# Toward a Unified Intercloud Interoperability Conceptual Model for IaaS Cloud Service

Tahereh Nodehi[1], Sudeep Ghimire[2] and Ricardo Jardim-Gonçalves[3]
[1]*Departamento de Engenharia Electrotecnica, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (UNL), Campus de Caparica, Caparica, Portugal*
[2]*UNINOVA, Departamento de Engenharia Electrotecnica, Faculdade de Ciencias e Tecnologia, UNL, Lisboa, Portugal*
[3]*CTS, UNINOVA, Departamento de Engenharia Electrotecnica, Faculdade de Ciências e Tecnologia, UNL, Lisboa, Portugal*
*t.nodehi@campus.fct.unl.pt, sudeepg545@gmail.com, rg@uninova.pt*

Keywords: Cloud Computing, Intercloud Interoperability, Service Level Agreements (SLA), Model Driven Architecture (MDA).

Abstract: The concept of interoperation between cloud providers is a recent research challenging objective. Current cloud systems have been developed without concerns of seamless cloud interconnection, and actually they do not support intercloud interoperability. The paper proposes a conceptual model for Intercloud Interoperability, to enable schedule dynamic operation for Infrastructure as a Service (IaaS) between different clouds. The paper is providing a better understanding of elaborates on the cloud computing architecture, appropriate metrics for Service Level Agreements (SLA) and Quality of Service (QoS) models that are required for seamless integration and interoperability between cloud environments. Then, a conceptual model for the Intercloud Interoperability Framework for Workload Migration is proposed. The novel component of the framework that provides interoperability is the Transformation Engine that maps workload between heterogeneous cloud providers, whilst Model Driven Architecture (MDA) is adopted as an applicable method for developing the Transformation Engine module.

## 1 INTRODUCTION

Cloud computing is a recent computational paradigm that many large software industries are adopting. Current cloud systems include several individual, but heterogeneous clouds with finite physical resources. With time, it is expected that expansion of the application scope of cloud services would require cooperation between clouds from different providers that have heterogeneous functionalities (Jardim-Goncalves, Popplewell, et al. 2012)(Coutinho et al. 2013). This seamless interworking mechanism between clouds is called "Intercloud". Interoperability, and can provide better Quality of Service (QoS), avoidance of vendor lock-in, whilst enable inter-cloud Resource Sharing and reduce power consumption and/or labour costs due to delivering services from various locations and different sources. However, most of the current cloud environment does not support intercloud interoperability and cloud computing needs more research work to provide sufficient functions to enable seamless collaboration between cloud services (Jardim-Goncalves, Agostinho, et al. 2012).

The paper elaborates on the cloud computing architecture and analysis relevant requirements to propose a novel Intercloud Interoperability framework, addressing cooperation between clouds that entails negotiated and agreed contract between intercloud service providers, metrics for Service Level Agreement (SLA) and QoS for Intercloud systems. A conceptual model is proposed to support the Intercloud Interoperability Framework for Workload Migration, tackling the essential technical requirements of a cloud operational environment. One of the important components introduced in the framework to support interoperability is the Transformation Engine that is able to map the workload between heterogeneous cloud providers. Model Driven Architecture (MDA) is the approach taken for developing the Transformation Engine module. The proposed framework is in validation using simulation experiments.

The rest of this paper is organized as follows.

After this Introduction in Section 1, Section 2 discusses the current state of the art in Intercloud Interoperability and the motivation for the present research, addressing cloud architecture and required parameters for QoS and SLA. The conceptual model for Intercloud framework is proposed in section 3. Section 4 and 5 propose the framework and discuss implementation possibilities. Section 6 concludes the paper.

## 2 INTERCLOUD INTEROPERABILITY FOR IAAS CLOUDS

Cloud Computing is one of the fastest evolving technologies in computer science. There are many analysis and researches on the development of the general cloud architecture defined by National Institute of Standards and Technology (NIST) (Grance 2010). An IaaS cloud service provider may have limited computing resources that can be one challenge for cloud developers (Bernstein 2009)(Bernstein et al. 2009)(Parameswaran & Chaddha 2009). The provisioning of the computing resources using IaaS multi-providers in an inter-cloud environment can be a powerful approach to solve this issue (Demchenko et al. 2013). However, cloud computing needs more research work to provide sufficient functions to enable seamless collaboration between IaaS cloud services.

