# Semantic Interoperability Framework for IAAS Resources in Multi-Cloud Environment

**Karima Benhssayen and Ahmed Ettalbi**

IT Architecture and Model Driven Systems Development Team, ADMIR Laboratory
ENSIAS, Mohammed V University in Rabat, Morocco

**Summary**

Cloud computing has proven its efficiency, especially after the increasing number of cloud services offered by a wide range of cloud providers, from different domains. Despite, these cloud services are mostly heterogeneous. Consequently, and due to the rising interest of cloud consumers to adhere to a multi-cloud environment instead of being locked-in to one cloud provider, the need for semantically interconnecting different cloud services from different cloud providers is a crucial and important task to ensure. In addition, considerable research efforts proposed interoperability solutions leading to different representation models of cloud services. In this work, we present our solution to overcome this limitation, precisely in the IAAS service model. This solution is a framework permitting the semantic interoperability of different IAAS resources in a multi-cloud environment, in order to assist cloud consumers to retrieve the cloud resource that meets specific requirements.

***Key words:***
*Cloud Computing, Semantic Interoperability, IAAS Resources, Multi-Cloud Environment.*

## 1. Introduction

In recent years, enterprises tend to migrate their IT infrastructure to the cloud because of its benefits including cost-saving, scalability, high availability of services, and the pay-as-you-go manner.

Cloud computing is defined by NIST (National Institute of Standards and Technology) as “a model for enabling ubiquitous, 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”. [1]

NIST defines three service models of cloud computing presented as SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service) in which cloud providers manage and control the underlying infrastructure including physical services, networking and data centers.

Along with SaaS and PaaS, IaaS is one of the most important service models of cloud computing, because of its advantages in helping organizations to focus on their business growth. Consequently, the IaaS cloud market has known a considerable growth, offering several services and resources with different architectures’ grounding.

Therefore, according to a survey conducted in 2019 [2], by the RightScale Cloud Industry Research Team in coordination with Flexera’s technology asset management solutions provider, 84% of companies have a multi-cloud strategy, which means they rely on a combination of different cloud providers and their own data centers to host their IT infrastructure.

As a result, cloud consumers need to have more flexibility in the usage of cloud resources from cloud providers of their choice, if they need to scale there resources to others from providers fitting their requirements.

To this end, we propose a framework permitting the semantic interoperability between IAAS resources in a multi-cloud environment. This solution aims at maximizing the semantic interoperability between IaaS resources, thus, helping cloud consumers to have more visibility on available cloud resources fitting their requirements.

The rest of this paper is organized as follows: Section two provides existing definitions of interoperability and semantic interoperability in cloud computing environment. Section three presents an overview of the technology used. Section four illustrates the problematic and the related work. In Section five, we present the architecture, the sequence diagram and the benefits of our proposed framework. Lastly, we conclude our paper with perspectives and future work.

Manuscript received February 5, 2021
Manuscript revised February 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.2.1

## 2. Interoperability and semantic interoperability definitions in the cloud computing environment

### 2.1 Interoperability definition

In order to better understand the semantic interoperability definition in the cloud computing environment, it is important to understand firstly what interoperability means. Consequently, several definitions were proposed, either in industrial and research field, we cite from them the following definitions:

- Academic Definition [3]:"we interpret interoperability as the ability to federate multiple clouds to support a single application. In other words, interoperability involves software and data simultaneously active in more than one cloud infrastructure, interacting to serve a common purpose"

- Industrial Definition OMG [4]:"In the context of cloud computing, interoperability should be viewed as the capability of public cloud services, private cloud services, and other diverse systems within the enterprise to understand each other's application and service interfaces, configuration, forms of authentication and authorization, data formats, etc. in order to work with each other."

### 2.2 Semantic interoperability definition

Concerning the semantic interoperability definition, there are few definitions proposed especially in the academic research field. We cite from them the following one:

- "Semantic Interoperability is defined as automatically interpretation of the information exchanged meaningfully and accurately, in order to produce useful results (using a common information exchange reference model".[5]

## 3. Overview of technology used

### 3.1 RDF

RDF (Resource Description Framework): is a World Wide Web Consortium (W3C) semantic web standard that defines a language for describing resources (anything that can be identified on the web using URIs: physical objects, abstract concepts, etc) and relationships between them [6], thus insuring their publication and linking over the web.

