL) and Simple Object Access Protocol (SOAP) are World Wide Web Consortium (W3C) standards. Universal Description Discovery and Integration (UDDI) is an Organization for the Advancement of Structured Information Standards (OASIS) specifies the interoperability versions for UDDI, WSDL, and SOAP. It also defines the interoperability requirements in Basic Profiles. More specifically, the interoperability profile provides details and tests for interoperability using the following specific standards [171]:

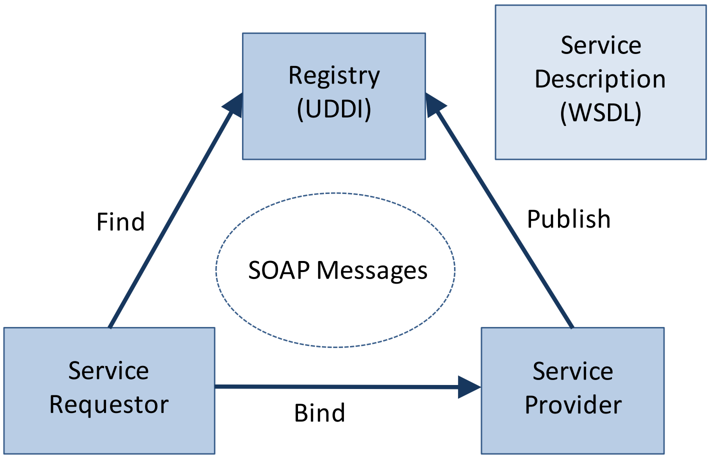


Figure 3-17 Web Services Architecture [167].

- SOAP including material related to: Namespaces in XML
- WSDL including material related to XML Schema Part 1: Structures and Part 2: Datatypes
- UDDI which includes support for UDDI interfaces

Following sections will describe the core standards for SOA.

3.2.4.1 ***XML***

Extensible Markup Language (XML) is a textual format to specify a set of encoding rules which is both human-readable and machine-readable. The main design goals of XML are simplicity, generality, and usability over the Internet. It is widely used to show arbitrary data structures, for example in web services. XML Web service is the most widely accepted and successful type of service. XML Web Services exploits XML messages that follow the SOAP standard. This type of service has two fundamental requirements [170]:

- It communicates via Internet protocols (often through HTTP).
- It sends and receives data formatted as XML documents.

3.2.4.2 ***WSDL***

Web Service Definition Language (WSDL) is an XML format to describe Web Services. Figure 3-18 illustrates the usage of WSDL which a service provider and a service consumer interact through WSDL based messages. Different steps to provide and consume a service are [166]:

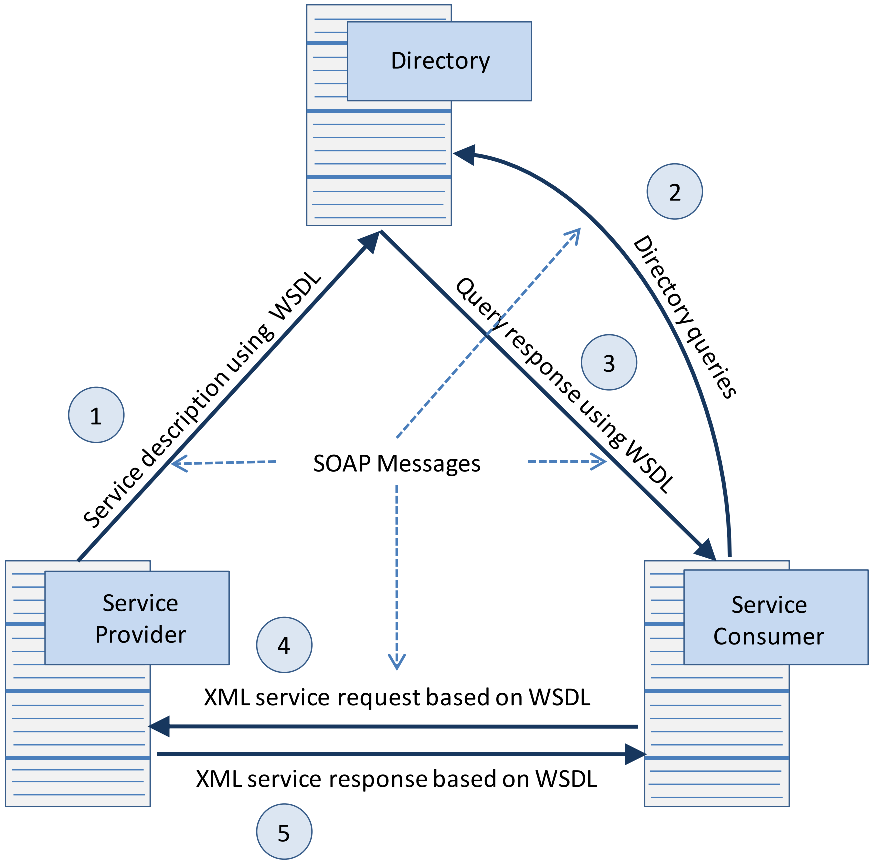


Figure 3-18 web services basics [166].

1. A service provider specifies its service using WSDL which is exported to a directory of services. The directory could use Universal Description, Discovery, and Integration (UDDI) as well as other forms of directories.
2. A service consumer sends out one or more queries to the service directory to locate a service and realize how to communication with it.
3. Part of the WSDL based message specified by the service provider is sent to the service consumer to inform the service consumer about the requests and responses from the service provider.
4. The service consumer uses the WSDL to pass a request to the service provider.
5. The service provider provides the adequate response to the service consumer.

Client of Web Services will use the WSDL based service descriptions to produce client proxies to access the service of the service provider. The service provider specifies the operations (i.e., what is the service), the binding (i.e., how to access the service), and the endpoints (i.e., where to access the service) all in the WSDL based document. WSDL enables the messages and the operations to access services independent of technical details and implementation. Table 3-2 shows the main elements of WSDL [171]. Moreover, Figure 3-19 depicts the relationships between essential elements of WSDL [171].

Table 3-2 Main Elements of WSDL [171].

WSDL Elements	Function
Types	Define the types of input and output parameter of operations; used in messages
Message	Specify the input and output parameters of operations. Messages have names and can have multiple parts.
PortType	A collection of operations
Operations	Use messages for their input and output parameters. Operations are Web services methods.
Binding	Specify how the operations of portTypes will be invoked.
Port	A specific invocation endpoint, containing the address (the where) of the service.
Service	A collection of endpoints.

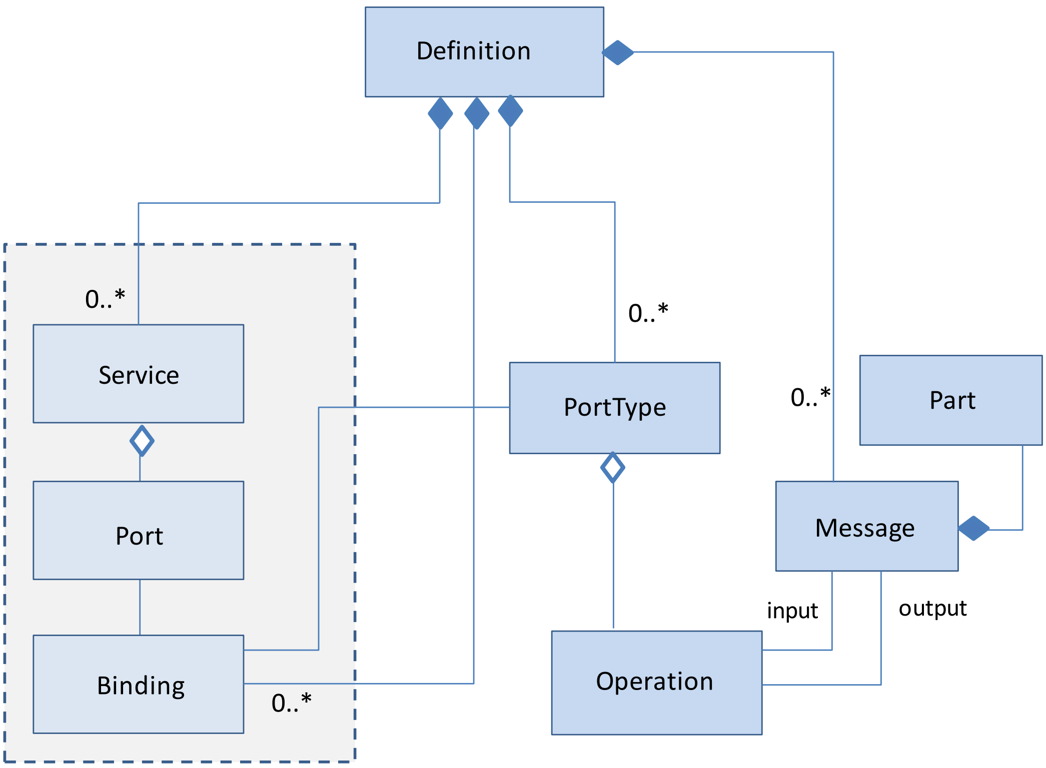


Figure 3-19 WSDL metamodel: relationships between WSDL elements [171].

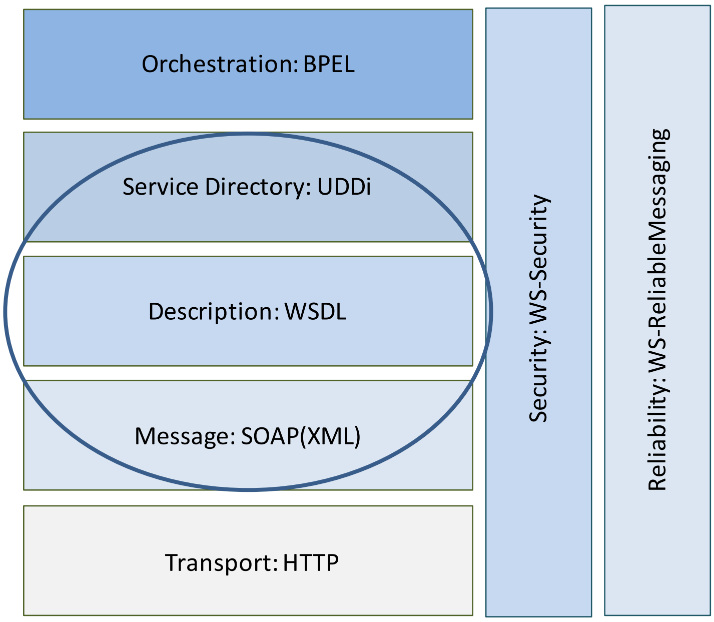


Figure 3-20 UDDI within the Web services stack [171].

3.2.4.3 ***UDDI***

Universal Description Discovery and Integration (UDDI) is a platform-independent, XML-based registry that enables businesses to define and spread their service offerings in UDDI registries, and to discover other businesses and services through those registries. UDDI registries provide the UDDI standard and are typically built on top of relational databases [171]. Figure 3-20 depicts where UDDI fits within the overall stack of Service-Oriented technologies that includes registration and discovery. A service, described using WSDL, has to be published to enable others to discover it from UDDI registries. Service requestors can invoke the services either individually or within business processes. The communication infrastructure is typically via SOAP messages over HTTP protocol.

3.2.4.4 ***SOAP***

Simple Object Access Protocol (SOAP) protocol [172] is an important core standard for Web Services. SOAP protocol is an XML message structure that supports distributed computing through providing an effective mechanism for exchanging the messages and accessing remote objects. Khoshafian [171] illustrates in **Error! Reference source not found.** the main role of SOAP in the triangle of registering, discovering, and request/response interchanges. The discovery and registration of services is also done through SOAP. The UDDI registry can have Web browser-based access; however, its API is through XML messages in SOAP envelopes. As shown in Figure 3-21, SOAP is the layer above Hypertext Transfer Protocol (HTTP) and other Internet transport protocols. A service requestor sends SOAP messages to the provider to request a service. SOAP messages are also sent from provider to requestor for service responds.

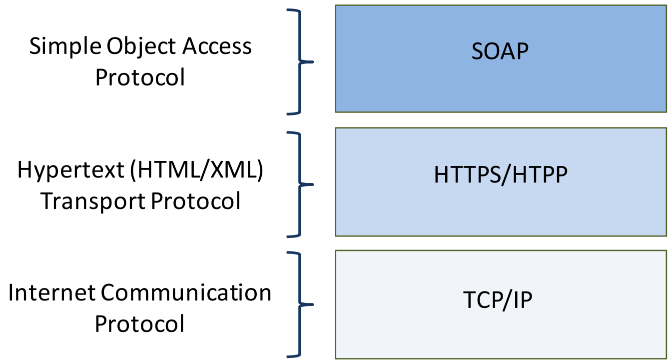


Figure 3-21 SOAP layer [171].

3.2.5 SOA Benefits

Applying SOA architecture to develop enterprise applications enables great flexibility through providing [173]:

- Greater interoperability
- Increased reuse
- More agile business processes
- Improved visibility
- Reduced maintenance costs
- Compliance and governance

It should be mentioned that a fundamental purpose of exploiting Service-Orientation methodology in application development process is establishing *interoperability* naturally and as an expected service design specifications. In other words, SOA architecture and the related standards enable existing applications to interoperate seamlessly with an easier maintenance way than traditional enterprise software solutions. Sharoff explains in "Enterprise Cloud Computing Technology, Architecture, Applications" book [70] that SOA approach provides interoperability through packaging application systems as bundles of published services and it is feasible to evolve their usage as business needs changed.

3.3 Actual practices using MDA, and SOA approaches

The aim of this PhD work is proposing a new framework to provide Intercloud Interoperability. In order to find the appropriate approaches to develop and design the Intercloud Interoperability framework to devise the best approaches for implementation of our framework, more than 300 papers, books, websites, and thesis were studied. Finally, the most relevant articles related to current research approaches towards Intercloud Interoperability, Cloud Computing, and MDA, MDI, and SOA approaches were selected. Following sections summarize existing research work in MDA, MDI and SOA approaches. Also section 3.3.2 details current work on MDA/SOA/MDA-SOA based solutions to develop Cloud Computing.

The aim of this PhD work is proposing a new framework to provide better Intercloud Interoperability. In order to devise the appropriate approaches to develop and design the Intercloud Interoperability framework, more than 300 papers, books, websites, and thesis were studied. As a result, Model Driven Architecture along with Service-Oriented Architecture are chosen as the most appropriate approaches.

Following sections summarize existing research work in MDA, MDI and SOA approaches. Also section 3.3.2 details current work on MDA/SOA/MDA-SOA based solutions to develop Cloud Computing.

3.3.1 MDA, and SOA solution for enterprise interoperability

Recently, SOA and MDA approaches are increasingly exploited to develop different frameworks to solve several problems such as interoperability in enterprises. For instance, Xu and colleagues described in [174] that providing service interoperability is feasible using a model driven paradigm along with service oriented systems.

The interoperability between applications and services is inherent to the system design using MDA approach because MDA supports defining services, facilities, and applications through PIM model. [175] and [176] have explored various dimensions of interoperability by making use of MDA and SOA.

SOA inherits the ability of a service to be invoked by any potential service consumer and are connected using standard, dependency reducing decoupled message based methods. This methodology guarantees that services are coarse-grained reusable components that expose their functionality through a well-defined interface, systems can be built as a composition of services and evolve through the addition of new services. So, SOA methodology supports and promotes interoperable system designs. [177] presents a paradigm of cloud-marketplace ecosystem, making use of SOA to achieve collaborative marketplace architecture for the domain of e-procurement. A key issue for enabling interoperability is to come to an agreement about which services can be provided by whom and which can be consumed by whom in a network of service. Han at al. in [178] discusses how the OMG standards Business Motivation Model (BMM) can support Organizational Interoperability by enabling a community or organization to work together using SOA services at a higher level of abstraction.

Table 3-3 describes briefly the latest articles in the area of MDA and SOA based solutions.

Table 3-3 Current state-of-the-art in MDA-SOA solutions.

Reference	**Author(s)**	**Year**	**Title**	**Area**	**What had been done?**
[179]	Roman et al.	2011	A Model-driven Approach to Interoperability in B2B Data Exchange	MDA	This paper proposed MDA-based data exchange framework for Business-to-Business (B2B) data exchange to enable automation in the end-to-end data exchange process.
[180]	Limyr et al.	2006	Semaphore – A Model-Based Semantic Mapping Framework.	MDA-interoperability	This paper specified state of the arts for applying the MDD approach to the interoperability problem and it proposed a MDA based framework and a tool devised to enhance information interoperability between enterprise applications.
[181]	Saleem et al.	2010	Model driven security frameworks for addressing security problems of Service Oriented Architecture.	MDA-SOA	In this paper, several Model Driven Security Frameworks attempted to solve security issues in developing an application using SOA are studied. The Model driven frameworks comprised Meta Object Facility (MOF), SECTEC Framework, SECTISSIMO Framework and SAP Research by Hasso-Plattner Institute
[182]	Delgado et al.	2010	A Model-driven and Service-oriented framework for the business process improvement.	MDA-SOA	SOA is increasingly exploited to develop business processes. Delgado and colleagues proposed MINERVA framework based on Model Driven Development (MDD) and Service Oriented Computing (SOC) paradigms to augment business processes.
[183]	Zdun et al.	2007	Modeling Process-Driven and Service-Oriented Architectures Using Patterns and Pattern Primitives.	MDA-SOA	This article, suggested a pattern language for process-driven SOAs. The patterns are represented by modeling elements called pattern primitives.

Reference	Author(s)	Year	Title	Area	What had been done?
[184]	Herold et al.	2008	A Seamless Modeling Approach for Service-Oriented Information Systems.	MDA-SOA	To exploit MDD approach in developing an enterprise, it has to be modified corresponding to the requirements of application. This paper represented a MDD method to apply to the business process to implement service-oriented distributed enterprise information systems.
[185]	Touzi et al.	2009	A model-driven approach for collaborative serviceoriented architecture design.	MDA-SOA	This article introduced a collaborative architecture that specifies interoperability at three levels of MDA principles (CIM, PIM, PSM). The proposed MDA methodology bridges the gap between the business collaborative process model and the IT developer level (collaborative SOA model).
[186]	Ricken	2007	Top-Down Modeling Methodology for Model-Driven SOA Construction.	MDA-SOA	In this paper, MDA approach is applied for modeling Business Process as a key to bridge the gap between business analysts and IT developers. The proposed method enables Business Process models to lead to service orientation of business functions.
[187]	Radhakrishnan & Wookey	2004	Model driven architecture enabling service oriented architectures.	MDA-SOA	This paper described the value proposition of applying MDA approach for SOAs through several aspects and instances. The paper clarified five perspectives that MDA approach can add value to services: • Grouping, interlinking and coupling services (brokered services) • Integrating services implemented with multiple underlying technologies • Building value-added data driven services • Delivering context sensitive and profile driven services • Total Business Integration
[188]	Benguria et al.	2007	A platform independent model for service oriented architectures.	MDA-SOA	At present, SOA technologies depend on the specific platforms where services are implemented. This paper described the challenge of developing a SOA modelling language able to distinguish the logical solution from the technical implementation.

Reference	Author(s)	Year	Title	Area	What had been done?
[189]	Kim	2008	Modeling of Distributed Systems with SOA & MDA.	MDA-SOA	This paper proposed a new approach to model and design heterogeneous distributed systems based on MDA and SOA. The approach attempted to reduce the essential human interaction for converting a PIM into a PSM and a PSM into code for a SOA. Integrating a service-oriented modeling architecture with MDA has two main advantages: The clear organization of models and information based on the stereotypes derived from the service-oriented architecture and Select Perspective as development process. The productivity, quality and impact analysis benefits of the use of MDA with its emphasis on automation, transformation and synchronization.
[190]	Rahmani et al.	2006	An MDA-Based Modeling and Design of Service Oriented Architecture.	MDA-SOA	Similar to [189], this paper described research which applies model driven approach for SOA. In this paper SOA-based PIM transformed to PSM for Web Services as a specific platform.
[191]	Bispo et al.	2010	Applying a model-driven process for a collaborative service-oriented architecture.	MDA-SOA	This paper represented the research which applies a MDA process to advance specific middleware services in Web-based Groupware Service-Oriented Architecture (WGWSOA) infrastructure. The deployment of MDA enabled WGWSOA to support heterogeneity and interoperability.
[192]	Rafe et al.	2009	Using MDA for Developing SOA-Based Applications.	MDA-SOA	MDA transformation among different abstraction layers is still an open challenge. This paper described the process of deploying driven modeling and aspect oriented programming to ease generating PSM from PIM which experimented for SOA as a target model.
[193]	Utomo	2011	Implementation of MDA Method into SOA Environment for Enterprise Integration.	MDA-SOA	This paper using a case study of e Shop application claimed SOA-MDA approach is a favored method for analysis, design and implement of enterprise integration.

Reference	Author(s)	Year	Title	Area	What had been done?
[194]	Hahn et al.	2010	Enhancing the Interoperability between Multiagent Systems and Service-Oriented Architectures through a Model-Driven Approach.	MDA-SOA	This paper presents a MDA based framework for the integration of SOA and multi-agent systems (MAS). Since there is not a standardized metamodel for characterizing services via MDA-based techniques, this paper utilized a model transformation from SoaML, as a metamodel for SOA, to PIM4AGENTS, as a platform independent metamodel for MAS, is deployed for integration.
[195]	Ali et al.	2010	Model driven support for the Service Oriented Architecture modeling language.	MDA-SOA	This paper presents a tool for modeling SOA using SoaML and partially generating DS XML to provide SoaML based MDA support. The tool is developed using Eclipse utilities like EMF, GMF and ADT to build a SoaML editor and an ATL configurator for generating DS models.
[196]	Sadovykh et al.	2010	Enterprise Architecture Modeling with SoaML using BMM and BPMN – MDA Approach in Practice	MDA-SoaML	This paper described the authors' experiment of utilizing oriented architecture Modeling Language (SoaML) [197] for arranging business models and enterprise IT systems implementation.
[160]	Berre et al.	2009	Model Driven Service Interoperability through Use of Semantic Annotations.	MDI	This paper presented an approach for analysing two ontology-based semantic annotation architectures for service interoperability, the EMPOWER architecture was developed according to platform specific XML-based technologies and the MEMPOWER architecture merged the EMPOWER architecture and Model Driven Interoperability (MDI) using UML and SoaML. The evaluation of both approaches was based on examples from interoperability between ERP-systems in a Buyer/Seller interaction context.

Reference	Author(s)	Year	Title	Area	What had been done?
[198]	Bézivin et al.	2010	Editorial to the Proceedings of the First International Workshop on Model-Driven Interoperability.	MDI	This paper represented the structure, objective and details of the First International Workshop on Model Driven Interoperability (MDI 2010).
[159]	Elvesæter et al.	2006	Towards an Interoperability Framework for Model- Driven Development of Software Systems.	MDI	This paper proposed an interoperability framework to address the business interoperability requirements of heterogeneous enterprise using model-driven development of for enterprise applications and software systems.
[174]	Xu et al.	2009	Model Driven Interoperability through Semantic Annotations using SoaML and ODM.	MDI-SOA	This article claimed service interoperability is feasible using a model driven paradigm with service oriented systems described in SoaML as well as semantic annotations to and from ontology models (in ODM)
[199]	Lemrabet et al.	2010	Model Driven Interoperability in practice : preliminary evidences and issues from an industrial project.	MDI-SOA	In this paper described a new practical perspective of interoperability based on MDA and ATHENA Interoperability Framework. The article established according to a technical project called ASICOM which intent to amend an interoperability platform between industrial partners. Consequently, the authors asserted that MDA and SOA together enhance interoperability.
[170]	Arzt	2010	Service-Oriented Architecture A Field Guide to Integrating XML and Web Services.	SOA	This book highlighted Extensible Markup Language (XML), Web services, and Service Oriented principles as problem-solving tools.
[171]	Khoshafian	2007	Service Oriented Enterprises	SOA	In this book, Setrag Khoshafian described technological foundations of Service Orientation Architecture and Service Oriented Enterprise becomes agile and extraordinary.

Reference	Author(s)	Year	Title	Area	What had been done?
[200]	Kavianpour	2007	SOA and large scale and complex enterprise transformation.	SOA	This paper presented practical experience of author regarding application of SOA to a very large and complex enterprise transformation. Consequently, the author proved MDA based approaches guided with supporting SOA governance are the key to success of large scale SOA transformation.
[178]	Han et al.	2009	Organizational Interoperability Supported through Goal Alignment with BMM and Service Collaboration with SoaML.	SoaML	This paper proposed an approach to represent inter-organizational services through the collaboration modelling support in SoaML. Furthermore, an approach for goal-driven identification of business services and service-centric organizational interoperability is introduced.
[201]	Elvesæter et al.	2011	Specifying services using the service oriented architecture modeling language (SoaML)	SoaML	In this paper had a survey study on the SoaML language constructs and reviewed three different methods with the appropriate practical modelling guidelines to specify services.

3.3.2 MDA, and SOA based solutions for Cloud Computing

As described in previous subsection, SOA and MDA approaches are two new methodology that increasingly exploited to develop different frameworks to alleviate issues like interoperability in enterprises [160][194][185][191][195][174][189]. Kim [189] specified main advantages to integrate a service-oriented modeling architecture with MDA:

- The clear organization of models and information based on the stereotypes derived from the SOA and Select Perspective as development process.
- The productivity, quality and impact analysis benefits of the use of MDA with its emphasis on automation, transformation and synchronization.

Cloud providers, mainly cloud Software-as-a-Service (SaaS), can use the advantages of MDA approach to develop the software applications. The interoperability between applications and services is the characteristic of a system designed based on MDA approach. Table 3-4 summarizes current research work on MDA-based solutions for Cloud Computing. Beside

MDA approach, SOA method is a recent methodology which has significantly influenced IT architectures. SOA is fundamentally an architecture framework that can immensely help cloud computing architecture to provide the required services model with agility and scalability [202]. Additionally SOA promised interoperability between applications by put up application systems as group of published services [70]. Dillon and colleagues [5] described several ways that SOA can help implementing cloud services:

- *Service Description for Cloud Services:* Web Services: WSDL language and the Representational State Transfer (REST) protocol are two broadly used interface languages to characterize Web services. Cloud API specification can be defined using these protocols.
- *Service Discovery for Cloud Services:* Various service discovery models can be exploited for cloud resource discovery, resource selection, and service-level agreement verification.
- *Service Composition for Cloud Service:* It is possible to exploit Web Services, which are created to implement business applications, for cloud service integration, collaboration, and composition.
- *Service Management for Cloud Service.* Cloud infrastructure management can adopt research and functions in SOA governance and services management.

Considering the high-level definition of cloud and SOA, Infosys [202] presented how SOA and cloud overlap (shown in Figure 3-22). Table 3-4 also shows the current research work on SOA-based solutions for Cloud Computing. In addition to leverage MDA or SOA based solutions separately to develop Cloud Computing, it is possible to merge SOA, and MDA in progress of optimal solutions for Cloud Computing (e.g Sharma's research work [203]). We are planning to exploit MDA-based SOA method to get the benefits of these technologies in implementing a novel framework for Intercloud Interoperability.

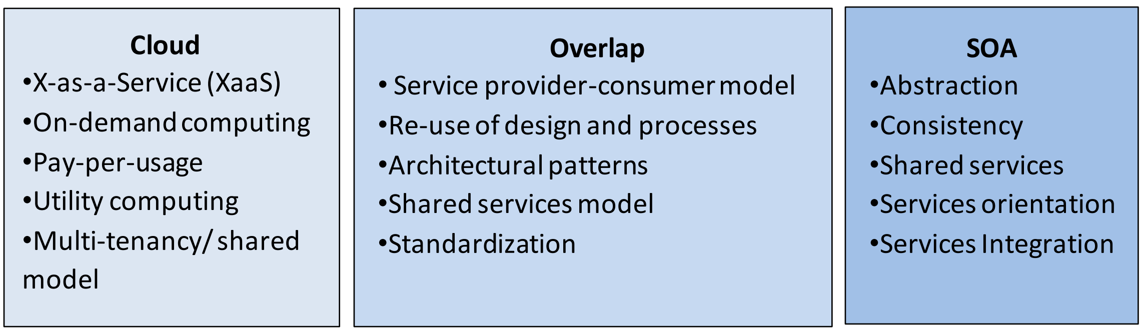


Figure 3-22 Cloud and SOA overlap in several architectural aspects [202].