Considering use cases proposed by NIST (Badger et al. 2010), Lewis (Lewis 2012) identified *"Workload Migration"* one of the four main cloud interoperability use cases that can benefit from current standards. Intercloud interoperability for IaaS service cloud providers should be able to allow IaaS cloud provider to migrate the workload to other

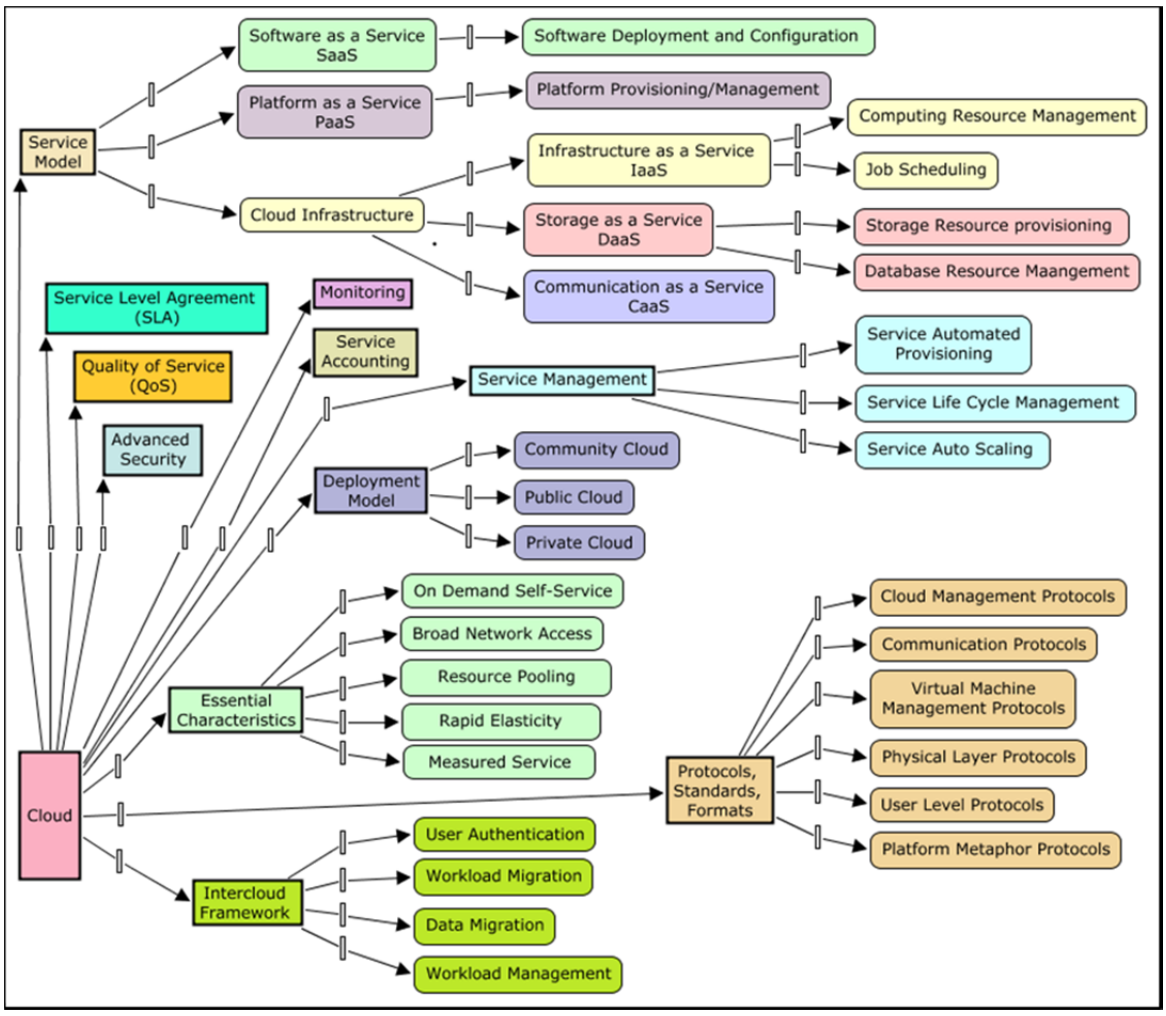


Figure 1: Intercloud Computing Generic Architecture.

selected IaaS providers to execute based on the appropriate requirements and then collect the results without a lock-in.