RDF is a widely used standard, based essentially in triple model, in which resources are described by triples in the following format:

**<Subject> <Predicate> <Object>**: where Subject and Object are two resources identified by URIs, and Predicate is the property describing the relationship between them.

Example:

<http://example.org/article.pdf> <is published by> <http://journal.org/ >.

RDF triples can be also presented as graphs, as follows:

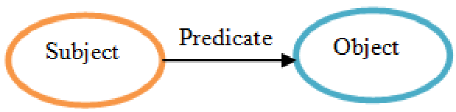


**Fig. 1:** Graph representation of RDF triples

### 3.2 OWL

OWL (Web Ontology Language): is a semantic web language introduced by W3C, and is developed as a vocabulary extension of RDF [7],in order to produce ontologies' documents that can be published over the web, and then referred by other ontologies.

An OWL ontology document "describes a domain in terms of classes, properties and individuals and may include rich descriptions of the characteristics of those objects."[8]

### 3.3 SPARQL

SPARQL (Sparql Protocol and RDF Query Language): is proposed by W3C in order to query and manipulate linked data stored in RDF databases.

SPARQL query language has a very simple and robust syntax, very similar to SQL. It has the power to access complicated databases using a full set of query operations such as SORT, JOIN, FILTER, etc.

### 3.4 Ontology mapping

Marc Ehrig and York Sure define ontology mapping as follow: "Given two ontologies A and B, mapping one ontology with another means that for each concept (node) in ontology A, we try to find a corresponding concept (node), which has the same or similar semantics, in ontology B and vice versa."[9]

## 4. Problem definition and related work

### 4.1 Problem definition

IaaS service model is one of the most important service models of cloud computing attracting more attention by researchers trying to answer different questions, such as, security, discovery and selection, management and so on. In addition, the interoperability research field in the cloud computing environment has known a considerable growth in the last few years. Consequently, several solutions were proposed targeting the IaaS layer, such as ontologies, like mOSAIC ontology [10], permitting the description of concepts related to IaaS services, and management standards providing description models for IaaS resources, including OCCI [11] (Open Cloud Computing Interface) and CIMI (Cloud Infrastructure Management Interface) [12].

Also, and to become more competitive, cloud providers use their own APIs, describing resources with different terminologies, thus forcing cloud consumers to use only these APIs, which confront them to the vender lock-in problem [13].

In addition, cloud consumers tend to use a multi-cloud environment instead of being locked-in to one cloud provider. Consequently, and because of the different description models proposed of IaaS resources, we find ourselves against another problem, which is how to semantically interconnect these different description models, especially in a multi-cloud environment.

Furthermore, to the best of our knowledge, there is no proposed solution trying to semantically interconnect IaaS resources in a multi-cloud environment where we have different representation models, including standardized resources and cloud providers' ones.

### 4.2 Related work

- Yongsiriwit et al. [14] presented a new framework permitting the semantic interoperability between heterogeneous Iaas cloud resources supporting the following standards: TOSCA, OCCI and CIMI. The framework defines three specific ontologies related to each standard using the OWL ontology; sTOSCA, sOCCI and sCIMI. Also, it allows the abstraction of the aforementioned specific ontologies using Linked-CR ontology, thus assuring their mapping using SWRL language (ex: sOCCI->Linked-CR and vice-versa). A public knowledge base is created in order to store the modeled triples of all used ontologies, thus facilitating the finding of resources using the SPARQL query language. A quantitative and qualitative evaluation of the framework is illustrated, in order to demonstrate the facility and the effectiveness of the approach.

- Di Martino et al [15] presented a scalable architecture for semantically interoperable resources at the IaaS level. This architecture relies on the IEEE P2302 standard for inter-cloud interoperability and federation. The goal is to ensure the information exchange based on an inter-cloud ontology which uses semantic web technologies such as OWL and SPARQL, and which takes into consideration economic entities by proposing a commercial model. In order to demonstrate the applicability of the architecture, a proof-of-concept prototype is illustrated based on an optimistic approach. According to this last, cloud consumer initiates a RESTful API call to the inter-cloud Resource Exchange for a desired inter-cloud resource. The inter-cloud Resource Exchange constructs then the related SPARQL Queries which then issued against the resource repository placed in the same component (inter-cloud Resource Exchange). Once the cloud provider verifies the received request it reserves the resources to be consumed by cloud consumer.

