D1.2
Cloud4SOA Cloud Semantic Interoperability Framework
Version 1.0 - 02/06/2011

WP 1 Cloud4SOA Reference Framework
Dissemination Level: Public

Lead Editor: Nikos Loutas, CERTH

Status: Final

Seventh Framework Programme of the European Community

Proposal/Contract no.: 257953
Context

<table>
<thead>
<tr>
<th>WP 1.1</th>
<th>Cloud4SOA Reference Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1.2</td>
<td>Cloud4SOA Cloud Semantic Interoperability Framework</td>
</tr>
</tbody>
</table>
| Dependencies | This deliverable studies and analyzes the semantic interoperability problems that are raised in the Cloud. Then, the semantic interoperability issues/concerns are categorized in different levels defining the Cloud4SOA Cloud Semantic Interoperability Framework. Additionally, the Cloud4SOA Cloud Semantic Interoperability Framework includes a roadmap comprising of a set of guidelines and good practices for building interoperable Cloud platforms.

It will be the basis for WP2 through WP7. |

Contributors: N. Loutas, E. Kamateri, K. Tarabanis (CERTH), Francesco D’Andria (Atos Origin)

Reviewers: P. Gouvas (SingularLogic), S. Bocconi (Cyntelix), F. Bosi (NUIG),

Approved by: QM

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Authors</th>
<th>Sections Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>10/01/11</td>
<td>N. Loutas, K. Tarabanis</td>
<td>ToC</td>
</tr>
<tr>
<td>0.2</td>
<td>24/01/11</td>
<td>E. Kamateri</td>
<td>All</td>
</tr>
<tr>
<td>0.3</td>
<td>24/02/11</td>
<td>N. Loutas</td>
<td>Comments and restructuring</td>
</tr>
<tr>
<td>0.4</td>
<td>18/03/11</td>
<td>E. Kamateri</td>
<td>All</td>
</tr>
<tr>
<td>0.5</td>
<td>24/03/11</td>
<td>N. Loutas</td>
<td>Second round of comments and revisions</td>
</tr>
<tr>
<td>0.6</td>
<td>27/04/11</td>
<td>E. Kamateri</td>
<td>All</td>
</tr>
<tr>
<td>0.7</td>
<td>03/05/11</td>
<td>N. Loutas</td>
<td>Edited all sections.</td>
</tr>
<tr>
<td>0.8</td>
<td>11/5/11</td>
<td>E. Kamateri</td>
<td>Updated Executive Summary, Introduction, and Conclusions</td>
</tr>
<tr>
<td>0.9</td>
<td>12/5/11</td>
<td>N. Loutas</td>
<td>Major revision. Preparation of final version and sent for internal review.</td>
</tr>
<tr>
<td>1.0</td>
<td>02/06/11</td>
<td>E. Kamateri, N. Loutas</td>
<td>Final version to be submitted to the EC.</td>
</tr>
</tbody>
</table>
# Table of Contents

Executive Summary .......................................................................................................................... 10

1. Introduction .................................................................................................................................... 11
   1.1. Document Scope .................................................................................................................. 11
   1.2. Methodology ............................................................................................................................ 11
   1.3. Overview ................................................................................................................................. 12

2. Information Systems Semantic Interoperability ........................................................................... 14
   2.1. Semantic Interoperability ......................................................................................................... 14
   2.2. Interoperability Frameworks and Models .................................................................................. 15
   2.3. Discussion .................................................................................................................................. 23

3. Cloud Computing Semantic Interoperability ............................................................................... 26
   3.1. Cloud PaaS Interoperability .................................................................................................... 26
   3.2. Standardization Activities ........................................................................................................ 28
   3.3. Cloud Semantic Interoperability Frameworks ........................................................................... 30
      3.3.1. Levels of Interoperability ............................................................................................... 31
      3.3.2. Types of Semantic Conflicts .......................................................................................... 32
      3.3.3. Types of Solutions ........................................................................................................... 33
   3.4. Discussion and Findings ........................................................................................................... 37

4. Cloud4SOA Cloud Semantic Interoperability Framework ............................................................. 43
   4.1. Basic Definitions ...................................................................................................................... 43
   4.2. CSIF Core dimensions ................................................................................................................ 43
      4.2.1. Fundamental PaaS Entities ............................................................................................ 44
      4.2.2. Types of Semantics ........................................................................................................... 46
      4.2.3. Levels of Semantic Conflicts .......................................................................................... 48

5. Cloud Computing Semantic Interoperability Use Cases .............................................................. 49
   5.1. End-user to Cloud: Application Deployment on a PaaS offering .............................................. 49
   5.2. Changing Cloud vendors: Migrating an application to a different PaaS offering ..................... 50
   5.3. Hybrid Clouds: PaaS systems/offers interoperation ................................................................. 50
   5.4. Enterprise to Cloud to Enterprise: Interoperation between applications deployed on different PaaS offerings.................................................................................................................. 51

6. Good Practices for Semantically Interoperable Cloud PaaS ......................................................... 54

7. Conclusions ........................................................................................................................................ 57

References ............................................................................................................................................... 58

Annex A. Full license ............................................................................................................................. 63
   A.1. License ......................................................................................................................................... 63
   A.1.1. Definitions .................................................................................................................................. 63
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1.2. Fair Use Rights</td>
<td>64</td>
</tr>
<tr>
<td>A.1.3. License Grant</td>
<td>64</td>
</tr>
<tr>
<td>A.1.4. Restrictions</td>
<td>64</td>
</tr>
<tr>
<td>A.1.5. Representations, Warranties and Disclaimer</td>
<td>65</td>
</tr>
<tr>
<td>A.1.6. Limitation on Liability</td>
<td>65</td>
</tr>
<tr>
<td>A.1.7. Termination</td>
<td>65</td>
</tr>
<tr>
<td>A.1.8. Miscellaneous</td>
<td>65</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1 Implementing through a Framework [1] ................................................................. 12
Figure 2 The Cloud4SOA CSIF Definition ........................................................................... 12
Figure 3 ATHENA Interoperability Framework [18] ............................................................. 16
Figure 4 AIF Conceptual Framework ..................................................................................... 17
Figure 5 IDEAS Interoperability Framework [19] ................................................................. 18
Figure 6 LISI Reference Model [20] ..................................................................................... 19
Figure 7 Enterprise Interoperability Framework [21] [22] ...................................................... 20
Figure 8 GridWise Interoperability Context-Setting Framework [17] ................................... 21
Figure 9 Perspective on Heterogeneity and Types of Interoperability [23] ......................... 21
Figure 10 Enterprise Interoperability and Collaboration (EI) and Digital Ecosystems (DE) clusters to be merged into Future Internet Enterprise Systems (FIInES) Cluster [24]........... 22
Figure 11: Interoperability Dimensions Summary ................................................................. 23
Figure 12 Partitioning of the Modeling Space for Clouds .................................................... 32
Figure 13 Four types of semantics for an Application ......................................................... 33
Figure 14 A new Cloud computing model [35] ................................................................. 34
Figure 15: Core and proprietary capabilities [47] ............................................................. 34
Figure 16: NIST model for Cloud capabilities ..................................................................... 35
Figure 17: Cloud4SOA CSIF ............................................................................................... 44
Figure 18 A Basic PaaS System Model ............................................................................... 45
Figure 19 Semantics of Core Artifacts .............................................................................. 46
Figure 20 End-user to Cloud use case [33] ........................................................................ 50
Figure 21 Changing Cloud vendors use case [33] .............................................................. 50
Figure 22 Hybrid Clouds use case [33] ............................................................................. 52
Figure 23 Enterprise to Cloud to Enterprise use case [33] ............................................... 53
Figure 24 A Standardized API ............................................................................................ 55
List of Tables

Table 1 Layers of Interoperability ................................................................. 24
Table 2 Semantic Interoperability Frameworks ............................................. 25
Table 3 Interoperability – Standardization Bodies Initiatives ...................... 30
Table 4 Analysis of Cloud Computing Interoperability Frameworks ............ 39
Table 5: Approaches for Addressing Semantic Interoperability .................. 41
### Abbreviations

A table with used abbreviations is strongly recommended

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIF</td>
<td>ATHENA Interoperability Framework</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BPEL</td>
<td>Business Process Execution Language</td>
</tr>
<tr>
<td>C4ISR</td>
<td>Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance</td>
</tr>
<tr>
<td>CCE</td>
<td>Cloud Computing Environment</td>
</tr>
<tr>
<td>CCIF</td>
<td>Cloud Computing Interoperability Forum</td>
</tr>
<tr>
<td>Cloud4SoA:</td>
<td>A Cloud Interoperability Framework and Platform for user-centric, semantically-enhanced service-oriented applications design, deployment and distributed execution</td>
</tr>
<tr>
<td>CMWG</td>
<td>Cloud Management Working Group</td>
</tr>
<tr>
<td>CSIF</td>
<td>Cloud Semantic Interoperability Framework</td>
</tr>
<tr>
<td>CSP</td>
<td>Cloud Service Provider</td>
</tr>
<tr>
<td>DE</td>
<td>Digital Ecosystems</td>
</tr>
<tr>
<td>DMTF</td>
<td>Distributed Management Task Force</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>EFII</td>
<td>European Future Internet Initiative</td>
</tr>
<tr>
<td>EI</td>
<td>Enterprise Interoperability</td>
</tr>
<tr>
<td>EIF</td>
<td>European Interoperability Framework</td>
</tr>
<tr>
<td>ENISA</td>
<td>European Network and Information Security Agency</td>
</tr>
<tr>
<td>eTOM</td>
<td>enhanced Telecom Operations Map</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>FInES</td>
<td>Future Internet Enterprise Systems</td>
</tr>
<tr>
<td>FR</td>
<td>Functional Requirements</td>
</tr>
<tr>
<td>GICTF</td>
<td>Global Inter-Cloud Technology Forum</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>GWAC</td>
<td>GridWise Architecture Council</td>
</tr>
</tbody>
</table>

© Cloud4SoA consortium
IaaS  
Infrastructure as a Service

ICT  
Information and Communication Technology

IDEAS  
Interoperability Developments for Enterprise Application and Software - roadmaps

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

LISI  
Levels of Information Systems Interoperability

NFR  
Non-Functional Requirements

NIST  
National Institute of Standards and Technology

NoE  
Network of Excellence

OGF  
Open Grid Forum

OMG  
Object Management Group

OS  
Operating System

OVF  
Open Virtualization Format

PaaS  
Platform as a Service

PAID  
Procedures, Applications, Infrastructure and Data

QoS  
Quality of Service

RDF  
Resource Description Framework

REST  
Representational State Transfer

SaaS  
Software as a Service

SID  
Shared Information/Data Model

SLAs  
Service Level Agreements

SMEs  
Small and Medium Enterprises

SOA  
Service Oriented Architecture

SOAP  
Simple Object Access Protocol

TAM  
Telecom Application Map

UCC  
Unified Cloud Computing

UCI  
Unified Cloud Interface

US  
United States
<table>
<thead>
<tr>
<th>VM</th>
<th>Virtual Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS</td>
<td>Web services</td>
</tr>
</tbody>
</table>
Executive Summary

The present document is Deliverable 1.2 “Cloud4SOA Cloud Semantic Interoperability Framework” (henceforth referred to as D1.2) of the Cloud4SOA project. The main objective of this document is to deliver an open, innovative and generic framework for facilitating the implementation and deployment of semantically interoperable Cloud platforms. To make this possible, a comprehensive study and analysis of semantic interoperability problems that are raised when heterogeneous Cloud platforms try to exchange information is necessary. This process will result in a topology of semantic interoperability problems that are raised in Cloud computing environments and, thus, in the definition of the Cloud4SOA Cloud Semantic Interoperability Framework (CSIF).

The development of the Cloud4SOA CSIF constitutes the first step towards the creation of the Cloud4SOA system.