Celesti (Celesti et al. 2010) proposed a three-phase (discovery, match-making, and authentication) cross-cloud federation model for a general cloud architecture.

(Bernstein & Vij 2010) investigated that many-to-many mechanisms such as Messaging and Presence Protocol (XMPP) for transport, and Semantic Web techniques such as Resource Description Framework (RDF) can be the appropriate approaches for inter-cloud environments.

In this paper, the conceptual perspective for Intercloud Interoperability is studied for the second use case defined by NIST: Dynamic Operation Dispatch to IaaS Clouds, where many questions concerning intercloud interoeprability are still open. The focus is on the definition of all required parameters and in the proposal of the conceptual model for Intercloud Interoperability for workload migration. In future, we intend to design and implement the proposed framework followed by simulation.

## 2.2 Required Concepts for Intercloud Interoperability Framework

NIST proposed a cloud computing definition (Mell & Grance 2009) as follows: "Cloud computing 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". Thus, the term "Cloud" is used to describe the networks that incorporate various technologies, without the user knowing it.

Considering regular Cloud Architecture, Intercloud Architecture has to introduce an extra module for Intercloud Interoperability. Figure 1 depicts the intercloud conceptual model adopted in this research. The main Service Models are:

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Storage as a Service (DaaS)
- Communication as a Service (CaaS)
- Infrastructure as a Service (IaaS).

The proposed interoperability framework for IaaS cloud service providers forwards the workload to selected IaaS cloud providers. Thus, the proposed conceptual model considers the collected protocols, standards, formats, and common mechanisms by Bernstein (Bernstein et al. 2009) that can be useful for intercloud architecture.

Moreover, this paper refers to other references that formally describe QoS (Wang et al. 2012)(Salama et al. 2012)(Goyal et al. 2012) and SLA (Rubach & Sobolewski 2009)(Sun et al. 2013). Then, it identifies the required parameters and metrics for SLA and QoS modules that are fundamental for intercloud interoperability.

### 2.2.1 Appropriate QoS-SLA Metrics

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

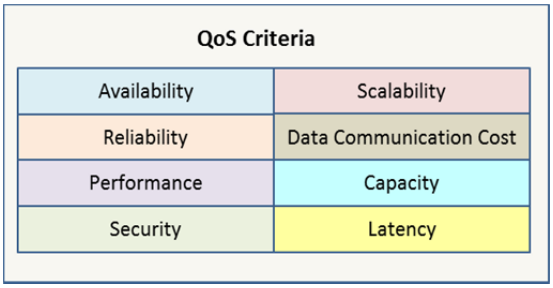


Figure 2: Required QoS Parameters for IaaS services.

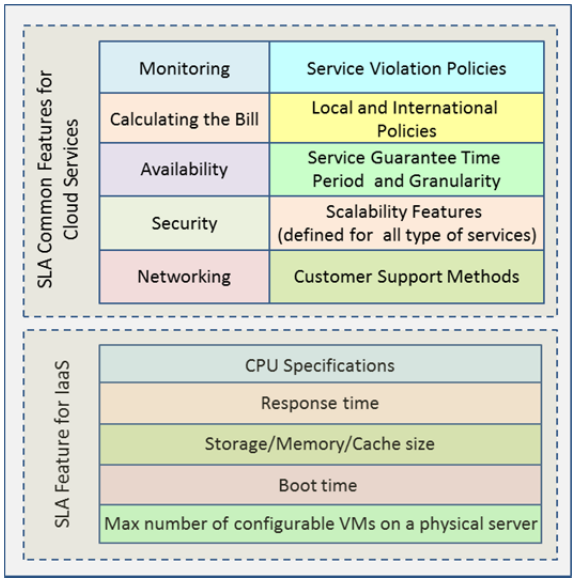


Figure 3: Required SLA Metrics for IaaS over Intercloud.

environment (Wang et al. 2012)(Salama et al. 2012)(Goyal et al. 2012) that can be beneficial to our proposed model. Additionally, research on defining a formal model for SLA has been considered in various systems (Rubach & Sobolewski 2009)(Sun et al. 2013).

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

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

Moreover, the appropriate SLA metrics for IaaS cloud services for all types of requirements are listed in Figure 3. The SLA metrics 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 metrics, we investigated some previous research work (Rubach & Sobolewski 2009)(Sun et al. 2013) as well as some dominant IaaS cloud service providers, such as Amazon's EC2 (Amazon n.d.), Windows Azure (Microsoft n.d.), and Rackspace Cloud (Rackspace n.d.).