Table 3-4 Current state-of-the-art for MDA-based, and SOA-based solutions of Cloud Computing

	Author(s)	Year	Title	Area	What had been done?
[204]	Sharma & Sood	2011	Cloud SaaS and Model Driven Architecture.	MDA-Cloud	Incorporating MDA reduces the impact of applying software technological advancements on software applications and it augments the rigor, durability and reusability of the cloud services. In this paper, MDA approach was deployed to develop cloud SaaS.
[205]	Sharma & Sood	2011	A Model-Driven Approach to Cloud SaaS Interoperability	MDA-Cloud	This paper introduced an MDA-based approach to provide interoperability among the software services in the cloud.
[206]	Sharma & Sood	2011	Enhancing Cloud SaaS Development With Model Driven Architecture	MDA-Cloud	In order to have robust, flexible and agile software solutions for advanced cloud software applications, this paper studied the MDA approach to develop software systems
[202]	Infosys	2011	Connecting the dots : Cloud and SOA	SOA-Cloud	Infosys released a whitepaper in 2011 to present the overlap between SOA and Cloud Computing and explain how SOA has being connected and enhanced cloud.
[207]	Maule	2012	SoaML and UPIA Model Integration for Secure Distributed SOA Clouds	SoaML-Cloud	This paper described the required information for SOA modelling techniques and some methods to exchange between U.S. Department of Defence (DoD) and commercial tools.
[203]	Sharma	2011	Modelling Cloud SaaS with SOA and MDA	MDA-SOA-Cloud	This paper highlighted merging Cloud Computing, SOA, and MDA in progress of optimal business solutions.
[208]	Zhang and et al.	2012	On-Demand Service-Oriented MDA Approach for SaaS and Enterprise Mashup Application Development	MDA-SOA-Cloud	This proposed an On-Demand Service-Oriented Model Driven Architecture approach that applies Service Oriented Architecture (SOA) elements into MDA to develop an enterprise mashup prototype.

Chapter 4

The InterCloud Interoperability Framework (ICIF)

4 The InterCloud Interoperability Framework (ICIF)

This chapter proposes a novel framework based on Model Driven Architecture and Service Oriented Architecture which support intercloud interoperability in a heterogeneous computing resource cloud environment. Aa generic architecture for intercloud framework with four layers is proposed. Then, considering the four layer architecture, a detailed model proposed. In the model, Cloud Subscriber requires to exploit computing resources from another Cloud Provider with the propose of delivering better services to its applications with the lower cost. Different aspects of the proposed ICIF for the chosen scenario are explained.

A job-scheduler is required to be defined as a functionality component of our interoperability framework. As a traditional problem, it is proven that finding an optimised job scheduling solution for distributing the multiple job operations, with QoS constraints in a distributed environment is a nondeterministic polynomial time (NP-complete) problem. Therefore, this thesis proposes a heuristic solution for Genetic Algorithm based job scheduler as a part of interoperability framework offering workload migration with the best performance at the least cost. The Job-Selection module integrates the Job-Scheduler Generic Enabler from FI FUTURE INTERNET Core Platform (FI-WARE) cloud that is exploited by the MANufacturing industries (FITMAN) Portugal trial. and its adopted cloud hosting architecture.

This chapter includes six main sections: First, considering literature review chapter, required concepts considered in proposed Intercloud Interoperability Framework are discussed. Second, a generic architecture for InterCloud Interoperability Framework is proposed. Third, the ICIF for Computing Resource Cloud Providers is discussed in detail. Fourth, a new Genetic Algorithm based job-scheduler is proposed. Fifth, a short introduction to FITMAN Portugal trial is presented. Finally, the content of this chapter is summarized.

4.1 Underlying Assumptions of the Proposed the Intercloud Interoperability Framework

Cloud computing is a buzzword in the area of information technologies which delivers on-demand services ranging from software to platform or infrastructure services over the internet. In previous chapters, many challenges are discussed in the area of cloud computing. This thesis identified “intercloud interoperability” as a research challenge. To develop the solution and simulation process presented in this thesis, it is fundamental to identify cloud system appropriately. We consider following characteristics of Cloud Computing, specified by the National Institute of Standards and Technology (NIST) [43], to distinguish Cloud from other computing paradigms:

1. *On Demand Self-Service:* A cloud computing vendor has to provide computing resources automatically according to the customer requirements.
2. *Broad Network Access:* A cloud corporation should be able to provide its available services for any heterogeneous client platforms (e.g., Smart-phones, and laptops), regardless of specifications, from any Internet connected location.
3. *Shared Resource Pooling:* The cloud computing vendor provides a pool of computing resources to serve multiple consumers using a multi-tenant model, with various physical and virtual resources. The allocation and reallocation of resources is dynamic and in accordance with consumer demand.
4. *Rapid Elasticity:* A Cloud Provider (CP) should be able to rapidly and elastically include or exclude computing resources according to the client's changing needs. The cloud consumer should be able to purchase the provided cloud services in any quantity at any time. In reality, the CP does not have unlimited resources, hence, based on the SLA contract, provider has to pay a penalty for not meeting current requirements of the clients. To reduce the penalty cost and increase the QoS, InterCloud Interoperability can support CPs to provide better rapid elasticity.
5. *Measured Service:* A cloud system should have a number of appropriate mechanisms to monitor, control, and report automatically the utilizing the computing resources that can provide transparency between the cloud service consumers and provider. Therefore, cloud services exploit a metering application to control, monitor and optimise the resource consumption. As a result, the customer pays only for the time of utilizing the cloud services.

The well-known three layered cloud architecture covers the three level of service model in cloud computing including Software, Platform and Infrastructure as a Services [43]. This thesis subdivides infrastructure cloud services into three sub-layers as shown in Figure 4-1:

1. Communication as a Service (CaaS).
2. Storage as a Service (DaaS).
3. Computational Resources as a Service (IaaS).

The focus of this thesis is supporting interoperability between CPs that provide Computational Resources as a Service (IaaS). IaaS CPs deliver scalable, secure, and accessible computing resources such as variant types of computing processors (CPUs) and memory with different network bandwidth qualities over the Internet. IaaS vendors help customers to reduce infrastructure investment cost and increase efficiencies of modernizing and developing IT capabilities. The cloud consumer can manage the allocated cloud infrastructures to develop, deploy and run applications. Amazon's EC2 [59], Windows Azure Virtual Machines [60], and Rackspace Cloud [61] are some popular available IaaS CPs.

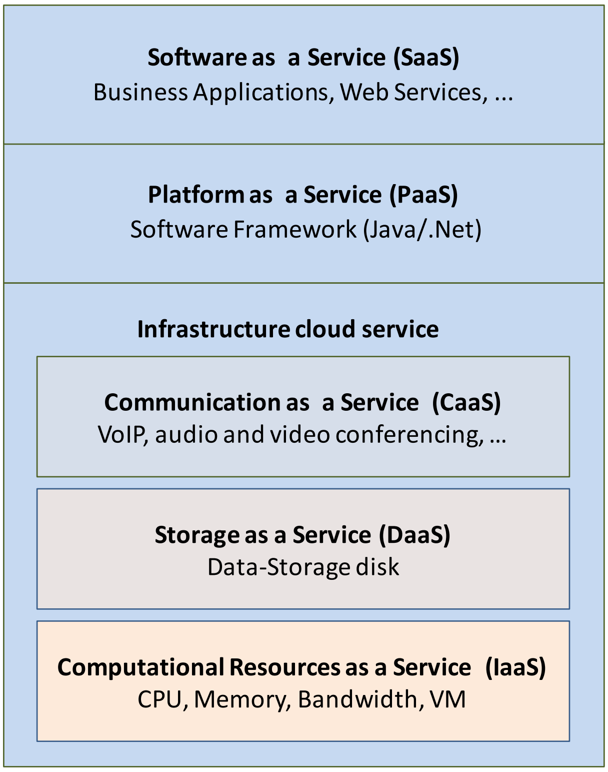


Figure 4-1. Three layered cloud architecture: Software, Platform, and Infrastructure service models. Infrastructure cloud service is subdivided in Communication, Storage, and Computational Resources as Services.

The proposed interoperability framework for IaaS cloud service providers forwards the workload to selected IaaS cloud providers. Thus, the proposed framework considers the collected protocols, standards, formats, and common mechanisms by Bernstein [45] that can be useful for intercloud architecture. Moreover, through studding the literatures, we diagnose following approaches are the appropriate methods for developing ICIF (discussed in next section):

- MDA as a software design approach can be used to develop and integrate enterprise applications using automated tools to provide system-independent models and transform them to the efficient implementations.

- SOA is an architectural solution in which software is constructed as combined applications made up of services running on various nodes, interact through exchanging messages.
- Since scheduling jobs with multiple QoS requirements in a distributed environment is a complex problem, a Genetic Algorithm based solution can be an appropriate method job-scheduler considering multi-criteria constrains that will be discussed in detail.

Following sub-section refers to other references that formally describe QoS [209][210][211] and SLA [212][213] and identifies the required parameters and characteristics for SLA and QoS modules that are fundamental for intercloud interoperability.

4.1.1 Appropriate QoS-SLA characteristics

Numerous cloud services with different pricing and Quality of Services (QoS) exist in an intercloud environment which makes it complicated to select the best composition of services based on consumer requirements. To distinguish the most appropriate combination of services, Intercloud Interoperability framework should consider QoS criteria and Service level agreements (SLAs) as a contract negotiated and agreed between the service provider and the consumer.

Some previous research work have been studied the appropriate models for QoS in cloud environment [209][210][211] that can be beneficial to our proposed model. Additionally, research on defining a formal model for SLA has been considered in various systems [212][213].

In this section, we are aiming to present suitable SLA-QoS characteristics for IaaS cloud service providers and consumers.

In our interoperability framework, the following QoS requirements have been considered: availability, reliability, performance, security, scalability, data communication cost, capacity, and latency parameters for IaaS cloud service (Figure 4-2).

Moreover, the appropriate SLA characteristics for IaaS cloud services for all types of requirements are listed in Figure 4-3. The SLA characteristics include Common SLA Features which are general requirements for all cloud services and the Specific SLA features which are required for delivering IaaS cloud services. To propose appropriate SLA characteristics, we investigated some previous research work [212][213] as well as some dominant IaaS cloud service providers, such as Amazon's EC2 [59], Windows Azure [60], and Rackspace Cloud [61].

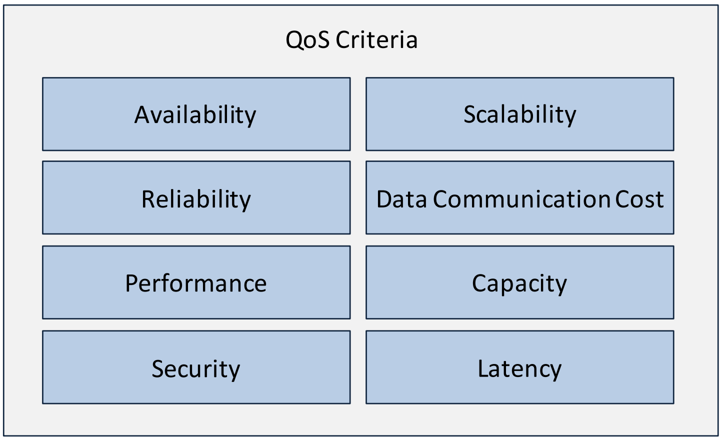


Figure 4-2 Required QoS Parameters for IaaS services.

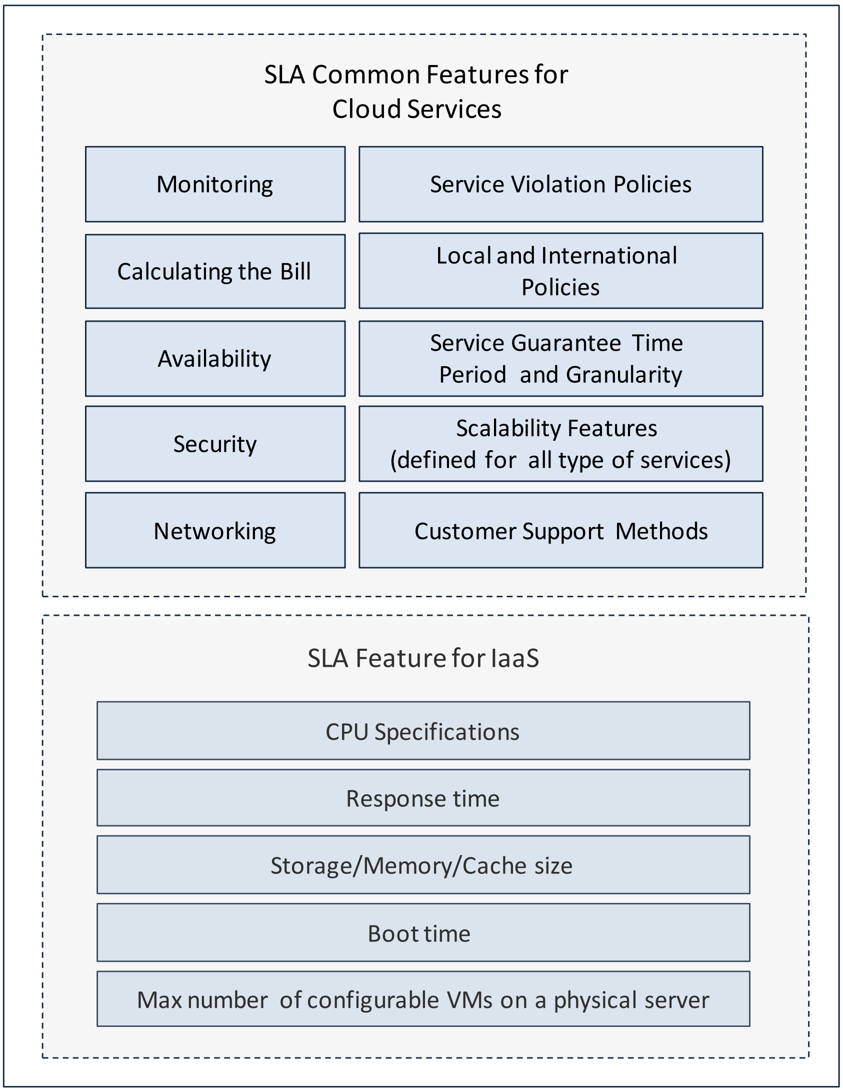


Figure 4-3 Required SLA characteristics for IaaS over Intercloud.

4.2 *Generic Architecture for InterCloud Interoperability Framework (ICIF)*

We consider a generic architecture for our framework shown in Figure 4-4 that aims to resolve interoperability incompatibilities between heterogeneous Cloud computing Platforms. This architecture utilizes the knowledge driven from emerging IT trends such as MDA, SOA, semantics and also provides an interface for integration of other applications being developed to perform various tasks in the paradigm of cloud computing. It comprises of two horizontal layers, the MDA-SOA Layer, the Enablers-Integration Layer and two vertical layers, namely the Semantics Layer and the InterCloud Layer, that span across all the horizontal ones. A high-level view of the generic architecture is as shown in Figure 4-4:

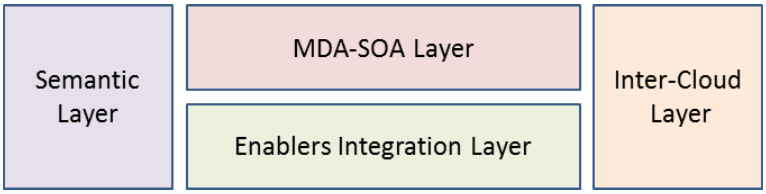


Figure 4-4 A high-level view of the generic architecture.

The *MDA-SOA Layer* implements the core functionalities offered by the overall framework that will support major interoperability related operations. The *Enablers Integration layer* provides the interfaces for integration of third party cloud-based applications into the generic architecture, so as to achieve some specific tasks. The *Semantics Layer* provides the functionality to maintain and utilize the semantic models that will be necessary to obtain interoperability. The *InterCloud Layer* puts in place the technical infrastructure related to independent clouds, which provides necessary information for all the horizontal layers. All of these functionalities will be exposed through well define interfaces like web service which provides an easy access for the MDA-SOA Framework functionalities:

4.2.1 Semantic Layer

Semantic Layer is an important layer of ICIF to clarify semantic interoperability conflicts between Cloud Subscriber and Cloud Providers. As shown in Figure 4-5 it has four components named Application Model, Data Model and Cloud Offering Model, span the entire architecture resolving semantic interoperability conflicts that are raised between different clouds. A data model should clearly specify the structure of data. A data model consists of data elements and their relations.

It is to be noted that Cloud Offering Model is the top level abstraction component to generalize different models of cloud offering. In any instance, this can be implemented by

SaaS, PaaS or IaaS Offering Model, based on the use-case(s) for which the intercloud interoperability framework is being used. In this thesis we are considering only IaaS Offering Model.

The application model is fundamental for SaaS or PaaS intercloud interoperability environment. Application model should be able to provide enough information about the structure of applications exist in both Cloud Subscriber and Cloud Provider. It should provide details on classes, controllers, and other elements that provide or affect application functionality.

Semantics are used by the MDA-SOA Layer in order to provide the means for developing interoperability related mechanisms.

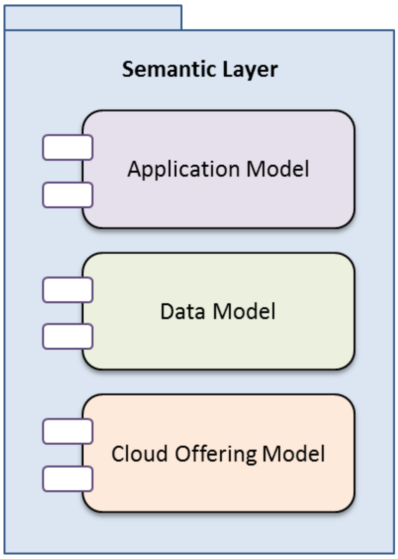


Figure 4-5 Semantic Layer.

4.2.2 GE Integration Layer

Interoperability between clouds will arise because of different use-case scenarios, which will require providing various implementations based on the problem domain that are discussed in previous section. This thesis proposed interoperability solution for migrating the workload operations to other Cloud Providers available. Hence, for the use case chosen in this PhD work, the ICIF requires a queue of job operations from Cloud Subscriber as input to distribute among other Cloud Providers. “Generic Enabler (GE) Integration Layer” of ICIF architecture deploys the Job-Scheduler GE of Cloud Subscriber and selects the jobs that are not depend on a unique computing resources of CS and waiting to receive computing resources. The fifth section of this chapter introduce Job-Scheduler GE of FI-WARE platform cloud that is integrated as part of a job selection module of ICIF.

This, layer acts as the point of integration for such implementations which are termed as enables in this thesis. So, the lower layer of the architecture provides an open space to integrate third party implementations. The components being integrated in this layer virtually can be anything -service or application and will communicate with other layers or are used by other layers through well-defined interfaces. So, in the generic architecture this layer is just an abstraction layer, and doesn't require any predefined components, because this layer doesn't implement any specific functionality.

4.2.3 InterCloud Layer

One of the vertical layers of the generic architecture intercloud layer involves the appropriate capabilities that enhance the selection of specific providers form the network of cloud providers. This layer makes use of the SOA and Cloud computing principles to provide different functionality of InterCloud Layer. An abstract view of the InterCloud Layer is presented in Figure 4-6. This layer has four main components that will be discussed in next section for the case of migrating the workload operations from Cloud Subscriber to the IaaS Cloud Providers.

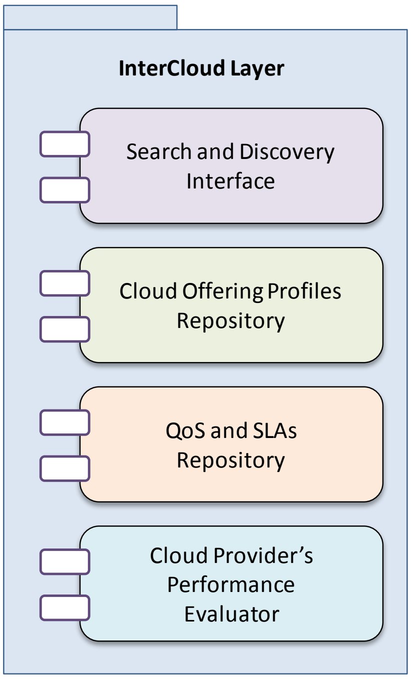


Figure 4-6 InterCloud Layer.

Its main components support search and discovery mechanisms with the help of repositories. At the same time they support the selection mechanism by providing the profile of the cloud providers through QoS and SLAs repositories.

4.2.4 MDA-SOA Layer

The MDA-SOA Layer is the top layer of ICIF architecture that acts as the arbiter layer between the other layers. It includes the essential components to improve the semantic annotation of the Semantic Layer and the functionalities of the InterCloud layer. Additionally, this layer makes use of GE integration layer to select job operation waiting for resource allocation.

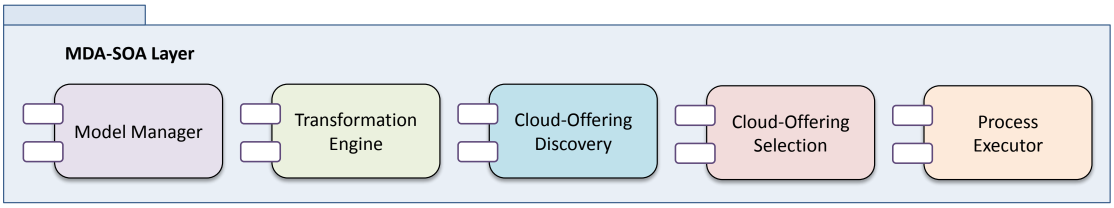


Figure 4-7 The MDA-SOA Layer of ICIF.

The MDA-SOA Layer,lies on top of the Enablers Integration Layer, and comprises of a components that will be accessible from top layer application interface layer, with well-defined interfaces. Its components capitalize on the semantic annotation of the Semantic Layer and the functionalities of the intercloud layer to offer various cloud resources discovery and selection based on the requirements of the service consumer application which is obtained through the top layer i.e. application interface layer. At the same time, this layer makes use of enablers' integration layer to achieve some specified tasks, based on the functionality provided by the enabler. On the whole, MDA-SOA layer acts as the mediator layer between all the other layers. This layer makes extensive use of the concepts and principles that have been discussed in the literature review. MDA-SOA Layer and its components are depicted in Figure 4-7.

4.3 ICIF for Computing Resource Cloud Providers

In order to further explain the proposed framework, for PhD work we select "Workload Migration" as an interoperability use case, which is for workloads independent from unique resources of a specific cloud-provider and its task is dynamically dispatch the operations to the clouds. In other words, the goal of proposing ICIF for this use case is to support interoperability between an IaaS CS and IaaS CPs to deliver services to the users of CS with better performance at the least cost. The ICIF focuses on dynamic dispatching of the operations on the most appropriate CPs available based on the job requirements.

The ICIF vision [15] is shown in Figure 4-8. The framework opens an account between IaaS CS and each available IaaS CP based on related Service Level Agreement (SLA) contract. The list of charges and QoS promises of each available CP is updated periodically. The ICIF considers a test workload, with specified CPU power, and memory or network performance requirements. The framework operates the test workload a few times on each CP, to arrange the CPs by availability, and performance and price aspects. IaaS CS is using FI-WARE Platform that will be introduced in section 4.5. The Job-Selection module of ICIF integrates the Job-Scheduler Generic Enabler (GE) to select the job operations waiting to receive required computing resources. Only the operations that are independent of unique resources of IaaS CS can be selected to forward and execute on other IaaS CPs. The framework selects the most effective IaaS CPs, maps the job model accordingly, and dispatches the job to the selected CP. Finally ICIF collects the operation results from selected CP. All data and model transformation and mapping tasks between CS and CPs are happening through the ICIF.

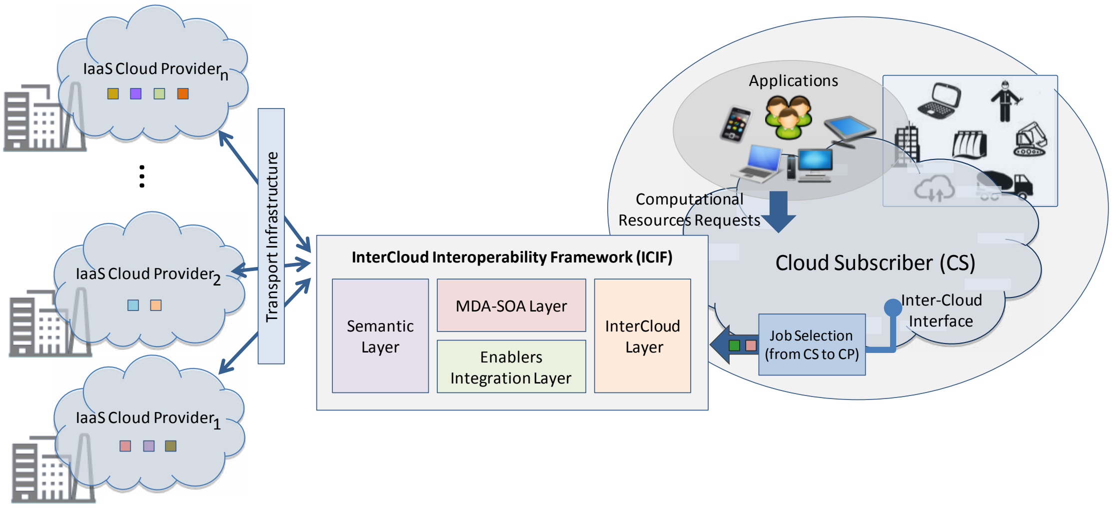


Figure 4-8. InterCloud Interoperability Framework (ICIF) vision.

4.3.1 Formal Model

This sub-section is describing the formal model for the ICIF. To propose a formal model, it is necessary to specify the job model:

Job Model: The input of the ICIF from CS is a finite set $J = \{j_i | i = 1, \dots, x\}$ of jobs j_i. The job production is dynamic and each job j_i is based on the specified requirements of applications. Each j_i has a set of requirements $R_i = \{t_i, cp_i, b_i, m_i, d_i, p_i\}$ which t_i is serving time, cp_i is computing power requirement, b_i is bandwidth requirement, m_i is memory requirement, d_i is maximum possible waiting time, and finally p_i is number of related pricing policy based on the service price and the SLA contract between CS and the application which requested computing resources. In the evaluation section, the possible choice for cp_i, b_i and m_i for the case in this thesis are specified. *Job-Selection Module* selects the jobs from waiting queue in CS, considering the deadline d_i of job j_i is longer than network delay to get service from other CPs.

For our use case, as shown in Figure 4-9, each layer has few module explained as follow:

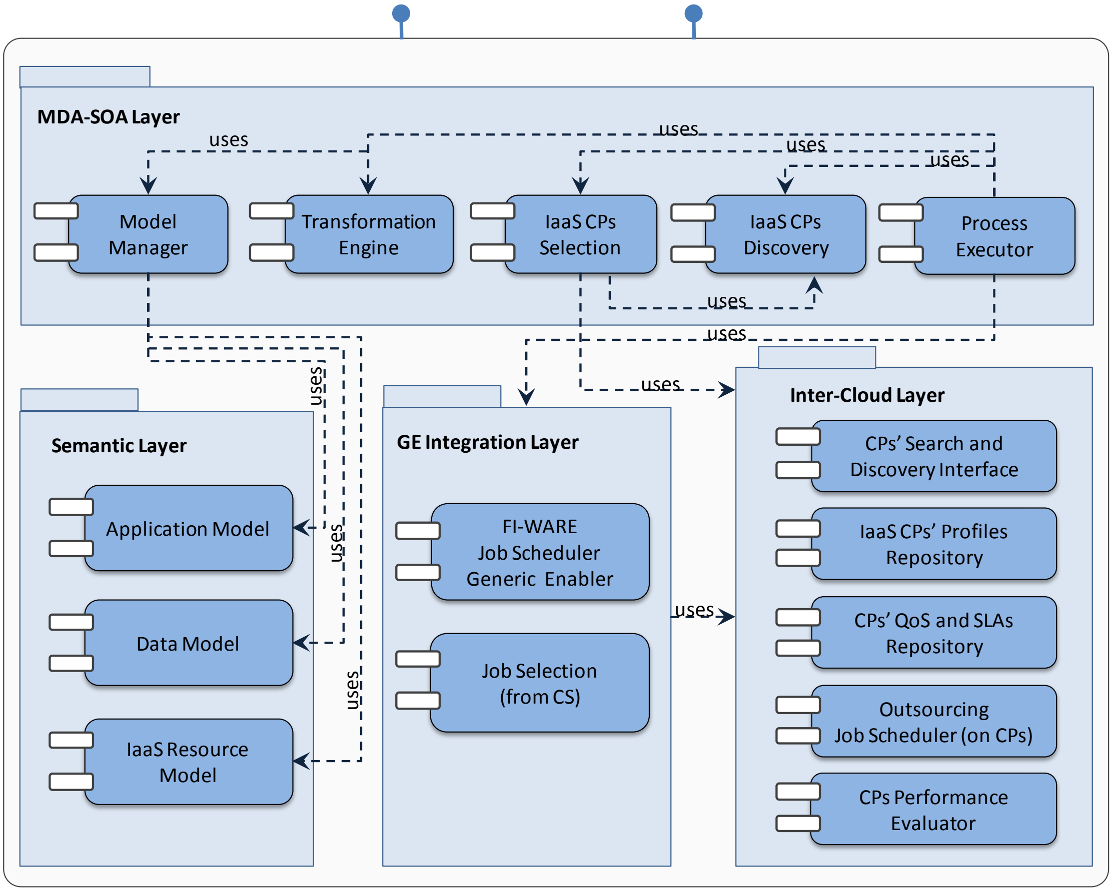


Figure 4-9 The fundamental components of ICIF's four layered architecture

- ***Intercloud-Interface Module*** *≡ φ₀(In : Job-Events, Out : Job-Queue):* ICIF Framework receives job operations for workload migration through φ_0. The jobs are independent from any unique computing recourse of CS.