- Di Martino et al [16] proposed another solution which is a multi-layer ontology architecture in order to semantically model cloud resources and services. The architecture is composed of three mainly layers presented as follows: The agnostic layer contains the agnostic service description ontology in order to represent services, resources, methods and their parameters in an abstract way. The service categorization layer contains the cloud services categorization ontology in order to classify services and resources in a provider-centric way. At last, we find the proprietary layer, which contains several cloud providers' ontologies, and their representation in an OWL-S ontology. All these aforementioned ontologies are interrelated with each other, in such a way to facilitate their interrogation using SPARQL queries. The semantic web technologies used by the authors are: OWL, OWL-S, SPARQL and SWRL.

- Francesco Moscato et.al [10] presented the mOSAIC ontology which aims at solving the semantic interoperability problem in the cloud computing environment by permitting the description of concepts related to services of different deployment models (IaaS, PaaS and SaaS). The ontology is developed using OWL and some existing standards like NIST and IBM proposals, and can be used for semantic retrieval and composition of cloud services in the mOSAIC project. An example of the mOSAIC ontology is illustrated in order to describe a simple data storage service implemented with Google App Engine.

## 5. The Proposed Framework

### 5.1 General presentation of the architecture

The proposed solution is a framework aiming at maximizing the level of semantic interoperability between IaaS resources, in a multi-cloud environment. It consists of:

- Components interacting with cloud providers (IaaS Resource Requestor) and cloud consumers (IaaS Resource Request Listener components),

- Components ensuring the storage of received cloud providers' responses (Resource Persistency), and their mapping using standards' ontologies (Resource Mapping components)

- Components used to store cloud providers' responses and their corresponding ontologies (Knowledge Base Component), and the resulting RDF triples after the ontology mapping step (Resource DataBase Component).

Figure 2 presents the general architecture of the proposed framework and figure 3 describes the flow of possible interactions between components of this framework.

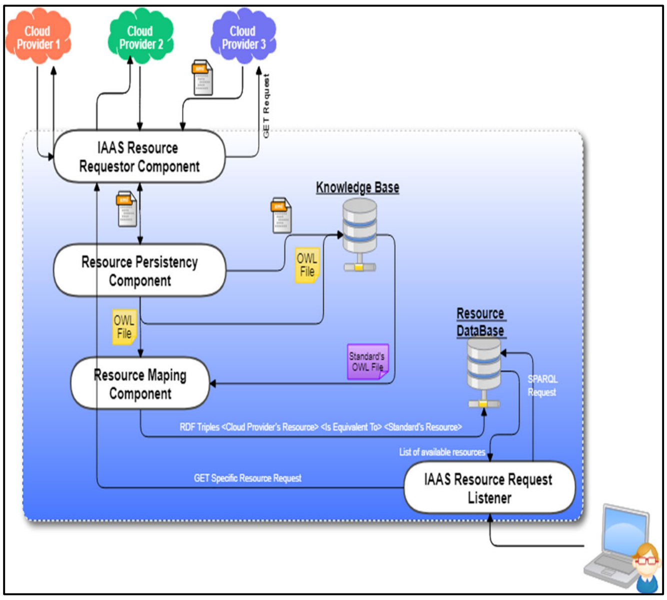


**Fig. 2:** IaaS Resource Semantic Interoperability Architecture

### 5.2 Description of components

#### a) IAAS Resource Requestor Component

Constitutes the interface of the framework with cloud providers. Its main functions are:

- Request available IaaS resources from cloud providers periodically.
- Send the received responses (XML format) to the IaaS Resource Persistency component in order to create OWL related ontologies.
- Get requests from the IaaS Resource Request Listener component looking for specific resources meeting cloud consumer's requirements, if no available results are found in the Resource Database.