# 3 CONCEPTUAL MODEL OF INTERCLOUD INTEROPERABILITY FOR WORKLOAD MIGRATION

Previous sections specified a conceptual model for cloud architecture and identified different requirements such as appropriate QoS-SLA metrics to resolve interoperability incompatibilities between heterogeneous IaaS cloud service providers.

This section proposes the detailed interoperability framework to dispatch part of workload between other selected IaaS cloud service providers based on the discussion in the section 2 that enhances our previous work (Nodehi et al. 2013). The conceptual model for the intercloud framework for IaaS is explained in Figure 4 that includes following fundamental components:

- *Intercloud Interface Module:* The intercloud framework receives events for workload migration through Intercloud Interface.
- *Model Manager Module:* Model Manager receives tasks from Intercloud Interface and provides the required details of the task (Object Models, Operation Models, and Data Model) using Semantic module accordingly.
- *QoS and SLAs Repository Module:* This module specifies QoS parameters (proposed in previous section) for the requirements of each task. Moreover, the SLA criteria between the current cloud and other IaaS cloud providers and user profiles are identified in this module.
- *Process Executor Module:* This component is responsible for the execution of the business process based on the details and requirements of all tasks. This module specifies the appropriate operations which should be executed to achieve the defined task. The activity of the process model is evaluated to choose and perform the appropriate ones for the current work-flow. This component also keeps track of all the activities and adds events to the workload queue.
- *IaaS Resource Discovery Module:* This module provides the functionality for IaaS cloud providers discovery. It would exploit information offered by semantic models and SLA agreements and the QoS specifications in order to find IaaS Cloud Resources in other available clouds which meet the current work-flow requirements.
- *IaaS Resource Selection Module:* Resource selection component selects appropriate providers from the network of cloud providers. This module considers information from SLA-QoS requirement module and discovered resource providers from Resource Search and Discovery Module to select the set of clouds for migrating and dispatching workloads. It also exploits the information from Model Manager to make the best suited selection.
- *Transformation Engine Module:* Transformation Engine performs the necessary model transformation to map the task details obtained in Model Manager as per the specifications of the selected IaaS resources that discovered and selected in Resource Discovered and Selected modules. It also uses the semantic module to make the necessary transformations.

Transformation Engine is the key component of our framework that can provide interoperability through mapping workload to other selected cloud providers.

- *Semantic Module:* Intercloud Semantic is the most essential module of the architecture with three components: Object Model, Operation Model, and Data Model. Semantic layer provides the functionality to maintain and utilize the semantic models that will be necessary to obtain interoperability.