- ***Job-Selection Module*** *≡ φ_1(In : Job-Queue Out : Job-Queue):* This module received the Job-Queue through φ_0, and evaluate the possibility of outsourcing a job j_i on other IaaS CPs. A job can be selected if the deadline d_i of job j_i is longer than network delay to allocate computing resource from other IaaS CPs.

- ***Model-Manager Module*** *≡ φ_2(In : Job, Out : Job-ObjectModels, Job-OperationModels, Job-DataModels, Job-Requirments):* It receives jobs from φ_1 and provides the required details of the job using φ_8 accordingly. Each job j_i is specified by data model, operation model, object model and set of requirements $R_i = \{t_i, cp_i, b_i, m_i, d_i, p_i\}$.

- ***QoS-SLAs-Repository Module*** *≡ φ_3(Out: QoS-SLA Lists for each IaaS CP):* Service-Level-Agreement (SLA) is a part of service contract defined by each cloud vendor.

SLA repository represents an agreement between the IaaS CS and each IaaS CP. Each SLA defines recovery actions if agreed requirements cannot be satisfied. Moreover, QoS properties for each service of the cloud provider are provided by this repository which will be used for making the correct selection of the cloud provider based on the job requirements. The CS opens an account with each discovered IaaS CP based on CP's SLA. QoS-SLAs-Repository module holds the list of charges and QoS promises of each CP. Then the CS considers a test workload, with specified CPU power, memory and network performance requirements. The CS operates the test workload a few times on each CP, to arrange the CPs by availability, and performance and price aspects. Moreover, the CS evaluates the CPs for the price and QoS characteristics such as availability, and forwards the workloads accordingly.

- ***Process-Executor Module*** ≡ *φ4(In : { ∀i : φ2(Job-Queue(i)) | Job-Queue(i) ≠ Ø}, Out : Operation-Series)*: It is responsible for the execution of the business process based on the details and the requirements of all the jobs available in a Job-Queue. Process Executor defines the sequence of operations to be performed to achieve some specific job. Every activity of the process model will be evaluated and the ones that satisfy the business conditions for the current work-flow would be executed. It also keeps track of all the activities and adds events to the workload queue.

- ***Resource-Search-Discovery Module*** ≡ *φ_5()*: It provides the functionality for IaaS CPs discovery. It would exploit information offered by semantic models φ_8 and SLA/QoS specifications φ_3 in order to find IaaS Cloud Resources in other available clouds which meet the current work-flow requirements.

- ***Resource-Selection Module*** ≡ *φ_6()*: Resource selection component selects appropriate IaaS CPs from available cloud providers. This module considers information from SLA-QoS-Repository Module φ_3() and discovered CPs from Resource-Search-Discovery Module φ_5() to select the set of clouds for migrating and dispatching IaaS workloads. It also exploits the information from Model-Manager Module φ_2() to make the best suited selection.

- ***Transformation-Engine Module*** ≡ *φ_7(In : Job, Out : Job`)* : φ7 performs the necessary model transformation to map the "Job" details obtained in Model-Manager Module $\varphi_2$() to "Job`" (shown in Figure 4-10). It also uses the Semantic Module φ_8() to make the necessary transformations. φ_7() is the key component of the framework to support interoperability through mapping workload from IaaS CS to other selected IaaS CPs.

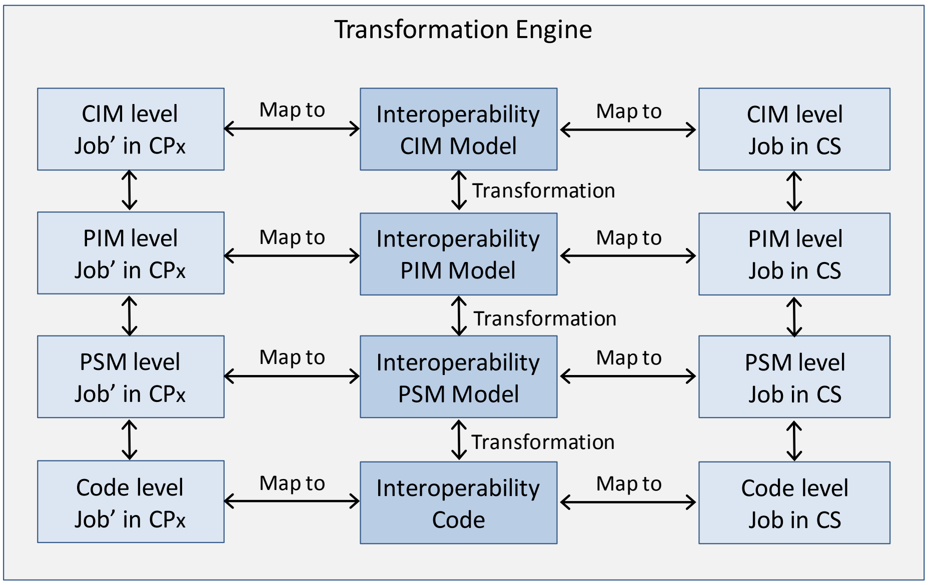


Figure 4-10 Transformation-Engine Module

- ***Semantic Module*** ≡ *φ_8(Out : ObjectModel, OperationModel, DataModel*): Intercloud Semantic is an essential module of the framework with three components : ObjectModel, OperationModel, and DataModel. Semantic layer provides the functionality to maintain and utilize the semantic models that will be necessary to obtain interoperability.

- ***GA based Outsourcing-Job-Scheduler Module*** ≡ *φ_9()*:This module dispatches all selected jobs coming from φ_1 to outsource on selected CPs effectively and exploits module φ_7 to map the job accordingly. This module exploits the GA based Job-Scheduler proposed in section 4.4).

- ***Job-Results Module*** ≡ *φ_{10}(Out : Result` → Result)* : This module collects the results, performs necessary transformations and maps and sends back the results through an Interface component.

- ***CPs-Performance-Evaluator Module*** ≡ *φ_{11}(In : ph(t) list → ph(t+1) list) :* Each IaaS Cloud Provider has a "performance history variable" at time t called $ph(t)$. ICIF framework sends a test workload to each Cloud Provider CP_k periodically and updates the performance variable ph_k according to the CP_k's resource availability and CP_k's response time.

4.4 A new Genetic Algorithm Based Job-Scheduler

The Cloud Subscriber workloads consist of multiple jobs, with QoS constraints and the job model defined in section 4.3.1. The Cloud Subscriber has limited number of resources, thus, it requires migrating the workload on the other Cloud Providers to deliver better QoS. We assume each job has a number of QoS requirements that should be full filled within a specified deadline; otherwise there is a pre-decided penalty cost. Hence, a job-scheduler is required to be defined as a functionality component of our interoperability framework. The job-scheduler should be able to distribute the jobs to the available Cloud Providers effectively.

As a traditional problem, it is proven that finding an optimised job scheduling solution for distributing the multiple job operations, with QoS constraints in a distributed environment is a nondeterministic polynomial time (NP-complete) problem [18][214]. Therefore, we must use a heuristic job scheduling solution to reduce the overall cost and increase the performance. This PhD thesis proposes a job scheduler based on iterative Genetic Algorithm (GA) [215].

We assumed the job that has smaller "MaxWaitingTime" or "deadline" should receive service sooner. Figure 4-11 is showing one example: the queue holds the ID number of jobs and we assumed the ID of new job with "MaxWaitingTime" between d_S and d_{S+1} should be add accordingly to the right place in the queue between Job ID_S and Job ID_{S+1}. If there exist jobs with the same "MaxWaitingTime", the ID of new job should be added to the end of the series. It should be mentioned that the value of "MaxWaitingTime" variable reduces by time.

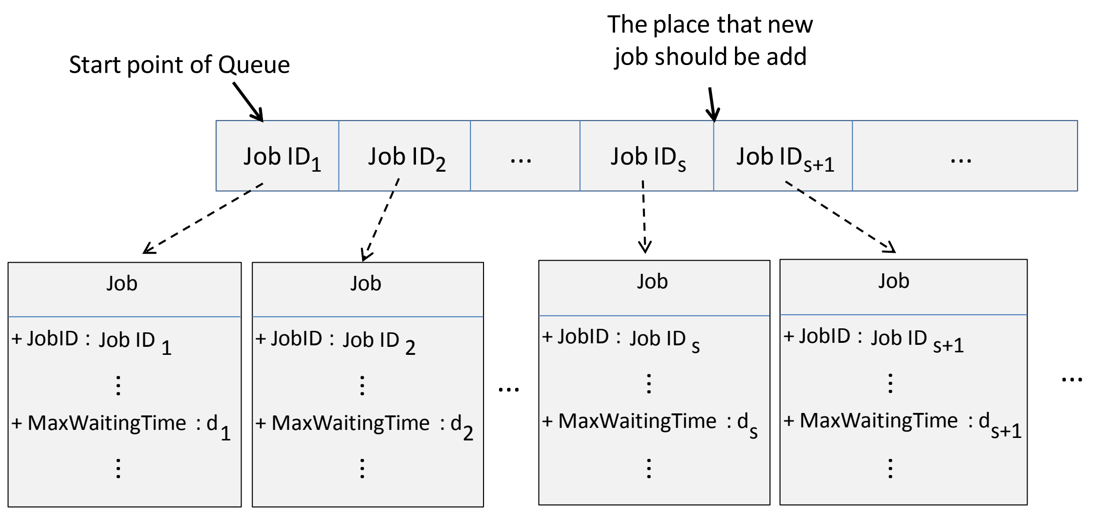


Figure 4-11 It is assumed the ID of new job with "MaxWaitingTime" between d_S and d_{S+1} should be add accordingly to the right place in the queue between Job ID_S and Job ID_{S+1}.

The proposed job scheduling algorithm is shown in Figure 4-12. In this algorithm, there are a number of input variables:

1. A Job-Queue with priorities that each job_i is specified with a set of Requirements: $\{t_i$,

cp_i, b_i, m_i, d_i, p_i}. Where the t_i is serving time, cp_i is computing power requirement, b_i is bandwidth requirement, m_i is memory requirement, d_i is maximum possible waiting time, and finally p_i is number of related pricing policy based on the service price and the SLA contract between CS and the application which requested computing resources.

2. The list of dynamic performance history of each Cloud Provider.
3. “x”: It is the number of jobs for each step that will be explain in an example.
4. “n”: It is iteration number will be explain in an example.
5. “crossover-rate”: It is a rate of a process that takes more than one parent solutions and producing a child solution from them. Here the parents are the pattern of series of CPs allocated to the jobs at each step.
6. “mutation-rate”: In GA, "mutation" is a genetic operator that modifies some gene values in a chromosome from its original state. This operator intends to maintain genetic diversity with mutation-rate from one generation of a population of genetic algorithm chromosomes to the next.
7. “acceptable queuing time”.

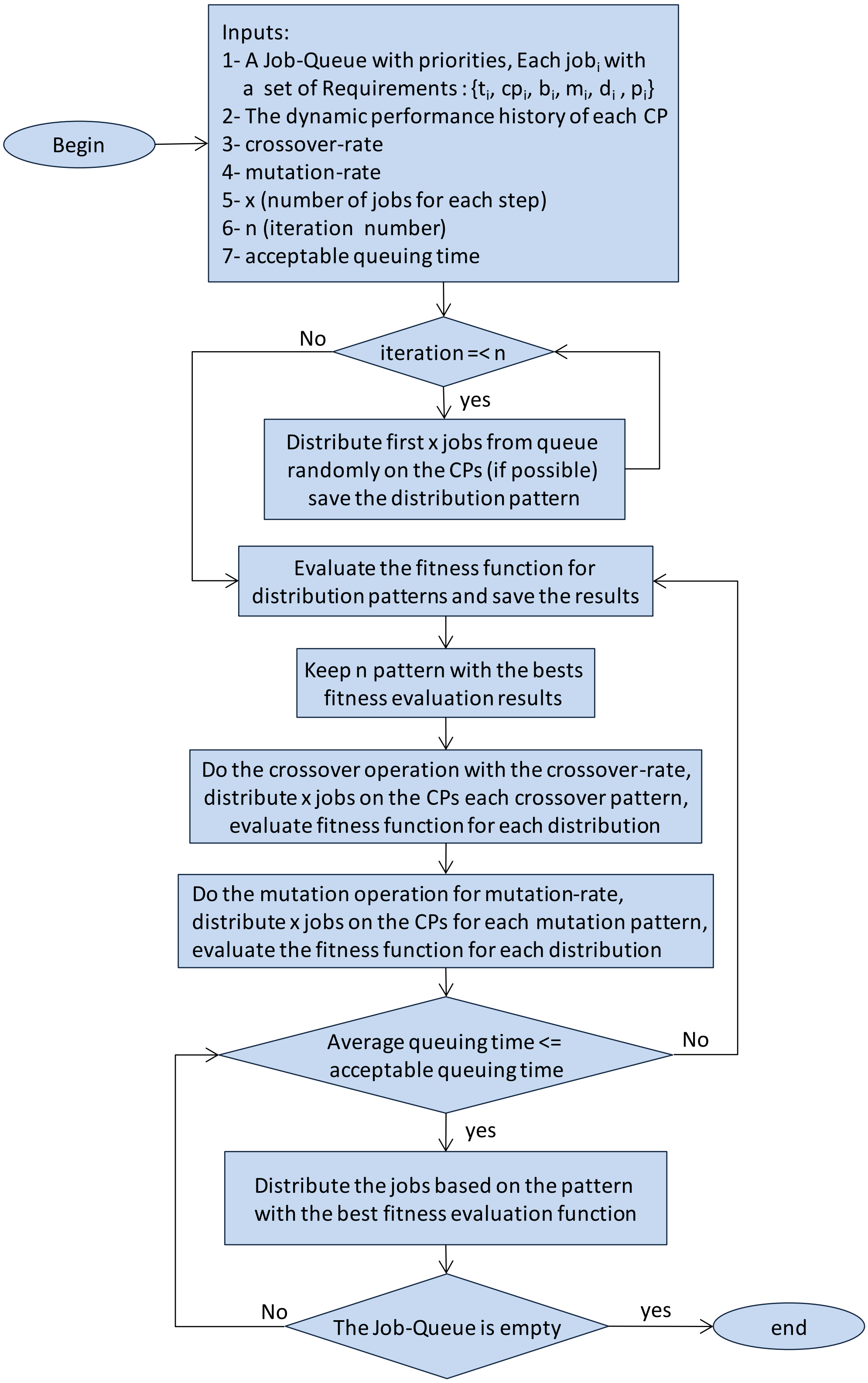


Figure 4-12. The Genetic Algorithm based model for distributing jobs on the selected Cloud Providers.

Defining an applicable *fitness function* is essential and having strong effect on the convergence rate of GA and achieving the optimal solution. This thesis considered two main factors to define the fitness function:

1. The Performance of each IaaS CP: The framework allocates a performance history variable ph_k to each IaaS Cloud Provider CP_k. ICIF framework sends a test workload to each Cloud Provider CP_k periodically and updates the performance variable ph_k according to the CP_k's resource availability and CP_k's response time. The variable ph_{k_t} is the average of $ph_{k_{t-1}}$ with factor of (m-1) and the last resource availability and response time:

$$ph_{k_t} = \frac{((m-1) * ph_{k_{t-1}} * \frac{\text{Availability}_{k_t}}{\text{ResponseTime}_{k_t}})}{m}$$

   The GA solution should maximize the sum of ph_k for all k (available CPs).

2. The Cost: The ICIF framework has the SLA repository based on the agreement between CS and CPs that includes the price lists for different computing resource offering. The $cost_{ik}$ is the cost of computing resource offering from Cloud Provider CP_k for the requirement of job j_i. The GA solution should minimize the overall cost.

   In our GA based solution, the fitness function is defined as:

$$f = \frac{\sum ph_{j_i}}{\sum cost_{j_i}}$$

   Where ph_{j_i} is the performance of the CP which provides resources for job j_i with the $cost_{j_i}$.

To describe the algorithm, we give a small example. To simplify the example we assume all Cloud Providers (CPs) are able to deliver all job requirements. In this example we consider there are 4 CPs available to corporate with Cloud Subscriber, the x is 50 jobs, and the iteration number n is 20 (shown in Figure 4-13). Additionally, we assumed the $\text{crossover} - \text{rate} = 0.1$ and the $\text{mutation} - \text{rate} = 0.02$. In real scenario we run the simulation model for different values for crossover-rate and mutation-rate, and finally we achieve the values of these factors of GA based solution for the best performance/cost results.

Figure 4-13 An example for proposed GA-based job-scheduler to distribute jobs from Cloud Subscriber to 4 other Cloud Providers. It is assumed that number of jobs for each step is x=50,the iteration number is n=20, the crossover-rate is 0.1 and the mutation-rate is 0.02.

As shown in Figure 4-13, the algorithm starts with random allocation of CPs to a series of 50 jobs and evaluates the fitness function for each series of allocation. The algorithm repeats this step for n=20 times and save the allocation patterns and the fitness evaluation results. Then, algorithm applies the crossover operation with the rate of 0.1 on the series of CPs allocation and distributes 50 jobs on the CPs for each crossover pattern and evaluates fitness function for each distribution. Similarly, algorithm applies the mutation operation with the rate of 0.02 on the series of CPs allocation and distributes 50 jobs on the CPs for each mutation pattern and evaluates fitness function for each distribution. Afterward, if the average queuing time is smaller than acceptable queuing time, the algorithm distributes the jobs based on the pattern with the best fitness evaluation functions, otherwise repeats the crossover and mutation operation steps. The algorithm repeats this process of distributing the jobs on CPs till the queue is empty.

4.5 *Job-Scheduler GEs from FI-WARE cloud*

In the PhD thesis, we are proposing an Intercloud Interoperability Framework that integrates Job-Scheduler Generic Enablers (GE) from FI-WARE[1] (FUTURE INTERNET Core Platform) cloud. The overall vision for MDA-SOA based inter-cloud interoperability is shown in Figure 4-14. A cloud based application makes use of the proposed framework to interoperate with other clouds. Application accesses the functionality of the framework through the interfaces defined by the framework. Following subsections introduce MANufacturing industries (FITMAN) and its adopted cloud hosting architecture.

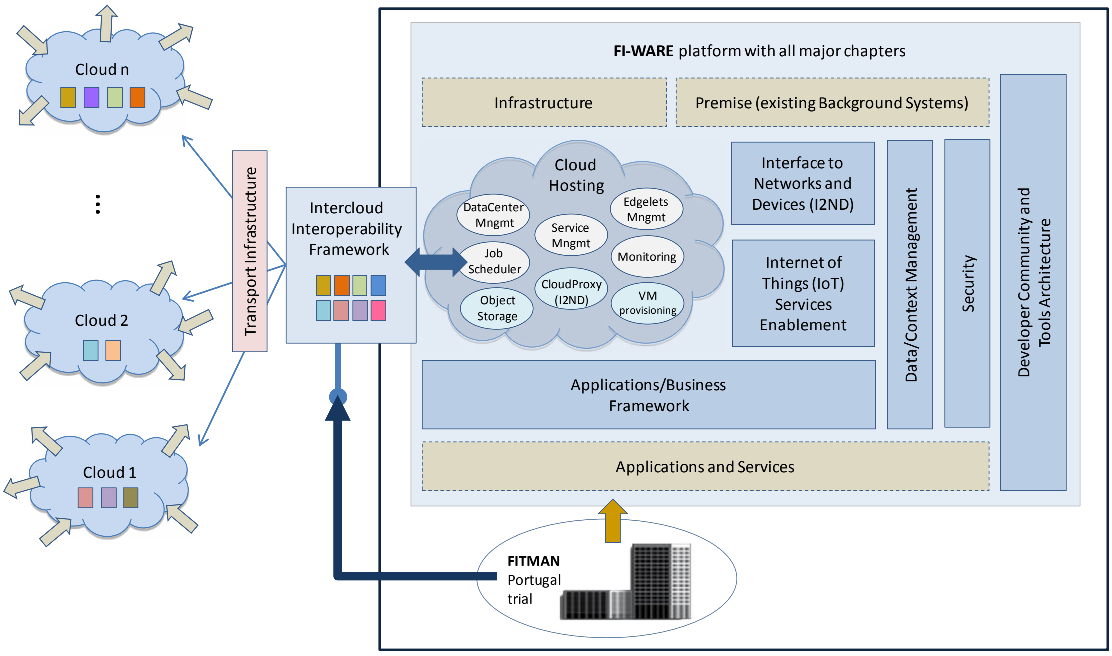


Figure 4-14 MDA-SOA Intercloud Interoperability Framework.

4.5.1 FITMAN

Future ICT technologies should leverage enterprises to respond more effectively to current challenges faced by the enterprises, such as global competition, reducing energy consumption and waste generation and constant need for innovation. Future Internet Technologies for FITMAN is one of the Use Case Trials projects selected in the 2nd phase of the Future Internet Public Private Partnership (FI PPP) programme. FITMAN is developing new capabilities to provide improved technologies for interoperability, connectivity, mobility and intelligence, which make enterprises smarter, more agile, mobile and collaborative. FITMAN defined 11 trials to trigger the use of Future Internet technologies in the factories of the future. It provides the FI PPP Core Platform which test and assess the suitability, openness

[1] http://www.fi-ware.eu/

and flexibility of FI-WARE [2] Generic Enablers (GEs)[2] [2]. The FITMAN use case trials belong to several manufacturing sectors such as automotive, aeronautics, construction, and manufacturing assets management. FITMAN findings reveal that the general business objectives behind FI technology adoption seek to

1. Improve communications/collaboration
2. Reduce production costs
3. Reduce time to market
4. Improve the usefulness of the information
5. Increase production capacity.

The FITMAN Portugal trial addresses the development of projects related to construction industry in the context of the Future of Internet. There are certain requirements that will be fulfilled by FI-WARE and other projects at UNINOVA to realize the FITMAN Portugal trial. The high-level goal of the FI-WARE project is to build the Core Platform of the Future Internet. The mission of the FITMAN proposal is:

- Provide the *Future Internet Public Private Partnership (FI PPP)* Core Platform with a set of industry-led use case trials in the manufacturing domain, in order to test and assess the suitability, openness and flexibility of FI-WARE GEs.
- Contribute to the social-technological-economical-environmental-political (STEEP) [216] objectives included in the "ICT for Manufacturing" and "Future Internet Enterprise Systems" [217] EU research roadmaps, by integrating FI generic and specific enablers with key business processes and enterprise applications currently running in Smart, Digital and Virtual Factories of the Future.

It is envisaged that an improvement in the communication and collaboration processes is likely to have considerable impact on the success of the construction project which is measured in terms of project deviation, i.e. total cost, duration and quality. The portable service workspace intends to increase the sharing of electronic information about the construction plans on-site though Internet, promoting a common understanding between planning and construction teams with the support of communications and other technologies to provide exact location of physical objects, thus feeding Augmented Reality technologies. Furthermore, the introduction of the collaboration technology through the usage of FI-WARE GEs may result in reengineering of the problem solving process, leading to further increase in

[2] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Cloud_Hosting

productivity. Figure 4-15 shows the overall ecosystem of FITMAN-Portugal trial which shows all the stakeholders and the role of FI-WARE.

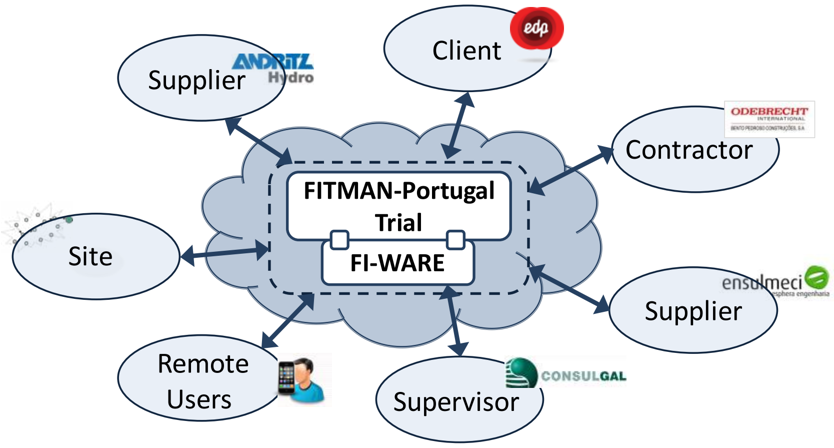


Figure 4-15 FITMAN Portugal trial eco-system.

4.5.2 FI-WARE Cloud Hosting Architecture

The aim of FI-WARE project is to design, develop and implement the Core Platform within the European FI-PPP Program defined under the ICT FP7 Work Programme. It includes six Reference Architectures, Cloud Hosting, Data/Context Management, Internet of Things (IoT), Services Enablement, Applications/Services Ecosystem and Delivery Framework, Security, and Interface to Networks and Devices (I2ND).

The Cloud Chapter of FI-WARE offers Generic Enablers that includes the foundation to establish a cloud hosting infrastructure to develop Future Internet applications and services. The architecture includes a set of GEs to provide hosting capabilities at several levels of resource abstraction with the goal of providing the requirements of different applications. The cloud hosting architecture include following GEs (shown in Figure 4-16):

- *FIWARE.ArchitectureDescription.Cloud.JobScheduler_V2:* This GE offers the application to submit and manage computational jobs in a unified and scalable manner.
- *FIWARE.ArchitectureDescription.Cloud.DCRM (DCRM: Data Center Resource Maangement):* This GE is offering provisioning and life cycle management of virtualized resources (compute, storage, network) associated with virtual machines, which can run general purpose Operating Systems as well as arbitrary software stacks. Application developers and providers can use these virtual machines to develop and deploy their own software components that comprise their application stacks.

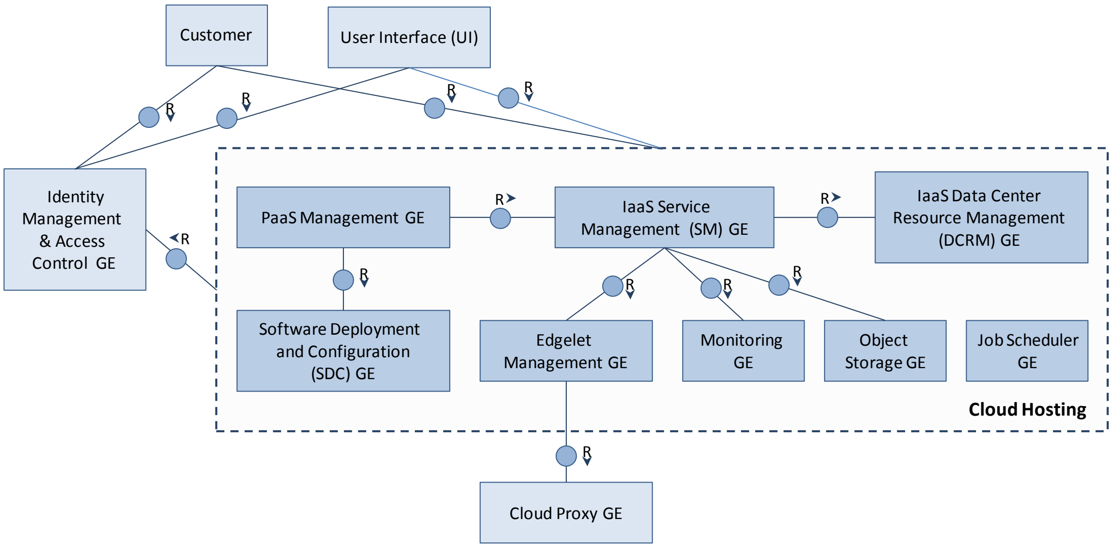


Figure 4-16 FI-WARE Cloud Hosting Architecture

- *FIWARE.ArchitectureDescription.Cloud.ObjectStorage:* Object Storage GE offers provisioning and life cycle management of object-based storage containers and elements, which can be efficiently used to store unstructured fixed content (such as images, videos, etc) as well as accompanying metadata.
- *FIWARE.ArchitectureDescription.Cloud.CloudEdge:* Edgelet Management GE offers the capability to host lightweight application components, called edgelets, on devices typically located outside of the Data Center, such as those provided by the *Cloud Proxy GE* (developed jointly by the Cloud chapter and the Interfaces to Network and Devices chapter)
- *FIWARE.ArchitectureDescription.Cloud.SM (SM: Service Management):* SM GE provides the means to host complex applications potentially comprising multiple virtual machines and other runtime components (as outlined above), by automated provisioning and life cycle management of such compound applications (also called services), including elasticity and auto-scaling based on characteristics collected by the Monitoring GE.
- FIWARE.OpenSpecification.Cloud.Monitoring:
- *FIWARE.OpenSpecification.Cloud.PaaS:* PaaS Management GE uses the above capabilities to offer provisioning and management of complete PaaS environments, leveraging
- FIWARE.OpenSpecification.Cloud.SDC: the Software Deployment and Configuration (SDC) GE which offers a flexible framework for installation and customization of software products within individual virtual machines.

