ETSI SR 003 382 V2.1.1 (2016-02)

Cloud Standards Coordination Phase 2;
Cloud Computing Standards and Open Source;
Optimizing the relationship between standards and Open Source in Cloud Computing

Reference
DSR/NTECH-00031

Keywords
Cloud, Cloud computing, Open Source Software

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

List of figures/list of tables

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Special Report (SR) has been produced by ETSI Technical Committee Network Technologies (NTECH).

The present document is approved by the NTECH Technical Committee and for publication on the Cloud Standards Coordination website (http://csc.etsi.org).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

Cloud Computing is increasingly used as the platform for ICT infrastructure provisioning, application/systems development and end user support of a wide range of core services and applications for businesses and organizations.

Cloud Computing is drastically changing the way ICT is delivered and used. However, many challenges remain to be tackled. Concerns such as security, vendor lock-in, interoperability and accessibility, service level agreements more oriented towards users are examples of issues that need to be addressed.

In February 2015, the Cloud Standards Coordination Phase 2 (CSC-2) was launched by ETSI to address issues left open after the initial Cloud Standards Coordination Phase 1 (CSC-1) work was completed at the end of 2013, with a particular focus on the point of view of the Cloud Computing users (e.g. SMEs, Administrations).

The present report investigates the relationship and the interactions between standardization and Open Source based software and solutions in Cloud Computing. This question was not addressed in Cloud Standards Coordination Phase 1 (see [i.1]). In the meantime, Cloud Computing has emerged as one of the domains of Information and Communication Technology where Open Source development plays a very important role and changes significantly the status quo and, amongst other, the traditional approach to standardization.

1 Scope

The present report presents the results of the analysis of the relationship between Standards and Open Source in the context of Cloud Computing.

In February 2015, the Cloud Standards Coordination Phase 2 (CSC-2) was launched by ETSI to address issues left open after the Cloud Standards Coordination Phase 1 (CSC-1) work was completed at the end of 2013. Cloud Standards Coordination Phase 2 is investigating some specific aspects of the Cloud Computing standardization landscape, in particular from the point of view of the Cloud Computing users (e.g. SMEs, Administrations). It will also generate a new snapshot regarding the state of standards and investigate the interaction and relation between standardization and Open Source based software and solutions.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Cloud Standards Coordination, Final Report, November 2013.

NOTE: See http://csc.etsi.org/resources/CSC-Phase-1/CSC-Deliverable-008-Final_Report-V1_0.pdf.

[i.2] Regulation (EU) No 1025/2012 of the European Parliament and of the Council, on European standardization, 25 October 2012.

NOTE: See http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32012R1025.

[i.3] Implementing FRAND standards in Open Source: Business as usual or mission impossible?, European Commission, November 2012.

NOTE: See http://ec.europa.eu/DocsRoom/documents/15601.

[i.4] Open requirements for standards, Open Source Initiative.

NOTE: See http://opensource.org/osr.

[i.5] ETSI SR 002 960 (V1.0.1): "Working in ETSI within an OSS context: Guidance and recommendations, including usage of OSS within ETSI Secretariat, adoption/usage of elements of OSS in the elaboration of ETSI Standards and adoption of ETSI Standards within the OSS communities".

[i.6] Comparison of free and open-source software licenses, Wikipedia.

NOTE: See https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses.

[i.7] Top 20 Open Source licenses, Black Duck.

NOTE: See https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses.

[i.8] The architecture of Open Source Applications, A. Brown & G. Brown, The AOSA editors.

[i.9] The OPNFV Release 1 'Arno'.

NOTE: See https://www.opnfv.org/sites/opnfv/files/opnfv_arno_overview_diagram.jpg.

[i.10] ISO/IEC Guide 2:2004: "Standardization and related activities - General vocabulary".

[i.11] OpenStack Application Programming Interface (API).

NOTE: See http://developer.openstack.org/api-ref.html.

[i.12] UK Government Open Standards Principles.

NOTE: See https://www.gov.uk/government/publications/open-standards-principles/open-standards-principles.

[i.13] "Compatibility Of The Licensing Of Embedded Patents With Open Source Licensing Terms", Iain G. Mitchell QC, Stephen Mason.

NOTE: See http://www.ifosslr.org/ifosslr/article/view/57.

[i.14] ISO/IEC Draft 19941: "Cloud Computing - Interoperability and Portability".