Cloud computing interoperability does not differ in principle from traditional system’s interoperability. In this light, D1.2 starts with a review of the existing interoperability frameworks that will form the basis for the development of the Cloud4SOA CSIF providing significant knowledge about how interoperability has been already tackled. Meanwhile, several definitions illustrate the concept of semantic interoperability and how it is perceived in information systems.

The definition of the Cloud4SOA CSIF relies heavily on the work carried out in D.1.1. More specifically, the extended review of the State-of-the-Art in the research fields covered by Cloud4SOA identified gaps, deficiencies, needs and problems relative to Cloud semantic computing interoperability. However, the establishment of a semantic interoperability framework requires a more in-depth analysis of Cloud computing semantic interoperability issues. A deep investigation of semantic conflicts between heterogeneous PaaS offerings is performed by standardization bodies. At the same time, several studies can be found in the literature that discuss and try to address semantic conflicts in the field of Cloud PaaS. D1.2 reviews and analyzes these works in order to build an all-inclusive and comprehensive semantic interoperability framework for Cloud PaaS. Specifically, the Cloud4SOA CSIF aims to identify where semantic conflicts arise, categorize them and propose solutions.

Once these two parallel steps were completed, Cloud4SOA CSIF is structured. Cloud4SOA introduces a three-dimensional semantic interoperability framework that encloses any semantic incompatibility conflict arising at PaaS layer while it enables each semantic conflict to be matched with the appropriate PaaS entity and the type of semantics. Last, the Cloud4SOA CSIF introduces a set of guidelines and good practices for building interoperable Cloud PaaS systems.

The Cloud4SOA CSIF includes a set of definitions for basic concepts, such as PaaS semantic interoperability, and is structured according to the following core dimensions:

1. **Fundamental PaaS Entities**, i.e. PaaS System, PaaS offering, API and Application.
2. **Types of Semantics**, i.e. functional, non-functional, execution.
3. **Levels of Semantic Conflicts**, i.e. information model and data.
1. Introduction

The aim of this section is to present the background of the work pursued during Task 1.2. The scope and the main objectives which have guided this work are introduced in section 1.1. The methodology followed is described in section 1.2. Last, section 1.3 presents the organization of the current deliverable.

1.1. Document Scope

The present document is Deliverable 1.2 “Cloud4SOA Cloud Semantic Interoperability Framework” (henceforth referred to as D1.2) of the Cloud4SOA project. The main objective of this document is to deliver an open, innovative and generic framework for facilitating the implementation and deployment of semantically interoperable Cloud platforms. To make this possible, a comprehensive study and analysis of semantic interoperability problems that are raised when heterogeneous Cloud platforms try to exchange information is necessary. This process will result in a topology of semantic interoperability problems that are raised in Cloud computing environments and, thus, in the definition of the Cloud4SOA Cloud Semantic Interoperability Framework (CSIF). Additionally, the Cloud4SOA CSIF includes a set of guidelines and good practices for building interoperable Cloud PaaS offerings.

The vision of Cloud4SOA is to open up the Cloud market to small-medium European PaaS providers and strengthen their market position and to treat the vendor-lock in problem. Cloud4SOA will thus enhance Cloud-based application development, deployment and migration by semantically interconnecting heterogeneous Platform as a Service (PaaS) offerings both within the same as well as across different Cloud PaaS providers and will facilitate the access of Cloud-based application developers to the PaaS offering that best matches their computational needs.

The Cloud4SOA consortium

1.2. Methodology

The development of the Cloud4SOA CSIF constitutes the first step towards the creation of the Cloud4SOA system. The methodology followed in order to develop the Cloud4SOA CSIF uses a layered approach as presented in Figure 1.

According to [1], the framework is considered as something at a high level, offering neutral ground upon which a community of stakeholders can talk about issues and concerns related to integrating parts of a large, complex system. More specifically, a framework sits at a broad, conceptual level and provides context (policies and process guidelines) for more detailed technical aspects of interoperability. On top of this, a model identifies a particular problem space and defines a technology independent analysis of requirements. In the context of Cloud4SOA, the model is expressed through the Cloud4SOA Reference Architecture specified in detail in D1.3. The design maps the model requirements into a particular family of solutions based upon standards and technical approaches. Finally, a solution manifests a design into a particular vendor software technology, ensuring adherence to designs, models, and frameworks. In the context of Cloud4SOA the design and the solution will be realized through the implementation of the Cloud4SOA system.
Cloud computing interoperability does not differ in principle from traditional system’s interoperability [2], where its major factors are how a Cloud-based system will work with what someone already has, how it will work with future solutions and how someone will get their data and/or services back once the contract is over and move it/them to another service provider. In this light, D1.2 starts with a review of the existing interoperability frameworks that will form the basis for the development of the Cloud4SOA CSIF providing significant knowledge about how interoperability has been already tackled. Meanwhile, several definitions illustrate the concept of semantic interoperability and how it is perceived in information systems.

The definition of the Cloud4SOA CSIF relies heavily on the work carried out in D.1.1. More specifically, the extended review of the State-of-the-Art in the research fields covered by Cloud4SOA identified gaps, deficiencies, needs and problems relative to Cloud semantic computing interoperability. However, the establishment of a semantic interoperability framework requires a more in-depth analysis of Cloud computing semantic interoperability issues. A deep investigation of semantic conflicts between heterogeneous PaaS offerings is performed by standardization bodies. At the same time, several studies can be found in the literature that discuss and try to address semantic conflicts in the field of Cloud PaaS. D1.2 reviews and analyzes these works in order to build an all-inclusive and comprehensive semantic interoperability framework for Cloud PaaS. Specifically, the Cloud4SOA CSIF aims to identify where semantic conflicts arise, categorize them and propose solutions.

1.3. Overview

The remainder of D1.2 is divided into six sections.
Section 2 reviews the existing literature in semantic interoperability and interoperability frameworks providing significant knowledge about how semantic interoperability has been already tackled in other domains of research.

Section 3 investigates the interoperability issues that emerge across heterogeneous Cloud computing environments with an emphasis on semantic aspects as well as on how these issues are confronted.

Section 4 introduces the Cloud4SOA CSIF and discusses in detail its core dimensions.

Section 5 presents the most common interoperability usage scenarios validating the Cloud4SOA CSIF.

Section 6 summarizes good practises and guidelines for building a semantically interoperable Cloud architecture.

Finally, Section 7 discusses our findings and concludes the document.
2. Information Systems Semantic Interoperability

Cloud computing interoperability should be viewed as any other Web-based information system’s interoperability [2]. To this end, this section reviews existing literature in semantic interoperability and interoperability frameworks (in the domain of information systems) providing significant knowledge about how semantic interoperability has been already tackled. The outcome of this review will support and feed the development of the Cloud4SOA CSIF.

In particular, section 2.1 investigates semantic interoperability as well as its main aspects/characteristics. Section 2.2 presents well-known interoperability frameworks in the domain of information systems. Section 2.3 discusses and concludes the section.

2.1. Semantic Interoperability

Semantic interoperability is a commonly used concept in information systems and it has been investigated through several studies. This section illustrates interoperability though several definitions, identifying its basic dimensions including semantic interoperability. As the document’s goal is to develop a semantic interoperability framework, an in-depth analysis of semantic interoperability follows detecting semantic aspects and dimensions.

Interoperability can mean many things to many people and is often interpreted in many different ways with different expectations [3]. Without a shared understanding on interoperability, research and development efforts cannot be efficiently carried out and coordinated. Interoperability can be considered as compatibility in terms of communication protocols, communication interfaces, data accesses, data types, data semantics, application functionality, dynamic behavior [4], and database schemas while it implies the uniform information systems integration [5].

Generally speaking, interoperability can be defined “as a measure of the ability of performing interoperation between two or more different entities (that can be pieces of software, processes, systems, business units, etc) [6].”

In the domain of information systems, interoperability [7] is specified “as the ability for two (or more) systems or components to exchange information and to use the information that has been exchanged.” In an attempt to enhance this definition, Vernadat [8] introduces the concept of exchange of functionality i.e. service, saying that interoperability is “the ability for a system to communicate with another system and to use the functionality of the other system.”

Several definitions have also been proposed introducing different aspects of information systems’ interoperability such as Duval’s definition [9], where interoperability “means that independently developed software components can exchange information, so that they can be used together.” Another definition which emphasizes on the applicability of the exchanged information [10] describes interoperability “as the ability to transfer and use information in a uniform and efficient manner across multiple organizations and information technology systems.” Additionally, INTEROP Network of Excellence [11] introduces the concept of data exchange instead of information saying that interoperability can be assumed as “the ability of information systems to operate in conjunction with each other encompassing communication protocols, hardware, software, application, and data compatibility layers.”

The aforementioned definitions do not make a clear distinction between different interoperability dimensions. However, one can infer the existence of at least two dimensions, one referring to technical interoperability, e.g. interoperable functionalities and processes, and one referring to semantic interoperability, e.g. information, data and knowledge interoperability.
Semantic interoperability has a key role in information systems interoperability with semantic issues being a central part of the interoperability definitions and the proposed interoperability frameworks. As the scope of this document is to introduce a semantic interoperability framework, a deep investigation for extracting the main semantic aspects, problems and solutions is necessary.

European Interoperability Framework (EIF) [12] defines semantic interoperability as the ability of organizations to process information from external sources in a meaningful manner. It ensures that the precise meaning of exchanged information is understood and preserved throughout exchanges between parties.

Vernadat [6] investigates the semantic aspects of the interoperability arising when two information systems decide to collaborate. The semantic concerns deal with information/data integration and granularity in order to enable interoperation, collaboration and especially knowledge and information sharing. He also defines semantic interoperability as the ability to share, aggregate or synchronize data/information across heterogeneous information systems making sure that the shared information is interpreted by the communicating systems in a consistent way. Thus, semantic interoperability is the ability to automatically interpret the exchanged information meaningfully and accurately and collaborate towards a joint purpose. To achieve this, both sides must refer to a common information exchange reference model [13].

Nowadays, businesses make use of the Service Oriented Architecture (SOA) and Web services (WS) for developing interoperable information services.

In the context of SOA technology, semantic interoperability means the capability of the interaction and behavior collaboration between software services. In the following two representative definitions are presented illustrating better the directions of semantic interoperability between services. The first definition specifies semantic interoperability from a general perspective mentioning that it is “the capability that two software modules or systems can exchange the data with precise meaning, and the receiving party can accurately translate or convert the information carried by the data, including the knowledge, i.e., information and knowledge that can be understood, and ultimately produce an effective collaborative results” [14] [15]. The second definition expands the idea of common interpretation of knowledge to the consistent, flexible exchange of data between consumers and providers in such a way that fulfills the non-functional requirements (NFRs) including performance, reliability and scalability, regardless the diverse information involved [16].

2.2. Interoperability Frameworks and Models

Interoperability is a multidimensional concept which can be looked at from numerous perspectives and approached from various different directions [3]. To conciliate all these perspectives, approaches and directions, which usually are different by nature, a reference framework is necessary.

The “framework” constitutes a useful mechanism/tool that can be used to compare concepts, principles, methods, standards, models and tools in a certain domain of concern [6]. In particular, the “interoperability framework” is an agreed approach that can enable interoperability between entities that wish to work towards a joint purpose. To achieve this, the interoperability framework specifies a set of common elements such as vocabulary, concepts, principles, policies, guidelines, recommendations, standards, specifications and practices [12]. The development of a well-structured framework for illustrating and identifying the critical interoperability issues will provide a successful context for the integration within complex systems [17].

Several frameworks with the intention to guide the seamlessly interoperation and integration can be identified in literature. This section reviews well-known frameworks which mainly focus on
service and/or application interoperability that is raised between enterprises, information systems etc. including:

- ATHENA Interoperability Framework (AIF) [18]
- IDEAS Interoperability Framework [19]
- LISI Reference Model [20]
- Enterprise Interoperability Framework [21] [22]
- GridWise Interoperability Context-Setting Framework [17]
- Perspective on Heterogeneity and Types of Interoperability [23]
- Future Internet Enterprise Systems (FINES) [24]

ATHENA Interoperability Framework (AIF) [18] provides a compound framework along with an associated reference model which represents the required and provided artifacts of two collaborative enterprises. To succeed in this, it adopts a holistic perspective on interoperability in order to analyze and address the business needs and the technical requirements as well as a multidisciplinary and a model-driven approach to solve the interoperability problems.