- *FIWARE.ArchitectureDescription.Cloud.SelfServiceInterfaces*
- *FIWARE.ArchitectureDescription.Cloud.Identity*
- *FIWARE.ArchitectureDescription.Cloud.Accounting*

4.6 Summary

This section gives a brief statement of the main points introduced in this chapter. The main focus of this chapter is proposing a novel framework called "InterCloud Interoperability Framework" (ICIF) to support interoperability in a heterogeneous computing resource cloud environment. The proposed ICIF provides enough functionality to provide the ability to select the workloads independent from unique resources of the cloud subscriber and dynamically dispatching the operations to the most effective cloud providers available at runtime. To achieve the goal, during this chapter several tasks in five sections have been done:

- *Underlying Assumptions of the Proposed the Intercloud Interoperability Framework*: The characteristics of cloud computing considered to develop the solution and simulation process of the PhD work are described in this section. Additionally, the appropriate QoS-SLA parameters for proposed ICIF are described.
- *Generic Architecture for Inter-Cloud Interoperability Framework (ICIF)*: The ICIF generic architecture is proposed in this section. It includes four layers: Semantic Layer, GE Integration Layer, Inter-Cloud Layer, and MDA-SOA Layer.
- *ICIF for Computing Resource Cloud Providers*: In order to further explain the proposed framework, for PhD work we select "Workload Migration" as an interoperability use case, which is for workloads independent from unique resources of a specific cloud-provider and its task is dynamically dispatch the operations to the clouds. This section explains the details of proposed solution for this particular use case.
- *A new Genetic Algorithm Based Job-Scheduler:* The proposed ICIF requires an appropriate process for effective IaaS-CP discovery and selection. This section proposes a novel Job-Scheduler that is based on Genetic Algorithm. The GA-based solution offers job scheduling algorithm to dispatch the selected jobs to the available cloud providers with the best performance at the least cost.
- *Job-Scheduler GEs from FI-WARE cloud:* The FITMAN Portugal trial engaged with the development of projects related to construction industry with the goal of initiating the use of Future Internet technologies in the factories of the future. In this project,

there are certain requirements that will be fulfilled by FI-WARE platform. Cloud Hosting is one fundamental layer of FI-WARE which manages and indeed provides cloud services. FI-WARE consists of several Generic Enablers (GEs) including Job-Scheduler GE. This GE is integrated with the proposed ICIF to select the job operations waiting to receive required computing resources.

Chapter 5

A New Agent Based Simulation Model for InterCloud Environment adopted ICIF and the Validation Process

5 A New Agent Based Simulation Model for InterCloud Environment adopted ICIF and the Validation Process

This chapter will validate the proposed solution for computing resource Inter-cloud Interoperability problem. As discussed earlier, the Job-Selection module of proposed ICIF integrates the Job-Scheduler GE from FI-WARE Platform that is exploited by FITMAN Portugal trial. In other words, the job queue will get input from FITMAN project. Hence, it is important to analyse the type of input rate for the evaluation process.

In this chapter, first, we give a short description of FITMAN and then the Agent Based Simulation model is proposed for simulation of the inter-cloud environment that uses ICIF. Finally, the results of simulations are discussed.

5.1 Agent Based Simulation Modeling Approach adopted ICIF

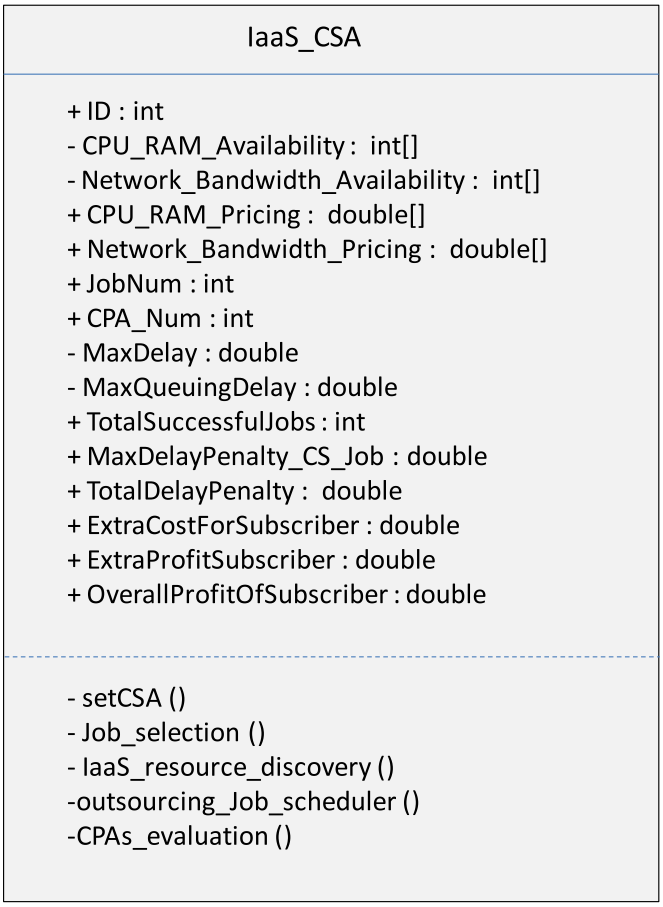
Interactions in Inter-Cloud Environment fall under the category of complex non-linear systems for which simple, intuitive, analytical solutions are not readily available Hence, this thesis discusses an ABS approach to simulate an extendable Inter-Cloud environment that uses proposed the IaaS ICIF. ABS approach is a powerful modeling and simulation technique for a large variety of research topics and has advantages over conventional approaches in many cases [218] and [219]. ABS can simulate a dynamic model in which agents interact repeatedly over the time to achieve an optimized solution. Agents in ABS represent actors, objects, or processes of a system that behave based on the interaction rules of the modelled system. Recent computer technology enables simulation of millions of such agents, which can be analysed to make scientific conclusions. The overall simulation is modelled within the scope of the scenarios being implemented for FITMAN Portugal trial. Moreover, it is considered that the ICIF supports appropriate functions for IaaS inter-cloud interoperability.

In our ABS simulation model, there are three types of agents discussed as follow:

5.1.1 IaaS Cloud Subscriber Agent (IaaS CSA):

IaaS CSA agent is representing IaaS CS assumed in the intercloud environment that uses the proposed ICIF. IaaS CSA is a cloud computing resource provider that has limited number of the computing resources, hence, requires interoperating with the IaaS Cloud Provider Agents (IaaS CPAs) to provide better QoS for the users. IaaS CSA is based on the functionality of the ICIF to dispatch the operations on the most appropriate IaaS CPAs available based on the Job Agents' requirements. The IaaS CSA opens an account with the available IaaS CPAs based on related SLA contract. The list of charges and QoS promises for each available IaaS CPA has to be updated periodically. The ICIF considers a test workload, with specified CPU power, and memory or network performance requirements. The CSA sends the test workload a few times on each CPA, to arrange the CPAs by availability, and performance and price aspects.

In our framework, the Job-Selection module of ICIF integrates the Job-Scheduler GE of FIWARE platform to select the job operations waiting to receive required resources. In reality, migrating a job from CS to a CP costs considerable overheads. Additionally, migrating a job with short life time is wasteful. This raises the question of selecting the best process to migrate. In our model, only the jobs that are independent of unique resources of CSA with a large enough life time are selected to forward and execute on the IaaS CPAs. The framework selects the most effective IaaS CPAs to dispatch the jobs and uses GA based Outsourcing-Job-Scheduler module $\varphi_9()$. In the proposed simulation model, CSA agent has number of attributes and is related to a number of operations described as follows (shown in Figure 5-1):



IaaS_CSA = IaaS Cloud Subscriber Agent

Figure 5-1. Attributes and operation associated to IaaS_CSA agent.

- **ID:** It is a unique number associated to the CSA that is produced at the time of generating the CSA.
- **CPU_RAM_Availability:** It is a list of available cpu and memory resource to resemble a virtual resources. During our simulation process, we considered several types of computing resources including SingleCore, DualCore, QuadCore, and OctoCore processors with a variety of attached memory. Allocating a resource to a job reduces the number of available resources accordingly.
- **Network_Bandwidth_Availability:** It is a list shows the amount of network bandwidth.
- **CPU_RAM_Pricing:** It is a list of pricing policies for available resources. It is possible to provide different combination of available resources for the costumers (as Jobs) that are specified in the SLA of CS. The price chosen is a random value between +10% and -10% of Amazon EC2 Cloud's real price [59].
- **Network_Bandwidth_Pricing:** It is a list of pricing policies for different network bandwidth speeds. It is possible to provide different combination of available CPU, RAM resources with different network bandwidth speeds for the costumers (as jobs) that are specified in the SLA of CS. Similarly, the price chosen is a random value between +10% and -10% of Amazon EC2 Cloud's real price [59].
- **JobNum:** It is the number of available job agents in the simulation environment that are produced based on costumer request to the CS.
- **CPA_Num:** It is the number of available Cloud Provider Agents. The agent defined for Cloud Provider will be introduced later.
- **MaxDelay:** In a real scenario, allocation resources to a job can happen with a small amount of delay that can be varied. The proposed model assumed, if CS has enough amount of resources it will allocate to the job within a random time that can be maximum equal to MaxDelay time.
- **MaxQueuingDelay:** In real scenario, evaluating the current conditions and then queuing the job accordingly can happen with a small amount of delay. The proposed model assumes a MaxQueuingDelay time for queuing process.
- **TotalSuccessfulJobs:** This variable contains the total number of jobs that successfully received the required resources since the time simulation process stats to run.

- **MaxDelayPenalty_CS_Job:** Cloud Subscriber should be able to include or exclude computing resources according to the customer's changing requirements. In reality the cloud subscriber does not have unlimited resources, hence the cloud provider has to arrange appropriate resources to assure fulfilling the current requirements of the costumer based on the SLA contract with the costumers, otherwise it may be specified that cloud provider has to pay a penalty for not meeting the correspondent service level agreement. The penalty can vary for different services. This simulation model, considers a random value for amount of penalty that can be maximum equal to the amount of MaxDelayPenalty_CS_Job. This amount can be easily replaced with a list that contains the numbers of different amount or policies for penalty according to the case study scenario.
- **TotalDelayPenalty:** This variable contains the overall amount of penalty that CS should pay since the time simulation process stats to run.
- **ExtraCostForSubscriber:** When cloud subscriber does not have possibility to provide requested computing resources from its own resource pool, it will forward the request to a selected resource cloud provider. Hence, it might pay more/less for providing service through the resources offered from another cloud. The variable "ExtraCostForSubscriber" contains the extra cost for subscriber through exploiting a particular resource from another cloud. The amount of extra cost can be calculated using the pricing policy between CS and its customers and difference between the pricing policy between CS and selected CP.
- **ExtraProfitSubscriber:** Sometimes, providing the resources through another cloud provider is cheaper than the actual cost of the service inside the cloud subscriber. This simulation model considers a variable called "ExtraProfitSubscriber" that contains the extra profit that subscriber can gain through exploiting a particular resource from another cloud. The amount of extra profit can be calculated using the pricing policy between CS and its costumer and difference between the pricing policy between CS and selected CP.
- **OverallProfitOfSubscriber:** This variable contains the overall profit that CS can gain since the time simulation process stats to run.

Furthermore, there are a number of operations associated to the CSA that are based on the functionality of interoperability framework described in section 4.2:

- **setCSA():** When simulation starts, this function initialize the all variables and states related to Cloud Subscriber Agent.
- **Job_selection():** This function is based on φ_1(In : Job-Queue Out : Job-Queue) module defined as a functionality of ICIF. This function evaluates the possibility of outsourcing a job according to the current situation of CSA and its resource availability as well as job requirements. Only jobs with the requirement independence of a particular resource of CSA that have deadline longer than network delay can forward to receive computing resource from other CPAs.
- **IaaS_resource_discovery():** It is based on φ_5() module of ICIF that provides the functionality for resource discovery from other clouds. It detects other available clouds which meet the current job requirements.
- **outsourcing_Job_scheduler():** It is based on φ_6() and φ_9() modules of ICIF. It allocates resources to the jobs waiting to receive resources from other cloud providers. This module considers information from SLA contract between CSA. This function exploits the GA based Job-Scheduler proposed in section 4.4 and changes the status of jobs and updates the amount of related variables.
- **CPAs_evaluation():** This function is based on φ_{11} module of ICIF. Each IaaS Cloud Provider has a "performance history variable" at time t called $ph(t)$. CSA sends a test workload to each Cloud Provider CP_k periodically and updates the performance variable to $ph_k(t+1)$ according to the CP_k's resource availability and CP_k's response time.

5.1.2 IaaS Cloud Providers Agent (IaaS CPA)

In proposed ABS model, there are predefined numbers of agents for CPs called IaaS CPA. The purpose is cooperation between CSA and CPAs to provide services with higher QoS and lower price to the consumers of CSA. The interoperation between CSA and each CPA is possible through the functionality supported by ICIF. There is a SLA contract between each CPA and CSA that gets updated periodically. A CPA provides computing resources based on the service request from CSA. Each CPA is specified by different service combinations and prices. Each IaaS CPA_i has a number of performance variables. The CSA sends workload test periodically to all available CPA_i and updates the performance variables. The variables associated to each CPA_i are describe as follow (shown in Figure 5-2):

- **ProviderID**: It is a unique number associated to each CPA that is produced at the time of generating.

- **CPU_RAM_Availability**: It is a list of available cpu and memory resource to resemble a virtual resources in each CPA.
- **Network_Bandwidth_Availability: :** It is a list shows the amount of network bandwidth in each CPA.
- **CPU_RAM_Pricing**: It is a list of pricing policies for available resources. It is possible to provide different combination of available resources for the jobs that are outsourcing from CSA. The pricing policies are specified in the SLA contract between CSA and each CPA. The price chosen is a random value between +10% and -10% of Amazon EC2 Cloud's real price [59].
- **Network_Bandwidth_Pricing**: **:** It is a list of pricing policies for different network bandwidth speeds for each CPA available. It is possible to provide different combination of available CPU, RAM resources with different network bandwidth speeds for jobs outsourced by CSA that are specified in the SLA contract between CSA and each CPA. Similarly, the price chosen is a random value between +10% and -10% of Amazon EC2 Cloud's real price [59].
- **PerformanceDelay**: As described before, each Cloud Provider has a number of variables indicating performance of each CP. CSA sends test workload periodically to each CPA and evaluate the different values for the performance variable associated to each CPA. Finally CPAs_evaluation() operation from CSA updates the amount of each performance variable. "PerformanceDelay" variable of each CPA shows the delay of delivering the required resources of the test workload sent by CSA.
- **PerformanceResponseTime**: It is another performance variable associated to each CPA available. It can be upgraded periodically by CPAs_evaluation() operation from CSA. This variable of each CPA represents the response time of delivering the required resources of the test workload sent by CSA.
- **PerformanceServiceAvailability**: Similarly, it is another performance variable associated to each CPA available. It can be upgraded periodically by CPAs_evaluation() operation from CSA. This variable of each CPA represents the service availability of delivering the required resources of the test workload sent by CSA.
- **PerformanceCost**: Again, it is another performance variable associated to each CPA available. It can be upgraded periodically by CPAs_evaluation() operation from CSA.

This variable of each CPA contains the price of the delivered resources that requested by the test workload sent by CSA.

- **setCPA**: When simulation starts, this function initialize the all variables and states related to each Cloud Provider Agent.

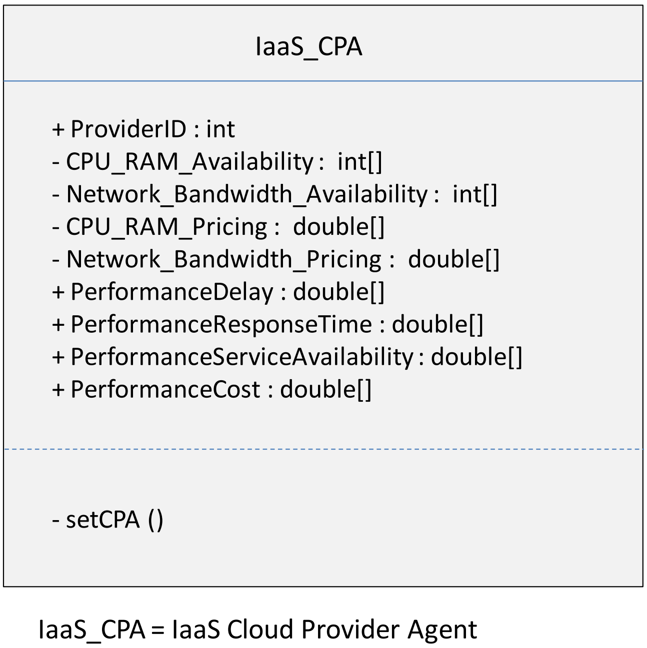


Figure 5-2. Attributes and operation associated to IaaS_CPA agent.

5.1.3 Job Agent

In the simulation model, there is an agent called Job that represents the dynamic workload in CSA. The workload is generated by cloud applications in CS. The job model is based on job definition in ICIF. Each Job agent is based on the requirements of applications and has set of variables shown in Figure 5-3:

- **JobID:** It is a unique number associated to each Job agent that is produced at the time of generating.
- **JobIsSet:** If a Job agent receives the required resources, the value of "JobIsSet" variable will be set to "true".
- **ServiceSupplierID:** If a Job agent receives the required resources from cloud x, the value of "ServiceSupplierID" variable will be set to the ID of cloud x.
- **ServiceTime**: It is a variable that contains the required serving time requested by each Job agent.

- **MaxWaitingTime**: It is a variable that contains the maximum possible delay to get the service requested by each Job agent.
- **CurrentWaitingTime**: It is a variable that contains the current waiting time of each Job agent.
- **CPURequirements**, **RAMRequirements**, and **BDRequirements:** These variables are showing the required computing power, memory, and Network Bandwidth requested by each Job agent.
- **Dependency:** It is a variable that specifies if the Job requirements are depend on a unique resource of CSA or not.
- **DelayPenalty** and **Cost:** These two variables are based on service price and the SLA contract between CSA and the application which requested computing resources.

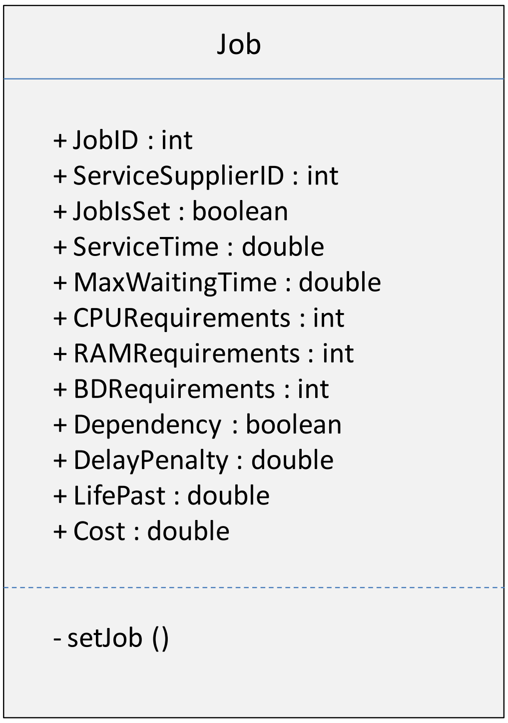


Figure 5-3 and operation associated to Job agent.

Selecting the best process (Job) to migrate from the CS to one of the CPs is fundamental to achieve high performance. In the simulation model, the Job-selection() operation is based on Job-Selection Module from ICIF that selects the jobs from waiting queue in CSA to forward to the CPAs. In the model, only the Jobs that are independent from a specified resource of CSA can be forwarded to get service from CPAs. Moreover, since the required service time can be milliseconds, sub-second to minutes, and hours, only Jobs with a "ServiceTime" variable bigger than few minutes and “MaxWaitingTime” variable bigger than the network delay can be forwarded to the CPAs to receive required computing resources.

Simulation model considers m/g/n queue for modelling the inter-cloud environment with the Poisson process for the Job arrival. Poisson distribution considers a period of time T during which events occur at an average rate of λ events per time unit. Additionally, since the workload is heterogeneous and dynamic, the “ServiceTime” is considered with the general distribution. The reason of exploiting this model will be discussed in next section.

Section 5.3 will discuss the simulation results of our ABS simulation model.

5.2 FITMAN workload

As mentioned in previous chapter, TRIAL 7 of FITMAN project is FITMAN Portugal trial and addresses the development of projects related to construction industry in the context of the Future of Internet. There are certain requirements that will be fulfilled by FI-WARE and other projects at UNINOVA to realize the FITMAN trial. The trial aims at optimizing the management of the construction projects, by early identification of design and technical mistakes, including on line detection and real time fixing of incompatibility using remote collaboration. Middle ware utilizes the collection of FI-WARE GEs for achieving specific tasks like identity management and cloud storage.

The implementation of the trial will help in the better management of the construction management process helping in decision making process and avoid (decrease) future risks during the project life cycle. The work flow remains the same and the information generated at various phases remains the same. But there will be a significant change in the way the generated information is stored, retrieved, processed and distributed.

In this trial, a common web platform will be developed for all the stake holders to store and retrieve information and documents generated at different stages of the work flow. Thus a collaborative workspace will be created using standard web and storage technologies. Concerned authorities have access to the results through platform based on their access rights. Moreover, the physical objects which are important part of the overall work flow is identified and connected to information system and accessed/tracked using new technologies

Various sources of data that produce information regarding concrete class, concreting plan, slum test result, and concrete sample test results are integrated in the central information system. The front end provides web application for entering and viewing information as required and accessibility of the user. Based on the profile of the user (i.e. designer or supervisor or contractor), the application provides varying work spaces to meet their needs.

At the same time front end is also supported by mobile application for on-site user. The overall application is supported by backend apps like statistical analysis and deviation assessment. These apps will implement the statistical methods used in the construction industry based on the defined rules and knowledge of the involved stake holders. At the same time deviation assessment application will be like a decision support tool that will help the supervisors to take further actions based on the results of the test procedures.

5.2.1 Job Production Rate

As explained, FITMAN Portugal Trial has various applications, from web application to statistical analysis and deviation assessment applications. Thus, the case study has very heterogeneous and dynamic workloads. Moreover, the job arrival is not dependent on workload history and the probability of job arrival is independent of the time. Hence, one possible way to model this type of workload is through Markov process [220][221]. Simulation model considers m/g/n queue for modelling the inter-cloud environment with the Poisson process for the Job arrival. Poisson distribution considers a period of time T during which events occur at an average rate of λ events per time unit. Additionally, since the workload is heterogeneous and dynamic, the "ServiceTime" is considered with the general distribution.

5.3 Simulation Results

The three type of agents in our simulation process are explained in previous subsection: CSA, CPA, and Job. Each agent has a number of attributes and operations shown in Figure 5-1. In the simulation model, it is possible to initialize the number of CPAs and average job production. We consider three scenarios (shown in Figure 5-4, Figure 5-5, and Figure 5-6) that in all of them the Job agents are generated with the Poisson distribution with the average job production rate (λ = 37) and each job has a number of requirements with normal distribution for service time. In the diagrams shown in Figure 5-4, Figure 5-5, and Figure 5-6, there are two lines: a line showing the number of jobs that received computing resources and the other one showing the waiting queuing jobs. Moreover, there are three output boxes showing total number of produced jobs, total profit for CSA, and total response time to deliver the service to jobs. In all three scenarios, the CSA agent has a predefined SLA with specified service prices and penalty rules for not delivers the promised QoS and according to the SLA agreement between CSA and the user that submits jobs, if CSA cannot provide the promised QoS there will be a penalty cost. In all scenarios, the price for different services is a random value between +10% and -10% of Amazon EC2 Cloud's real price [59]. Following are our three scenarios:

5.3.1 Single cloud provider environment

In this scenario, there is a CSA agent as an IaaS cloud subscriber with limited number of resources. According to the SLA agreement between CSA and the user that submits jobs, if CSA cannot provide the promised QoS there will be a penalty cost. The price for different services is a random value between +10% and -10% of Amazon EC2 Cloud's real price [59]. Figure 5-4 shows the results for this scenario.

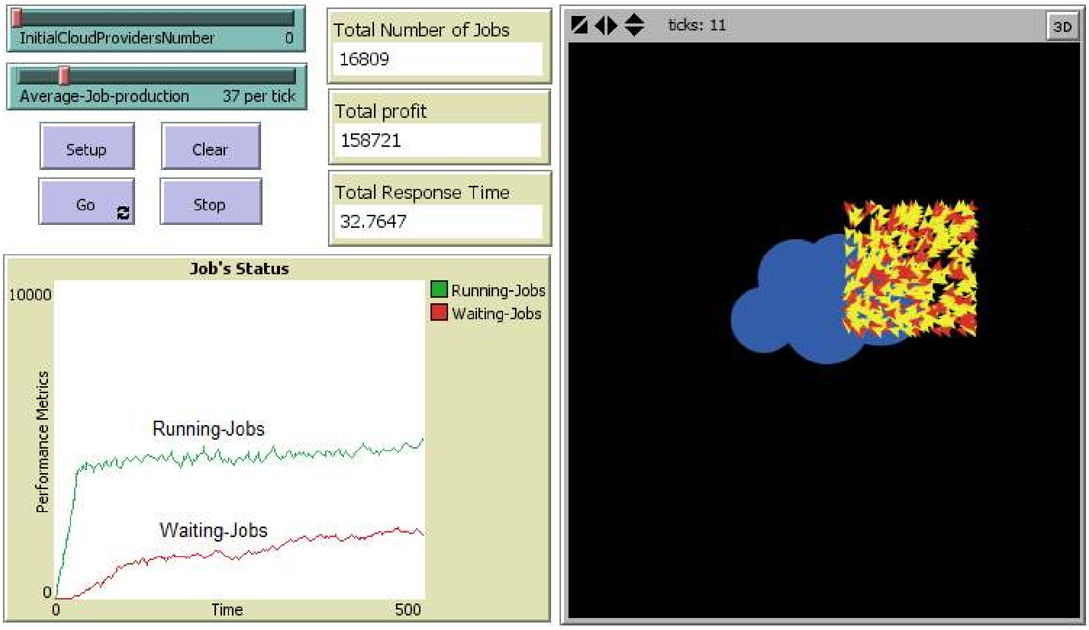


Figure 5-4 The simulation results for single cloud provider environment.

5.3.2 Multi-cloud provider environment without using GA based job schedule

In this scenario, there is a CSA agent as an IaaS CS with limited number of resources and there are four CPA agents as IaaS CPs. There is a SLA contract between CSA and each CPA that specifies the costs of each service with different assured QoS and the predefined fine as penalty for not delivering the promised QoS. Figure 5-5 shows the results for this scenario.