#### b) Resource Persistency Component

Permits the persistency of received XML responses, and their OWL ontologies following the steps bellow:

1- Get the XML responses sent from cloud providers to the IaaS Resource Requestor Component,
2- Foreach XML file, it checks the Knowledge Base,
3- If the XML file already exists, the component proceeds to the comparison process to verify eventual updates of related IaaS resources,
4- If there are no changes between the two versions of XML files, it ignores the received XML file and skips to step 2 to continue with the next XML file,
5- Else it updates the existing XML file by the new one,
6- Else, it stores the XML file in the Knowledge Base component,
7- Annotate the XML file using the OWL ontology language in order to create ontology related to each cloud provider,
8- Store the aforementioned ontologies in the Knowledge Base Component and send a copy of each created ontology to the Resource Mapping Component.

#### c) Resource Mapping component

- Receives the OWL ontology related to each new or updated cloud provider response from the IaaS Resource Persistency Component
- Proceeds to the mapping of each OWL ontology using the IAAS Standards' ontologies stored in the Knowledge Base Component such as OCCI ontology, in order to retrieve the corresponding semantic description of each cloud resource in the standard's ontologies (OCCI, mOSAIC ...).
- The findings of the previous step are stored in the Resource DataBase as RDF triples:
<Cloud Provider's Resource> <Is Equivalent To> <Standard's Resource>

d) Knowledge Base Component

- Is used to store the XML responses files and their related OWL ontologies with other useful information such as cloud provider name, in order to facilitate the access of the requested resources by cloud consumers.
- It contains also ontologies related to available IaaS Standards, such as OCCI and mOSAIC ontologies.

e) Resource DataBase Component

- Is an RDF DataBase containing available equivalences between cloud providers' resources and standard's ones:

<Cloud Provider's Resource> <Is Equivalent To> <Standard's Resource>.

- It contains also equivalences between cloud standard's resources. For example: **< OCCI Resource><Is Equivalent To> < mOSAIC Resource>.**

f) IAAS Resource Request Listener Component

Constitutes the interface of the framework with cloud consumers. Its main functions are:

- Get requests sent by cloud consumer trying to find available resources with specific requirements.
- Verify if the requested resources exist in the Resource Database Component using SPARQL queries.
- If there are no available resources, it requests the IaaS Resource Requestor Component in order to get new or updated resources with desired requirements.
- Else, it returns all available resources to the consumer, with other information such as cloud provider name and the link to access available resource.