More specifically, the ATHENA Interoperability Framework structures the interoperability issues and solutions at three levels: conceptual, technical and applicative integration. The conceptual integration focuses on concepts, metamodels, languages and model relationships offering a modeling foundation for structuring various aspects of interoperability. The applicative integration emphasize on methodologies, standards and domain models providing guidelines, principles and patterns that can be used to tackle interoperability issues. Technical interoperability focuses on technical development and ICT environments while it provides ICT tools and platforms for developing and running enterprise application and software systems [25].

Moreover, the ATHENA reference model defines that interoperations between collaborating enterprises can take place at four layers [18]: enterprise/business, processes, services and data/information. Interoperability at the enterprise/business level is considered as the organizational and operational ability of an enterprise to cooperate with another organization overcoming differences in working practices, legislations, cultures and commercial approaches. Interoperability of processes aims to make various processes of different organizations work together creating cross-organizational business processes. Interoperability of services is the ability to identify, compose and execute various applications (designed and implemented independently). Lastly, interoperability of information/data is related to the management,
exchange and processing of different documents, messages and/or structures by different collaborating entities. For each of these levels a *model-driven interoperability* approach is needed to illustrate the models that should be used to formalize and exchange the provided and required artifacts that must be negotiated and agreed upon. Moreover, to overcome the semantic barriers which emerge from different interpretations of syntactic descriptions, precise, computer processable meaning must be associated with each concept using *ontologies* and *semantics*.

Figure 4 AIF Conceptual Framework

The IDEAS Interoperability Framework [19] underlines the need for a structured approach which brings together academic and industrial communities in order to achieve consensus on the roadmap for interoperability of enterprise applications and software. To this end, the framework captures and interrelates the current state of the art, vision statements, and research challenges from many perspectives and proposes three main layers for structuring interoperability issues of enterprise applications: *business, knowledge* and *ICT*, with two additional vertical dimensions: *semantics* and *quality attributes*. The business layer addresses the organization-oriented aspects as well as the management specifications of an enterprise. It consists of the decisional model, the business model and the business processes. The knowledge layer focuses on acquiring, structuring and representing the knowledge of an enterprise. It consists of the organization model which defines the roles within an organization, the skills competency model and the knowledge assets. The ICT layer deals with the ICT solutions that allow an enterprise to operate, make decisions, exchange information within and outside its boundaries. It includes solution management, workplace interaction, application logic, process logic and data logic. The semantic dimension and the quality attributes cross the aforementioned layers. The semantic dimension deals with capturing and representing the actual meaning of concepts, and thus promoting understanding. Lastly, the quality attributes contain business considerations that must be accommodated in a system, such as (i) security, (ii) scalability, (iii) portability, (iv) performance, (v) availability and (vi) evolution.
The LISI approach or “levels of information systems interoperability” has been proposed by the Architecture Working Group of the US Department of Defense (DoD) (on Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance – C4ISR). Its aim is to provide a maturity model and a process for determining the interoperability needs. The interoperability model is based on a set of requisite capabilities that must be inherent to all information systems that wish to interoperate. Moreover, each level of capabilities must cover all four enabling attributes of interoperability known as PAID, namely: procedures, applications, infrastructure (hardware, communications, security and system services) and data. The five levels of interoperability are listed below [20]:

Level 0 – isolated systems (manual extraction and integration of data).
Level 1 – connected interoperability in a peer-to-peer environment.
Level 2 – functional interoperability in a distributed environment.
Level 3 – domain-based interoperability in an integrated environment.
Level 4 – enterprise-based interoperability in a universal environment.

Figure 5 IDEAS Interoperability Framework [19]
The Enterprise Interoperability Framework, developed in the context of INTEROP Network of Excellence [21] [22], aims to define the research domain of enterprise interoperability and helps to identify and structure the knowledge of the domain. It is based on a barrier-driven approach taking into account the basic concepts addressed in existing frameworks and models. Indeed, it defines three basic dimensions: the dimension of interoperability barriers, the dimension of interoperability concerns and the dimension of interoperability approaches. Figure 7 illustrates the interoperability framework along with its three dimensions.

The dimension of interoperability barriers takes into account three categories of interoperability problems: conceptual, technological and organizational. Conceptual barriers refer to syntactic and semantic differences of information exchanged. Technological barriers emerge with the usage of information technologies and concern the standards that are used to present, store, exchange, process and communicate the data through the use of computers. Lastly, the organization barriers are related to the definition of responsibilities and authorities and the incompatibility of organization structures.

Interoperability takes place at various levels of enterprise including data, services, processes and business level (these levels are based on ATHENA Technical Framework). The data level refers to making different data models and query languages working together. The service level deals with identifying, composing and making functions together with various applications by solving semantic and syntactic differences, as well as finding connections to various heterogeneous databases. The process level aims at making processes working together while the business level refers to working in a harmonize way so that business can be developed and shared between companies.

There are three basic ways to relate entities (systems) together in order to establish interoperability [26]: the integrated approach, the unified approach and the federated approach. In the integrated approach, there exists a standard format for all models which has to be agreed by all parties to elaborate models and build systems. In the unified approach, there is a common meta-model across partners’ models. It provides the means for solving semantic differences allowing mapping between diverse models. Lastly, the federated approach is not making use of any common format. This implies that no partner imposes its models, languages and methods of work. Therefore, partners establish interoperability dynamically without using a predetermined meta-model.
Identifying the importance and the need for developing an interoperability framework that will facilitate the creation of open and interoperable solutions among entities that interact by the means of electric power systems, GridWise Architecture Council (GWAC) introduces the GridWise Interoperability Context-Setting Framework [17]. The framework consists of eight interoperability categories and a set of cross-cutting issues. The interoperability categories are divided based on their perspective into organizational, informational and technical.

The basic connectivity, the network interoperability and the syntactic interoperability constitute the technical aspects of the framework focusing on the establishment of physical and logical connections between systems, exchange of messages and understanding of data structure in messages exchanged, respectively.

The semantic understanding and the business context are the two core sub-categories of the informational interoperability. The semantic understanding of the types of things relevant to information exchange as well as description of how these entities are related to one another are concepts involved in the semantic understanding. On the other hand, how they are related to similar entities across different business domains consists part of the business context.

Interoperability is also driven by the needs of businesses to share information. Business procedures, business objectives and economic/regulatory policy constitute the organizational aspects of the framework. The business procedures support the alignment between operational business processes and procedures. The business objectives refer to the strategic and tactical objectives shared between businesses while the economic/regulatory policy refers to the political and economic objectives.

The cross-cutting issues represent the areas that should be investigating for advancing interoperability across the web of electric concerns, including (i) shared meaning of content, (ii) resource identification, (iii) time synchronization and sequencing, (iv) security and privacy, (v) logging and auditing, (vi) transaction and state management, (vii) system preservation, (viii) performance, reliability and scalability, (ix) discovery and configuration and (x) system evolution.
A classification of interoperability types which is based on the heterogeneity that exists in information systems is presented in [23]. According to this categorization, four different levels of interoperability can be detected: syntactic, structural, system and semantic. Syntactic heterogeneity is considered to be relevant to differences in machine-readable aspects of data presentation, also referred to as formatting. Representational heterogeneity involves data modeling constructs while schematic heterogeneity appears in structured databases. Both representational and schematic heterogeneity constitute aspects of structural heterogeneity. Heterogeneities that appear at the information system and at the platform level constitute the system interoperability. Lastly, semantic interoperability refers to the ability of systems to exchange data in such a way that the precise meaning of the data is readily accessible and the data itself can be translated by any system into a form that it is understandable.

![Figure 8 GridWise Interoperability Context-Setting Framework [17]](image)

Figure 8 GridWise Interoperability Context-Setting Framework [17]

![Figure 9 Perspective on Heterogeneity and Types of Interoperability [23]](image)

Figure 9 Perspective on Heterogeneity and Types of Interoperability [23]
Solving the issue of general interoperability and enabling the easy creation of services and applications are considered as high priorities for the Future Internet. Towards this direction, the Future Internet Enterprise Systems (FInES)\(^1\) Standardization effort has recently released a report [24], whose objective is to establish a standard framework for overall interoperability identifying barriers and concerns through a comprehensive analysis of the relevant work derived from existing groups and projects. In particular, the areas of interest for FInES are Cloud computing, Virtualization and Semantic Enterprise Interoperability. The analysis will result in identifying the needs and then developing new user-oriented standards in FInES.

![Diagram of Enterprise Interoperability and Collaboration (EI) and Digital Ecosystems (DE) clusters to be merged into Future Internet Enterprise Systems (FInES) Cluster [24]](image)

The aforementioned frameworks investigate the dimensions of interoperability as a whole. However, since the aim of the current document is to provide a semantic interoperability framework, a more focused review is needed presenting how semantic interoperability has been addressed and structured by the following well-accepted frameworks:

- **EIF [12]**
- **Ram and Park [27]**
- **Sheth and Gomadam [28]**
- **outGrid [29]**

In EIF [12], semantic interoperability encompasses two basic concepts: semantic and syntactic aspects. Semantic interoperability is referred to the meaning of data elements and the relationship between them. It entails developing vocabulary to describe data exchanges, and ensures that the data elements are understood in the same way by communicating receivers. On the other hand, syntactic interoperability is about describing the exact format of the information to be exchanged in terms of grammar, format and schemas.

Ram and Park [27] provide a categorization of semantic interoperability conflicts which are the result of different data models used by different platform owners, and usually raised either at the data or at the schema level. Data-level conflicts are differences in data domains caused by the multiple representations and interpretations of similar data. Examples of data-level conflicts are data-value conflicts, data representation conflicts, data-unit conflicts, and data precision conflicts.

Schema-level conflicts are characterized by differences in logical structures and/or inconsistencies in metadata (i.e., schemas) of the same application domain. Examples of such conflicts are naming conflicts, entity-identifier conflicts, schema-isomorphism conflicts, generalization conflicts, aggregation conflicts, and schematic discrepancies [30].

Sheth and Gomadam [28] define four types of semantics to capture the entire life cycle of a Web service, data semantics, functional semantics, non-functional semantics, and execution semantics. Data semantics are the formal definition of data in input and output messages of a web service. They are used in service discovery and interoperability between WS. Functional semantics are the formal definition of the capabilities of a web service. They can help the discovery and composition of WS. Non-functional semantics are the formal definition of quantitative or non-quantitative constraints like Quality of service (QoS) requirements like minimum cost and policy requirements like message encryption. They are used in discovery, composition and interoperability of WS. Execution Semantics are the formal definition of the execution or flow of services in a process or of operations within a service. They are used in process verification and exception handling.

In [29], semantic interoperability is defined as the understanding and treatment of exchanged data. In particular, two systems that are able to process exchanged data and produce meaningful results out of them are considered to be semantically interoperable. Moreover, semantic interoperability can be located at data, control and presentation level. Data semantic refers to structure/schema/standard being used to store, query, understand, process data and thus produce meaningful results from data hosted in given infrastructures. Control Semantic refers to structure/schema/standard being used to create, validate and execute application(s) in given infrastructures and corresponding execution environment(s). Presentation/Visual semantic refers to standard/technology being used to design, develop and harmonize Graphical User Interfaces (GUI) and overall access to platform functionality.