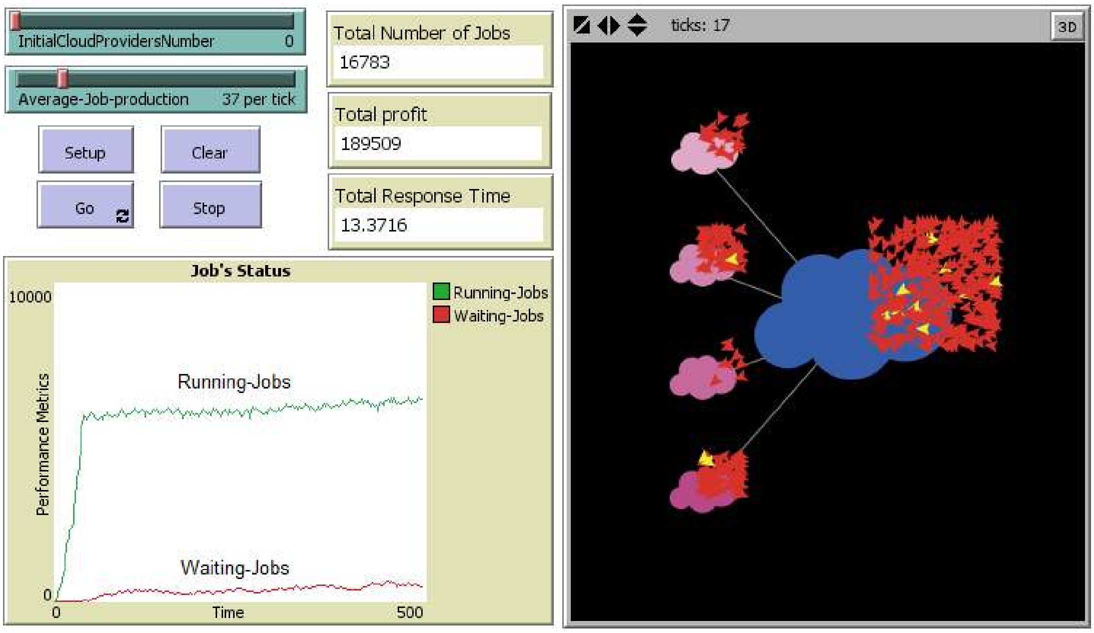


Figure 5-5 The results for multi-cloud provider environment without using GA-based job-scheduler.

5.3.3 Multi-cloud provider environment using GA based job scheduler

Similar to previous scenario, there is a CSA agent as an IaaS CS with limited number of resources and there are four CPA agents as IaaS CPs. There is a SLA contract between CSA and each CPA that specifies the costs of each service with different assured QoS and the predefined fine as penalty for not delivering the promised QoS. The Outsourcing-Job-Scheduler operation uses GA solution presented in presented in section 4.4 at chapter 4, to dispatch the jobs on the available CPAs. For this scenario, the GA solution considered $\text{crossover} - \text{rate} = 0.1$, $\text{mutation} - \text{rate} = 0.02$, the number of jobs for each step $x = 50$ and repeating steps $n = 20$. Figure 5-6 shows the results for this scenario.

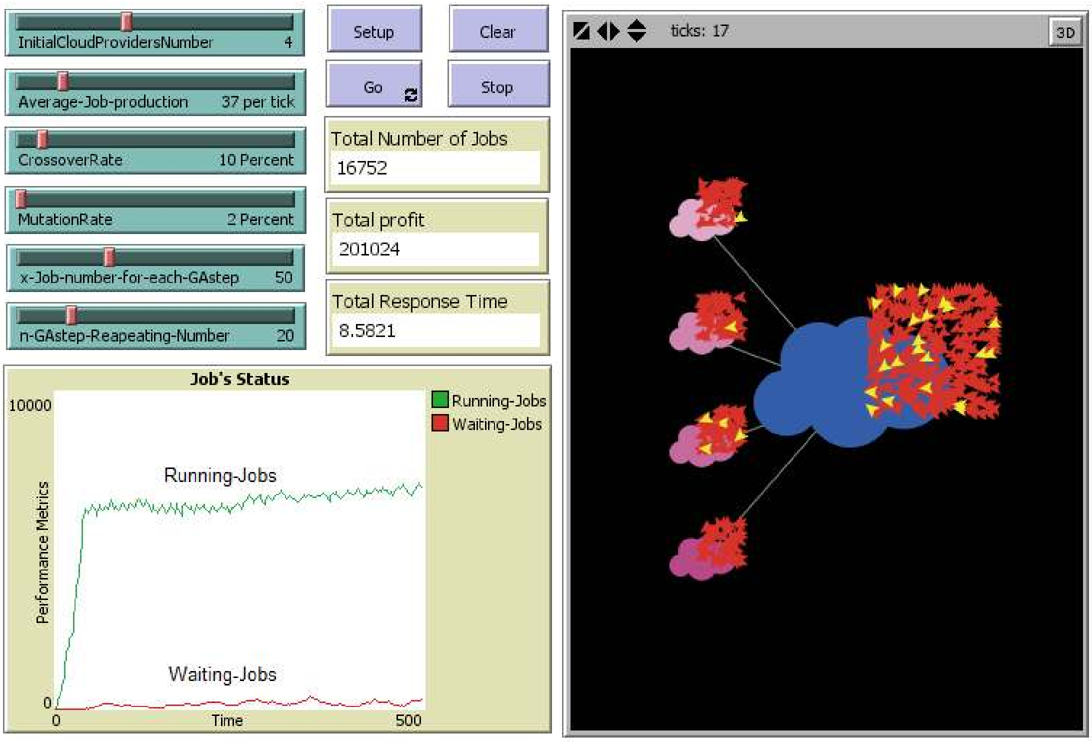


Figure 5-6 The results for multi-cloud provider environment using GA based solution

The simulation results for all scenarios are shown in Table 5-1. It can be seen that the total response time reduces 28.66% using Scenario 2 compared to Scenario 1. This implies that the Quality of Service is improving using the Multiple Cloud Scenario compare to the single Cloud Scenario. Additionally, total profit for the CS increases 2.34% using Scenario 2 compared to Scenario 1. Moreover, Table 5-1shows the total response time reduces 16.21% using Scenario 3 compared to Scenario 2 implying that the Quality of Service is improving using the Multiple Cloud with GA based solution compared to Multiple Cloud without GA based solution. Additionally, the total profit for the CS increases 14.19% using Scenario 3 compared to Scenario 2, thus the total cost for CS considerably reduced using ICIF with GA based solution.

Table 5-1 The simulation results for three scenarios.

Environment	Total Number of Jobs	Total Profit	Total Response Time
Single Cloud	16809	158721	32.7647
Multiple Cloud without GA based job scheduler	16783	189509	13.3716
Multiple Cloud with GA based job scheduler	16752	201024	8.5821

Chapter 6

Discussion and Final Consideration

6 Discussion and Final Consideration

This chapter summarises the PhD thesis work, providing an overview of "what is the problem and motivation?", "how this thesis deals with the problem and the contribution?", and "what are the considerations to develop the solution Finally, areas for further development and research are discussed.

6.1 What is the problem and motivation?

Cloud computing has emerged as a new and promising paradigm and includes managing heterogeneous clouds and delivering services over the Internet. Today, many small and large enterprises around the world have leveraged cloud computing services instead of traditional on-site alternatives. There are number of reasons discussed in the thesis why cloud services are universally used among different sectors today, such as the reduction of costs in using infrastructures and software applications, ability pay based on usage, and providing more flexibility and scalability.

The research on cloud computing is still at an early stage. The current growth of cloud and its considerable advantages are limited by challenges that exist in ongoing state. These issues can be obstacles to the growth of cloud computing for outsourcing applications from heterogeneous organizations. Present state of the art shows intercloud interoperability challenge is key to exponential growth of cloud computing.

The intercloud concept is based on the fact that each single cloud service provider has limited number of computing resources. Intercloud aims to support interoperability between heterogeneous cloud computing providers that can be in two levels:

1. Enabling cloud providers to collaborate together. The cloud collaboration can allow cloud service providers to deliver better quality of services, avoid data lock-in, and reduce scaling/producing costs.
2. Enabling cloud customers to migrate in and out of a particular cloud vendor and switch between providers based on their requirements grow or shrink, and move their data, applications or workloads around as their business demand change, without a lock-in.

However, most of current cloud systems are developed without interoperability concerns and available standards in cloud environment do not support inter-cloud interoperability and will take years to fully develop. Thus, more research work is required to provide sufficient functions to enable global seamless collaboration between cloud services. Hence, Intercloud Interoperability is selected as a general research topic for the PhD thesis.

There are many use cases defined for intercloud interoperability, such as Intercloud Interoperability for:

- Dispatching dynamic operations to IaaS cloud providers
- Copying data objects between cloud service providers
- Cloud bursting from data center to cloud service providers
- Migrating a queuing-based Application
- Migrating VMs from a cloud service provider to another

This PhD thesis selected the first mentioned use case, dispatching dynamic operation to IaaS cloud providers, as a research question. A solution for this use case should reinforce interoperability for IaaS service providers that should be able to allow an IaaS Cloud Subscriber (CS) to migrate the workload to the other selected IaaS Cloud Providers (CPs) through dispatching operations dynamically from CS to a available CPs.

The first chapter of this thesis states the current challenges in the area of cloud computing and emphasis on the "Intercloud Interoperability" problem, clarifies the research question and corresponding hypothesis which are addressed in the PhD thesis, and finally describes the adopted research method.

6.2 How thesis deals with the problem and the contribution

This PhD thesis proposed a novel framework called "InterCloud Interoperability Framework" or ICIF to support interoperability in a heterogeneous computing resource cloud environment. The main objective of ICIF is the ability to select the workloads independent from unique resources of the cloud subscriber and dynamically dispatching the operations to the most effective cloud providers available at runtime. The framework opens an account between IaaS Cloud Subscriber (CS) and each available IaaS Cloud Provider (CP) based on related Service Level Agreement (SLA) contract. The list of charges and QoS promises of each available CP is updated periodically. The ICIF considers a test workload, with specified CPU power, and memory or network performance requirements. The framework operates the test workload a few times on each CP, to arrange the CPs by availability, and performance and price aspects. To achieve these aims, various tasks as detailed below are done:

6.2.1 Studying the state of the art

The first step involves conducting a comprehensive literature review that would gather all research findings in multiple domains that are fundamental for the developing our IaaS intercloud solution. Relevant areas for literature reviews include:

- Cloud Computing
- IaaS Inter-cloud Interoperability
- Application development approaches like SOA and MDA
- Genetic Algorithm Systems
- Agent Based Simulation Model

6.2.2 Select the most appropriate approach to develop the interoperability framework architecture that can clarify semantic interoperability conflicts between IaaS-Cloud Subscriber and IaaS-Cloud Providers.

The thesis proposed a generic architecture for our framework that aims to resolve interoperability incompatibilities between heterogeneous cloud computing platforms. It is fundamental to adopt the most appropriate methods for developing the architecture of such framework. Through a literature review of different methodologies that have been applied to resolve various scenarios of interoperability, Model Driven Architecture (MDA) and SOA methods are selected as possible approaches to support Intercloud Interoperability.

- The Object Management Group (OMG) announced the MDA initiative as a software development approach to system-specification and interoperability based on the use of formal models. MDA focuses on the development of models rather than detailed, platform-specific code which can be generated when needed. Instead of requiring developers to define every detail of a system's implementation using a programming language, it lets them model what functionality is needed and what overall architecture the system should have. The MDA approach gives the facility to understand complex and real-world systems while providing an abstraction of the physical system. MDA specifies three level of modeling abstractions: Computation Independent Model, Platform Independent Model and Platform Specific Model. Transformation techniques play a key role in making MDA successful. Transformations can be categorized based on the type of source and destination they operate on. At top level, model transformation approaches can be identified as model-to-code transformations or model-to-model transformations.

- SOA is a new architectural style to develop applications through services. It is defined as a collection of independent services which communicate with each other. The communication can include a simple data passing or two or more services coordinating the same activity. The connection for exchanging request and subsequent response messages between service customer and provider are specified in an understandable way to both the service consumer and provider. SOA is a paradigm for solution architects to facilitate developing new value-added solutions by incorporating different solution artifacts such as business processes, services, packaged applications, and manageable attributes all over their lifecycle.

6.2.3 Developing appropriate process for selection of operations to migrate to other clouds

It is a process that analyses current state of the workload in Cloud Subscriber (CS) and evaluate the possibility of outsourcing operations on other IaaS Cloud Providers (CPs). It is fundamental to select operations for migration that are not dependent on a unique computing resource of CS. This research work assumed the workload is series of job operations specified with following requirements:

- Serving Time
- Maximum Response Time
- Computing Power Requirement
- Memory Requirement
- Minimum Network Bandwidth Requirement
- Priority (That is based on the service price and the SLA contract between CS and the application which requested computing resources)

A job can be selected if it is independent from a unique computing resource of CS and its Maximum Response Time is longer than network delay to allocate computing resource from other IaaS CPs. ICIF has a module called Model-Manager Module that provides the required details of the job. Each job is specified by data model, operation model, object model and set of requirements.

6.2.4 Developing appropriate processes for effective IaaS-CP discovery and selection

It is fundamental to provide enough functionality for IaaS CPs discovery. Furthermore, the IaaS-CP selection process is necessary to select appropriate IaaS CPs from available cloud providers. According to our study, distributing the operations in a cloud-based environment is a nondeterministic polynomial time (NP-complete) problem, a Genetic Algorithm (GA) based job scheduler proposed as a part of interoperability framework, offering workload migration with the best performance at the least cost.

This process considers the workload requirements and the SLA repository between IaaS-CS and IaaS-CPs. SLA repository represents an agreement between the IaaS CS and each IaaS CP. Each SLA defines recovery actions if agreed requirements cannot be satisfied. Moreover, QoS properties for each service of the cloud provider are provided by this repository which will be used for making the correct selection of the cloud provider based on the job requirements. The CS opens an account with each discovered IaaS CP based on CP's SLA. This process holds the list of charges and QoS promises of each CP. Moreover, the CS evaluates the CPs for the price and QoS characteristics such as availability, and forwards the workloads accordingly.

6.2.5 Developing appropriate processes for mapping dynamic workload from IaaS Cloud Subscriber to other selected IaaS Cloud Providers

The Transformation-Engine module of interoperability framework performs the necessary model transformation to map the "Job" details to "Job`". The necessary transformations can be made by applying the principles of MDA approach combined with a Semantic model of workload. This process is the key component of the framework to support interoperability through mapping workload from IaaS Cloud Subscriber to the selected IaaS Cloud Providers.

6.2.6 Developing a novel model for analyzing the interactions between IaaS-CS and IaaS-CPs to outsourcing the dynamic workload to them.

Interactions in Inter-Cloud Environment fall under the category of complex non-linear systems for which simple, intuitive, analytical solutions are not readily available Hence, this thesis developed an Agent Based Simulation (ABS) model to simulate an extendable Inter-Cloud environment that uses the proposed IaaS Inter-cloud Framework. ABS approach is a powerful modeling and simulation technique for a large variety of research topics and has advantages over conventional approaches in many cases. ABS can simulate a dynamic model in which agents interact repeatedly over the time to achieve an optimized solution. Agents in ABS represent actors, objects, or processes of a system that behave based on the interaction rules of the modeled system. Recent computer technology enables simulation of millions of such agents, which can be analysed to make scientific conclusions. The proposed an ABS approach includes three types of agents: Cloud Subscriber Agent (CSA), Cloud Provider

Agent (CPA), and Job agent. Each agent is defined with set of specify attributes and operations according their rule in our described InterCloud environment. Three scenarios are defined to run the ABS model: (1) Single Cloud, (2) Multiple Cloud without using proposed GA based job-scheduler, and (3) Multiple Cloud using proposed GA job-scheduler.

6.2.7 Select a case study and validate the proposed framework

GRIS group (Group from Research in Interoperability of Systems) is a research group from UNINOVA at Universidade Nova de Lisboa that contributes to various system interoperability research projects. It works with several enterprises to provide better solution. Hence, we selected one appropriate enterprise from GRIS partners as a case study called FITMAN[3]. The FITMAN Portugal trial addresses the development of projects related to construction industry with the goal of triggering the use of Future Internet technologies in the factories of the future. In this project, there are certain requirements that will be fulfilled by FI-WARE platform and other projects at GRIS research center.

Cloud Hosting is one fundamental layer of FI-WARE which provides the computation, storage and network resources, upon which services are provisioned and managed. It includes several Generic Enablers (GEs). The Job-Selection module of proposed InterCloud Interoperability Framework (ICIF) integrates the Job-Scheduler GE to select the job operations waiting to receive required computing resources. Only the operations that are independent of unique resources of IaaS CS can be selected to forward and execute on other IaaS CPs. The framework selects the most effective IaaS CPs, maps the job model accordingly, and dispatches the job to the selected CP. Finally ICIF collects the operation results from selected CP. All data and model transformation and mapping tasks between CS and CPs are happening through the ICIF.

6.3 The considerations to develop the proposed solution

Previous part discussed about the various tasks that have been done to develop a solution for IaaS intercloud interoperability issue. There are many challenges in developing a framework that supports migrating the operations from one cloud to another cloud. This part addresses a number of challenges and considerations during development process of the proposed solution in this thesis:

[3] Future Internet Technologies for MANufacturing industries

6.3.1 Converting the job operation requirements from Cloud Subscriber environment to the target Cloud Provider environment:

As mentioned before, the ICIF is an InterCloud Interoperability Framework that supports dynamic workload dispatch from Cloud Subscriber to the selected IaaS Cloud Providers. In this case, handling job operation movement to the target cloud is fundamental factor of successful interoperability framework.

The thesis proposed a four layer architecture framework. Two layers of it are collaborating to solve this problem: Semantic Layer, and MDA-SOA Layer. Semantic layer provides Application Model, Data Model and IaaS Cloud Offering Model. Semantics are used by the MDA-SOA Layer in order to provide the means for developing interoperability related mechanisms. Model-Manager Module of MDA-SOA Layer provides the required details of each job. Hence, each job can be specified by data model, operation model, object model and set of requirements. Afterward, Transformation-Engine Module of MDA-SOA Layer uses the basic principal of MDA approach. It performs the necessary model transformations to map the "Job" details obtained in Model-Manager Module to "Job`". Hence, these processes enable the framework to support interoperability through mapping workload from IaaS CS to other selected IaaS CPs.

6.3.2 The effective method for ICIF to use the QoS-SLA Agreements

The proposed ICIF should consider QoS criteria and Service level agreements (SLAs) as a contract negotiated and agreed between: a. Cloud Subscriber (CS) and the consumer, b. CS and other Cloud Providers (CPs). The presented model includes a "QoS-SLAs-Repository Module" that represents an agreement between the IaaS CS and each IaaS CP. Each SLA defines recovery actions if agreed requirements cannot be satisfied. Moreover, QoS properties for each service of the cloud provider are provided by the repository which is used for making the correct selection of the cloud provider based on the job requirements. The CS opens an account with each discovered IaaS CP based on CP's SLA. QoS-SLAs-Repository module holds the list of charges and QoS promises of each CP. Then the CS considers a test workload, with specified CPU power, memory and network performance requirements. The CS operates the test workload a few times on each CP, to arrange the CPs by availability, and performance and price aspects. Moreover, the CS evaluates the CPs for the price and QoS characteristics such as availability, and forwards the workloads accordingly.

Numerous cloud services with different pricing and Quality of Services (QoS) exist in an intercloud environment which makes it complicated to select the best composition of services based on consumer requirements. Suitable SLA-QoS characteristics are presented as:

availability, reliability, performance, security, scalability, data communication cost, capacity, and latency parameters for IaaS cloud service. However, considering and measuring all parameters separately were not feasible. In this thesis, the developed Genetic Algorithm Based Job-Scheduler considers a mixture of QoS-SLA parameters for effective IaaS-CP discovery and selection:

- The Performance of each IaaS CP: The framework allocates a performance history variable to each IaaS Cloud Provider. ICIF framework sends a test workload to each Cloud Provider periodically and updates the performance variable according to the resource availability and response time of each Cloud Provider. This variable is the average of previous value with factor of (m-1) and the last resource availability and response time. The GA solution maximizes the sum of the history variable for all available CPs.
- The Cost: The SLA repository based on the agreement between CS and CPs that includes the price lists for different computing resource offering. The proposed GA solution minimizes the overall cost.

In our GA based solution, the fitness function is defined as sum of performance history variable divided by sum of overall cost.

6.3.3 How the proposed Agent based Simulation Model demonstrates the InterCloud environment assumed during the thesis

As described before, this thesis developed an Agent Based Simulation (ABS) model to simulate the Inter-Cloud environment that uses the proposed IaaS InterCloud Framework. There are a number of assumptions for our intercloud environment that should be demonstrated using the ABS model:

6.3.3.1 *Cloud Subscriber*

IaaS CSA agent is defined to represent IaaS Cloud Subscriber. This agent is a cloud computing resource provider that has limited number of the computing resources, hence, requires interoperating with the IaaS Cloud Provider Agents (IaaS CPAs) to provide better QoS for the users. IaaS CSA is based on the functionality of the ICIF to dispatch the operations on the most appropriate IaaS CPAs available based on the Job Agents' requirements. The IaaS CSA opens an account with the available IaaS CPAs based on related SLA contract. The list of charges and QoS promises of each available IaaS CPA has to be updated periodically. The ICIF considers a test workload, with specified CPU power, and memory or network performance requirements. The CSA sends the test workload a few times on each CPA, to arrange the CPAs by availability, and performance and price aspects. Our simulation model assumed CSA agent has number of variables to show the amount of available resources and their prices, and other SLA-QoS policies between the CS and customers that provided Jobs. It also has a number of variables to calculate the penalty costs, profits, number of running jobs, and other Cloud Providers. Additionally, this agent has a number of operations for: setting up the CS, Job selection process, IaaS resource discovery, IaaS resource selection, and outsourcing Genetic Algorithm based Job scheduler.

6.3.3.2 *Cloud Providers*

IaaS CPA agent is defined to represent IaaS Cloud Providers. In the ABS model, there are predefined numbers of CPA agents. The purpose is cooperation between CSA and CPAs to provide services with higher QoS and lower price to the consumers of CSA. The interoperation between CSA and each CPA is possible through the functionality supported by ICIF. There is a SLA contract between each CPA and CSA that gets updated periodically. A CPA provides computing resources based on the service request from CSA. There are a number of variables associated to each CPA to specify different service combinations and prices. Additionally, each CPA has a number of performance variables. The CSA sends workload test periodically to all available and updates the performance variables.

6.3.3.3 *Job operations*

An agent called "Job" is defined to represent the dynamic workload in Cloud Subscriber. It is based on the requirements of applications and has set of variables {“ServiceTime”, "MaxWaitingTime”, “CPURequirements”, “RAMRequirements”, “BDRequirements”, “Dependency”, “Priority”, “DelayPenalty” and “Cost”}. It also has an operation called setJob() to show the job received the required computing resources or not. In each time slot a number of Job agents are produced and waiting to get the resources.

6.3.4 The security concerns

In this use case, the interoperability is in job operation level. Each operation is the smallest sequence of programmed instructions that can be managed independently. It means the result of execution of each job is part of bigger program and does not have a significant meaning by itself. Therefore, we considered the security cannot be an issue in this level of intercloud interoperability.

6.3.5 Analysis

The fifth chapter of this thesis is validating the proposed framework. The proposed ICIF integrates the Job-Scheduler GE from FI-WARE Platform that is exploited by the FITMAN Portugal trial. The aim is supporting interoperability between FI-WARE Platform Cloud and CPs through ICIF to improve the performance of FITMAN.

The Job-Selection module of ICIF integrates the Job-Scheduler Generic Enabler (GE) to select the job operations waiting to receive required computing resources. Only the operations that are independent of unique resources of IaaS CS can be selected to forward and execute on other IaaS CPs. The framework selects the most effective IaaS CPs, maps the job model accordingly, and dispatches the job to the selected CP. Finally ICIF collects the operation results from selected CP. All data and model transformation and mapping tasks between CS and CPs are happening through the ICIF.

Running the ABS simulation model with different values for GA solution factors results in more appropriate values for: crossover-rate, mutation-rate, number of jobs for each step x, and repeating steps n in GA based job-scheduler solution. It can be used to predict the more suitable values for the GA solution factors in different case studies. The thesis considered three scenarios during the simulation process:

1. Single cloud provider environment: In this scenario, there is a CSA agent as an IaaS cloud subscriber with limited number of resources.
2. Multi-cloud provider environment without using GA based job scheduler: In this scenario, there is a CSA agent as an IaaS CS with limited number of resources and there are four CPA agents as IaaS CPs.
3. Multi-cloud provider environment using GA based job scheduler: Similar to previous scenario. The only difference is considering the Genetic Algorithm outsourcing job scheduler during the simulation process. For this scenario, the GA solution considered crossover-rate=0.1, mutation-rate=0.02, the number of jobs for each step x=50 and repeating steps n=20.

The simulation results show that the response time improves using Scenario 2 compared to Scenario 1 and improves further in Scenario 3. Additionally, the simulation results imply that the overall profit for CS increases 2.34% using Scenario 2 compared to Scenario 1 and increases 14.19% using Scenario 3 compared to Scenario 2, thus the total cost for CS considerably reduces using ICIF with GA based solution.

6.4 Areas for Further Development and Research

Based on our research, it can be predicted that InterCloud Interoperability will become a determinant of success. This research work focused on InterCloud Interoperability for dynamic dispatching operations to IaaS cloud providers use case. For future work we can:

- Apply the proposed Agent Base Simulation model in the business scenarios to deduce which cloud vendors are appropriate to interoperate with.
- Improve the Genetic Algorithm (GA) based outsourcing job-scheduler by adding more factors to the fitness function.
- Study the impact of various workloads on them and adjust their values accordingly By applying different values for the properties of GA solution.
- Adjust the values of the GA solution's factors, by applying different values for these factors, and studying the impact of various workloads on them.
- Propose an intercloud framework which supports inter-cloud interoperability in a computing resource cloud environment with the goal of workload migration using Virtual-Machine (VM) migration. The workload can be specified as VMs with various requirements, like: {Required Virtual CPUs, required memory, a unique IP address, the Domain Name System resolver configuration, the list of virtual network interfaces, the subnet mask and identifier for each subnet attached to the VM, the MAC address assigned to the VM, the list of virtual block devices the VM assumes, the list of attached storage devices, minimum required network bandwidth}. The framework has to seamlessly migrates a stopped VM from Cloud-Subscriber (CS) to the most effective Cloud Providers available. It is necessary to analyse and address the challenges about adaptive VM migration.
- Develop an intercloud framework for migrating data between CS and CPs. Ensuring data security, managing data movement and encryption to the target cloud, and data synchronization are a few number of important challenges for this use case.

References

[1] R. Jardim-Goncalves, K. Popplewell, and A. Grilo, "Sustainable interoperability: The future of Internet based industrial enterprises," *Comput. Ind.*, vol. 63, no. 8, pp. 731–738, 2012.

[2] "FI-WARE : FUTURE INTERNET Core Platform," *Funded by: Seventh Framework Programme (FP7) and European Commission*. [Online]. Available: http://www.fi-ware.eu/.

[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *J. Internet Serv. Appl.*, pp. 7–18, Apr. 2010.

[4] V. Delgado, "Exploring the limits of cloud computing," KTH - Royal Institute of Technology, 2010.

[5] T. Dillon, C. Wu, and E. Chang, "Cloud Computing: Issues and Challenges," *24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010.

[6] L. LaManna, "Top 9 Challenges in Cloud Computing," 2012. [Online]. Available: http://blogs.sap.com/innovation/cloud-computing/top-9-challenges-in-cloud-computing-that-are-slowing-its-adoption-011918.

[7] Jack Rosenblum, "Top Five Challenges Of Cloud Computing," *Business, Cloud Computing, Computing, IT, Lists, News, SaaS, Storage, Technology, Trends*. .

[8] T. Rings, J. Grabowski, and S. Schulz, "On the Standardization of a Testing Framework for Application Deployment on Grid and Cloud Infrastructures," *2010 Second Int. Conf. Adv. Syst. Test. Valid. Lifecycle*, pp. 99–107, Aug. 2010.

[9] N. Borenstein and J. B. Mimecast, "Cloud Computing Standards," *IEEE Internet Comput.*, 2011.

[10] S. O. Jr, A. Luis, and D. Moreover, "The Problem with Cloud-Computing Standardization," *IEEE Comput. Soc.*, no. July, pp. 13–16, 2011.

[11] P. Mell and T. Grance, "Perspectives on Cloud Computing and Standards," *Natl. Inst. Stand. Technol.*, 2009.

[12] F. Feldhaus, "Cloud Interoperability."

[13] Jr Robert H. Carpenter, "Walking from Cloud to Cloud: The Portability Issue in Cloud Computing," *Washingt. J. Law, Technol. Arts*, pp. 1–14, 2010.

[14] T. J. Mühlbauer, "Formal Specification and Analysis of Cloud Computing Management," 2012.

[15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds : A Berkeley View of Cloud Computing," 2009.

[16] "Cloud Security Alliance, Top Threats to Cloud Computing V1.0," *Underwrit. by HP*, no. March, pp. 1–14, 2010.

[17] H. Huang and L. Wang, “P & P : a Combined Push-Pull Model for Resource Monitoring in Cloud Computing Environment,” *IEEE 3rd Int. Conf. Cloud Comput.*, 2010.

[18] Q. Li and Y. Guo, “Optimization of Resource Scheduling in Cloud Computing,” *2010 12th Int. Symp. Symb. Numer. Algorithms Sci. Comput. IEEE Comput. Soc.*, no. 1, pp. 315–320, Sep. 2010.