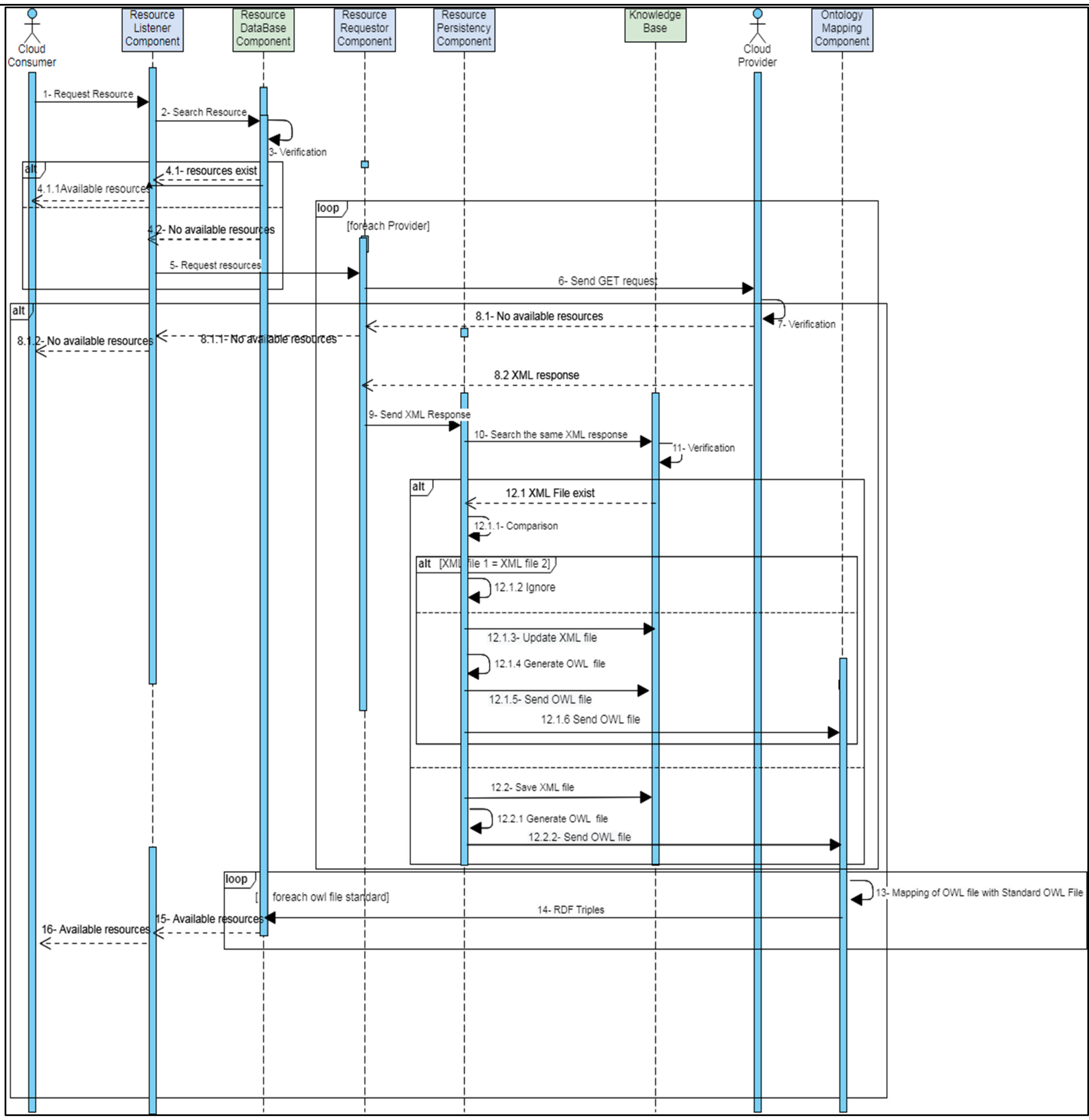


**Fig 3:** Sequence Diagram of proposed architecture

## 5.3 Benefits of the proposed framework

a) Fast Retrieval of resources

Imagine a use case where cloud consumer is using different solutions from different cloud providers. Using resource A from cloud provider CPA is no more satisfactory for cloud consumer, who needs to scale to another resource in order to make his application more efficient.

By using our proposed framework, cloud consumer can request the cloud resource he wants to scale to, and the framework lists to him all available resources from different cloud providers responding the expressed requirements.

b) Different description models

The framework is designed to support all description models of IaaS resources existing in the cloud computing market, and proposed by research efforts targeting the semantic interoperability in the IaaS service model. Consequently, the knowledge base can be updated in order to support new standards' ontologies, in addition of its periodically updates based on cloud providers' ontologies.

c) Periodic update of resources

The IaaS Resource Requestor Component requests periodically cloud providers in order to send their available resources. These lasts can include also updated resources in addition to the new ones which are all stored in the Knowledge Base by the IaaS Resource Persistency Component.

d) Discovery and selection

OWL, RDF and SPARQL query languages are widely used and efficient semantic web languages, permitting an easy access of linked data even in a more complicated use cases. Consequently, queried data is fatly returned to cloud consumer who can choose the access of the desired resource easily, by using its URI returned in the SPARQL query.

## 6. Conclusion and future work

The growing interest of cloud consumers to adhere to a multi-cloud environment reveals several concerns including interoperability of cloud services of SaaS, PaaS, and IaaS service models. Consequently, several research efforts including standards such as OCCI and CIMI were proposed in order to deal with the interoperability problem of IaaS resources, providing different representation models. As a result, the need to semantically interconnect IaaS resources in a multi-cloud environment is a crucial and important task to ensure. In this vein, we have proposed in this paper our framework composed of different components, interacting with each other in order to assume the fast retrieval of requested resources from cloud consumers with specific requirements.