![Interoperability Dimensions Summary](image)

**2.3. Discussion**

Several definitions have been proposed trying to capture the basic characteristics of semantic interoperability. Among them, we distinguish the definition proposed by [14] [15] where semantic interoperability is defined as “the capability that two systems can exchange the data with precise meaning, and the receiving party can accurately translate or convert the information carried by the data, including the knowledge”.
The interoperability frameworks in the domain of information systems that have been proposed so far attempt to identify the interoperability dimensions and locate the interoperability conflicts that emerge when collaborating enterprises try to interoperate in an appropriate framework so that they can be easily analyzed and confronted. However, interoperability is a controversial concept perceived differently within the research community. As a result, the interoperability frameworks vary significantly in the way that they address the interoperability conflicts. Specifically, there are frameworks such as GridWise Interoperability Context-Setting Framework [17] and IDEAS Interoperability Framework [19] which address interoperability from an abstract perspective detecting interoperability conflicts in organizational, conceptual and technical layer. Frameworks like ATHENA Reference Model [18] and LISI Reference Model [20] define as the main layers of their interoperability the entities where interoperability problems emerge. In Table 1, the aforementioned interoperability frameworks are analyzed to their core dimensions. Summing up, interoperability conflicts are identified in business, process, application, data, infrastructure and knowledge layer. The six layers of Table 1 represent a “metaframework” capable of covering any dimension/category of the discussed interoperability frameworks.

Table 1 Layers of Interoperability

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Core Dimensions</th>
<th>Business</th>
<th>Process</th>
<th>Application</th>
<th>Data</th>
<th>Technical Infrastructure</th>
<th>Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATHENA Reference Model [18]</td>
<td>Business, Processes, Application, Data</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>LISI Reference Model [20]</td>
<td>Procedures, applications, infrastructure and data</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Enterprise Interoperability Framework [21] [22]</td>
<td>Business, Processes, Application, Data</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Perspective on Heterogeneity and Types of Interoperability [23]</td>
<td>Syntactic, structural, system and semantic</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Semantic interoperability frameworks follow different directions in the establishment of a common understanding for achieving semantic interoperability. Specifically, there are frameworks that focus on detecting the layer of semantic conflicts such as the framework presented in [29] which locates semantic conflicts in data, control and presentation layer. On the other hand, there are frameworks that specify the types of semantics needed in order to capture the knowledge/information for an entity. Such a framework is presented in [28] where Sheth and Gomadam detect four types of semantics (data semantics, functional semantics, non-functional semantics, and execution semantics) for capturing the entire life cycle of a Web service. Last, there are frameworks that concentrate on the reasons why semantic interoperability has caused such as the framework proposed by Ram and Park [27]. In this framework, the semantic conflicts are caused due to data or logic (schema) incompatibilities.

Table 2 presents the aforementioned semantic interoperability frameworks analyzed to their main concepts. The creation of an efficient metaframework capable of mapping the categories of the reviewed semantic interoperability frameworks is not feasible this time since they follow different approaches in addressing the semantic interoperability issue.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Core Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIF [12]</td>
<td>Semantic and syntactic</td>
</tr>
<tr>
<td>Ram and Park [27]</td>
<td>Data and schema</td>
</tr>
<tr>
<td>Sheth and Gomadam [28]</td>
<td>Data, functional, non-functional, and execution</td>
</tr>
<tr>
<td>outGrid [29]</td>
<td>Data, control and presentation</td>
</tr>
</tbody>
</table>

From the above discussion, we can deduce that a semantic interoperability framework can vary in purpose. It can define the levels of semantic interoperability or else the entities where semantic interoperability takes place, e.g. semantic conflicts may emerge at the application due to different descriptions of the same application’s feature. Moreover, a semantic interoperability framework can distinguish semantic conflicts based on the reason that caused them e.g. data or schema conflicts. Last, a semantic interoperability framework can introduce different types of semantics that will capture and organize the information that describe an entity, as proposed in [28].
3. Cloud Computing Semantic Interoperability

Semantic interoperability issues emerge also in Cloud computing environments since current Cloud computing solutions have not been built with interoperability as a primary concern [31]. The objective of this section is to review the literature focusing on the semantic interoperability between heterogeneous Cloud PaaS offerings and provide the foundation that will support the development of Cloud4SOA CSIF.

To this end, section 3.1 explores definitions of Cloud computing interoperability focusing on semantic aspects/characteristics. In section 3.2, initiatives derived from Cloud computing standardization bodies are investigated working towards interoperability at the PaaS layer. Section 3.3 reviews literature with the aim to present Cloud computing interoperability frameworks which define standardization approaches and directions for resolving semantic interoperability. Lastly, section 3.4 discusses and summarizes an initial set of requirements that a Cloud computing semantic interoperability framework should enclose.

3.1. Cloud PaaS Interoperability

PaaS offerings offer a higher-level platform, hiding the low-level details from the developer, such as the operating system, the load balancing, the data storage and access. This platform enables programmers to write code and deploy applications without worrying about software versioning or limited infrastructure resources. However, taking a closer look into the Cloud market, limited development tools are offered by the current Cloud PaaS providers, most of which use their proprietary runtime frameworks, programming languages and APIs supporting also on-premise toolsets and SDKs, while APIs come short to provide the underling middleware capabilities. These incompatibilities obstruct the uniform interoperability of Cloud PaaS providers and their PaaS offerings as well as the consistent migration of the application.

There are several attempts in the literature to scope, address and define Cloud computing interoperability. In the following a more in depth investigation of these definitions is provided focusing on interoperability that is raised in PaaS layer.

Generally speaking, the interoperability between two different Cloud providers refers to their ability to cooperate or else interoperate [32], thus establishing a federation of Clouds. Therefore, interoperability is a prerequisite for cooperation.

The Use Case Cloud Computing Discussion Group2 emphasizes that interoperability requires the conveyed information to be understood by the receiving systems. In the case of Cloud computing platforms, this means that “interoperability is the ability to write code that works with more than one Cloud provider simultaneously, regardless of the differences between the providers” meaning that there is the need of an API in order for the code to interact with any kind of system [33].

While the previous definition comes short to explain what “code” is, Foley provides a detailed description of “code” and the need to be integrated for the successful interoperation of different Cloud providers. Specifically, he states that Cloud computing interoperability is “the ability to move

2 http://groups.google.com/group/Cloud-computing-use-cases
data, applications and virtual servers from one Cloud computing environment to another” [34]. Since the current investigation is positioned on PaaS layer, the migration of virtual servers is out of scope.

Most of the Cloud providers, especially the big players in the Cloud market, use different management APIs to differentiate their products and prevent customers from moving to another Cloud provider. Therefore, this aspect should be included in a definition for Cloud computing interoperability, i.e. “interoperability refers to customers’ ability to use the same artifacts, such as management tools, virtual server images, and so on, with a variety of Cloud computing providers and platforms” [35].

Another approach investigating interoperability at the management level of Cloud computing platforms is presented in [2]. In this case, interoperability is defined as “modularity and flexibility to interface easily with any service or technology in the virtualization and Cloud ecosystem” and “standardization to avoid vendor lock-in and to create a healthy community”.

However, Cloud computing interoperability includes more characteristics than these reported above. For example, Goyal gives the following definition, identifying a number of basic concepts: “Cloud computing interoperability includes the ability of some application code to run on more than one provider CCE, regardless of the differences between the CCE. It also includes process execution, security, portability, migration/cloning control, standards, transparency, and manageability and regulatory compliance” [36] [37].

Moreover, these features must span interconnectivity, including inter-resource communication controls, resource isolation, resource security, visibility control (of private resources), resource discovery, resource mobility and control.

The European Telecommunications Standards Institute (ETSI) has recently introduced TC CLOUD3 which defines interoperability as the cooperation of multiple Clouds to support an application: “Interoperability is closely related to portability… Interoperability involves software and data simultaneously active in more than one Cloud infrastructure, interacting to serve a common purpose” [38].

In a similar definition, interoperability is considered as a synonym of integration “enabling products/software components to work with or integrate with each other seamlessly, in order to achieve a desired result ... This is enabled by either integrating through standard interfaces or by means of a broker that converts one product interface to another” [39].

According to the literature review, Cloud computing interoperability, compatibility and portability are closely related terms and often confused [38]. Therefore, Cohen clarifies the similarities and the differences among these terms in an attempt to exemplify and differentiate them: “Cloud computing interoperability is the ability for multiple Cloud providers to work together or interoperate. Cloud Compatibility and Portability answer to the question “how”? Cloud Compatibility means application and data that work with the same way regardless of the Cloud provider, whereas Cloud Portability is the ability of data and application components to be easily moved and reused regardless of the provider, operating system, storage, format or API” [32].

Expressing the Open Grid Forum’s (OGF)4 directions, Lee differentiates portability from interoperability [40]. According to his definition, interoperability means “being able to avoid “Cloud silos” that are non-interoperable since they are built on different APIs, protocols, and software stacks”. On the other hand, portability means “being able to perceive critical properties, such as performance, numerical stability and monitoring”.

---

3 http://www.etsi.org/WebSite/Technologies/GRID_CLOUD.aspx
4 http://www.ogf.org/

© Cloud4SoA consortium
European Network and Information Security Agency (ENISA) focuses on the lock-in problem at each level of the Cloud stack separately, since each level deals with different services [41]. According to ENISA, PaaS lock-in occurs at both the API (i.e. platform specific API calls) and the component level (i.e. a PaaS provider may offer a higher efficient back-end data store). Therefore, even if a compatible API is offered, the data may not be portable across PaaS offerings, as different data access models may exist. For security reasons PaaS environments often use heavily customized runtimes which also affect interoperability.

The aforementioned definitions deduce that interoperability between different Cloud PaaS providers involves the transparent exchange of “code”, including data and applications. However, interoperability may entail several additional features such as the management (API) compatibility or the capability of seamlessly integration. Furthermore, the analysis of Cloud computing interoperability shows that interoperability conflicts mainly emerge on semantic level, since most of the issues arise due to the different modeling and notation of the same features across different Cloud providers. Hence, semantic technology, such as common data models, standardization of functional and non-functional description of Cloud services and standardization of management (API) functions, can play a key role for Cloud computing interoperability.

However, limited work is available in the studied literature on Cloud computing semantic interoperability. Nevertheless, the semantic aspects of Cloud computing interoperability can be identified in several works in the area of Cloud computing. Thus, the following sections report the State of the Art in the domain of Cloud computing focusing on semantic interoperability as well as the semantic aspects that emerge.

3.2. Standardization Activities

There are several initiatives trying to address interoperability through standardized Cloud models and APIs. Standardization bodies, non-profit groups and member operated organizations work on advancing Cloud computing interoperability standards, with the collaboration of academia, researchers, governments and vendors. This section concentrates on the standardization initiatives addressing Cloud computing interoperability at the PaaS layer.

The Open Cloud Manifesto, an initiative supported by tenths of companies, including major software and infrastructure vendors such as IBM, SAP, Siemens and Telefonica, argues that Cloud computing should capitalize on open standards. It tries to address standards around security, integration, portability, interoperability, governance etc [42].

The Distributed Management Task Force (DMTF) a group dealing with standards that promote interoperable IT management among multi-vendor systems, tools and solutions, has introduced the Open Cloud Standards Incubator, which aims to standardize the interactions among Cloud environments by developing Cloud management use cases, architectures and interactions. Recently, DMTF has introduced the Cloud Management Working Group (CMWG) to continue the work of Open Cloud Standards Incubator. CMWG will develop a set of prescriptive specifications delivering architectural semantics as well as implementation details to achieve interoperable management between service requestors/developers and service providers. The

---

5 http://www.enisa.europa.eu/
6 http://www.openCloudmanifesto.org/index.htm
7 http://www.dmtf.org/home
8 http://www.dmtf.org/standards/Cloud

© Cloud4SoA consortium  page 28 of 65
CMWG will propose a resource model that will capture the key artifacts identified by “Use Cases and Interactions for Managing Clouds” document [43] produced by the Open Cloud Incubator. The document focuses on use cases, interactions, and data formats while describing how standardized interfaces and data formats can be used to manage Cloud environments.