[19] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, “Dynamically scaling applications in the cloud,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 45–52, 2011.

[20] M. Kretzschmar and S. Hanigk, “Security management interoperability challenges for collaborative clouds,” *4th Int. DMTF Acad. AllianceWorkshop Syst. VirtualizationManagement, IEEE*, pp. 43–49, 2010.

[21] K. Hamlen, M. Kantarcioglu, L. Khan, and B. Thuraisingham, “Security Issues for Cloud Computing,” *Int. J. Inf. Secur. Priv.*, vol. 4, no. 2, pp. 36–48, 2010.

[22] M. Kantarcioglu, A. Bensoussan, and S. R. (Celine) Hoe, “Impact of security risks on cloud computing adoption,” *2011 49th Annu. Allert. Conf. Commun. Control. Comput.*, pp. 670–674, Sep. 2011.

[23] IEEE, *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. 1990.

[24] J. O’Brien and G. Marakas, *Introduction to Information Systems*, 13 edition. McGraw-Hill/Irwin, 2005.

[25] E. Wilson, “Consilience: The Unity of Knowledge,” *New York Vintage. ISBN 0-679-76867-X*, 1999.

[26] C. G. Morris, “Academic Press Dictionary of Science and Technology,” *Gulf Prof. Publ. 1992*.

[27] G. Dodig-Crnkovic, “Scientic methods in computer science,” *Proc. Conf. Promot. Res. IT New Univ. Univ. Coll. Sweden, 2002*.

[28] D.-M. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” *Comput. Networks ISDN Syst.*, 1989.

[29] H. Fan, X. Shan, J. Yuan, and Y. Ren, “Covariances of Linear Stochastic Differential Equations for Analyzing Computer Networks,” *Tsinghua Sci. Technol.*, vol. 16, no. 3, pp. 264–271, Jun. 2011.

[30] M. Dräxler, “Computer Networks – Important Formulas,” 2008.

[31] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, 2005.

[32] OMG, “MDA Guide Version 1.0.1,” no. June, 2003.

[33] OMG, A. B. Ormsc, C. Burt, D. Dsouza, K. Duddy, W. El Kaim, W. Frank, S. Iyengar, J. Miller, J. Mischkinsky, J. Mukerji, J. Siegel, R. Soley, S. Tyndal-, A. Uhl, A. Watson, and B. Wood, “Model Driven Architecture (MDA) Document number ormsc / 2001-07-01,” *Architecture*, pp. 1–31, 2001.

[34] Merriam-Webster, "Scientific Method," *www.merriamwebster.com/dictionary/scientific%20method*, 2010.

[35] F. L. H. Wolfs, "Introduction to the Scientific Method," *teacher.pas.rochester.edu/phy_labs/appendixe/appendixe.html*, 2010.

[36] A. Carpi and A. Egger, "CHED 392-Teaching scientific process and the nature of science," *Proc. 236th Natl. Meet. Am. 2008*.

[37] S. D. Schafersman, "Scientific Thinking and the Scientific Method," *www.freeinquiry.com/intro-to-sci.html*, 2010.

[38] A. Carpi and A. Egger, "The Process of Science," *www.visionlearning.com/process_science.php*, 2010.

[39] J. Randolph, G. Julnes, E. Sutinen, and S. Lehman, "A Methodological Review of Computer Science Education Research," *J. Inf. Technol. Educ.*, vol. 7, 2008.

[40] L. M. Camarinha-Matos, "Scientific Research Methodologies and Techniques – slides from Unit 2 (Scientific Method)," *www.uninova.pt/~cam/teaching/srmt.htm*, 2010.

[41] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Version 15," *Natl. Inst. Stand. Technol.*, 2009.

[42] R. K. Chellappa, "Intermediaries in Cloud-Computing: A new Computing Paradigm.," *Present. INFORMS Meet. Dallas*, 1997.

[43] T. Grance, "The NIST Cloud Definition Framework," *Natl. Inst. Stand. Technol.*, 2010.

[44] CSA, "Security Guidance for Critical Areas of Focus in Cloud Computing V3.0," *Cloud Secur. Alliance*, pp. 0–176, 2011.

[45] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," *2009 Fourth Int. Conf. Internet Web Appl. Serv.*, no. Mdi, pp. 328–336, 2009.

[46] D. C. Plummer, D. M. Smith, T. J. Bittman, D. W. Cearley, D. J. Cappuccio, D. Scott, R. Kumar, and B. Robertson, "Five Refining Attributes of Public and Private Cloud Computing," *gartner*, 2009.

[47] J. Geelan, "Twenty one experts define cloud computing," *Cloud Comput. J.*, pp. 1–5, 2009.

[48] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," *10th IEEE Conf. High Perfomance Comput. Commun. IEEE*, 2008.

[49] D. Catteddu and G. Hogben, "Cloud Computing: Benefits, risks and recomendations for information security," 2009.

[50] James Staten, "Is cloud computing ready for the enterprise?," *Forrester Res.*, 2008.

[51] D. C. Plummer, T. J. Bittman, T. Austin, D. W. Cearley, and D. M. Smith, "Cloud Computing : Defining and Describing an Emerging Phenomenon," *Gartner, Inc.*, 2008.

[52] LizheWang and G. von Laszewski, “Scientific cloud computing: Early definition and experience,” *10th IEEE Conf. High Perform. Comput. Commun. Dalian, IEEE*, 2008.

[53] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A Break in the Clouds : Towards a Cloud Definition,” *ACM SIGCOMM Comput. Commun. Rev.*, 2009.

[54] “Open cloud manifesto,” 2009.

[55] “XenSource Inc, Xen.” [Online]. Available: http://www.citrix.com/products/xenserver/overview.html.

[56] “Kernal Based Virtual Machine (KVM).” [Online]. Available: http://www.linux-kvm.org/page/MainPage.

[57] “VMWare ESX Server.” [Online]. Available: http://www.vmware.com/.

[58] GSA, “Infrastructure as a Service (IaaS),” *General Services Administration*, 2012. [Online]. Available: http://www.gsa.gov/portal/content/112063.

[59] Amazon, “Amazon Elastic Compute Cloud (Amazon EC2).” [Online]. Available: http://aws.amazon.com/ec2/.

[60] Microsoft, “Windows Azure, a rock-solid cloud platform for blue-sky thinking.” [Online]. Available: http://www.windowsazure.com/en-us/.

[61] Rackspace, “Rackspace Cloud.” [Online]. Available: http://www.rackspace.com/cloud/.

[62] A. Folch, “Interface development for Eucalyptus based cloud,” Vilnius Gediminas Technical University, 2011.

[63] Google, “Google Compute Engine.” [Online]. Available: https://cloud.google.com/products/compute-engine.

[64] Amazon, “Amazon Web Services (AWS) Elastic Beanstalk.” [Online]. Available: http://aws.amazon.com/.

[65] Google, “Google Apps : It provids independently customizable versions of several Google products under a custom domain name.,” 2011. [Online]. Available: http://www.google.com/apps/index1.html.

[66] and employees. Salesforce.com: An enterprise cloud computing leader that offers social and mobile cloud technologies that helps companies connect with customers, partners, “Salesforce.” [Online]. Available: http://www.salesforce.com/eu.

[67] M. Ahronovitz, D. Amrhein, P. Anderson, A. de Andrade, J. Armstrong, E. A. B, J. Bartlett, R. Bruklis, K. Cameron, M. Carlson, R. Cohen, T. M. Crawford, V. Deolaliker, P. Downing, A. Easton, R. Flores, G. Fourcade, T. Freund, T. Hanan, V. Herrington, B. Hosseinzadeh, S. Hughes, W. J. Huie, N. Q. Hung, P. Isom, S. R. J, S. Johnston, R. Kulkarni, A. Kunjunny, E. Lau, T. Lukasik, B. Marcus, G. Mazzaferro, C. McClanahan, M. Medley, W. Melo, A. Monroy-Hernandez, A. Nassar, D. Nicol, L. Noon, S. Padhy, G. Parann-Nissany, G. Pfister, T. Plunkett, L. Qian, B. Ramachandran, J. Reed, G. Retana, B. P. Rimal, D. Russell, M. F. Rutkowski, C. Sanford, K. Sankar, A. O. Sanz, M. B. Sigler, W. Sinclair, E. Sliman, P. Stingley, P. Straton, R. Syputa, R. J. Taylor, D. Tidwell, K. Walker, K. Williams, J. M. Willis, Y. Sasaki, M. Vesace, E. Windisch, P. Yara, and F. Zappert, “Cloud Computing Use Cases White Paper, Version 4.0,” 2010.

[68] G. Caryer, T. Rings, J. Gallop, S. Schulz, J. Grabowski, I. Stokes-Rees, and T. Kovacikova, “Grid/cloud computing interoperability, standardization and the Next Generation Network (NGN),” *2009 13th Int. Conf. Intell. Next Gener. Networks*, pp. 1–6, Oct. 2009.

[69] M. Dikaiakos and D. Katsaros, “Cloud computing: Distributed Internet computing for IT and scientific research,” *IEEE Comput. Soc.*, Jan. 2009.

[70] G. Shroff, *Enterprise Cloud Computing Technology, Architecture, Applications*. 2010.

[71] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li, “Comparison of Several Cloud Computing Platforms,” *2009 Second Int. Symp. Inf. Sci. Eng.*, pp. 23–27, Dec. 2009.

[72] P. Fan, J. Wang, Z. Zheng, and M. R. Lyu, “Toward Optimal Deployment of Communication-Intensive Cloud Applications,” *2011 IEEE 4th Int. Conf. Cloud Comput.*, pp. 460–467, Jul. 2011.

[73] S. S. A. Ahson and M. Ilyas, *Cloud computing and software services: Theory and techniques*. CRC Press, Inc., 2010.

[74] D. Bernstein, “The Intercloud : Cloud Interoperability at Internet Scale,” *Sixth IFIP Int. Conf. Netw. Parallel Comput.*, 2009.

[75] A. Parameswaran and A. Chaddha, “Cloud interoperability and standardization,” *SETlabs briefings*, 2009.

[76] N. Grozev and R. Buyya, “InterCloud architectures and application brokering: taxonomy and survey,” *Softw. Pract. Exp.*, 2012.

[77] L. Badger, R. Bohn, R. Chandramouli, T. Grance, T. Karygiannis, R. Patt-Corner, and J. Voas, “Cloud Computing Use Cases,” *National Institute of Standards and Technology,*, 2010. [Online]. Available: http://www.nist.gov/itl/cloud/use-cases.cfm.

[78] G. A. Lewis, “The Role of Standards in Cloud- Computing Interoperability,” 2012.

[79] D. Bernstein, S. Clara, and N. Court, “Using Semantic Web Ontology for Intercloud Directories and Exchanges,” *Int. Conf. Internet Comput. ICOMP’10*, 2010.

[80] “IEEE P2302 Working Group (Intercloud).” [Online]. Available: http://grouper.ieee.org/groups/2302/.

[81] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, “How to Enhance Cloud Architectures to Enable Cross-Federation,” *IEEE 3rd Int. Conf. Cloud Comput.*, Jul. 2010.

[82] GICTF, “Intercloud Interface Specification Draft (Intercloud Protocol),” *Glob. Inter-Cloud Technol. Forum, Whitepaper*, 2012.

[83] GICTF, “Use Cases and Functional Requirements for Inter-Cloud Computing,” *Glob. Inter-Cloud Technol. Forum, Whitepaper*, 2010.

[84] GICTF, “Technical Requirements for Supporting the Intercloud Networking,” *Glob. Inter-Cloud Technol. Forum, WhitePaper*, 2012.

[85] GICTF, “Intercloud Interface Specification Draft (Cloud Resource Data Model),” *Glob. Inter-Cloud Technol. Forum, Whitepaper*, 2012.

[86] D. Bernstein and D. Vij, “Using XMPP as a transport in Intercloud Protocols,” *2nd USENIX Work. Hot Top. Cloud Comput.*, 2010.

[87] V. S. K. Nagireddi and S. Mishra, “An ontology based cloud service generic search engine,” in *2013 8th International Conference on Computer Science & Education (ICCSE)*, 2013, pp. 335–340.

[88] A. Abouzamazem and P. Ezhilchelvan, “Efficient Inter-cloud Replication for High-Availability Services,” in *IEEE International Conference on Cloud Engineering (IC2E)*, 2013, pp. 132–139.

[89] F. Pop, V. Cristea, N. Bessis, and S. Sotiriadis, “Reputation Guided Genetic Scheduling Algorithm for Independent Tasks in Inter-clouds Environments,” in *27th International Conference on Advanced Information Networking and Applications Workshops*, 2013, pp. 772–776.

[90] Y. Demchenko, C. Ngo, C. de Laat, J. AntoniGarcia-Espin, S. Figuerola, J. Rodriguez, L. M. Contreras, G. Landi, and N. Ciulli, “Intercloud Architecture Framework for Heterogeneous Cloud Based Infrastructure Services Provisioning On-Demand,” *27th Int. Conf. Adv. Inf. Netw. Appl. Work.*, pp. 777–784, Mar. 2013.

[91] C. Coutinho, A. Cretan, and R. Jardim-Goncalves, “Cloud-based negotiation for sustainable enterprise interoperability,” *2012 18th Int. ICE Conf. Eng. Technol. Innov.*, pp. 1–10, Jun. 2012.

[92] CloudStandards, “Cloud Standards Customer Council,” 2013. [Online]. Available: http://cloud-standards.org/wiki/index.php?title=Main_Page.

[93] D. Bernstein and D. Vij, “Intercloud directory and exchange protocol detail using XMPP and RDF,” *CBSA Cloud-based Serv. Appl. Work.*, 2010.

[94] D. Bernstein, D. Vij, and S. Diamond, “An Intercloud Cloud Computing Economy - Technology, Governance, and Market Blueprints,” *2011 Annu. SRII Glob. Conf.*, pp. 293–299, Mar. 2011.

[95] P. DE ALMEIDA, “MDA – Model Driven Architecture : Improving Software Development Productivity in Large-Scale Enterprise Applications,” University of Fribourg, Switzerland, 2008.

[96] OMG, “Unified Modeling Language Specification Version 2.1.1,” 2007.

[97] OMG, “Meta Object Facility (MOF) Core Specification,” 2006.

[98] OMG, “Common Warehouse Metamodel (CWM) Specification,” 2001.

[99] OMG, “XML Metadata Interchange Specification Version 2.1,” 2005.

[100] OMG, “Object Management Group’s General Glossary,” 2012.

[101] F. Chitforoush, M. Yazdandoost, and R. Ramsin, “Methodology Support for the Model Driven Architecture,” *14th Asia-Pacific Softw. Eng. Conf.*, pp. 454–461, Dec. 2007.

[102] D. S. Frankel, “Model Driven Architecture Applying MDA to Enterprise Computing,” 2003.

[103] N. Koch, “Classification of model transformation techniques used in UML-based Web engineering,” *Software, IET*, 2007.

[104] A. Jilani, M. Usman, and Z. Halim, "Model Transformations in Model Driven Architecture," *UniCSE - Univers. J. Comput. Sci. Eng. Technol.*, 2010.

[105] A. M. Jimenez, "Change propagation in the MDA: A model merging approach," The University of Queensland, 2005.

[106] M. Kay, "Xsl transformations (xslt) version 2.0," *W3C Recomm.*, 2007.

[107] Eclipse, "JET tutorial part 1 (introduction to jet)," 2004. [Online]. Available: http://www.eclipse.org/articles/Article-JET/jet_tutorial1.html.

[108] Eclipse, "Jet tutorial part 2 (write code that writes code)," 2004.

[109] OMG, "MOF 2.0/XMI Mapping, Version 2.1.1," 2007.

[110] OMG, "Human-Usable Textual Notation (HUTN) Specification," 2004.

[111] K. Czarnecki and S. Helsen, "Classification of Model Transformation Approaches," *Proc. 2nd OOPSLA Work. Gener. Tech. Context Model Driven Archit.*, pp. 1–16, 2003.

[112] "Jamda: The Java Model Driven Architecture 0.2," 2003.

[113] "AndroMDA 3.4," 2012.

[114] "Compuware Brings the Power of Model-Driven Development to Eclipse," 2006. [Online]. Available: http://www.compuware.com/.

[115] L. Grunske and L. Geiger, "Using graph transformation for practical model driven software engineering," *Model. Softw. Dev.*, 2005.

[116] K. Czarnecki and S. Helsen, "Feature-based survey of model transformation approaches," *IBM Syst. J.*, 2006.

[117] L. A. Rahim, S. Bahiyah, R. Syed, and U. T. Petronas, "Proposed Design Notation for Model Transformation," *19th Aust. Conf. Softw. Eng.*, 2008.

[118] OMG, "Meta Object Facility (MOF) 2 . 0 Query / View / Transformation Specification," 2011.

[119] D. van der Meij, "A Metamodeling Approach to Incremental Model Changes," University of Twente, 2009.

[120] "ATL : Atlas Transformation Language ATL Starter's Guide," no. December, ATLAS group LINA & INRIA Nantes, 2005.

[121] H. Z. Zhenjiang Hu, Hong Mei, Hui Song, Masato Takeichi, Yingfei Xiong, "Beanbag: A Language for Operation-based Heterogeneous Data Synchronization." [Online]. Available: http://code.google.com/p/synclib/.

[122] E. Foundation, "Epsilon Transformation Language." [Online]. Available: http://www.eclipse.org/epsilon/.

[123] A. Agrawal, "GReAT : A Metamodel Based Model Transformation Language."

[124] A. Agrawal, G. Karsai, and F. Shi, "Graph Transformations on Domain-Specific Models," 2003.

[125] A. Agrawal, A. Vizhanyo, Z. Kalmar, F. Shi, A. Narayanan, and G. Karsai, "Reusable Idioms and Patterns in Graph Transformation Languages," *Electron. Notes Theor. Comput. Sci.*, Mar. 2005.

[126] A. P. Antonio Cicchetti, Davide Di Ruscio, Romina Eramo, "JTL - JANUS TRANSFORMATION LANGUAGE," 2010. [Online]. Available: http://jtl.di.univaq.it/index.php.

[127] A. Cicchetti, D. Di Ruscio, R. Eramo, and A. Pierantonio, "JTL : a bidirectional and change propagating transformation language," *3rd Int. Conf. Softw. Lang. Eng.*, 2010.

[128] P. Muller, F. Fleurey, and J. Jézéquel, "Weaving Executability into," *Model Transform. Pract. Work. Montego Bay, Jamaica*, 2005.

[129] G. B. Brice Morin, Frank Fleurey, Nelly Bencomo, Jean-Marc Jézéquel, Arnor Solberg, Vegard Dehlen, "An Aspect-Oriented and Model-Driven Approach for Managing Dynamic Variability," *Proc. ACM/IEEE 11th Int. Conf. Model Driven Eng. Lang. Syst. (MoDELS 08), Toulouse, Fr.*, 2008.

[130] T. Marc Pantel, ACADIE-team, OLC-team, "The TOPCASED project - a Toolkit in OPen source for Critical Applications & SystEms Design," *TOOLS Eur. / Model. Dev. Tool Implementers Forum (MDD-TIF), Zurich, Switz.*, 2007.

[131] D. V. Zaid Altahat, Tzilla Elrad, "Using Aspect Oriented Modeling to localize implementation of executable models," *Model. Asp. Work. ECOOP*, 2007.

[132] Triskell team, "Kermeta - Breathe life into your metamode." [Online]. Available: http://www.kermeta.org/.

[133] S. R. Janis Barzdins, Audris Kalnins, Karlis Podnieks, Edgars Rencis, "The Lx transformation language," 2007. [Online]. Available: http://lx.mii.lu.lv/.

[134] Eclipse, "Eclipse Model-to-Model Transformation (M2M)," 2006. .

[135] OMG, "MOF Model to Text Transformation," 2008.

[136] "MOLA - MOdel transformation LAnguage, Research Laboratory of Modeling and Software Technologies of Institute of Mathematics and Computer Science University of Latvia," 2004. [Online]. Available: http://mola.mii.lu.lv/.

[137] A. Kalnins, J. Barzdins, and E. Celms, "Basics of Model Transformation Language MOLA," *ECOOP 2004 (Workshop Model Transform. Exec. Context MDA) , Oslo, Norway,*, 2004.

[138] A. Kalnins, J. Barzdins, and E. Celms, "Model Transformation Language MOLA : Extended Patterns," *Proc. Balt. DB&IS, Riga, Latv.*, 2004.

[139] A. S. Audris Kalnins, Edgars Celms, "Tool support for MOLA," *Fourth Int. Conf. Gener. Program. Compon. Eng. (GPCE'05). (Workshop Graph Model Transform. , Tallinn, Est.*, pp. 1–12, 2005.

[140] A. Kalnins, E. Celms, and A. Sostaks, "Model Transformation Approach Based on MOLA 2 Brief Description of MOLA Language," *ACM/IEEE 8th Int. Conf. Model Driven Eng. Lang.*

Syst. (MoDELS/UML "2005). (MoDELS/UML"05 Work. Model Transform. Pract. , Montego Bay, Jamaica, 2005.

[141] L. Tratt, "The MT model transformation language," *ACM*, no. Department of Computer Science, King's College London, Strand, London, WC2R 2LS, U.K., 2006.

[142] B. Bordbar, "SiTra : Simple Transformer," 2011. [Online]. Available: http://www.cs.bham.ac.uk/~bxb/Sitra/index.html.

[143] E. Visser, "Program Transformation with Stratego / XT : Rules , Strategies , Tools , and Systems Eelco Visser," *Domain-Specific Progr. Gener. Vol. 3016 Lect. Notes Comput. Sci. pages 216--238. Spinger-Verlag*, no. February, 2004.

[144] S. Vermolen and E. Visser, "Heterogeneous Coupled Evolution of Software Languages," *11th Int. Conf. Model Driven Eng. Lang. Syst. (MODELS 2008), Vol. 5301 Lect. Notes Comput. Sci. pages 630--644, Heidelberg, Springer*, 2008.

[145] J. Steel and M. Lawley, "Model-Based Test Driven Development of the Tefkat Model-Transformation Engine," *15th Int. Symp. Softw. Reliab. Eng.*, pp. 151–160.

[146] M. Lawley and J. Steel, "Practical Declarative Model Transformation With Tefkat," *Satell. Events Model. 2005 Conf. LNCS Vol. 3844. Jamaica*, 2005.

[147] E. Balland, P. Brauner, R. Kopetz, P. Moreau, and A. Reilles, "Tom : Piggybacking Rewriting on Java," *Conf. Rewriting Tech. Appl.*, 2007.

[148] K. Lano and S. K. Rahimi, "UML RSDS Model Transformation and Model-Driven Development Tools Supports." [Online]. Available: http://www.dcs.kcl.ac.uk/staff/kcl/uml2web/.

[149] K. Lano, "The UML-RSDS Manual," 2012.

[150] OptXware Research and Development LLC, "The Viatra-I Model Transformation Framework Users ' Guide," 2007.

[151] "VIATRA2 : VIsual Automated model TRAnsformations framework, Supported by Budapest University of Technology and Economics, HUN OptXware Research and Development LLC, and HUN YourKit LLC, RUS," [Online]. Available: http://eclipse.org/gmt/VIATRA2/.

[152] OMG, "OMG Meta Object Facility (MOF) Core Specification (Version 2.4.1)," 2011.

[153] D. Di Ruscio, "Specification of Model Transformation and Waving in Model Driven Engineering," University of L'Aquila, Italy, 2007.

[154] T. Mens, K. Czarnecki, and P. Van Gorp, "A taxonomy of model transformation," *Electron. Notes Theor. Comput. Sci.*, 2006.

[155] D. W. Stephen J. Mellor, Kendall Scott, Axel Uhl and D. W. By Stephen J. Mellor, Kendall Scott, Axel Uhl, *MDA Distilled: Principles of Model-Driven Architecture*. Addison Wesley, 2004.

[156] "INTEROP NoE (Interoperability Research for Networked Enterprises Applications and Software Network of Excellence)." [Online]. Available: http://www.interop-vlab.eu/.

[157] "ATHENA IP (Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications Integrated Project)." [Online]. Available: http://cordis.europa.eu/projects/rcn/72762_en.html.

[158] R. Ruggaber, "ATHENA–advanced technologies for interoperability of heterogeneous enterprise networks and their applications," 2006.

[159] B. Elvesæter, A. Hahn, A. Berre, and T. Neple, "Towards an Interoperability Framework for Model- Driven Development of Software Systems," *Interoperability Enterp. Softw. Appl.*, 2006.

[160] A.-J. Berre, F. Liu, J. Xu, and B. Elvesaeter, "Model Driven Service Interoperability through Use of Semantic Annotations," *Int. Conf. Interoperability Enterp. Softw. Appl. China*, Apr. 2009.

[161] D. Chen and G. Doumeingts, "European initiatives to develop interoperability of enterprise applications—basic concepts, framework and roadmap," *Annu. Rev. Control*, Jan. 2003.

[162] A. Berre, B. Elvesæter, N. Figay, C. Guglielmina, S. G. Johnsen, D. Karlsen, T. Knothe, and S. Lippe, "The ATHENA Interoperability interoperability framework," ... *Interoperability II*, pp. 1–12, 2007.

[163] IBM, "SOA," 2008. [Online]. Available: http://www-01.ibm.com/software/solutions/soa/.

[164] "Organization for the Advancement of Structured Information Standards(OASIS)." [Online]. Available: http://www.oasis-open.org/committees/tc_cat.php?cat=soa.

[165] IBM, "Defining SOA as an architectural style," 2007. [Online]. Available: http://www.ibm.com/developerworks/architecture/library/ar-soastyle/#resources.

[166] D. Barry, "Web services and service-oriented architecture: the savvy manager's guide," 2003.

[167] S. Güner, "Architectural Approaches, Concepts and Methodologies of Service Oriented Architecture," Technical University Hamburg Harburg, 2005.

[168] A. Arsanjani, "Service-oriented modeling and architecture How to identify, specify, and realize services for your SOA," 2004. [Online]. Available: http://www.ibm.com/developerworks/library/ws-soa-design1/.

[169] "World Wide Web Consortium (W3C), Web Services Architecture," 2004. [Online]. Available: http://www.w3.org/TR/ws-arch/.

[170] N. H. Arzt, "Service-Oriented Architecture A Field Guide to Integrating XML and Web Services," *J. Healthc. Inf. Manag.*, vol. 24, no. 2, pp. 45–52, Jan. 2010.

[171] S. Khoshafian, *Service Oriented Enterprises*. 2007.

[172] D. Winer, D. Box, B. Atkinson, and M. Al-Ghosein, "Simple Object Access Protocol (SOAP)," *maintained by the XML Protocol Working Group of the World Wide Web Consortium*. [Online]. Available: http://www.w3.org/tr/soap/.

[173] K. Chu, O. Cordero, M. Korf, C. Pickersgill, R. Whitmore, C. Author, W. Bathurst, R. Dobrik, R. Kodali, and R. Zimmermann, "Oracle ® SOA Suite Developer ' s Guide," no. September, 2006.

[174] J. Xu, Z. Bai, A. J. Berre, and O. C. Brovig, “Model Driven Interoperability through Semantic Annotations using SoaML and ODM,” *Inf. Control Probl. Manuf.*, 2009.

[175] R. Jardim-Goncalves, A. Grilo, and A. Steiger-Garcao, “Challenging the interoperability between computers in industry with MDA and SOA,” *Comput. Ind.*, vol. 57, no. 8–9, pp. 679–689, Dec. 2006.

[176] R. Jardim-Goncalves, K. Popplewell, and A. Grilo, “Sustainable interoperability: The future of Internet based industrial enterprises,” *Comput. Ind.*, vol. 63, no. 8, pp. 731–738, Oct. 2012.

[177] A. Grilo and R. Jardim-goncalves, “Cloud-Marketplaces : Distributed e-procurement for the AEC sector,” *Adv. Eng. Informatics*, vol. 27, no. 2, pp. 160–172, 2013.

[178] F. Han, E. Moller, and A. J. Berre, “Organizational Interoperability Supported through Goal Alignment with BMM and Service Collaboration with SoaML,” *2009 Int. Conf. Interoperability Enterp. Softw. Appl. China*, pp. 268–274, Apr. 2009.

[179] D. Roman, B. Morin, S. Wang, and A. J. Berre, “A Model-driven Approach to Interoperability in B2B Data Exchange,” in *Advanced Result on MDI/SOA innovation workshop (at IWEI 2001)*, 2011.

[180] A. Limyr, T. Neple, A. Berre, and B. Elvesæter, “Semaphore–a model-based semantic mapping framework,” *Bus. Process Manag. Work. LNCS 4103, Springer-Verlag Berlin Heidelb.*, pp. 271–280, 2006.