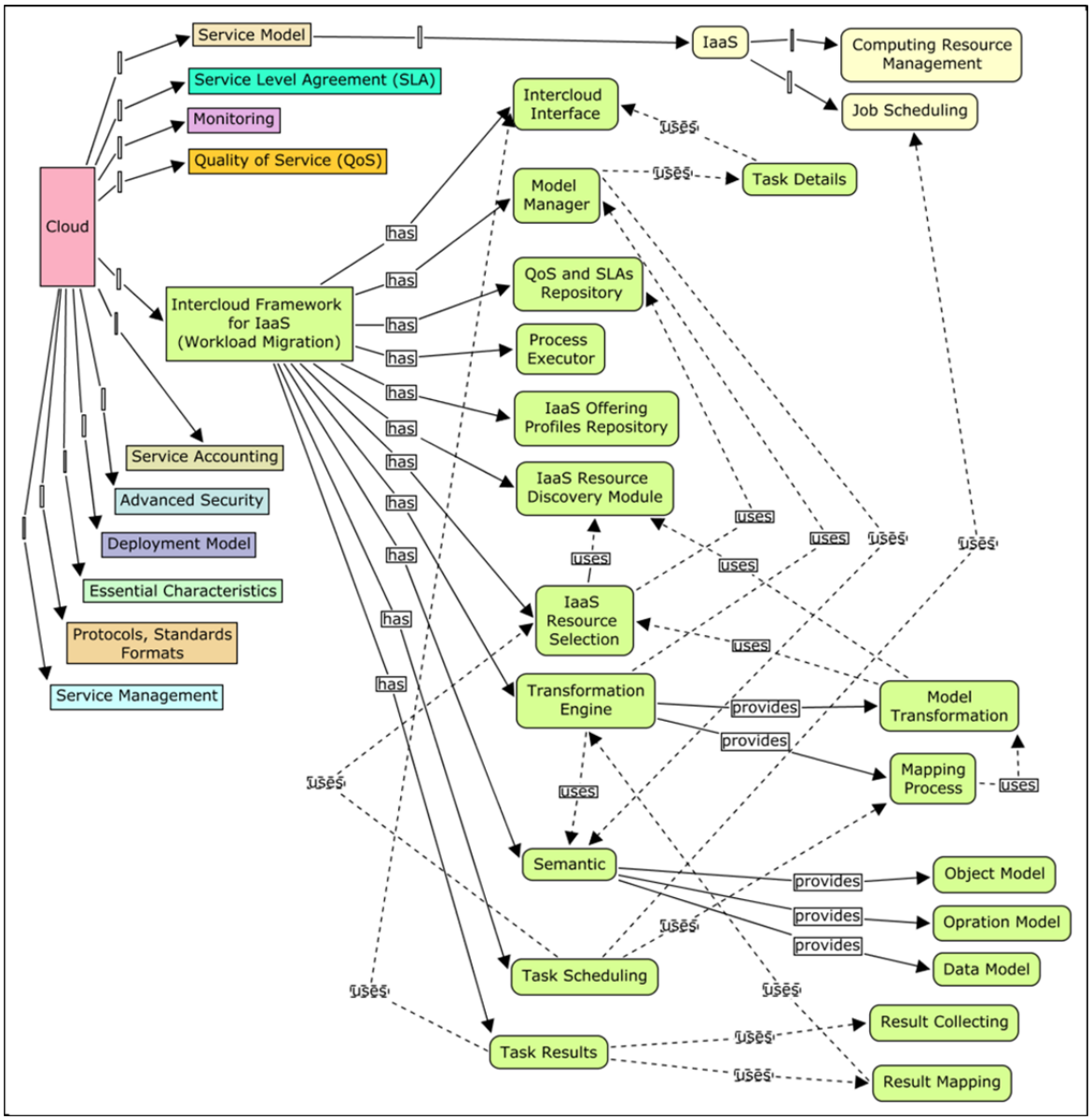


Figure 4: Intercloud Computing Generic Architecture.

- *Task Scheduling:* Considering the conceptual cloud architecture in Figure 1, this component exploits the job-scheduling techniques used in the host IaaS cloud to dispatch all tasks on the other selected IaaS clouds through IaaS Resource Selection component.
- *Task Results Module:* This module collects the results of performing the dispatched tasks from selected IaaS clouds, performs the necessary transformations and maps and sends back the results through an Interface component to host IaaS cloud.

# 4 MDA AND INTERCLOUD INTEROPERABILITY

The proposed framework is under validation through simulation experiments. In section 3, the Transformation Engine module is the key component of our framework that can provide interoperability through mapping workload to other selected cloud providers. Based on our research, a potential architecture for the implementation of Transformation Engine component is the Model

Driven Architecture (MDA) (Cretan et al. 2012). MDA is introduced by Object Management Group (OMG) as a software development approach to system-specification and interoperability based on the use of formal models (OMG 2003). MDA focuses on the development of models rather than platform-specific code which can be generated when needed through three levels of modeling abstractions (shown in Figure 5):

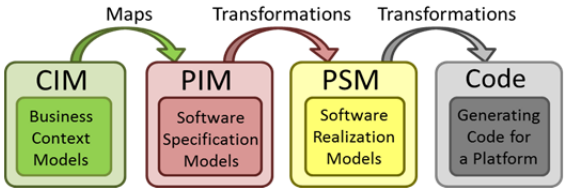


Figure 5: Model Driven Architectures levels.

- *Computation Independent Model (CIM)* represents what the business actually does or wants to do in future, independent of technology specifications.
- *Platform Independent Model (PIM)* provides a formal definition of the functionality of the software system and defines data, dependencies and architectural realizations.
- *Platform Specific Model (PSM)* provides the details that specify how the system uses a particular type of platform and ease generating corresponding code from the PIM that fits the operating platform.

Transformation techniques play a key role in making the MDA method successful. It can be categorized based on the type of source and destination they operate on (Jimenez 2005):

- *Code to Code*: Here the source and target are textual artifacts.
- *Model to Code*: This kind of transformation can produce source code from models, such as converting PSM to code corresponds to the model-to-code transformation.
- *Code to Model*: Code to model transformations generate models from textual representations.
- *Model to Model*: It 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.