The scope of the Cloud Computing Interoperability Forum (CCIF)\(^\text{10}\) is to enable a global Cloud computing ecosystem, where the organizations are able to work together seamlessly. The creation of a common agreed upon framework/ontology, which enables the ability of two or more Cloud platforms to exchange information in a unified way, is of primary importance. The Unified Cloud Computing, the project that will implement the above characteristics, will develop an open and standardized Cloud interface for the unification of several Cloud API's (an API for all other APIs). Furthermore, this unified Cloud interface will use Resource Description Framework (RDF) for the description of semantic Cloud data models. In order to achieve its scope, CCIF focuses on building community consensus, exploring emerging trends, and advocating best practices/reference architectures for the purposes of standardized Cloud computing platforms.

The Global Inter-Cloud Technology Forum (GICTF)\(^\text{11}\) aims to promote the standardization of network protocols and interfaces through which Cloud systems interwork with each other, promote international interworking of Cloud systems, enable global provision of highly reliable, secure and high-quality Cloud services, contribute to the development of Japan’s ICT industry and the strengthening of its international competitiveness.

The growth of a vibrant commercial marketplace for Cloud based services, where major buyers and sellers collaborate to define a range of common approaches, processes, metrics and other key service enablers, is the primary objective of TM Forum’s Cloud Services Initiative\(^\text{12}\). For this purpose, TM Forum has introduced the Frameworx Integrated Business Architecture, with the following components: Business Process Framework (eTOM)\(^\text{13}\), Information Framework (SID)\(^\text{14}\), Application Framework (TAM)\(^\text{15}\), and Integration Framework\(^\text{16}\). The eTOM framework defines an industry’s common process architecture for business and functional processes, the SID offers a common reference model that service providers, software providers, and integrators use to describe management information, the TAM framework provides a common language between service providers and their suppliers to describe systems and their functions, as well as a common way of grouping them and the Integration Framework provides a service-oriented integration approach with standardized interfaces and supporting tools.

Table 3 shows the standardization bodies of Cloud computing PaaS layer classified based on their focus. As it is observed, most of the existing work on Cloud computing interoperability emphasizes on a standardized interface (API). Moreover, standardization of management and business processes seems to play a key role for Cloud computing interoperability. Furthermore, a common resource, management and data model are going to supplement the creation of an interoperable Cloud computing system.

\(^{10}\) http://www.Cloudforum.org/
\(^{11}\) http://www.gictf.jp/index_e.html
\(^{12}\) http://www.tmforum.org/
\(^{13}\) http://www.tmforum.org/BusinessProcessFramework/1647/home.html
\(^{14}\) http://www.tmforum.org/InformationFramework/1684/home.html
\(^{15}\) http://www.tmforum.org/ApplicationFramework/2322/home.html

© Cloud4SoA consortium
### Table 3 Interoperability – Standardization Bodies Initiatives

<table>
<thead>
<tr>
<th>Stand. Groups</th>
<th>Standardization approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Management model/Business Process Framework</td>
</tr>
<tr>
<td><strong>Open Cloud Manifesto</strong></td>
<td></td>
</tr>
<tr>
<td>(standards for security, integration, portability, interoperability, governance)</td>
<td></td>
</tr>
<tr>
<td><strong>DMTF/CMWG</strong></td>
<td>✓</td>
</tr>
<tr>
<td>(standards for management, a resource model)</td>
<td></td>
</tr>
<tr>
<td><strong>CCIF</strong></td>
<td>✓</td>
</tr>
<tr>
<td>(a common ontology, standardized interface, data model)</td>
<td></td>
</tr>
<tr>
<td><strong>GICTF</strong></td>
<td>✓</td>
</tr>
<tr>
<td>(standards for network protocols and interfaces)</td>
<td></td>
</tr>
<tr>
<td><strong>cTOM</strong></td>
<td>✓</td>
</tr>
<tr>
<td>(common business process architecture)</td>
<td></td>
</tr>
<tr>
<td><strong>SID</strong></td>
<td>✓</td>
</tr>
<tr>
<td>(common description of management info)</td>
<td></td>
</tr>
<tr>
<td><strong>TAM</strong></td>
<td>✓</td>
</tr>
<tr>
<td>(Common language to describe systems and their functions)</td>
<td></td>
</tr>
<tr>
<td><strong>Integration Framework</strong></td>
<td>✓</td>
</tr>
<tr>
<td>(standardized interfaces)</td>
<td></td>
</tr>
</tbody>
</table>

### 3.3. Cloud Semantic Interoperability Frameworks

This section presents several Cloud computing interoperability frameworks which define the main directions for the investigation and advancement of Cloud computing interoperability. The presented frameworks are organized based on their focus. In particular, first we present frameworks that detect and demonstrate the levels of interoperability in the context of Cloud computing. Then, we introduce interoperability frameworks that identify the types of semantics needed to capture the characteristics of a cloud computing application. Lastly, we review interoperability frameworks that identify types of solutions which can address and resolve the interoperability problem in Cloud computing systems.

#### Levels of Interoperability
- Urquhart [44]
Types of Semantic Conflicts

- Sheth and Ranabahu [31] [45]

Types of Solutions

- Urquhart [46] and Sambyal et al. [35]
- Sperb Machado et al. [47]
- Mell and Grance (NIST) [48]
- Jha et al. [49]
- Cohen [32]
- Endo et al. [50]
- Hoff [51]
- Cerf [52]
- Soley (OMG) [53]
- CCIF\(^\text{17}\) (ref. [54])
- Bozman and Chen [55]
- European Future Internet Initiative (EFII) [56]
- ENISA [57]
- Govindarajan and Lakshmanan [39]
- Llorente [2]

3.3.1. Levels of Interoperability

Urquhart [44] claims that Cloud computing interoperability is an issue that arises in i) application/service, ii) management and iii) image/data level. In this categorization, PaaS and SaaS are both referred as applications. Application/service interoperability is the ability of developers to create loosely coupling applications which are platform independent, while management interoperability depends on APIs compatibility. Specifically, management interoperability refers to the capability of controlling multiple Cloud environments—both public and private. This includes how images can be delivered between providers, how servers and/or applications can be started and stopped etc. Image or data interoperability is based on how a virtual server image, a Java application or a database is defined, in order to be deployed on another provider. Indeed, image/data interoperability can be further divided into two subcategories: portability and mobility. Portability is the ability to move an image in a "down" state, and boot it at its destination. The image, in this case, can be as simple as a file system or as advanced as an Open Virtualization Format (OVF)\(^\text{18}\) portable Virtual Machine (VM) image with sophisticated metadata. On the other hand, mobility is the ability to move a live compute workload without losing client connections.

\(^{17}\) http://www.Cloudforum.org/
\(^{18}\) http://www.dmtf.org/standards/published_documents/DSP0243_1.1.0.pdf
Similarly, Llorente [2] identifies two interoperability levels: i) management and ii) service interoperability.

### 3.3.2. Types of Semantic Conflicts

Sheth and Ranabahu propose a model in order to address the semantic conflicts of Cloud computing interoperability [31]. Their semantic model has four dimensions: functional and non-functional definitions, data modeling and service description enhancement. Semantic interoperability problems occur as there are different functional and non-functional definitions which refer to application functionality and quality-of-service details, respectively. Furthermore, data store models can vary from schema-less data store to schema-driven data store, such as a relational database. Porting data across Cloud providers, who support different data-schemas, presents a significant challenge. Lastly, Clouds expose their operations via APIs or else via service interfaces which differ between vendors. However, the semantics of APIs’ operations are similar.

Furthermore, in [45], they enhance the previous model proposing a top-down methodology for the application development process based on a semantic partitioning which will provide higher level of abstraction, facilitating interoperability and portability.

![Figure 12 Partitioning of the Modeling Space for Clouds](image)

The semantic partitioning of the application which is proposed in this publication was inspired by the four types of semantics for a service, namely data semantics, functional semantics, non-functional semantics, and execution semantics. Figure 4 illustrates this partitioning. The identified types of semantics are data, logic and process, non-functional and system.

Data semantics: Data semantics address the data aspects of an application. This includes that definitions of data structures, relationships across multiple data structures as well as restrictions on the access of some of the data items.

Logic and process semantics: These are the semantics pertaining to the core functionality (commonly referred to as the business logic) of the application. Unlike in a service, the functional and execution semantics are tightly tied together for an application.

Non-functional semantics: These are semantics not-directly relevant to the business logic such as access control and logging. While these are not part of the core functions, an application nevertheless requires them to be defined.

System semantics: System semantics govern the system related concerns of the application. Relevant considerations include deployment descriptions and dependency management. These
considerations are neither relevant for the business logic nor the non-functional considerations but become important when the application starts running on a system.

![Figure 13 Four types of semantics for an Application](image)

### 3.3.3. Types of Solutions

Urquhart in [46] and Sambyal et al. [35] claim that there are only two interface points that PaaS (and IaaS) services need to standardize:

1. The management interfaces that enable a wide variety of tools to monitor and manipulate the resources and services being offered.
2. The "unit of delivery" that includes the software to be hosted and any required supporting data, configuration, and policy required to allow that software to work.

Moreover, they propose a Cloud computing model, illustrated in Figure 14 that consists of four layers:

- Metadata which describes the manifest of the package as well as any other metadata required for processing the package such as the specification version, the application classification, etc.
- The “bits” are the software and the data which are delivered. This can be in any applicable format, such as OVF.
- The deployment/configuration layer contains the information required to successfully get the application up and running in the target Cloud. This layer could include a lot of information, such as server and storage configurations, network connections as well as information concerning to acceptable pricing and billing terms.
- The runtime orchestration layer combined with the service level policies is required to handle the automated run-time operation of the application bits.
A framework where Cloud computing capabilities/features are divided into core and proprietary is presented in [47]. The authors state that Cloud interoperability can be achieved by standardizing these two types of capabilities. Core refers to basic Cloud features, such as management and provision of VM, storage or network components, while proprietary refers to advanced characteristics, such as a load balance component which offers a better traffic management for VMs. Standardizing these features will enable different Cloud providers to interoperate. Meanwhile, Cloud providers will have the opportunity to add their proprietary functions on their APIs, but not change the core capabilities. They also propose that research should be conducted for defining the core functions [58] of each layer of the Cloud stack.

The National Institute of Standards and Technology (NIST)\(^\text{19}\), proposes a similar standardization model, in order to address Cloud computing interoperability [48]. In the NIST model Cloud capabilities fall into two categories: core and advanced capabilities. Core refers to portable features while advanced to proprietary capabilities. Furthermore, they argue that each Cloud model (IaaS, PaaS, SaaS) has its own specifications and need to be focused separately.

\(^{19}\)http://www.nist.gov/itl/cloud/index.cfm
Jha et al. [49] propose to summarize the exposed core capabilities and achieve interoperability with standardization at two levels: at the infrastructure level (core capabilities) and at the Cloud interface level. The semantics of a resource/service/system constitute the core capabilities along with the ability to manage these capabilities (provisioning, availability, QoS, security). The core capabilities are usually variable. Services expose their capabilities via interfaces and these interfaces can be accessed through APIs. Interface incompatibility arises since different core capabilities are exposed from different Cloud providers, in order to support their Cloud’s usage modes.

Similarly, Cohen [32] states that Cloud providers can interoperate when they share a common set of application interfaces (APIs) as well as a consensus on the terminology/taxonomies that describe them.

Endo et al. [50] claim that APIs include basic/core and additional functionalities. The basic functions range from control of VM to programming primitives used to develop distributed applications in the Cloud, while the additional functions can include service quality assessment, load balance, elastic application growth and backup strategies.

Hoff [51] states that the Cloud community rotates around three ways in addressing Cloud computing interoperability: service brokers, semantics and APIs. More specifically, service brokers/intermediaries help customers of services declare their requirements using some methodology. These declarations of service definitions are translated across different service providers, who may have their own proprietary interfaces.