[181] M. Q. Saleem, J. Jaafar, and M. F. Hassan, “Model driven security frameworks for addressing security problems of Service Oriented Architecture,” in *International Symposium in Information Technology (ITSim 2010)*, 2010, vol. 3, pp. 1341–1346.

[182] A. Delgado, F. Ruiz, and I. G. De Guzmán, “A Model-driven and Service-oriented framework for the business process improvement,” *J. Syst. Integr.*, no. Mdd, pp. 45–55, 2010.

[183] U. Zdun, C. Hentrich, and S. Dustdar, “Modeling Process-Driven and Service-Oriented Architectures Using Patterns and Pattern Primitives,” *ACM Trans. Web*, vol. 1, no. 3, 2007.

[184] S. Herold, A. Rausch, A. Bösl, J. Ebell, C. Linsmeier, and D. Peters, “A Seamless Modeling Approach for Service-Oriented Information Systems,” *5th Int. Conf. Inf. Technol. New Gener.*, 2008.

[185] J. Touzi, F. Benaben, H. Pingaud, and J. P. Lorre, “A model-driven approach for collaborative serviceoriented architecture design,” *Int. J. Prod. Econ.*, 2009.

[186] J. Ricken, “Top-down modeling methodology for model-driven soa construction,” *OTM Confed. Int. Conf. move to meaningful internet Syst.*, pp. 323–332, 2007.

[187] R. Radhakrishnan and M. Wookey, “Model driven architecture enabling service oriented architectures,” *Whitepaper SUN Microsystems*, 2004.

[188] G. Benguria, X. Larrucea, B. Elvesæter, T. Neple, A. Beardsmore, and M. Friess, “A platform independent model for service oriented architectures,” *Enterp. Interoperability*, pp. 23–32, 2007.

[189] H. Kim, “Modeling of Distributed Systems with SOA & MDA,” *IAENG Int. J. Comput. Sci.*, 2008.

[190] A. T. Rahmani, V. Rafe, S. Sedighian, and A. Abbaspour, “An MDA-Based Modeling and Design of Service Oriented Architecture,” *ICCS 2006*, pp. 578–585, 2006.

[191] C. P. Bispo, R. S. P. Maciel, J. M. N. David, I. Ribeiro, and R. Conceicao, “Applying a model-driven process for a collaborative service-oriented architecture,” *Int. Conf. Comput. Support. Coop. Work Des.*, Apr. 2010.

[192] V. Rafe, R. Rafeh, P. Fakhri, and S. Zangaraki, “Using MDA for Developing SOA-Based Applications,” *2009 Int. Conf. Comput. Technol. Dev.*, pp. 196–200, 2009.

[193] W. H. Utomo, “Implementation of MDA Method into SOA Environment for Enterprise Integration,” *IJCSI Int. J. Comput. Sci.*, vol. 8, no. 6, pp. 10–18, 2011.

[194] C. Hahn, S. Jacobi, and D. Raber, “Enhancing the Interoperability between Multiagent Systems and Service-Oriented Architectures through a Model-Driven Approach,” *IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, Aug. 2010.

[195] N. Ali, R. Nellipaiappan, R. Chandran, and M. A. Babar, “Model driven support for the Service Oriented Architecture modeling language,” *Int. Work. Princ. Eng. Serv. Syst. - PESOS*, 2010.

[196] A. Sadovykh, P. Desfray, B. Elvesæter, A. Berre, and E. Landre, “Enterprise Architecture Modeling with SoaML using BMM and BPMN – MDA Approach in Practice,” in *7th Central and Eastern European Software Engineering Conference, CECR 2010*, 2010.

[197] OMG, “Service oriented architecture Modeling Language (SoaML) Specification,” 2012.

[198] J. Bézivin, R. M. Soley, and A. Vallecillo, “Editorial to the Proceedings of the First International Workshop on Model-Driven Interoperability,” *Computer (Long. Beach. Calif).*, pp. 4–5, 2010.

[199] Y. Lemrabet, M. Bigand, and D. Clin, “Model Driven Interoperability in practice : preliminary evidences and issues from an industrial project,” *First Int. Work. Model. Interoperability (MDI 2010)*, 2010.

[200] M. Kavianpour, “Soa and large scale and complex enterprise transformation,” *Serv. Comput. 2007*, pp. 530–545, 2007.

[201] B. Elvesæter, A.-J. Berre, and A. Sadovykh, “Specifying services using the service oriented architecture modeling language (soaml),” in *The 1st International Conference on Cloud Computing and Services Science, CLOSER 2011,*, 2011.

[202] Infosys, “Connecting the dots : Cloud and SOA,” 2011.

[203] R. Sharma, M. Sood, and D. Sharma, “Modeling Cloud SaaS with SOA and MDA,” *Adv. Comput. Commun. - Commun. Comput. Inf. Sci.*, 2011.

[204] R. Sharma and M. Sood, “Cloud SaaS and Model Driven Architecture,” *ACCT Int. Conf. Adv. Comput. Commun. Technol.*, 2011.

[205] R. Sharma and M. Sood, “A Model-Driven Approach to Cloud SaaS Interoperability,” *Int. J. Comput. Appl.*, 2011.

[206] R. Sharma and M. Sood, “Enhancing Cloud SaaS Development With Model Driven Architecture,” *Int. J. Cloud Comput. Serv. Archit.*, 2011.

[207] R. W. Maule, “SoaML and UPIA Model Integration for Secure Distributed SOA Clouds,” *IEEE Eighth World Congr. Serv.*, Jun. 2012.

[208] X. Zhang, K. He, J. Wang, J. Liu, C. Wang, and H. Lu, “On-Demand Service-Oriented MDA Approach for SaaS and Enterprise Mashup Application Development,” *Int. Conf. Cloud Serv. Comput.*, Nov. 2012.

[209] G. Wang, H. Wang, S. Arroyo, R. Rencher, and J. Tjelle, “Compositional QoS Modeling and Analysis of Cloud-based Federated Ecosystems,” *2012 IEEE 16th Int. Enterp. Distrib. Object Comput. Conf.*, pp. 173–182, Sep. 2012.

[210] M. Salama, A. Shawish, A. Zeid, and M. Kouta, “Integrated QoS Utility-Based Model for Cloud Computing Service Provider Selection,” *2012 IEEE 36th Annu. Comput. Softw. Appl. Conf. Work.*, pp. 45–50, Jul. 2012.

[211] M. K. Goyal, A. Aggarwal, P. Gupta, and P. Kumar, “QoS based trust management model for Cloud IaaS,” *2012 2nd IEEE Int. Conf. Parallel, Distrib. Grid Comput.*, pp. 843–847, Dec. 2012.

[212] P. Rubach and M. Sobolewski, *Dynamic SLA Negotiation in Autonomic Federated Environments*. Springer, 2009.

[213] Y. Sun, W. Tan, L. Li, G. Lu, and A. Tang, “SLA detective control model for workflow composition of cloud services,” *Proc. 2013 IEEE 17th Int. Conf. Comput. Support. Coop. Work Des.*, pp. 165–171, Jun. 2013.

[214] M. Garey, D. Johnson, and R. Sethi, “The complexity of flowshop and jobshop scheduling,” *Math. Oper. Res.*, 1976.

[215] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, 1989, p. 432.

[216] *FACTORIES OF THE FUTURE PPP STRATEGIC MULTI-ANNUAL ROADMAP*. Prepared by the Ad-hoc Industrial Advisory Group, 2010.

[217] “Future Internet Enterprise Systems,” *funded under the European Commission FP7 Programme*. [Online]. Available: http://www.fines-cluster.eu/fines/jm/.

[218] C. M. Macal and M. J. North, “Agent-based modeling and simulation,” *Proc. 2009 Winter Simul. Conf.*, pp. 86–98, Dec. 2009.

[219] A. Zutshi, A. Grilo, and R. Jardim-Gonçalves, “A Dynamic Agent-Based Modeling Framework for Digital Business Models: Applications to Facebook and a Popular Portuguese Online Classifieds Website.,” in *Digital Enterprise Design and Management*, 2014.

[220] D. Feitelson, *Workload modeling for computer systems performance evaluation*, First. Cambridge University Press, 2014.

[221] S. M. Ross, *Introduction to Probability Models*, 5th ed. Academic Press, 1993.

Appendix A

Current alternatives in the cloud computing market

Nowadays cloud environments include hundreds of independent, heterogeneous, private/hybrid clouds, but many business operators have predicted that the process toward interoperable cloud scenarios will begin in the near future. In order to analyzing the actual platform, Table 0-1 introduces a number of the existing cloud computing offers and specify the type of provided services according to the service/delivery model classification.

Table 0-1 Current alternatives in the cloud computing market.

	Company	Service	Company's description
1	**37Signals**	Software as a Service	Basecamp is a versatile project management tool that can be used for business as well as technical projects. Simplicity and ease of use are strengths suits of Basecamp. Features of Basecamp include wiki style document editing, file sharing, message boards, to-do list, and milestone management.
2	**3tera AppLogic**	Platform as a Service	CA 3Tera AppLogic is a turnkey cloud computing platform for composing, running and scaling distributed applications. It uses advanced virtualization technologies to be completely compatible with existing operating systems, middleware and web applications. Billions of lines of tried and true infrastructure software, middleware and application code can be used with CA 3Tera AppLogic unaltered.CA 3Tera AppLogic operates on the logical structure of the application, enabling you to package an entire N-tier application into a logical entity and manage it as a single system. This approach also makes it very easy to assemble, deploy, monitor, control and troubleshoot applications visually in a browser.
3	**3X Systems**	Infrastructure as a Service (Storage Vendors)	The 3X Remote Backup Appliance offers three critical applications in one easy to implement and affordable solution – secure data vault, remote backup, and rapid disaster recovery. The 3X Systems patent-pending Locator service allows organizations to build a "private cloud" that automatically backs up data from local and remote devices regardless of the location of either the appliance or the data. When needed, this technology makes disaster recovery quick and easy.
4	**Adaptive Computing**	Infrastructure as a Service	Moab® Adaptive Computing Suite provides the policy-based intelligence for cloud infrastructures to be successful and ensure

	Company	Service	Company's description
5			that each cloud is: **Agile**—with fast delivery of the IT services in minutes via user self-service, catalog-based service definitions, optimal resource provisioning to avoid failures, and chargeback based on usage **Automated**—across all resource decisions, provisioning and management processes to optimize resource utilization and capacity to reduce costs, meet service level guarantees, and reduce IT staff burden so IT staff can scale with cloud services **Adaptive**—so cloud resources self-optimize and respond to changing conditions, without manual intervention, to optimize service delivery to the business Moab Adaptive Computing Suite is an intelligent cloud management system that automates the decisions and process of provisioning diverse resources against diverse incoming workloads and changing conditions based on business policies and service level goals with usage billing and a self-service user request interface.
5	**Agathon Group**	Infrastructure as a Service	Agathon Group is a technology studio specializing in cloud hosting and custom software solutions. Cloud Hosting allows us to take a big bunch of hardware, make it look like one big piece of hardware ("the cloud"), and run virtual servers within the cloud. Using 3tera's AppLogic virtualization software, you get multiple servers working transparently behind the scenes to provide the horsepower to run your site. It's like a vacation timeshare, where you get the use of a much larger piece of property without having to pay full price for that property. Unlike a vacation timeshare, you're guaranteed to be able to use your space whenever you need it, not just whenever it happens to be available. We're not going to stick you with that "October week during rainy season" timeshare; it's always sunny with Agathon Group Cloud Hosting. With Cloud Hosting, the growing pains are eliminated. Even with the smallest Cloud Hosting package, your site is spread out across our large cloud of servers. As your needs grow, we simply dedicate more of that cloud for your use and you've increased capacity in minutes, without expensive hardware or new staff hires. The power of a dedicated server, the reliability and scalability of a farm of servers, the cost of a shared

	Company	Service	Company's description
			server. Good, fast, cheap: pick three.
6	**AllenPort**	Infrastructure as a Service	AllenPort's "hybrid SaaS" technology is making waves. It replicates a traditional Microsoft Windows network and makes it possible to download data or applications such as Microsoft Word to wherever the user is working. This is all done securely using AllenPort's Virtual File Cabinet offering. Overall, AllenPort's technology handles file management chores like backup, file sharing, disaster recovery, remote access and managing user requirements.
7	**Amazon EC2**	Infrastructure as a Service	Since staking its claim with Amazon Web Services in early 2006, Amazon.com has established itself as a pioneer. Amazon EC2 (Elastic Compute Cloud) users obtain and configure capacity and control computing resources while running them on Amazon's environment. The real draw is the ability to add capacity and scale in seconds, or reduce capacity as needed while customers only pay for what they use. It also is designed for use with other Amazon Web Services.
8	**Appirio**	Software as a Service	Appirio provides technology-enabled professional services that help companies do more with cloud applications and platforms like salesforce.com, Google, and Workday. Our services range from cloud strategy to cloud migration to cloud development to cloud management. Our technology helps enterprises build, manage, and connect cloud applications and platforms. Our offerings are supported by more than 400 cloud experts and CloudSpokes, a 15,000 person-strong global cloud developer community. Founded in 2006, Appirio has offices in the U.S. and Japan, and is backed by Sequoia Capital and GGV Capital.
9	**Appistry**	Platform as a Service	When it was founded in 2001, Appistry was already eyeing the cloud as the next big thing. Appistry's CloudIQ platform has become known for delivering a run-time application platform that complements existing technology to create scalable, service-oriented applications with lower investments. Appistry lives at the nexus of grid computing, virtualization and SOA, and delivers real-time IT apps and infrastructure that takes the best attributes of each of those three components.

	Company	Service	Company's description
10	**appnexus**	Platform as a Service	AppNexus has built a platform designed as core ad technology infrastructure that our clients can seamlessly plug their businesses into. From onboarding data to buying and selling both real-time and direct inventory, to sophisticated analytics and simplified client billing, we build toolsets for our clients and then get out of their way so they can put their media savvy to work. —A View of the Entire Internet: Finally single point access to all user and reporting data. Our clients can track and target users across their managed inventory — and real-time inventory to every single ad impression on the Internet. —Valuation Tools: Our algorithms are optimized to inventory, user and media — for any goal our clients have, from CPA to CPC to CPM and more. —Granular Analytics: From impressions served to clicks and conversions, clients can precisely track campaigns to raise CPMs, accurately bill clients, and manage financial relationships. —Superior Data Integration: AppNexus is singularly proficient at integrating our clients' unique data into our platform — on almost every impression, no matter the supply source, through our server-side user data store. Our clients can also plug in any third-party data. —Global Reach: AppNexus opened a third data center internationally in June to support our growing community of international clients and reduce latency for global inventory. Features like multiple currency and timezone capabilities and 24-hour support round out our commitment to worldwide operations. —Quality and Safety: AppNexus has a human auditing team to protect our clients' advertisers and reputation, as well as Sherlock, an automated creative quality control system. Sherlock detects fraud, malware, and viruses, enforces blacklists, and checks ad tags to ensure they are from approved third-party ad servers. —Built on a Cloud: We've invested millions of dollars to build top-of-the-line, highly scalable, three-data center architecture specifically for ad serving. We can process billions of impressions a day with 100% uptime, and our massive data warehousing power supports sophisticated analytics and reporting.
11	**AppRiver**	Security	Founded in April of 2002, AppRiver has entered the SaaS space

	Company	Service	Company's description
		Vendors	with an array of spam and virus services, including its SecureTide Spam and Virus Protection, Archiving and Compliance, and CipherPost Email Encryption. The company offers 24/7 support, no contracts, no cancellation penalties and a free, 30-day trial. Specifically, SecureTide, a fully managed e-mail protection service, eliminates up to 99 percent of unwanted e-mail, while its CipherPost service helps users achieve and maintain regulatory compliance by providing encryption for all e-mail and mobile messaging.
12	**AppScale**	Platform as a Service	Open-source community, rejoice! AppScale is an open-source platform for Google App Engine applications. AppScale lets users deploy and monitor their App Engine applications in an open-source environment while providing mechanisms to debug and profile applications as needed. AppScale has already developed quite the following and, as cloud platforms continue their market penetration, an open-source alternative will surely gain traction.
13	**Apptix**	Software as a Service	Apptix is the premier provider of cloud-based Unified Communications services including Microsoft hosted Exchange email, web conferencing, business VoIP, and Microsoft SharePoint. Every day, you combat a myriad of issues to run your business - servers crashing, broken phones, salespeople unable to work remotely, a lack of team collaboration. Apptix provides hosted communications services so you can focus on what's important - growing your business - instead of your IT.
14	**AppZero**	Infrastructure as a Service	AppZero may be a newbie but it wants to break down the walls of the traditional approach to virtualization and make it easier to move applications to the cloud. With that in mind, AppZero offers OS-free Virtual Application Appliances that are self-contained, portable units, meaning enterprises can experiment with moving applications to the cloud while avoiding cloud lock-in. Meanwhile, for cloud providers, AppZero offers services that offer fast and easy application provisioning.
15	**Asigra**	Infrastructure as a Service	Asigra provides WAN-optimized software to help customers leverage public and/or private clouds through a single interface

	Company	Service	Company's description
16		(Storage Vendors)	for remote data protection and archiving. The agentless software also allows management of backup life cycle. Asigra recently moved away from basing partner margins solely on sales volume, and now bases them on the commitment partners make to Asigra, giving smaller partners the same margin opportunities as larger partners.
16	**AT&T Synaptic**	Platform as a Service	Its Synaptic Compute as a Service offering delivers pay-as-you-go cloud computing, allowing access to virtual servers. AT&T provides and manages the virtualization infrastructure, including the network, servers and storage. The user provides and manages the database and applications. AT&T does it all with no upfront fees, no commitment and no termination fees, which means no lock-in.
17	**Axcient**	Infrastructure as a Service (Storage Vendors)	Startup Axcient offers a hybrid cloud storage model that includes a combination of a storage appliance and Internet-based storage service that lets customers back data up both locally for fast restores and online for safe archiving. SMBs can purchase Axcient's storage appliances with capacity between 500 GB and 10 TB through channel partners, who then connect them through the Internet to cloud-based storage infrastructures that are owned and managed by Axcient.
18	**Barracuda Networks**	Security Vendors	Barracuda recently emerged as a stronger player in the Software-as-a-Service space with the acquisition of Purewire, a SaaS-based secure Web gateway provider, putting the company on par with many established SaaS providers. Down the road, Barracuda plans to offer an array of SaaS and hybrid security services that will eventually integrate Purewire's Web Security Services into Barracuda's SaaS portfolio and existing product line, with a special emphasis on remote user support.
19	**Birst**	Software as a Service	Birst targets its cloud-based business intelligence tools toward mid-market businesses and underserved departments of big companies that may not have the money or expertise to implement large-scale, complex business intelligence systems. Businesses use Birst to extract data from multiple sources, build data warehouses and author reports. The Birst Live Access

	Company	Service	Company’s description
20			feature in the latest release of the software lets businesses tap into pre-existing data warehouses. And the Birst Advanced ETL Services helps companies handle complex data transformations.
20	**Boomi**	Infrastructure as a Service	If we're learning anything about the cloud it's that it's all about self-service. And Boomi, with its AtomSphere offering, takes self-service a step into the future. Founded in 2000, Boomi and its AtomSphere connect any combination of cloud and on-premise applications without software or appliances. But rest assured that with AtomSphere, system integrators, ISVs and businesses handle any combination of SaaS, cloud and on-premise app integration without the burden of installing or maintaining software or appliances.
21	**CA**	Infrastructure as a Service	CA's acquisitions of NetQoS in September, 2009, and Cassatt earlier last year have put CA to the forefront of cloud computing in 2010. The two acquisitions helped CA bulk up its cloud-based network and systems management tools. By adding NetQoS's monitoring prowess and Cassatt's data center automation and policy-based optimization expertise, CA can boost the functionality of its Spectrum Automation Manger to let it manage network and systems traffic in both public and private cloud computing environments.
22	**Callidus Software**	Software as a Service	In "Glengarry Glen Ross" Alec Baldwin "motivates" sales representatives by offering a car, a set of steak knives -- and getting fired -- as performance incentives. Thankfully sales managers today have better tools. With its on-demand sales performance and incentive compensation management applications, Callidus seeks to do for sales compensation management what Saleforce.com has done for CRM. The applications help businesses manage incentive payments to employees, distributors, brokers and -- yes -- channel partners, to help align those incentives with corporate goals.
23	**Carbonite**	Infrastructure as a Service (Storage Vendors)	Carbonite is a pioneer in online backup and the first to offer unlimited backups for a fixed price of $55 a year. The company also offers its storage cloud offering to systems vendors for bundling purposes, and has a program to let solution providers

	Company	Service	Company’s description
24			add it to their list of services. Carbonite's software backs up data changes automatically when the PC is idle, and encrypts the data for security purposes. Customers can restore individual files or complete data sets.
24	**Caringo**	Infrastructure as a Service (Storage Vendors)	Caringo's CAStor lets solution providers build a cloud storage infrastructure by plugging a USB key onto multiple industry-standard servers. Those servers are clustered, and all files stored in the CAStor cluster are replicated. The company also offers a content router to distribute content from the CAStor cloud, the ability to serve files from that cloud and a desktop archive utility that allows individual end-users to store files and make them available without disrupting workflows.
25	**Cast Iron Systems**	Infrastructure as a Service	Calling itself a SaaS integration company, Cast Iron offers an option for integrating SaaS applications with the enterprise. That method, which involves configuration, not coding, can in some cases slash integration costs up to 80 percent. The approach also gives SaaS providers the ability to offer their customers a rapid data migration and integration service.
26	**Citrix**	Infrastructure as a Service	Citrix Cloud Center (C3) ties together virtualization and networking products, arming cloud providers with a virtual infrastructure platform for hosted cloud services. The service, which is available on a monthly, usage-based pricing model and support mode, is an architecture comprising five key components: a platform powered by Citrix XenServer; applications and desktop services via Citrix XenApp; delivery powered by Citrix NetScaler; a bridge using Citrix Repeater; and orchestration through Citrix Workflow Studio.
27	**Ctera**	Infrastructure as a Service (Storage Vendors)	Ctera provides an appliance which includes everything needed to get storage to the cloud in minutes. The company's CloudPlug is a full-fledged Linux-based appliance about the size of an AC adapter which plugs into a power outlet, a router, and a PC to automatically handle backups to a cloud-based storage provider without the need for additional hardware or software. The company also offers a small two-bay NAS appliance which also automatically backs data up to the cloud.

	Company	Service	Company's description
28	**Doyenz**	Infrastructure as a Service (Storage Vendors)	The Doyenz ShadowCloud platform can help solution providers restore servers in the cloud for low-cost disaster recovery, failover and data migration. The company allows the building and testing of servers as virtual machines using VMware's ESXi server virtualization software and StorageCraft's ShadowProtect backup software for physical environments. These servers can then be restored and deployed in minutes at the client site, or on managed hosted infrastructure from players such as Savvis or Rackspace. Customers pay the equivalent to about one hour per month of the solution provider's consulting fee, while solution providers get recurring revenue and better margins for their investment.
29	**eFolder**	Infrastructure as a Service (Storage Vendors)	eFolder helps SMB solution providers offer cloud-based storage and e-mail archiving. The company also lets its solution providers offer a hybrid cloud storage offering which includes a local disk-based backup appliance for fast restores while connecting that appliance to the cloud for remote backups. eFolder last year also acquired the DoubleCheck e-mail management and security business of Network Management Group in a bid to combine storage, e-mail archiving and e-mail security into a channel-only, integrated services offering.
30	**Elastra**	Infrastructure as a Service	Elastra makes software that enables enterprises to automate modeling, deployment and policy enforcement of the application infrastructure. Its products tie in with provisioning and virtualization tools. Elastra's Enterprise Cloud Server software handles the management and provisioning of complex systems. Users can quickly model and provision application infrastructure; automate changes to the system deployment process; efficiently utilize internal, external and virtualized resources on demand and enforce IT policy rules. Elastra Cloud Server can also run on Amazon Web Services.
31	**EMC**	Infrastructure as a Service (Storage Vendors)	EMC's Mozy online backup offering, which it acquired in late 2007, still remains one of the most popular cloud-based backup offerings. EMC in late 2008 combined its Mozy offering with Pi, a provider of services for personal information management it acquired, into a new subsidiary, Decho. Decho is aimed at

	Company	Service	Company's description
			providing a platform on which to build cloud-based storage and other services, particularly the ability to store and manage the full gamut of personal information.
32	**EMC**	Infrastructure as a Service	With its Atmos and Atmos onLine offerings, EMC is evangelizing its approach to the cloud to deliver scalability, elasticity and cost savings by building, virtualizing and deploying services and applications. Atmos onLine is a cloud storage service built on Atmos, EMC's policy-based information management platform. EMC Atmos onLine provides Cloud Optimized Storage, or COS, capabilities for moving and managing large amounts of data with reliable service levels and in a secure fashion.
33	**Engine Yard**	Platform as a Service	Engine Yard got its start in 2006 at a time when customers were developing Rails applications but didn't want to worry about deploying, managing and scaling them. Engine Yard Cloud is a Rails application cloud for Web developers and teams running on top of cloud computing infrastructure. With $37.5 million in funding raised with backing from Amazon.com, New Enterprise Associates and Benchmark Capital, Engine Yard is barreling down on the competition.
34	**Enomaly**	Platform as a Service	With the launch of its Elastic Computing Platform in 2004, Enomaly was one of the world's first Infrastructure-as-a-Service platforms for service providers. Today, more than 15,000 organizations are using Enomaly's Elastic Computing Platform and rely on it to deliver infrastructure-on-demand services to customers. In its more than five years in the game, Enomaly has become regarded for its unlimited scalability, self-service capabilities, multi-tenant security, automated provisioning and integration into existing infrastructure.
35	**FinancialForce.com**	Software as a Service	FinancialForce.com, formerly known as Coda, is another cloud-computing app vendor that's following the Salesforce.com model. In fact Salesforce holds a minority stake in the company, and its on-demand accounting applications are built on Salesforce's Force.com cloud-computing platform. Last year FinancialForce unveiled the Winter 10 release of its software with enhanced debt

	Company	Service	Company’s description
36			management capabilities, the ability to pull non-financial data into the application and a new "Launchpad" feature that helps users navigate through complex financial processes.
36	**FlexiScale**	Platform as a Service	FlexiScale started as a “utility hosting platform” but morphed to offer a cloud computing infrastructure that offers a flexible, scalable and automated public cloud infrastructure. It arms customers with the power to flex their requirements up and down on-demand and only pay per service used. FlexiScale offers all of the power and storage resources needed; the ability to scale in real-time; a pay-as-you-go pricing with no lock-in; multi-OS support and automatic self-healing and 100 percent SLA.
37	**GCloud3**	Platform as a Service	GCloud3, is offering a turnkey private cloud solution in its gPlatform. The startup’s gPlatform enables deployment of servers, desktops, firewalls/routers and SAN storage. The six-component platform comprises gVirtual, a private cloud solution deployed at the client site; gClient, a thin-client line for integration into gVirtual or gHosted solutions; gBackup, a backup of the private cloud infrastructure; and gHosted, a data center-style deployment that uses gClient to connect to the data center.
38	**Gizmox**	Platform as a Service	Its Visual WebGUI platform is a way to enable enterprises to create rich and responsive Web applications with solid performance and security. And while Visual WebGUI is best suited for developing new Web apps and approving old ones it can also modernize legacy apps using standard technologies like ASP.net, DHTML and Silverlight. The open source offering has gained traction, and recently surpassed 30,000 deployments. Now Gizmox is taking Visual WebGUI commercial to make Web-dependent cloud applications.
39	**GoGrid**	Platform as a Service	Do you have only minutes to build an enterprise-grade cloud infrastructure? GoGrid’s got you covered. The GoGrid platform lets users deploy Web and database cloud services, mount infinite-volume cloud storage, add load-balancing and create, save and deploy custom cloud server images. GoGrid makes it even easier by tying in API libraries and tools.