Figure 6 shows the basic model transformation pattern applied at the model level to convert source model elements to target model elements. Source model and target model may represent the same data with two different formats.

Various transformation languages and tool suites have been developed, although most of them are at

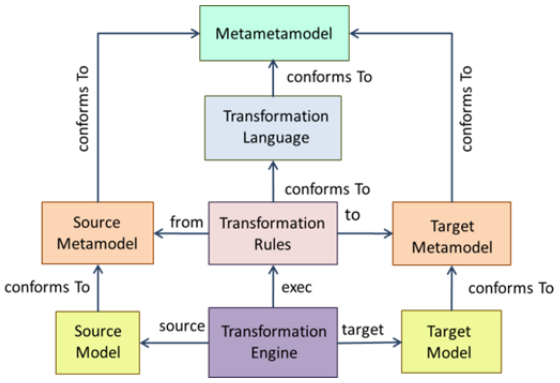


Figure 6: Model transformation pattern (Koch 2007).

experimental stage yet to be applied to industrial practice. For instance, Query/View/Transformation (QVT) is defined by the OMG to describe the requirements of a standard language for the specification of model transformation (OMG 2011a), or Graph Rewrite And Transformation language (GReAT) (Agrawal et al. 2005) is a metamodel-based graph transformation language that is designed to deal with the high-level complexity model transformation programs.

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 the real world.

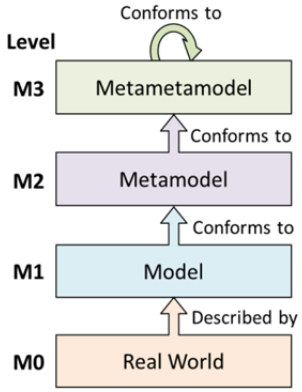


Figure 7: The four layer meta-modeling architecture.

In this regard, OMG has introduced a 4-layer

architecture called the MOF metamodeling stack (OMG 2011b) as shown in Figure 7 , 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.

# 5 VISION FOR INTERCLOUD INTEROPERABILITY FRAMEWORK FOR IAAS CLOUDS

This paper presented the required concepts and QoS-SLA criteria for a conceptual model for intercloud interoperability to address workload migration to IaaS clouds, which is identified by NIST as one major use case for cloud computing interoperability (Badger et al. 2010).

Transformation Engine is the key component of our framework that supports interoperability. As discussed in section 4, a potential architecture for the implementation of core Transformation Engine component is MDA. Figure 8 shows the overall picture for intercloud interoperability for IaaS clouds. Cloud "A" schedules and executes the specified workload on the other IaaS clouds and receives the results from them exploiting the interoperability framework. Application accesses the functionality of the framework through the interfaces defined by the framework.

As discussed in section 2, XMPP is a transport protocol and RDF is a standard model for data interchange both appropriate for the inter-cloud environments. The proposed conceptual model makes use of such standards and other communication infrastructure as the Transport infrastructure.

# 6 CONCLUSIONS

Intercloud Interoperability is tempting when individual clouds have limited computing resources in a restricted geographic area. Moreover, Intercloud Interoperability enables cloud providers to deliver better quality of services, avoid data lock-in, and reduce scaling/producing costs. Today, existing cloud environment does not fully support intercloud interoperability, and defining a conceptual model for