According to Cerf [52], Tim Berners-Lee argues that semantically linking data may be “the missing part of the vocabulary needed to interconnect computing Clouds”. Semantics of data and of the actions one take on the data and the vocabulary in which these actions are expressed appear to constitute the beginning of an InterCloud computing language. APIs allow Cloud operation, management, assurance and governance. As Cloud vendor sees its API functions as a first layer of competitive differentiation, an increasing number of incompatible APIs have emerged.

Expressing Object Management Group (OMG)20 directions, Soley [53] stresses that the Cloud community should cover 6 standardization areas in the context of Cloud computing interoperability: a) interfaces to IaaS, b) PaaS and deployment model formats for Cloud applications (e.g. resource descriptions, service and SLA models), c) management frameworks (e.g. governance and policy enforcement and SLA formats), d) portable component descriptions (e.g. description of VM’s), e) data exchange formats (to and from Clouds), f) Cloud taxonomies and reference models. Besides a standardized way to describe and manage the Cloud services, Soley introduces the necessity for standardized data exchange formats which will play significant role in Cloud computing interoperation.

---

20 http://www.omg.org/
CCIF is planning to come up with a global Cloud computing ecosystem, where the organizations are able to work together seamlessly. The creation of a common agreed upon framework/ontology, which enables the ability of two or more Cloud platforms to exchange information in a unified way, is of primary importance. The Unified Cloud Computing (UCC), the project that will implement the above characteristics, aims to achieve interoperability by the means of the following features: (i) Standardized Cloud interface: a key feature of UCC is the creation of a Cloud API that will unify the rest Cloud APIs and will provide an efficient abstraction of their functionality. (ii) Common interface for the interaction: it should behave as Cloud broker serving a common interface for the interaction between remote platforms, networks, systems, applications, services, identity and data. (iii) Common set of Cloud definitions: it constitutes an important factor that would enable vendors to exchange management information between distant Cloud providers. (iv) A specification and a schema: the model descriptions are provided by the schema and the details for integration with other management models are defined by the specification. These are important parts of unified Cloud interface (UCI) or of a Cloud broker. (v) Standards for NFRs: when an application migrates from one Cloud to another, apart from functionality, it is also important to ensure that NFRs will be satisfied in the new migrated environment. Therefore, there is a need for standards to define the NFRs. This common understanding will enable to compare and exchange data between different Cloud providers. However, this task could be complex considering the fact that there are several NFRs pertaining to security, availability, reliability, performance, scalability, etc., that requires compliance. Furthermore, the Cloud model should address both the platforms as service offerings and infrastructure Cloud platforms [54].

Governance plays a key role in Cloud computing interoperability since systems with different governance models cannot work together. The following works introduce the concept of governance/management in order to better address the semantic interoperability aspects.

During their study [55], Bozman and Chen present three key enablers towards seamlessly move of a workload from one Cloud to another (i) standardized programming interfaces, (ii) layers of abstraction and (iii) management capabilities. The standardized interface will allow applications written to those interfaces to be moved to a new Cloud. The abstraction layers will hide infrastructure from the Cloud services, reducing dependencies on underlying processors, operating systems and virtualization software and making ease service portability between Cloud providers. The new management capabilities have to work well with existing management products, both for managing physical servers and virtual servers in a unified way.

The European Future Internet Initiative (EFII) proposes, in [56], to develop a unified platform, named Future Internet Core Platform which will support many different applications and business processes. Therefore, it should adhere to some basic architectural principles that will ensure portability and interoperability and specifically it will support (i) a common communication model and operational constraints following the next generation network architecture (data control, management plane), (ii) a unified set of interface specifications, (iii) interfaces invoked in a standard manner and (iv) open standard metadata specifications defined for all those pieces where standardization of metadata will be required to support portability and interoperability (e.g., service specifications to be used during marketplace registration).

Lastly, ENISA summarizes a set of basic (interface) interoperability attributes that need to be defined [57] in order to enable interoperability and easy migration of Cloud services from one Cloud Service Provider (CSP) to another. These interoperability attributes include interface

21 http://www.Cloudforum.org/
interoperability/interface complexity, data format exchange capabilities, means of transfer/exchange, identity system and policy interoperability.

Govindarajan and Lakshmanan [39] report that besides APIs and brokers, interoperability should be investigated through control, data and other additional issues, such as policy management, security management and deployment/provisioning aspects which are to be managed across all the interfacing environments. Moreover, they propose to build relevant layers of abstraction in order to help interoperability and portability.

Llorente [2] identifies two interoperability levels: management and service interoperability. He focuses on management interoperability level saying that the “Cloud OS” represents the basic functions in a Cloud and requires a well defined communication with the underlying devices as well as an interface to expose administration and user functionality.

3.4. Discussion and Findings

In section 0, we studied the existing literature on Cloud computing interoperability focusing mostly on semantic interoperability problems and solutions at the PaaS layer.

We started by reviewing existing definitions for Cloud computing interoperability at PaaS layer. While providing a good understanding of the interoperability problem, they come short to provide insights on the different dimensions of Cloud computing interoperability, e.g. technical or semantic. Therefore, the need for a comprehensive and more detailed definition of Cloud computing emerges. Similarly, semantic interoperability also remains unspecified in the domain of Cloud computing.

Thereafter, we investigated the standardization initiatives addressing Cloud computing interoperability at the PaaS layer. Standardization groups mainly support the development of common frameworks and models, or extensions of existing standards which will enable the seamless interoperation and the unified management of heterogeneous Cloud PaaS systems.

After that, an extended review of several Cloud computing interoperability frameworks followed. The presented frameworks were organized based on their focus. In particular, frameworks that detect and demonstrate the levels of interoperability were discussed first. Then frameworks that identify the types of semantics and frameworks that identify the types of solutions in the context of Cloud computing. Based on the analysis of Cloud computing interoperability frameworks, the semantic interoperability conflicts can be located at management, application/service and data layer in a Cloud-based system. Moreover, the characteristics as well as the required information for describing the entire life cycle of a Cloud computing application are captured by the means of a set of semantics categorized into functional or logic, non-functional, data and system semantics.

Analyzing the frameworks’ characteristics, we can make some interesting observations. We can derive the semantic aspects that emerge, the layer of semantic conflict or else the entity which is affected by the semantic conflict, the types of semantics that are used for describing the controversial information/knowledge, as well as the proposed solution(s) that can address the specific semantic interoperability conflicts.

Table 4 summarizes the aforementioned interoperability frameworks analyzed to their central characteristics and semantic aspects. Moreover, each framework matches to one or more entities and types of semantics (according to [44] and [45], respectively). As an example, if a framework refer as central characteristic “Semantics of data and of the actions, standardization of APIs functions”, the entity or the layer is data and management while the types of semantics are data and system semantics.
Table 5 reviews the interoperability frameworks trying to detect the possible approaches that can address the semantic interoperability conflicts. In particular, there are four prevailing standardization approaches:

1. The standardization of a common description model,
2. The standardization of a common API,
3. The standardization of core and advanced functionalities and
4. The usage of common abstraction layers.

A common API should involve a set of core functionalities that will meet the basic needs of any Cloud Platform and will unify all different APIs (an API for all APIs). This will enable any existing Cloud API to be mapped to this common API. To succeed this, the Cloud community should agree on a standardized API (core and advanced functionality) and a set of management operations that should be similarly expressed across APIs. Furthermore, the description of applications and PaaS offerings should be capitalized on the same semantics, i.e. functional, non-functional, system and data. This common understanding will allow two (or more) different Cloud providers interoperate and co-operate for a joint purpose without problems. Moreover, customers can change Cloud provider more easily since their requirements can be understood similarly by different Cloud providers. Last, the adoption of abstraction layers is a method that can enhance interoperability hiding the differences of the underlining resources and services.
<table>
<thead>
<tr>
<th>Ref.</th>
<th>Characteristic</th>
<th>Types of Semantics</th>
<th>Layer/Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jha et al. [49]</td>
<td>Standardization at the infrastructure level (core capabilities) and at the interface level</td>
<td>✓</td>
<td>Management</td>
</tr>
<tr>
<td>Cerf [52]</td>
<td>Semantics of data and of the actions, standardization of APIs functions</td>
<td></td>
<td>Management, Data</td>
</tr>
<tr>
<td>Soley [53]</td>
<td>Interfaces to PaaS and deployment model formats, management frameworks, portable component descriptions, data exchange formats, Cloud taxonomies and reference models</td>
<td>✓ ✓ ✓ ✓</td>
<td>Management, Application, Data</td>
</tr>
<tr>
<td>Urquhart [46] and Sambyal et al. [35]</td>
<td>Management interface and the “unit of delivery” (Application, control, data)</td>
<td>✓ ✓</td>
<td>Management, Application, Data</td>
</tr>
<tr>
<td>CCIF [54]</td>
<td>Standardized Cloud interface, common interface for the interaction, common set of Cloud definitions, a specification and a schema, standards for NFRs</td>
<td>✓ ✓</td>
<td>Management, Application</td>
</tr>
<tr>
<td>Bozman and Chen [55]</td>
<td>Standardized programming interfaces, layers of abstraction and management capabilities</td>
<td>✓</td>
<td>Management</td>
</tr>
<tr>
<td>EFII [56]</td>
<td>A common communication model and operational constraints, a unified set of interface specifications, interfaces invoked in a standard manner and open standard metadata specifications (standard metadata specification s) (unified set of interface specifications)</td>
<td>✓ ✓ ✓</td>
<td>Management, Application</td>
</tr>
<tr>
<td>ENISA [57]</td>
<td>Interoperability/interface complexity, data</td>
<td>✓ ✓ ✓</td>
<td>Management, Data</td>
</tr>
<tr>
<td>Govindarajan and Lakshmanan [39]</td>
<td>Besides APIs and brokers, interoperability should be investigated through control, data and other additional issues (policy management, security management and deployment/provisioning)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ref.</td>
<td>Characteristic</td>
<td>Common data, resource and management models</td>
<td>Standardized APIs</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>---------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Sheth and Ranabahu [31] [45]</td>
<td>Functional and non-functional definitions, data modeling and service description enhancement</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>Urquhart [46] and Sambyal et al. [35]</td>
<td>Management interface and the “unit of delivery” (Application, control, data)</td>
<td>☑ ☑</td>
<td>☑</td>
</tr>
<tr>
<td>Sperb Machado et al. [47]</td>
<td>Standardized core and proprietary features/capabilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mell and Grance (NIST) [48]</td>
<td>Standardized core and advanced features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jha et al. [49]</td>
<td>Standardization at the infrastructure level (core capabilities) and at the interface level</td>
<td>☑ ☑</td>
<td>☑</td>
</tr>
<tr>
<td>Cohen [32]</td>
<td>Common application interfaces (APIs) and a consensus on the taxonomies</td>
<td>☑ ☑</td>
<td></td>
</tr>
<tr>
<td>Endo et al. [50]</td>
<td>Basic/core and additional functionalities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hoff [51]</td>
<td>Service brokers, semantics and APIs</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>Cerf [52]</td>
<td>Semantics of data and of the actions, standardization of APIs functions</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>Soley [53]</td>
<td>Interfaces to PaaS and deployment model formats, management frameworks, portable component descriptions, data exchange formats, Cloud taxonomies and reference models</td>
<td>☑ ☑</td>
<td></td>
</tr>
<tr>
<td>Entity</td>
<td>Description</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>CCIF [54]</td>
<td>Standardized Cloud interface, common interface for the interaction, common set of Cloud definitions, a specification and a schema, standards for NFRs</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Bozman and Chen [55]</td>
<td>Standardized programming interfaces, layers of abstraction and management capabilities</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EFII [56]</td>
<td>A common communication model and operational constraints, a unified set of interface specifications, interfaces invoked in a standard manner and open standard metadata specifications</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ENISA [57]</td>
<td>Interoperability/interface complexity, data format exchange capabilities, means of transfer/exchange, identity system and policy interoperability</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Govindarajan and Lakshmanan [39]</td>
<td>Besides APIs and brokers, interoperability should be investigated through control, data and other additional issues (policy management, security management and deployment/provisioning) and relevant layers of abstraction</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Llorente [2]</td>
<td>Management and service interoperability, well defined basic functions and an interface to expose functionality.</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

The aim of this section is to introduce the Cloud4SOA CSIF which includes a set of definitions for the basic concepts (section 4.1) as well as a three-dimensional semantic interoperability framework which captures any type of semantic interoperability conflict arising at the PaaS layer (section 4.2).