[i.15] "Open Standards and Open Source: Enabling Interoperability", F. Almeida, J. Oliveira, J. Crux.

NOTE: See: http://airccse.org/journal/ijsea/papers/0111ijsea01.pdf.

[i.16] ETSI GS NFV 002: "Network Functions Virtualisation (NFV); Architectural Framework".

[i.17] ETSI GS NFV 001: "Network Functions Virtualisation (NFV); Use Cases".

[i.18] ISO/IEC 17203: "Information technology - Open Virtualization Format (OVF) specification".

[i.19] ISO/IEC 19831: "Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol - An Interface for Managing Cloud Infrastructure".

[i.20] DMTF DSP0243: "Open Virtualization Format Specification".

[i.21] DMTF DSP0262: "Cloud Auditing Data Federation (CADF) - Data Format and Interface Definitions Specification".

[i.22] DMTF DSP0263: "Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol".

[i.23] DMTF DSP2038: "Cloud Audit Data Federation - OpenStack Profile (CADF-OpenStack)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Open Source license: copyright license for Open Source software

Open Source Software (OSS): computer software that is available in source code form

NOTE: The source code and certain other rights normally reserved for copyright holders are provided under an open-source license that permits users to study, change, improve and at times also to distribute the software.

source code: any collection of computer instructions written using some human-readable computer language, usually as text

standard: output from an SSO

NOTE: For the sake of simplicity, the meanings of "standard" and "specification" are not differentiated in the present report, unlike in the other CSC-2 reports.

Standards Setting Organization (SSO): any entity whose primary activities are developing, coordinating, promulgating, revising, amending, reissuing, interpreting or otherwise maintaining standards that address the interests of a wide base of users outside the standards development organization

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

3GPP	Third Generation Partnership Project
API	Application Programming Interface
ATIS	Alliance for Telecommunications Industry Solutions
CC	Cloud Computing
CCSL	Cloud Certification Schemes List
CDMI	Cloud Data Management Interface
CIMI	Cloud Infrastructure Management Interface
CSA	Cloud Security Alliance
CSC	Cloud Standards Coordination
CSC-1	Cloud Standards Coordination Phase 1
CSC-2	Cloud Standards Coordination Phase 2
CSI	Cloud Storage Initiative
CSMIC	Cloud Services Measurement Initiative Consortium
DMTF	Distributed Management Task Force
EC	European Commission
ENISA	European Union Agency for Network and Information Security
EPO	European Patent Office
FRAND	Fair, Reasonable And Non Discriminatory
GS	Group Specification
HP	Here we should take away the reference to HP in Clause B2 Table 2 Eucalyptus (see below)
IaaS	Infrastructure as a Service
ICT	Information and Communications Technology
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IP	Intellectual Property
IP	Internet Protocol
ISG	Industry Specification Group (an ETSI structure for open membership projects)
ISO	International Organization for Standardization
IT	Information Technology
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
JSON	JavaScript Object Notation
JTC	Joint Technical Committee
KVM	Kernel-based Virtual Machine
NFV	Network Function Virtualization
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NIST	National Institute of Science and Technology
OASIS	Advancing Open Standards for the Information Society
OCCI	Open Cloud Computing Interface
OCF	Open Certification Framework

ODCA	Open Data Center Alliance
OGF	Open Grid Forum
OMA	Open Mobile Alliance
ONF	Open Networking Foundation
OPNFV	Open Platform for NFV
OSS	Open Source Software
OVA	Open Virtual Appliance
OVF	Open Virtualization Format
PaaS	Platform as a Service
SaaS	Software as a Service
SDN	Software Defined Network
SDO	Standards Developing Organisation
SIIF	Standard for Intercloud Interoperability and Federation
SLA	Service Level Agreement
SME	Small or Medium Enterprise
SMI	Service Measurement Index
SNIA	Storage Networking Industry Association
SSO	Standards Setting Organization
STF	Specialist Task Force (an ETSI structure for internal projects)
TMF	TeleManagement Forum
UCD	Unified Cloud Disk
VIM	NVF Virtualised Infrastructure Management
VM	Virtual Machine
VNF	Virtualised Network Function
VNFC	VNF Component
W3C	World Wide Web Consortium
WS	Web Service

4 Standards and Open Source

4.1 Context

The Cloud Standards Coordination project (CSC)

Cloud Standards Coordination Phase 1 (CSC-1) took place in 2013 as a community effort supported by ETSI and primarily addressed the Cloud Computing standards roadmap. In December 2013 the results were publicly presented in a workshop organized by the European Commission (EC).