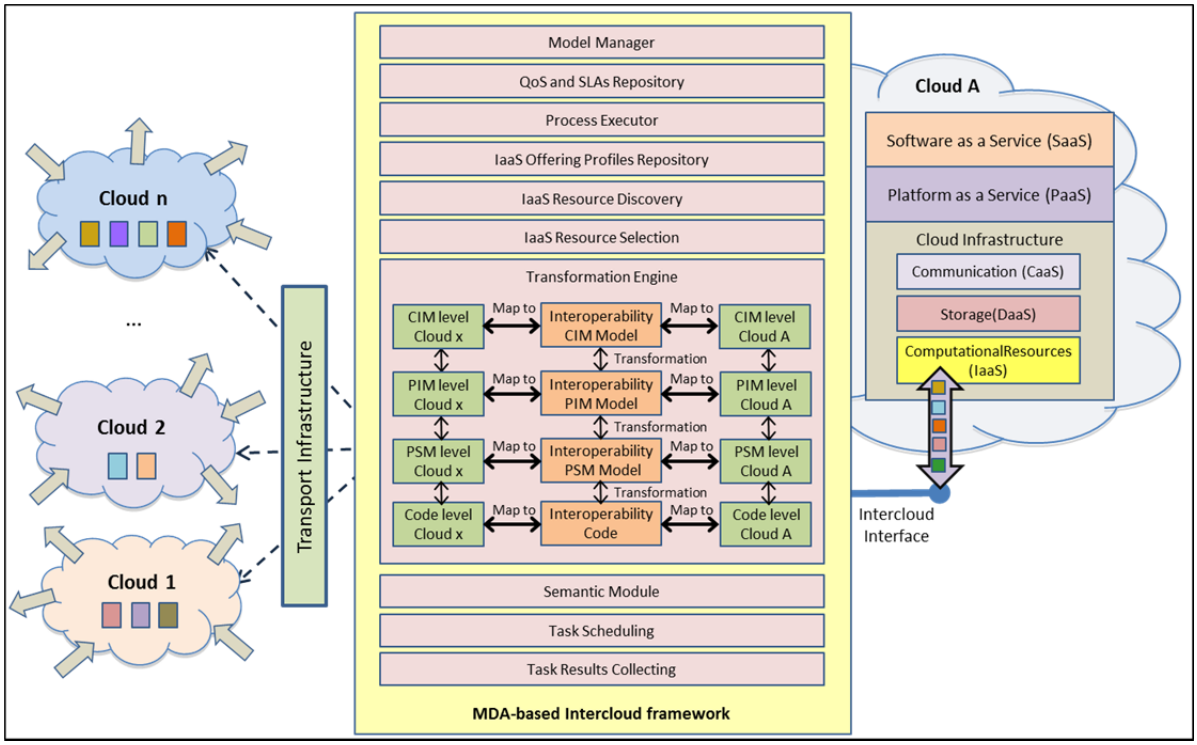


Figure 8: Vision for Intercloud Interoperability Framework.

Intercloud Interoperability to schedule dynamic operation for IaaS cloud providers is a requirement to achieve this objective.

Since, SLA and QoS models are important factors to introduce a comprehensive Intercloud Interoperability Framework, the paper refers to some useful references and proposes the essential metrics for SLA and QoS to integrate with cloud systems. Then, a conceptual model for the Intercloud Framework to dispatch workloads between various clouds is proposed. The Intercloud Framework receives tasks through Intercloud Interface; specifies the task details; considers all required SLA/QoS metrics; discovers and selects the available IaaS clouds; maps the task according to each offered cloud formats and requirements; dispatches the tasks between selected clouds; and finally collects and maps the results to an understandable format for the consumer cloud.

The Transformation Engine component of the framework is a core module to provide intercloud interoperability by mapping workloads to other cloud providers. MDA is identified as an appropriate approach to develop the Transformation Engine module.

As to continuing the work, it is planned to adjust the proposed framework according to the results of simulation experiments. Future work will expand the Intercloud framework to support interoperability in data migration and workload management between cloud providers.

## ACKNOWLEDGEMENTS

The authors sincerely acknowledge the financial support from the grant of the Portuguese Foundation of Science and Technology (FCT), the EC 7th Framework Programme under grant agreement n° FITMAN 604674 (http://www.fitman-fi.eu), and the Portuguese Projecto Mobilizador QREN PRODUTECH.

## REFERENCES

Agrawal, A. et al., 2005. Reusable Idioms and Patterns in Graph Transformation Languages. *Electronic Notes in Theoretical Computer Science*. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1571066105001143.

Amazon, Amazon Elastic Compute Cloud (Amazon EC2). Available at: http://aws.amazon.com/ec2/.

Badger, L. et al., 2010. Cloud Computing Use Cases. *National Institute of Standards and Technology,*. Available at: http://www.nist.gov/itl/cloud/use-cases.cfm.

Bernstein, D. et al., 2009. Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability. *2009 Fourth International Conference on Internet and Web Applications and Services*. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5072540(Accessed June 13, 2011).

Bernstein, D., 2009. The Intercloud : Cloud Interoperability at Internet Scale. *Sixth IFIP International Conference on Network and Parallel Computing*.

Bernstein, D. & Vij, D., 2010. Using XMPP as a transport in Intercloud Protocols. *2nd USENIX Workshop on Hot Topics in Cloud Computing*. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5577272 (Accessed November 11, 2012).