4.1. Basic Definitions

Capitalizing on the knowledge gained by reviewing and analyzing the literature, Cloud4SOA CSIF adopts the following definitions:

Cloud PaaS Interoperability [33]: Cloud PaaS interoperability boils down to the ability to write code that works with more than one Cloud PaaS offerings simultaneously and to be able to move code and data from one Cloud PaaS offering to another, regardless of the differences between them.

PaaS Semantic Interoperability: Semantic interoperability is the ability of Cloud PaaS systems and offerings to overcome the semantic incompatibilities and communicate. This refers to the ability of applications and their data to seamlessly be deployed on and/or migrated between Cloud PaaS offerings that are using the same technological background but different data (information) models and Application Programming Interfaces (APIs).

Heterogeneity [31]: Heterogeneity refers to the effect of numerous PaaS providers entering the Cloud market with each of them introducing their own platforms, models and programming paradigms, and marketing their own version of (incompatible and non-interoperable) services.

Portability [33]: Portability is the ability to move components, applications and data written for one Cloud PaaS offering to another one without being required to create new software, modify significantly the application being transported or reenter data descriptions. Achieving semantic interoperability is a fundamental requirement and a necessary precondition to be fulfilled in order to enable the seamless migration of applications across different Cloud PaaS offerings.

Cloud PaaS Broker [33]: A Cloud PaaS broker does not offer Cloud resources of its own, but matches consumers and providers based on the semantic profile of the consumer’s application and the semantic profiles of available PaaS offerings.

These definitions provided in this section will be used not only in this deliverable but in the upcoming deliverables of Cloud4SOA as well.

4.2. CSIF Core dimensions

Cloud4SOA introduces a three-dimensional semantic interoperability framework that aims to capture any type of semantic interoperability conflict arising at the PaaS layer. At the same time it enables every semantic conflict to be mapped to the appropriate PaaS entity and the type of semantics. In particular, Cloud4SOA CSIF is structured according to the following core dimensions, which are discussed in detail in the remainder of this section:

1. **Fundamental PaaS Entities**, i.e. PaaS System, PaaS offering, API and Application.
2. **Types of Semantics**, i.e. functional, non-functional, execution.
3. **Levels of Semantic Conflicts**, i.e. information model and data.
A semantic interoperability conflict is raised when during the deployment of an application on a PaaS offering or during the migration of an application from one PaaS offering to another, the semantic descriptions of any of the fundamental PaaS entities are incompatible. A semantic interoperability conflict may also be raised when two different PaaS Systems try to exchange information.

The first dimension allows us to locate where a semantic conflict is raised, i.e. between which fundamental entities. The second dimension allows us to identify the type of semantic conflict that has occurred. Finally, the third dimension allows us to identify nature of the semantic conflict that has occurred.

Information about all three dimensions needs to be collected in order to concretely define, fully understand and effectively treat a specific semantic conflict.

Error! Reference source not found. provides an overview of Cloud4SOA CSIF and its core dimensions.

Figure 17: Cloud4SOA CSIF

4.2.1. Fundamental PaaS Entities

Cloud4SOA CSIF adopts a basic PaaS system model. The objective of this model is to recognize the fundamental PaaS entities and consequently investigate the semantic interoperability problems that relate to each of them. The PaaS model defines the following fundamental entities (Figure 18):

- PaaS System
- PaaS Offering
- API
- Application
A PaaS System, such as cloudControl\(^\text{22}\), establishes agreements with different developers enabling their applications to be deployed on one of the PaaS offerings of the platform using the tools and API provided. Hence, a PaaS System may offer different offerings allowing developers select the one that best matches their needs, e.g. a Java-based, a PHP-based and a .NET-based offerings. The semantic description of the PaaS System contains general information about the PaaS provider, such as name, location and contact details, as well as technical specifications and characteristics of the system itself, including a short description of the available offerings and APIs.

**PaaS Offering**

A PaaS system publishes a number of PaaS offerings, i.e. platforms or programming platforms that Cloud-based application developers can use in order to implement and deploy their applications. The semantic descriptions of a PaaS offerings includes technical information about the development and execution environment, programming language, application server types and versions, security etc., as well as business information, e.g. details about services offered and pricing. A PaaS offering can host, depending on its capacity, more than one application.

**API**

Each PaaS offering offers has an API. The API constitutes a contact that defines how a developer will write his code in order to interact with the offering. More specifically, it contains a number of operations that support the management of the uploaded applications by the underlying system. Moreover, for each operation, the API specifies the information that should be sent to the system, the information that the system will send back and any error conditions that might occur.

**Application**

An application can vary from monolithic applications to n-tier applications, e.g. an ERP or a CRM. Cloud4SOA CSIF does not discriminate between the source code and the data of an application. As such, an application treated as a black box containing source code and/or data. Every application is described by a set of metadata regarding the applications type, version, owner/developer, technical details, size, information to access and process the application etc.

\(^\text{22}\) http://cloudcontrol.com/
A detailed specification of the fundamental PaaS entities will be delivered by WP2 – Cloud4SOA Semantic Layer.

### 4.2.2. Types of Semantics

Each of the fundamental PaaS entities can be described by three types of semantics:

- The **functional semantics** describe the core functionality/capabilities of a fundamental PaaS entity that is what an entity can offer or can do when it is invoked, e.g. services, tools, programming frameworks and development environment.

- The **non-functional semantics** model the non-functional aspects of a fundamental PaaS entity including reliability, security, pricing, availability, usability, performance and other QoS related information.

- The **execution semantics** are used to specify governance-related information of a fundamental PaaS entity (usually an application) that become important when it starts running or when it is invoked, such as information about deployment, management and control, discovery, selection, orchestration, etc.

Figure 19 provides an analysis of which types of semantics apply to a specific fundamental PaaS entity. This is detailed in the remainder of this section.

![Figure 19 Semantics of Core Artifacts](image)

**Fundamental PaaS Entities**

**PaaS System**

The effective semantic description of a PaaS System comprises of the following types of semantics:

- The **functional semantics** describe the PaaS System’s main capabilities or else what the PaaS system can offer including the underlying technological infrastructure, the available offerings etc.
• The **non-functional semantics** describe the PaaS System’s non-functional attributes, such as information about the provider (e.g. name, location and contact details), a short description of the PaaS system, its URL, technical details, QoS related information, e.g. about performance and security, and pricing policies.

• The **execution semantics** are not applicable in the case of the PaaS System.

**PaaS Offering**

An effective semantic description of a PaaS offering comprises of the following types of semantics:

• The **functional semantics** describe the PaaS offering’s capabilities or else what a specific PaaS offering can offer, including the programming language supported, the available tools and the available development environment, the API etc.

• The **non-functional semantics** describe the PaaS offering’s non-functional characteristics, such as technical details, software versions, performance, security and pricing policies. These attributes are redefined for the PaaS offering since a PaaS System may follow a general policy model which is common for all of its PaaS offerings or may follow a different policy for each of the PaaS offerings.

• The **execution semantics** describe how a PaaS offering is discovered, selected and utilized by a Cloud-based application developer.

**API**

An effective semantic description of an API comprises of the following types of semantics:

• The **functional semantics** define the API’s functionality or else what the API can offer including functions and operations, tools and access protocols etc.

• The **non-functional semantics** describe the API’s non-functional characteristics, such as the supported programming language(s), title, version, description, data formats used, security, backup, accessibility, usability, access rights etc.

• The **execution semantics** are not applicable in case of the API.

**Application**

An effective semantic description of an Application comprises of the following types of semantics:

• The **functional semantics** describe the core functionality of an application or else what an application can do.

• The **non-functional semantics** refer to non-functional characteristics of an application contains such as programming language, title, version, source (URL), security rights, performance and scalability considerations.

• The **execution semantics** model information produced when an application is running on a PaaS offering, including deployment description, load balancing, dependency management and elasticity considerations (rules and automatic resource allocation).
4.2.3. Levels of Semantic Conflicts

Cloud4SOA CSIF defines two levels of semantic conflicts:

- **Information model level** that refers to differences in logical structures, data structures, inconsistencies in metadata (i.e. schemata), relationships across multiple data structures as well as restrictions on the access of some of the data items.

- **Data level**: it refers to differences in data caused by the multiple representations and interpretations of similar or the same data.
5. **Cloud Computing Semantic Interoperability Use Cases**

As Cloud computing popularity continues to rise, more and more Small and Medium Enterprises (SMEs) enter the Cloud market with the aim to increase their efficiency and decrease their operational costs. The presence of numerous providers increases the need for interoperability since several federation scenarios can emerge among them. In this section, several use cases coming from literature review are described. We then investigate the semantic conflicts that can be raised each time. Hence, for each use case, we specify the fundamental PaaS entities, the type of semantics and the level of semantic conflicts that may arise. Finally, every identified semantic conflict is matched with an appropriate solution.

Cloud Computing Use Case group has released a white paper [33] [59] highlighting the requirements that need to be standardized to ensure interoperability in most typical scenarios of interaction in Cloud computing. Based on these scenarios, a set of use cases have been produced emphasizing at PaaS layer and, especially, the interoperation between PaaS providers and Cloud-based application developers (PaaS users).

From a more service/application-oriented perspective, the work, presented in [60], investigates the interoperability issues that emerge when developing, deploying and migrating applications/services on both public and private Cloud environments.

The use cases of the two efforts where reviewed, analyzed and merged. We thus concluded in the following use cases:

- **End-user to Cloud: Application Deployment on a PaaS offering**
- **Changing Cloud vendors: Migrating an application to a different PaaS offering**
- **Hybrid Clouds: PaaS systems/offerings interoperation**
- **Enterprise to Cloud to Enterprise: Interoperation between applications deployed on different PaaS offerings**

5.1. **End-user to Cloud: Application Deployment on a PaaS offering**

This scenario involves a Cloud-based application developer who is accessing data or applications located in a Cloud. This is the simplest version of usage scenarios. Focusing on the PaaS layer, a Cloud-based application developer can implement and deploy his application on a specific PaaS offering.

In this scenario, there are no semantic conflicts raised if the developer utilizes tools, services and models from one PaaS offering. However, semantic conflicts might be raised if the Cloud-based application developer attempts to deploy on the PaaS offering the application that he has developed somewhere else and/or has been described using a model different than the one supported by the specific PaaS offering.

In this case, the semantic conflict is raised due to differences in the semantic models of the Cloud-based application developer and the PaaS offering. Differences may exist in any of the functional, non-functional and execution semantics either at the information model or at the data level.
5.2. **Changing Cloud vendors: Migrating an application to a different PaaS offering**

This use case happens when an application migrates to a different PaaS offering within the same PaaS system or across different PaaS systems. Moreover, it can entail the case of an additional PaaS offering supporting part of the application’s running instance.

In this use case, several semantic conflicts can emerge. Some examples follow:

**Example 1:** A Cloud-based application developer, who has used in the code of his application functions from a PaaS offering’s API, e.g. the ‘connect a db’ call to bind a database, is not able to migrate his application and seamlessly deploy it on another PaaS offering that is using a different API, e.g. the new API is using a function called ‘insert a db’ to bind a database.

Using the Cloud4SOA CSIF, we will try to describe this semantic conflict. It is raised due to differences in the definitions of the two APIs; more specifically to the way that their functional
semantics are modelled. It is thus raised at the information model level, as it is caused by different naming for the same functionality. This semantic conflict can be resolved by means of a standardized API and/or a common PaaS offering model.