	Company	Service	Company's description
40	**Google**	Software as a Service	If there were any doubt that cloud computing -- and Google Apps in particular -- were ready for prime time, it dissipated last year when the Los Angeles city government adopted Google's e-mail and on-demand applications under a $7.25 million contract. L.A. chose Google Apps over Microsoft, which competed for the sale. What's more, in early 2009 the company began offering its Google Apps Premier Edition hosted office productivity software through solution providers for the first time.
41	**Google App Engine**	Platform as a Service	With Google App Engine, users can build, run and maintain their applications on Google's infrastructure with no servers to maintain. Apps can be served from their own domain or a free domain on Google's appspot.com domain. As with most platforms, App Engine is pay to play. It supports several programming languages and costs nothing to get started. Apps have up to 500 MB of storage and enough CPU bandwidth to support an app serving about 5 million page views a month.
42	**Hewlett-Packard**	Security Vendors	HP's Cloud Assure, launched in March 2009, was designed as a way to drive adoption of cloud services and also expanded HP's SaaS partner program to enable its resellers to provide more cloud-based services. HP Cloud Assure incorporates HP Application Security Center, HP Performance Center and HP Business Availability Center. HP also provides customers with a team of expert engineers that performs security scans, executes performance tests and deploys availability monitoring.
43	**i365**	Infrastructure as a Service (Storage Vendors)	i365, a subsidiary of Seagate, works with solution providers and managed service providers to help customers manage their storage infrastructure in a cloud environment. The company offers a full range of cloud storage services through partners, such as data protection, e-mail archiving, electronic discovery and retention management tools, including data restoration, migration, and erasure. i365 most recently introduced technology to help software vendors tie their applications directly into the i365 storage cloud.
44	**IBM**	Infrastructure as a Service	IBM's Smart Business Storage Cloud is a private cloud service that supports multiple petabytes of data and billions of files. It is

	Company	Service	Company’s description
45		(Storage Vendors)	based on IBM's blade server and XIV storage technologies. The service lets businesses build an on-site storage cloud managed by IBM, or back up data to one of IBM's own data centers. IBM also plans to build a business-grade public cloud for storage.
45	**IBM**	Software as a Service	Many industry observers have long viewed IBM's Lotus division as one more road-kill victim of the Microsoft juggernaut. But Lotus is meeting with some success with its LotusLive offerings, a collection of on-demand collaboration and communications applications that provide an alternative to on-premise applications such as Microsoft Office and cloud-computing personal productivity tools such as Google Apps.
46	**IBM**	Infrastructure as a Service	When it comes to the cloud, IBM isn't messing around. The proof is in the pudding with its Smart Business Cloud services and solutions. With its combination of services and systems, which comprises public and private clouds and cloud-based versions of some of IBM's most popular applications, IBM is looking to the cloud for everything from analytics and software and services delivery to services such as storage management and cloud-based e-mail, scheduling and contact information.
47	**InContact**	Software as a Service	InContact offers Software-as-a-Service call center and "agent optimization" applications that are used by some 650 customers. Founded in 1997 as UCN Inc., a reseller of telecommunications services, the company evolved through a number of acquisitions. In addition to applications used by service agents to assist customers, InContact's broad product line includes interactive voice response and automatic call distribution software, computer telephony integration, and even call center workforce management tools.
48	**Informatica**	Infrastructure as a Service	Informatica basically pioneered cloud computing for data integration, offering a host of offerings for customers of various shapes and sizes. It offers fast and easy pay-as-you-go and pay-for-use options that let users move data into or out of the cloud or manage data within the cloud of from one app to another.
49	**Intacct**	Software as a Service	During this recession many SMBs have discovered they lack visibility into their finances. They've outgrown spreadsheets and

	Company	Service	Company's description
			financial management applications designed for consumers and small businesses. But buying an enterprise-class ERP system would be overkill. Enter Intacct, which offers on-demand financial management and accounting software for businesses with 25 to 1,000 employees. Competing with Microsoft, NetSuite and Sage, the company has raised $29 million in venture funding since 2007, launched a channel program in 2008, and has more than 2,500 customers.
50	**Intronis**	Infrastructure as a Service (Storage Vendors)	Intronis Online Backup enables partners to offer their own branded cloud storage service. It features block-level online backups to ensure only changes to data are backed up and offers full security as well as a full set of compliance-ready archiving and recovery capabilities. It also manages backup, storage and restoration of Microsoft Exchange files. Solution providers can get commission and recurring revenue for referring customers or can provide the Intronis service as part of their own suite of managed services.
51	**LiveOps**	Software as a Service	LiveOps, competing in the same market space as InContact, takes the whole cloud-computing concept to another level. Yes, it offers an on-demand contact center platform that businesses use to run their customer service departments. But it goes even further by providing a network of some 20,000 independent at-home agents who use the system -- making it possible for a business to rely on the cloud not just for its call center software, but for its entire customer service operation.
52	**LongJump**	Platform as a Service	Formed in 2003, LongJump recently launched its Business Application Platform, a platform-as-a-service (PaaS) play that can be licensed for use in an enterprise data center or by ISVs seeking a platform to build and host their own SaaS- or cloud-based apps. And LongJump is actively rounding up ISVs to brand and host their own SaaS offerings in private clouds. LongJump also can convert data to private clouds with LongJump's PaaS licensing option.
53	**M86**	Security Vendors	Through the acquisitions of Marshal and Finjan, M86 has built out its cloud security offerings. In November, it launched

	Company	Service	Company’s description
			MailMarshal SMTP 6.7, one of the first cloud-based offerings that protects against both malware and blended threat attacks. The Finjan acquisition enhanced M86's existing Web and e-mail security technologies with real-time content inspection and code analysis technology, along with malware detection capabilities designed to address Web threats not recognized by traditional signature-based technologies.
54	**McAfee**	Security Vendors	In September, McAfee completed the acquisition of Security-as-a-Service e-mail and Web security company MX Logic, giving the second largest security company a huge leg up in the Security-as-a-Service market. McAfee launched a cloud-based e-mail gateway in October following the acquisition, which expanded its cloud portfolio with e-mail and Web security offerings, e-mail archiving and e-mail continuity services, along with the addition of 40,000 new customers and 1,800 channel partners.
55	**Mezeo Software**	Infrastructure as a Service (Storage Vendors)	Mezeo provides software that lets IT hosters, SaaS providers, MSPs, telcos and ISPs develop cloud-based storage for customers and resellers. The Mezeo Cloud Storage Platform includes a Web application, a native Windows desktop client and native applications for iPhone, BlackBerry and Windows Mobile. With the platform, service providers can provide sharing, collaboration, file tagging, nested files and folders and security.
56	**Microsoft**	Platform as a Service	Windows Azure is Microsoft’s cloud computing platform, available now for free. Set to debut Feb. 1 as a paid service, Azure offers an environment for developers to create cloud apps and services. The platform will also run alongside current Microsoft environments offering an OS as a service in Windows Azure, a relational database in the cloud in Microsoft SQL Azure and the Windows Azure platform AppFabric, which eases connections between cloud and on-premise apps.
57	**MyDials**	Software as a Service	Another player in the crowded Software-as-a-Service business intelligence space, MyDials focuses on providing users with operational performance management dashboards that display key performance indicators (KPIs) and other visual metrics. With

	Company	Service	Company's description
58			a strong presence in manufacturing, the vendor is working with channel partners who can develop useful KPIs for their clients. MyDials 3.0 offers new "what-if" scenario and problem analysis capabilities, as well as the ability to connect to a wider range of operational ERP, CRM and supply chain management applications.
58	**NetApp**	Infrastructure as a Service	Call it IT-as-a-Service (ITaaS) or call it an enterprise cloud infrastructure. Data ONTAP 8, NetApp's latest cloud computing infrastructure, ties together its two previously separate platforms: Data ONTAP 7G and Data ONTAP GX. It delivers improved data management functions and tighter integration with data center management systems. Ultimately, NetApp Data ONTAP 8 enables storage, server, network and applications layers to talk to each other.
59	**NetSuite**	Software as a Service	ERP applications were once developed by big companies (think Oracle and SAP) for big companies. Since its 1998 founding, NetSuite's forte has been providing integrated, on-demand ERP, CRM and e-commerce applications to SMBs, giving them many of the same process automation capabilities once available only to major corporations. NetSuite has also been building up its ISVs, whose products enhance and extend NetSuite's core applications through the company's SuiteCloud Ecosystem including development tools and an online application marketplace.
60	**New Relic**	Infrastructure as a Service	New Relic is running full throttle with its RPM offering, an on-demand performance management tool for Web applications. It takes only minutes to implement and offers visibility and code-level diagnostics for Web apps deployed in both private and public clouds, along with traditional and dedicated infrastructures, and any combination thereof. With RPM, New Relic delivers real-time metrics, unlocking the ability to monitor, troubleshoot and fine tune app performance in the cloud.
61	**Novell**	Infrastructure as a Service	Novell is looking to the cloud to tie together all things IT. It is combining products like Moblin, a cloud-centric desktop OS developed by Novell and Intel; the SUSE Appliance Program, a

	Company	Service	Company's description
			program for ISVs to build software appliances and receive go-to-market support; Novell Cloud Security Service; and PlateSpin Workload Management Solutions for IT managers.
62	**Open Nebula**	Infrastructure as a Service	This open-source toolkit fits snuggly into existing data center environments to build any type of cloud deployment. OpenNebula can be used to manage virtual infrastructure in the data center or to manage a private cloud. It also supports hybrid clouds to combine local infrastructure with public cloud infrastructure for hosting environments. Additionally, it supports public clouds by offering cloud interfaces to expose its functionality for virtual machine, storage and network management.
63	**OpSource**	Infrastructure as a Service	OpSource is all about "cloud operations," offering everything from an enterprise-grade cloud infrastructure to fully managed hosting and apps management. Essentially, OpSource Cloud is a virtual private cloud within the public cloud, giving users control over their degree of Internet connectivity. Meanwhile, OpSource On-Demand combines technical operations, application operations and business operations into a Web operations offering that includes application management, compliance and business services. Lastly, OpSource Billing CLM is a self-service offering for SaaS and Web customer on-boarding, subscription management and payment processing.
64	**Oracle**	Software as a Service	Tom Siebel essentially created the CRM industry when he launched Siebel Systems in 1993. The company began offering a SaaS version of its software in 2003, and Oracle acquired them in 2006 for $5.8 billion. Today Oracle offers both Oracle CRM On Demand, the SaaS version of the Siebel product, and Oracle Siebel CRM, an on-premise application. Businesses can implement both and link them using the Oracle Application Integration Architecture and Oracle Fusion Middleware.
65	**OrangeScape**	Platform as a Service	OrangeScape aims its Platform-as-a-Service offering at non-programmers. For five years, OrangeScape has been used to develop applications while reducing the learning curve by getting rid of the technology and infrastructure complexities. Developing

	Company	Service	Company's description
			with OrangeScape is as simple as using a spreadsheet. If that's too hard, maybe app development really isn't for you.
66	**Paglo**	Infrastructure as a Service	This IT search and management service startup recently launched its Log Management application to let IT managers capture and store their logs as well as search and analyze them in the cloud. Paglo compares it to a Google-like search for logs, collecting data from all network devices. Paglo has also recently launched a new application to monitor Amazon EC2 application instances, such as disk reads and writes, CPU utilization and network traffic. Users can access the cloud-based information from any Web browser.
67	**Panda Security**	Security Vendors	Panda has further immersed itself in the cloud computing arena after it unveiled cloud-based antimalware services for SMBs -- Panda Cloud Antivirus and Panda Cloud Protection, in November. Panda Cloud Protection, a fully hosted, managed security service for SMBs, provides hands-off protection of endpoints and e-mail. Specifically, the service, which relies on Panda's Collective Intelligence cloud-scanning system, incorporates endpoint cloud-based antimalware and firewall protection.
68	**Ping Identity**	Security Vendors	Ping Identity's PingConnect resolves one of the biggest problems in just about every organization -- multiple passwords. PingConnect's single sign-on service exponentially improves customers' security posture by eliminating passwords for virtually every major SaaS application, including Salesforce.com, Google Apps, Concur, SuccessFactors and Workday, among others. The company recently expanded its SaaS partner program by partnering with two SaaS providers: sales performance management company Callidus Software and sales enablement vendor Kadient.
69	**PivotLink**	Software as a Service	Business intelligence could be "The Next Big Thing" when it comes to cloud computing. PivotLink, with 15,000 paid subscribers generating 2 million analytic reports every month, is getting a lot of attention. The company raised $10 million in venture funding in 2009 when VCs were clinging tightly to their

	Company	Service	Company’s description
			wallets. In April of that year, the company debuted PivotLink Gadget, which lets developers add business intelligence features to Google Apps.
70	**Proofpoint**	Security Vendors	Proofpoint offers its comprehensive Enterprise service, based on the same platform powering its Messaging Security Gateway appliances and incorporates DLP, spam filtering and e-mail archiving services. Its hosted e-mail service, Proofpoint Protect, is also an easy-to-use, low-cost inbound e-mail security product designed for organizations that don't require outbound data privacy and e-mail encryption features. The company distinguishes itself with its single management and policy console powered by Proofpoint MLX technology, an advanced machine learning system.
71	**QlikTech**	Software as a Service	QlikTech's QlikView business intelligence software can be deployed on-premise, in the cloud, or on a mobile device. Founded in Lund, Sweden in 1993, the company went global in 2004 and now has more than 12,000 customers. QlikView uses an "in-memory, associative approach" to data analysis, allowing the software to access and analyze data in real time. In June the company debuted QlikView 9 with enhanced visualization and search capabilities, PDF reporting, and the ability to support huge data sets.
72	**Qualys**	Security Vendors	SaaS security risk and compliance management company Qualys offers a full range of on-demand services, with a specialty in customer adherence to a wide array of regulatory compliance mandates, such as PCI. The company serves thousands of subscribers around the world with its QualysGuard service, including 200 of the Forbes Global 2000, with real-time vulnerability management, policy compliance, PCI compliance and Web application scanning.
73	**RackSpace**	Platform as a Service	With its CloudServers offering, RackSpace delivers servers on-demand via a cloud-driven platform of virtualized servers. Users can add new instances and reduce instances within seconds while paying for what’s provisioned. It also offers CloudSites, a fully-managed Web hosting platform that lets the users code it and

	Company	Service	Company’s description
74			load it and offers patching and security, monitoring, redundancy, clustering and the power of the cloud. Add to that RackSpace’s CloudFiles file storage and hosting in the cloud, and the platform is complete.
74	**Reldata**	Infrastructure as a Service (Storage Vendors)	Reldata develops technology that consolidates iSCSI SAN, NAS and WAN into a single unified storage offering that solution providers can use to help customers build private storage clouds. The company's RELDATA 9240i/RELvos appliance lets customers independently scale disk storage capacity, replication storage services and network performance without disrupting their applications. The appliance uses Reldata's own RELvos virtualization operating system and includes a high-performance storage controller and integrated SAS disk storage.
75	**RightScale**	Infrastructure as a Service	RightScale's Cloud Management Platform eases deploying and managing apps in the cloud and enables automation, control and portability. The platform helps users get into the cloud quickly with cloud-ready ServerTemplates and best-practice deployment architectures. And users retain complete visibility into all levels of deployment by managing, monitoring and troubleshooting applications. Lastly, RightScale's Cloud Management Platform helps users avoid lock-in by letting them choose their deployment language, environment, stack, data store and cloud for portability.
76	**Robobak**	Infrastructure as a Service (Storage Vendors)	Robobak provides automated agentless data backup technology for remote and branch offices, giving managed service providers and traditional solution providers tools to help build cloud-based Storage-as-a-Service. Robobak's v9 Data Protection Suite uses technologies such as block-level incremental backups, compression and deduplication. It also includes a full set of encryption features as well as a complete set of tools for setting backup policies.
77	**Salesforce.com**	Platform as a Service	The cloud computing behemoth is kicking its presence up a notch. Its Force.com development platform lets users log in, build an app and push it out into the cloud. All told, it’s supposed to help build and run applications faster at a fraction of the cost of traditional software platforms. The platform includes a database,

	Company	Service	Company's description
			security, workflow, user interface and other tools to guide the process for building business apps, mobile apps and Web sites.
78	**Salesforce.com**	Software as a Service	What Salesforce.com has done is popularize the concept of cloud computing, turning a vague IT architectural concept into a mainstream computing practice and providing CRM SaaS applications that -- for many businesses --were their entre into cloud computing. Salesforce has sought to solidify its position as a SaaS/cloud computing leader with its Force.com platform and infrastructure tools for developing and running cloud computing applications. Yet Salesforce's on-demand CRM sales and customer service applications still account for the bulk of the company's sales.
79	**SAS Institute**	Software as a Service	The $2 billion behemoth is a leader in business intelligence and analytical software, providing software from basic reporting to complex data analysis. SAS has been growing the number of analytical applications it provides on a hosted basis. It offers on-demand versions of its marketing campaign management and drug development software. Anticipating growth in its cloud business, SAS is spending $70 million on a new facility to manage growth in data volumes generated by its SaaS-based and hosted applications.
80	**ScanSafe**	Security Vendors	Founded in 2004, SaaS-based Web filtering company ScanSafe has recently been turning heads in the industry for its ability to detect and block malicious and inappropriate Web sites in real time. The services are powered by its multilayered Outbreak Intelligence threat detection technology, which processes more than 20 billion Web requests and blocks more than 200 million sites per month. The company was acquired by networking giant Cisco in December.
81	**StillSecure**	Security Vendors	StillSecure's flagship ProtectPoint is a fully managed security service targeting SMBs and midsize companies, offering a wide range of SaaS services, including managed firewall, vulnerability scanning, managed VPN, intrusion detection, Web security and hosted e-mail security. In November, the company joined the HP ProCurve One Alliance as a partner to deliver managed security

	Company	Service	Company's description
			services, which allows HP ProCurve partners to earn recurring revenue without additional investments in their infrastructure.
82	**Stoneware**	Infrastructure as a Service	Stoneware's mission is simple: To enable organizations to move from a client-centric to a Web-based, private cloud computing environment. With products aimed specifically at core verticals -- education, healthcare, manufacturing, legal, financial and enterprise -- Stoneware offers private cloud technology that is being used to create solutions that enable organizations to access applications, content, data and services from anywhere in a secure fashion.
83	**SugarCRM**	Software as a Service	SugarCRM has been taking steps to play in the cloud computing world. In May, 2009, the company debuted Sugar Express, a low-cost ($10 per user, per month) on-demand version of its CRM application targeting small businesses, home offices and branches within larger corporations. In September, SugarCRM began making Sugar Community Edition available for developers on Amazon's Elastic Compute Cloud, allowing ISVs and in-house developers who build on SugarCRM to leverage the cloud for their development, testing and deployment chores.
84	**SyferLock**	Security Vendors	SyferLock, a SaaS authentication company founded in 2007, discovered a way to provide unique passwords while users only have to remember one. With SyferLock's GridOne, users create their own password, which remains static. Then users ultimately submit a different password for each login conducted through the cloud-based GridOne system that associates each letter of the original password to arbitrary letters and numbers selected by the application. And although SyferLock has primarily targeted the enterprise, as well as pharmaceutical, health care and government verticals, executives contend that GridOne can scale to fit the needs of just about any midmarket company.
85	**Symantec**	Infrastructure as a Service	In October 2008, Symantec's acquisition of MessageLabs gave the security giant a leg up over many of its competitors in the online messaging security market and a hefty boost to its existing SaaS portfolio. The acquisition and the growing popularity of SaaS-based Web security offerings also opened up some new

	Company	Service	Company’s description
			markets for channel partners, who had hoped to expand their service offerings in the SMB and midmarket.
86	**Symantec**	Infrastructure as a Service (Storage Vendors)	Symantec offers two cloud-based backup services. For businesses, Symantec offers the Symantec Protection Network, which solution providers can resell to customers. As part of Symantec Hosted Services, the company's cloud-based services platform can secure and manage information stored on endpoints and delivered via e-mail, Web, and instant messaging. For consumers, Norton Online Backup lets up to five PCs or Macs within a single household automatically back data up to the company's Internet-based data vault, with the ability to search and share that data.
87	**Symform**	Infrastructure as a Service (Storage Vendors)	Symform's Cooperative Storage Cloud breaks a copy of a customer's backup into 64-MB blocks, scrambles those blocks with AES-256 encryption, fragments those blocks into 1-MB fragments, adds 32 1-MB parity fragments for redundancy, and then scatters those 96 fragments to cloud storage nodes around the world. Those nodes come from each customer designating a small part if its own storage capacity to be used as storing fragments of other companies' data.
88	**Symplified**	Security Vendors	Touting breakthrough technology, Symplified has found its niche in SaaS security by integrating enterprise security policies and administration with cloud application management services, targeting financial services, health care, high tech, utilities and life sciences. In December, Symplified unveiled a new user provisioning service -- SinglePoint Cloud Identity Manager, which allows organizations to centralize the management of user accounts for multiple applications from within their firewall or by using the Salesforce.com platform.
89	**Taleo**	Software as a Service	When the economy recovers, businesses will start hiring again and will need to recruit new employees, bring them on board and manage their performance. Taleo offers on-demand applications to attract, hire and retain top talent. The company has approximately 4,200 customers, including some 3,500 SMBs. Recently it debuted Taleo 10 with an updated Web 2.0 interface,

	Company	Service	Company's description
			new staff development capabilities and access through mobile devices and social networks such as LinkedIn and Facebook.
90	**Terremark**	Platform as a Service	Terremark's vCloud Express is a pay-as-you-go scalable deployment platform that gives development teams quick setup and easy resource configuration. Meanwhile, Terremark's Enterprise Cloud offering gives users precise, dynamic allocation of computing resources such as security, scale and performance with multi-user capacity, a dedicated resource pool architecture and roll-based security with private network connectivity and physical device integration layered on top.
91	**Trend Micro**	Security Vendors	Trend Micro remains the pioneer of the SaaS arena and helped foster the cloud computing phenomenon with the launch of its Smart Protection Network. The company recently catapulted into the virtualization space with its new virtualized Web Gateway Security product and also released Trend Micro Deep Security 7.0 as part of its advanced server security effort intended to protect all aspects of the server, including the operating systems, network and application layers on physical as well as virtualized platforms.
92	**Ubuntu**	Platform as a Service	Want choice in cloud strategy? Look no further than Ubuntu. Ubuntu Enterprise Cloud (which is powered by Eucalyptus) is included in Ubuntu Server Edition and lets companies introduce private, in-house clouds behind the firewall. Available on Amazon EC2, Ubuntu also offers customers the benefits of pushing services to a public cloud.
93	**Vembu**	Infrastructure as a Service (Storage Vendors)	Vembu StoreGrid, which enables solution providers to quickly build their own branded storage cloud, now lets MSPs resell storage services through smaller reseller partners who do not have the ability to manage Storage-as-a-Service on their own but who have customers requiring the service.It also provides clustering and load balancing, as well as the ability to replicate data directly to the Amazon Simple Storage Service storage cloud.
94	**Verizon**	Platform as a Service	Like AT&T, Verizon is also getting in on the cloud game with Computing as a Service, an offering that lets users take advantage

	Company	Service	Company’s description
			of virtualization in a self-service model. With Verizon CaaS, unveiled last June, users can perform all standard cloud computing tasks with the peace of mind offered by one of the world’s most recognized providers. Using a Web-based user interface, customers can manage and deploy virtual servers, scale computing power and control private networks.
95	**VMware**	Infrastructure as a Service	Last August, VMware acquired SpringSource which provides Web application development and management services. SpringSource speeds the delivery of applications in the cloud using a process that has become known as "lean software." VMWare also acquired Hyperic, an open-source monitoring and troubleshooting vendor. The VMWare-SpringSource-Hyperic trifecta creates an amalgamation that ties together VMWare's virtualization vision, SpringSource's strong development tools and application servers as well as Hyperic's monitoring.
96	**VMware, Cisco Systems And EMC**	Platform as a Service	Last year the trio created a the Virtual Computing Environment (VCE) coalition to offer a complete virtual data center product set that takes the best bits and pieces from each vendor’s portfolio. VCE’s Vblock is a series of preconfigured, pretested solution sets based on Cisco’s Unified Computing Systems and networking switches, EMC’s Symmetrix or Clariion storage arrays, and VMware’s vSphere server virtualization platform.
97	**WatchGuard**	Security Vendors	WatchGuard prepared its partner base for an anticipated upsurge of managed service offerings in 2010 with the acquisition of e-mail messaging security company BorderWare in August, followed by the launch of a new Managed Security Services Provider partner program in December. The purchase of BorderWare, which specializes in e-mail and Web application security platform management for midsize companies, government organizations and solution providers, also opens up more doors for WatchGuard in the cloud computing space.
98	**Webroot**	Security Vendors	Webroot has recently plunged into the SaaS arena with the release of its on-demand Web, e-mail and archiving products hosted in the cloud. The company unleashed dozens of enhancements to its business Web security service, while

	Company	Service	Company's description
			unveiling new security services to help businesses with e-discovery and compliance initiatives. The launch was part of a broader strategy for Webroot to build out its cloud security services.
99	**Websense**	Security Vendors	Websense's Security-as-a-Service, which relies on real-time threat updates from the Websense ThreatSeeker Network, is positioned as a way to eliminate distribution, deployment and ongoing upgrades of on-premise equipment. Specifically, Websense SaaS incorporates Websense Hosted Web Security and Websense Email Security, designed to protect against numerous Web and e-mail threats without any impact to network performance. Both services provide full reporting and policy management capabilities.
100	**WhiteHat**	Security Vendors	Founded in 2001 by a former Yahoo information security officer, WhiteHat aims to address the rising tidal wave of financially motivated Web-based attacks at the application level, preventing hackers from infiltrating and planting malicious code on users' Web sites. To help combat growing Web threats, WhiteHat offers resellers its WhiteHat Sentinel, a SaaS-platform assessment tool and Web site vulnerability manager designed to evaluate and verify all classes of vulnerabilities.
101	**Xactly**	Software as a Service	Xactly, a competitor to Callidus, provides on-demand applications that help sales managers keep tabs on how well sales reps are meeting their quotas, manage incentives such as commissions and even define sales territories. All that may seem to come awfully close to Salesforce.com's applications, but Xactly and Salesforce actually play nice together: The Xactly Business Solutions software for SMBs is even built on Salesforce's Force.com cloud computing platform.
102	**Zenith InfoTech**	Infrastructure as a Service (Storage Vendors)	Zenith Infotech provides solution providers the technology to offer managed services, virtual help desk and disaster recovery to SMB customers. The company's Backup and Disaster Recovery offering pioneered the hybrid cloud storage concept. It includes a disk-based backup appliance for local backups and sends those backups in near-real-time to the cloud as often as every 15

	Company	Service	Company's description
			minutes.
103	**Zetta**	Infrastructure as a Service (Storage Vendors)	Zetta develops technology to help solution providers build a cloud-based storage infrastructure that can replace an SMB's primary storage hardware. Zetta Enterprise Cloud Storage delivers enterprise-class storage services like data snapshots, replication and full redundancy without the need for extra hardware, all the while acting like a primary storage array. It offers data protection, data integrity, security and privacy starting at 25 cents, per GByte, per month with minimum performance guarantees.
104	**Zeus Technology**	Infrastructure as a Service	Zeus gives users the ability to create, manage and deliver online services in cloud, physical or virtual environments, letting companies visualize and manipulate the flow of traffic to Web-enabled apps. And early this year, they will release the Zeus Cloud Traffic Manager so customers can monitor and control cloud usage, offering a single control point for distributed applications, reporting on datacenter usage and allowing for goals like cost, SLA, security and compliance to be applied.
105	**Zlago**	Software as a Service	For small businesses with limited IT resources, moving to the cloud can be a daunting prospect. Zlago seeks to overcome those limitations by offering SMBs packages of hosted applications, including desktop, e-mail, backup, security, data storage and other business applications. Zlago is eagerly recruiting VARs to private-label the hosted services for their customers. For anyone trying to figure out how the channel fits into the cloud, Zlago is one to keep an eye on.
106	**Zoho**	Software as a Service	Zoho, founded in 1996, offers 21 cloud applications ranging from word processing and spreadsheets to CRM and project management. One reason for Zoho's success is it doesn't force users to switch. It has integrated its project management software with Google Apps and offers plug-ins that make it possible to use applications with Microsoft Office and SharePoint. That's the kind of strategy that makes it easier for solution providers to help their customers adopt cloud computing.
107	**Zscaler**	Security	Zscaler offers its security business policy services via its Global

	Company	Service	Company's description
		Vendors	Cloud Infrastructure, and offers one of the most comprehensive arrays of Web security, management and compliance services around, including Web filtering, social network security, virus control, antivirus, Web policy and management, DLP, and PCI and HIPAA compliance. It also touts low latency, reduced risk, lowered cost, improved resource utilization and IT administration simplification.
108	**Zuora**	Software as a Service	Zuora for the Cloud offers an unrivaled solution for creating the right business model to succeed in the cloud. Built from Zuora's extensive experience with cloud customers, Zuora for the Cloud: --Offers robust functionality to meter, price, and bill for customer usage --Is offered in four editions to suit every cloud business model --Manages the subscription lifecycle, including purchases, renewals, upgrades, and cancellations --Provides an interface that can be used by your sales and service reps, or self-service by your customers --Can be used to transact directly with your customers or offered as an "in-a-box" solution for your partners --Includes a Blueprint for the Cloud of best practice considerations when launching your cloud business.