Celesti, A. et al., 2010. How to Enhance Cloud Architectures to Enable Cross-Federation. *IEEE 3rd International Conference on Cloud Computing*. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5557976 (Accessed November 9, 2012).

Coutinho, C., Cretan, A. & Jardim-Goncalves, R., 2013. Sustainable interoperability on space mission feasibility studies. *Computers in Industry*, 64(8), pp.925–937. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0166361513001309 (Accessed November 3, 2013).

Cretan, A. et al., 2012. NEGOSEIO: A framework for negotiations toward Sustainable Enterprise Interoperability. *Annual Reviews in Control36*, 36(2), pp.291–299. Available at: http://dx.doi.org/10.1016/j.arcontrol.2012.09.010 (Accessed November 3, 2013).

Demchenko, Y. et al., 2013. Intercloud Architecture Framework for Heterogeneous Cloud Based Infrastructure Services Provisioning On-Demand. *27th International Conference on Advanced Information Networking and Applications Workshops*, pp.777–784. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6550490 (Accessed November 11, 2013).

Goyal, M.K. et al., 2012. QoS based trust management model for Cloud IaaS. *2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing*, pp.843–847. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6449933.

Grance, T., 2010. The NIST Cloud Definition Framework. *National Institute of Standards and Technology (NIST)*. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:The+NIST+Cloud+Definition+Framework#3 (Accessed January 6, 2013).

Jardim-Goncalves, R., Agostinho, C., et al., 2012. Systematisation of Interoperability Body of Knowledge: The foundation for EI as a science. *Enterprise Information Systems Journal*.

Jardim-Goncalves, R., Popplewell, K. & Grilo, A., 2012. Sustainable interoperability: The future of Internet based industrial enterprises. *Computers in Industry*, 63(8), pp.731–738.

Jimenez, A.M., 2005. *Change propagation in the MDA: A model merging approach*. The University of Queensland. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Change+propagation+in+the+MDA:+A+model+merging+approach#0 (Accessed August 29, 2012).

Koch, N., 2007. Classification of model transformation techniques used in UML-based Web engineering. *Software, IET*. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4259295 (Accessed November 13, 2012).

Lewis, G.A., 2012. *The Role of Standards in Cloud-Computing Interoperability*,

Mell, P. & Grance, T., 2009. The NIST Definition of Cloud Computing Version 15. *National Institute of Standards and Technology (NIST)*.

Microsoft, Windows Azure, a rock-solid cloud platform for blue-sky thinking. Available at: http://www.windowsazure.com/en-us/.

Nodehi, T. et al., 2013. On MDA-SOA based Intercloud Interoperability framework. *Computational Methods in Social Sciences (CMSS)*.

OMG, 2003. MDA Guide Version 1.0.1. , (June). Available at: Group:http://www.omg.org/mda.

OMG, 2011a. Meta Object Facility ( MOF ) 2 . 0 Query / View / Transformation Specification. Available at: http://www.omg.org/spec/QVT/.

OMG, 2011b. *OMG Meta Object Facility ( MOF ) Core Specification (Version 2.4.1)*, Available at: http://www.omg.org/spec/MOF/2.4.1/PDF/.

Parameswaran, A. & Chaddha, A., 2009. Cloud interoperability and standardization. *SETlabs briefings*. Available at: http://www.infosys.com/infosys-labs/ publications/infosyslabs-briefings/documents/cloud-interoperability-standardization.pdf (Accessed April 30, 2012).

Rackspace, Rackspace Cloud. Available at: http://www.rackspace.com/cloud/.

Rubach, P. & Sobolewski, M., 2009. *Dynamic SLA Negotiation in Autonomic Federated Environments* R. Meersman, P. Herrero, & T. Dillon, eds., Springer.

Salama, M. et al., 2012. Integrated QoS Utility-Based Model for Cloud Computing Service Provider Selection. *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*, pp.45–50. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6341548 (Accessed November 27, 2013).

Sun, Y. et al., 2013. SLA detective control model for workflow composition of cloud services. *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp.165–171. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6580957.

Wang, G. et al., 2012. Compositional QoS Modeling and Analysis of Cloud-based Federated Ecosystems. *2012 IEEE 16th International Enterprise Distributed Object Computing Conference*, pp.173–182. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6337248 [Accessed November 27, 2013].