The CSC-1 Final Report [i.1] provides a snapshot on the Cloud Computing standardization landscape at the end of 2013. It is available at: http://csc.etsi.org/resources/CSC-Phase-1/CSC-Deliverable-008-Final_Report-V1_0.pdf.

Cloud Standards Coordination Phase 2

Given the dynamics of the Cloud Computing market and standardization, Cloud Standards Coordination Phase 2 (CSC-2) was launched in February 2015 with, in particular, the main objective of producing an updated version of the snapshot of the Cloud Computing standardization landscape. CSC-2 aims at better taking into account the needs of Cloud Computing customers on their Cloud related requirements and priorities. This will help CSC-2 to further assess the maturity of Cloud Computing standards and evaluate how standards can support the Cloud Computing customers' priorities.

Analyzing the relationship of Standards and Open Source

The question of Open Source has been alluded to in the Cloud Standards Coordination Phase 1 report [i.1], but not directly addressed:

> "Another aspect of the cloud computing environment that is worthy of consideration is the role of the various Open Source projects which are addressing many of the topics discussed in this report. While not formal standards, the Open Source projects are creating tried-and-tested APIs, protocols and environments which address aspects of interoperability, portability and security relating to cloud computing. It is possible that future specifications and standards may derive from one or more of the Open Source projects. Some examples of positive interaction have already been seen between standards bodies and Open Source projects that should be encouraged. The role of Open Source projects was not addressed in this report" (see [i.1], clause 6.1).

The present report addresses some of the points mentioned above, in particular regarding the positive interaction of Standards Setting Organizations (SSO) and Open Source communities.

4.2 Objectives

The present report will elaborate on the differences and overlaps between Open Source and standardization with the purpose of outlining areas where, despite these differences, Open Source communities and Standards Setting Organizations might come together to further add value to the Cloud Computing space.

The main objectives are to:

- Understand the relationship between Open Source and standards and vice-versa via the identification of a number of interaction scenarios involving Standard Setting Organizations and Open Source communities. These scenarios are not specific to Cloud Computing. Some of them are already visible and some only emerging.
- Clarify how these scenarios apply to Cloud Computing.
- Collect information upon the perceived strategies and visible actions of the SSOs regarding Open Source, and how they match the above scenarios.
- Collect information upon the perceived strategies and interactions of the Open Source projects towards standardization, especially when the interaction scenario involves one or more of the SSOs relevant in Cloud Computing.
- Propose recommendations to foster positive interaction, to suggest areas for collaboration between both communities on ways to support this interaction (e.g. technical frameworks, interoperability, intellectual property).

4.3 Approach

As it will be outlined a number of times in the remainder of the present report, standardization and Open Source are serving rather different purposes and have developed different ways to achieve their own goals. Therefore, the following is not going to be a debate on the respective merits (or lack of) of each approach.

The report is mostly focused on the relationship between standardization and Open Source in Cloud Computing. The understanding of this relationship may require that some consideration will be made of topics outside this precise scope. However, this has been limited to the maximum and the report is not addressing the following questions:

- The debate on the different meanings of "open". Different approaches to "openness" are coexisting, in particular regarding "open standards". The present report will refer to the EU regulation (see [i.2]), as was also the case for Cloud Standards Coordination Phase 1.
- The debate on the many options for intellectual property licensing. Different approaches are coexisting in Open Source communities as well as in standardization. Despite its importance, this question has been considered as outside of the scope of the present report.
- The debate on the respective merits of Open Source licenses. The same remark as above applies.

- The contributions of organizations that are not directly involved in standards making or Open Source projects in Cloud Computing that are outside the scope of the work., even if they are addressing important questions such as promotion, marketing, etc.

4.4 Content of the report

Clause 5 of the present document is a general analysis of the main differences between standards-making and Open Source (Software) and the related challenges. Though this analysis is not addressing Cloud Computing specifically, the remarks made apply also in this context.

Clause 6 is presenting a framework for the analysis of the interactions between standards (and in particular Standards Setting Organizations) and Open Source (and in particular Open Source organizations or projects). This framework is not specific to Cloud Computing but may be used in this context. It is used in the following clauses.