Example 2: In the same line, during migration the source PaaS offering might use the value “Java” for the ‘programming language’ attribute and the value “1.6” for the ‘version’ attribute, while the destination PaaS might use value “Java 1.6” for the ‘programming language’ attribute and might have no ‘version’ attribute.

The semantic conflict is raised due to differences in the definitions of the non-functional semantics of the two PaaS offerings. This conflict is raised both at the information model and data levels, as it is caused both by different representation for the same information and different interpretations of the same data. This semantic conflict can be resolved by means of a common PaaS offering model.

Example 3: Additionally, the two PaaS systems use different pricing models. The first one charges on a per hour basis (pay-per-hour), while the second charges on a per usage basis (pay-per-use). Thus, comparing the two pricing policies is a challenge.

The semantic conflict is raised due to differences in the definitions of the non-functional semantics of the two PaaS systems. This conflict is raised both at the information model level and at data level, a different representations are used for pricing. This semantic conflict can be resolved by means of a common pricing model for the two PaaS systems.

5.3. Hybrid Clouds: PaaS systems/offersings interoperation

This scenario refers to multiple Clouds working together for a common purpose. A hybrid Cloud can be delivered by (i) a federated Cloud provider that combines its own resources with those of other providers, and (ii) a broker with the difference that it does not have any Cloud resources of its own. Focusing on the PaaS layer, this scenario encloses multiple PaaS systems working together as a federation for delivering a joint purpose.

In this use case, several semantic conflicts can emerge. Some examples follow:

Example 1: Two PaaS systems decide to cooperate. However, their owners realize that the models that they use for describing the PaaS system, the PaaS offering and their APIs are incompatible. For example, the first one describes as ‘programming language’ both the specific language supported as well as its version, e.g. Java version 1.6, while the second uses ‘programming language’ and ‘version’ respectively.

Using the Cloud4SOA CSIF, we will try to describe this semantic conflict. It is raised due to differences in the semantic models of the two PaaS systems; more specifically to the way that their non-functional semantics are modelled. It is thus raised at the information model level, as it is caused by different representation for the same information. This semantic conflict can be resolved by means of a common PaaS system model.

Example 2: In the same line, two PaaS offerings, one from every PaaS system, need to interoperate to support the increasing needs of an application. However, the semantic models of the PaaS offerings describe the same characteristics differently. The first expresses uptime as “99.9% uptime guarantee SLA” while the second as “99.9% 24/7”.

© Cloud4SoA consortium
This semantic conflict is raised due to differences in the semantic models of the two PaaS systems; more specifically to the way that their non-functional semantics are expressed. It is thus raised at the data level, as it is caused due to different representations of the same value. This semantic conflict can be resolved by means of a common PaaS offering model.

![Hybrid Cloud](image)

Figure 22 Hybrid Clouds use case [33]

### 5.4. Enterprise to Cloud to Enterprise: Interoperation between applications deployed on different PaaS offerings

This scenario involves two enterprises located in different Clouds and, specifically, two Cloud applications which are running in different Clouds and need to interoperate for a joint purpose.

In the PaaS context, this is translated to the ability of an application which is developed in a specific PaaS offering to interoperate with an application developed in a different offering.

In this usage scenario, several interoperability conflicts may be raised. For example, two applications (A and B) need to collaborate for a common purpose. In particular, application “A” is first deployed and application “B” is deployed when the usage performance of “A” exceeds a predefined value. To this end, application “A” records constantly its usage experience including the current usage performance, while application “B” constantly reads these data.

**Example 1:** The interoperation cannot be established in case the two applications use different ways for representing information related to their performance, e.g. application “A” refers to it as performance while application “B” refers to it as CPU speed.

Using the Cloud4SOA CSIF, we will try to describe this semantic conflict. It is raised due to differences in the semantic descriptions of the two applications; more specifically to the way that their non-functional semantics are modelled. It is thus raised at the information model level, as it is caused by different naming for the same data. This semantic conflict can be resolved by means of a common application model.
Example 2: Elaborating on the previous example, semantic interoperability is also hampered when the two applications use different metrics for representing the same performance value. For example, the first one represents it with expressions such as “high”, “low” while the second one represents it with “the number of active VMs per second” or with the “CPU speed”.

Once again, the semantic conflict is raised due to differences in the semantic descriptions of the two Applications; more specifically in their non-functional semantics. It is thus raised at the information model level, as it is caused by different representations of the same value. In this case, a common abstraction layer can resolve this semantic conflict. The abstraction layer should provide a set of definitions which hide the multiple variations of the same or relevant performance factors.

Building a semantically interoperable Cloud PaaS architecture is a complex procedure. This section summarizes good practices and guidelines derived from both the above analysis as well as from the reviewed literature.

Initially, there is the necessity for a common, interoperable and open set of Cloud computing standards that will enable Cloud systems to work together in a common way [61]. This need is twofold; it encloses the creation of a common Cloud computing reference API and the adoption of a standard description model.

Furthermore, an ideal semantic interoperable Cloud computing PaaS architecture should support a semantic layer which will deliver a PaaS Offering Model and an Application Model for the common description of the available PaaS offerings as well as for the description of the developer’s applications and needs. This model should be enhanced by the above considerations. As a result, the publication of a PaaS offering by the PaaS provider and the creation of an application’s semantic profile by the Cloud-based application developer will be based on this standard description, facilitating interoperability and portability.

Moreover, a standardized API needs to be adopted by Cloud community. Standardization of its core functionalities is considered as a first step towards this direction.

Several studies have been published addressing this need. According to [62], an API consists of the management functionality such as control/management of Cloud application as well as the deployment functionality allowing developers to upload and deploy applications or manage them [62]. In [63] and [64], the core management services enclose (i) deployment and configuration, (ii) monitoring and reporting, (iii) SLA management, (iv) metering and billing and (v) provisioning services. Furthermore, a unified management approach will enable the seamlessly federation of different providers and the easily integration with third-party management applications (portability) [55]. NIST, in [65], proposes a Cloud computing taxonomy classifying the service provider’s functions into operational and business. In this taxonomy, the operational functions are further divided into provisioning/configuration, interoperability/portability and security capabilities. A similar categorization is proposed by [66] where the most common functions of a standardized platform are classified into technical and business.

Based on the above observations, a standardized API should involve the capabilities shown in Figure 24.
Meanwhile, reviewing the literature, additional good practices and guidelines arise. The following list summarizes a set of general guidelines that someone should also keep in mind when building semantic interoperable Cloud environments:

- Need for developing common used standards, a common taxonomy and a set of standardized APIs [32] [67] [68] [69] (this is also a notion shared by most of the standardization bodies focusing on Cloud computing interoperability).

- Use widely available tools, technologies, methodologies, and best practices [67], with a standard syntax, open APIs, and commonly used standards [68] [69].

- Support existing applications, development paradigms, programming models, and languages, as well as new ones [70].

- Applications should be portable. There should be no need to tailor the application to a given Cloud environment. Developers should write applications once and deploy them anywhere, to any Cloud or to on premises systems infrastructure, in one step [70] [71].

- Data should also be portable. This means that the customer must be able to export data from his current service provider’s database in a format that can be migrated to other databases [72].

- Applications should be built using model-driven methodologies in order to offer the ability to quickly adapt to the changing demands. Specifically, a platform must support workflow capabilities and allow for collaboration as part of the business transactions [71].

- Present developers a direct route to the market through a marketplace that effectively reaches the customer [73].

- The platform should provide the ability to include/embed/integrate other applications built on the same platform or others [70] [71].

- A benchmark that would standardize the SLAs provided by different Clouds is important for the uniform interoperability across them [74].

- Develop an application architecture with layers of abstraction to minimize access to proprietary modules [69].

- Transfer monitoring, logging, auditing and control functions to the new provider [69].

---

**Figure 24 A Standardized API**
• Develop ways to test and confirm interoperability and ensure that all the elements can work together and interoperate as required and in conformance with known standards and certification [24].

• Support SOA with representative functionalities: a) uniform naming resolution for any entity, b) efficient, scalable and distributed service repository, c) a common application communication infrastructure that supports multiple paradigms, such as request/response, publish/subscribe, multicast, d) meta-data indexing and inference to enable entity/service indexing, searching and discovery, e) modeling, composition, orchestration and execution of business processes and telco services [56].

• Standards are required for data exchange between services, composition and orchestration of services. Standards, such as SOAP and BPEL, will form the basis on which Cloud computing frameworks will be implemented [60].
7. Conclusions

The present document is Deliverable 1.2 “Cloud4SOA Cloud Semantic Interoperability Framework” of the Cloud4SOA project. The main objective of this document was to deliver an open, innovative and generic framework for facilitating the implementation and deployment of semantically interoperable Cloud platforms.

The Cloud4SOA CSIF includes a set of definitions for basic concepts, such as PaaS semantic interoperability, and is structured according to the following core dimensions:

1. **Fundamental PaaS Entities**, i.e. PaaS System, PaaS offering, API and Application.
2. **Types of Semantics**, i.e. functional, non-functional, execution.
3. **Levels of Semantic Conflicts**, i.e. information model and data.

Several use cases coming from literature review were selected and used for validating the Cloud4SOA CSIF. The semantic conflicts that can be raised each time analyzed based on the Cloud4SOA CSIF’s core dimensions: fundamental PaaS entities, the type of semantics and the level of semantic conflicts that may arise. Finally, every identified semantic conflict was matched with an appropriate solution.

Furthermore, the document summarized a set of good practices and guidelines for building a semantically interoperable Cloud PaaS architecture. Central role towards this initiative is the creation of a common, interoperable and open set of Cloud computing standards that will enable Cloud systems to work together in a common way. This encloses the creation of a common Cloud computing reference API and the adoption of a standard description model.

A semantic interoperable Cloud computing PaaS architecture should support a **semantic layer** which will deliver a PaaS Offering Model and an Application Model for the common description of the available PaaS offerings as well as for the description of the developer’s needs and applications. As a result, the publication of a PaaS offering by the PaaS provider and the creation of an application’s semantic profile by the Cloud-based application developer will be based on this standard description, facilitating interoperability and portability.

Such a **semantic layer** is going to be implemented by Cloud4SOA system. A better description will be provided in the context of the Cloud4SOA Reference Architecture (D1.3) and WP2 Cloud4SOA Semantic Layer. In the same context, Cloud4SOA CSIF will be revisited and enriched with details that will come from the detailed modeling of the fundamental PaaS entities that will take place in WP2.
References


Annex A. Full license

This is a public deliverable that is provided to the community under the license Attribution-NoDerivs 2.5 defined by creative commons http://www.creativecommons.org

This license allows you to
to copy, distribute, display, and perform the work
to make commercial use of the work
Under the following conditions:

**Attribution.** You must attribute the work by indicating that this work originated from the Cloud4SoA project and has been partially funded by the European Commission under contract number 216759

**No Derivative Works.** You may not alter, transform, or build upon this work without explicit permission of the consortium

For any reuse or distribution, you must make clear to others the license terms of this work.

Any of these conditions can be waived if you get permission from the copyright holder.

This is a human-readable summary of the Legal Code below:

A.1. License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED. BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICensor GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

A.1.1. Definitions

a. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
b. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
c. "Licensor" means all partners of the Cloud4SoA consortium that have participated in the production of this text.
d. "Original Author" means the individual or entity who created the Work.
e. "Work" means the copyrightable work of authorship offered under the terms of this License.
f. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

A.1.2. Fair Use Rights

Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

A.1.3. License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

b. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works.

c. For the avoidance of doubt, where the work is a musical composition:

d. Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

e. Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights society or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

f. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Derivative Works. All rights not expressly granted by Licensor are hereby reserved.

A.1.4. Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested.

b. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor, institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by
other reasonable means, the name of such party or parties; the title of the Work if supplied; and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

A.1.5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE MATERIALS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

A.1.6. Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

A.1.7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

A.1.8. Miscellaneous

a. Each time You distribute or publicly digitally perform the Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.