The components of the framework are: IaaS Resource Requestor and IaaS Resource Request Listener Components interacting mainly with cloud providers in order to get available cloud resources and cloud consumers with the aim of sending available resources with requested requirements respectively, Resource Persistency and Resource Mapping Components permitting the persistence of XML response files and their related OWL ontologies in the Knowledge Base Component, and the result of ontology mappings between OWL file response and OWL standard's ontologies in the Resource DataBase Component.

The framework offers several benefits including, fast retrieval and periodic update of resources in addition to discovery and selection. For our future work, we tend to develop each component with implementation and a case study illustration.

## References


[1] P. Mell and T. Grance, 'The NIST Definition of Cloud Computing', NIST Special Publication 800-145, p. 7, September 2011.

[2] 'RightScale 2019 State of the Cloud Report from Flexera', p. 50, 2019.

[3] K. Oberle and M. Fisher, 'ETSI CLOUD – Initial Standardization Requirements for Cloud Services', in *Economics of Grids, Clouds, Systems, and Services*, vol. 6296, J. Altmann and O. F. Rana, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 105–115.

[4] OMG 'Interoperability and Portability for Cloud Computing: A Guide Version 2.0', p. 39, 2017.

[5] D. Petcu, 'Portability and Interoperability between Clouds: Challenges and Case Study', in *Towards a Service-Based Internet*, vol. 6994, W. Abramowicz, I. M. Llorente, M. Surridge, A. Zisman, and J. Vayssière, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 62–74.

[6] B. McBride, 'The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS', in *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 51–65.

[7] 'OWL Web Ontology Language Reference', W3C Recommendation, p. 80, 10 February 2004.

[8] S. Bechhofer, 'OWL: Web Ontology Language', in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 2008–2009.

[9] M. Ehrig and Y. Sure, 'Ontology Mapping – An Integrated Approach', in *The Semantic Web: Research and Applications*, vol. 3053, C. J. Bussler, J. Davies, D. Fensel, and R. Studer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 76–91.

[10] F. Moscato, R. Aversa, B. Martino, T.-F. Fortis, and V. Munteanu, 'An Analysis of mOSAIC ontology for Cloud Resources annotation', p. 8, 2011.

[11] T. Metsch, A. Edmonds, and R. Nyren, 'Open Cloud Computing Interface - Core', p. 17, October 2010.

[12] 'Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol', p.

178 ,2012. Available at https://www.dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.1.pdf
[13] M. Toivonen, 'Cloud Provider Interoperability and Customer Lock-in', In Proceedings of the seminar (No. 58312107, pp. 14–19),2013-08-05.
[14] K. Yongsiriwit, M. Sellami, and W. Gaaloul, 'A Semantic Framework Supporting Cloud Resource Descriptions Interoperability', in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, Jun. 2016, pp. 585–592, doi: 10.1109/CLOUD.2016.0083.
[15] B. Di Martino *et al.*, 'Towards an Ontology-Based Intercloud Resource Catalogue -- The IEEE P2302 Intercloud Approach for a Semantic Resource Exchange', in *2015 IEEE International Conference on Cloud Engineering*, Tempe, AZ, Mar. 2015, pp. 458–464, doi: 10.1109/IC2E.2015.76.
[16] B. D. Martino, G. Cretella, A. Esposito, and G. Carta, 'An OWL ontology to support cloud portability and interoperability', *IJWGS*, vol. 11, no. 3, p. 303, 2015, doi: 10.1504/IJWGS.2015.070972.

**Karima BENHSSAYEN** PhD student in the IMS (IT Architecture and Model Driven Systems Development) Team of the ADMIR (Advanced Digital Enterprise Modeling And Information Retrieval) Laboratory AT the Higher National School for Computer Science and Systems Analysis (ENSIAS), Rabat, Morocco. Her research interests include Semantic Interoperability in the Cloud Computing.

**Ahmed ETTALBI** Professor at Software Engineering Department and member of the IMS (IT Architecture and Model Driven Systems Development) Team, ADMIR (Advanced Digital Enterprise Modeling And Information Retrieval) Laboratory of the Higher National School of Computer Science and Systems Analysis (ENSIAS) Rabat. His main research interests include Cloud Computing, WebServices, Object Modeling with Viewpoints and Software Oriented Architectures.