Clause 7 outlines some trends and open questions regarding the evolution of SSOs' and Open Source communities' expectations, strategies and perceived evolutions.

Clause 8 highlights conclusions and recommendations from the analysis done in the present report.

Clause 9 suggests some areas for further work.

Annex A is a compilation of information related to the undertakings of major SSOs in Cloud Computing related to Open Source.

Annex B is a compilation of information related to the undertakings of major Open Source communities and projects in Cloud Computing related to standardization.

Annex C introduces several examples of the scenarios outlined in Clause 6 in the context of Cloud Computing.

5 Standards and Open Source: definitions, objectives and interaction challenges

5.1 Definitions and objectives

5.1.0 Introduction

Clause 5 presents some generic characteristics of standards and Open Source (i.e. non-specific to Cloud Computing) and how standards and Open Source solutions together can help drive the development and uptake of Cloud Computing.

5.1.1 Standards

Definition

A *standard* is defined as "a document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context" (see [i.10]). This definition is in fact amplified by ETSI's rules for drafting standards (available via the ETSI Portal).

Standards are typically manifested as specifications that can be used as is or as elements of larger products, solutions or other standards. A standard can be compared to or said to constitute a reference based on which one can build products or services that all share the same specifications, and thus are "compatible" at some level. Standards are of various natures, may apply to different contexts and are not always directly related to an implementation.

A standard may be universal in nature, and is often used internationally and/or independent of a particular industry or vertical domain. Standards can also be developed to support a particular domain, vertical or industry sector.

Standards tend to be stable over time. Another frequently mentioned characteristic is that a standard should be technically agnostic/neutral, unless developed in support for a particular technology platform. This allows innovation to take place in implementations.

Standards Setting Organizations

A Standards Setting Organization (SSO) refers to any organization that develops and maintains standards. Some essential elements of the operation of an SSO (see [i.2], [i.4] or [i.12]) are:

- Transparent and publicly accessible decision-making processes.
- Collaborative consensus building, extensive consultation & review efforts.
- Formal procedures and mature processes.
- Fair access to standards at zero or nominal cost.
- Deliberate selection of future standards through pre studies, study groups or similar preceding any decision to develop a standard.
- Market support and usage.

Benefits

Whilst success and adoption of individual standards may vary, in general the known benefits that come from relying on standards are:

- Stability. The more standards are incorporated and used in ICT solutions, the more likely it is that the solution based on them will be stable over time.
- Focus on the core functionality. As a consequence of using standards, the developers of ICT solutions can spend the most of their efforts on creating support for the core functionality that is requested by the users.
- Widespread use and Interoperability. Using standards increases the probability of interoperability between solutions; standards for exchange of information for example are commonly based on specifications that are built for any technology platform and with support for different underlying technologies in mind.
- Technology/implementation neutrality. In particular, this is a significant factor to support avoidance of lock-in by allowing multiple implementations from different providers.
- Regulatory/Governmental Policies/Legal aspects. Standards are often used a support for regulation.

5.1.2 Open Source

Definition

Open Source refers to a way to develop solutions collaboratively. Open Source solutions rely on a "community" that is responsible for the development, provisioning and maintenance of the Open Source solution. Most OSS solutions have been developed with independence from the underlying environment as a main objective. Initially, Open Source solutions were particularly available on freely available technologies (such as Linux®, Apache, Java distributions, etc.) but are today available also on most commercially available technology platforms.

NOTE 1: "The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis."

NOTE 2: "The Apache Software Foundation owns all Apache-related trademarks, service marks, and graphic logos."

NOTE 3: "Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners."

Applied to the development of solutions, Open Source is characterized by the following:

- Decentralized production of source code.
- Collaboration across geographies and organizations.
- Variants, known as "forks", that sometimes are brought back into the originated version of the source code product or in other cases spun off into another product.

- Usage of Open Source licenses (see [i.6] and [i.7]).

Open Source communities

There are many Open Source communities in Cloud Computing. Some of them are listed and analysed in Annex B of the present document.

Benefits

Whilst success and adoption of individual open source project may vary, in general the known benefits that come with Open Source are:

- Shared co-development resources enabling collaboration and reducing development cost to each participant.
- Availability of many resources due to the collaborative and community-based nature of Open Source.
- A development model based on recurring and incremental releases and improvement that fits well with concepts such as "continuous delivery", "team based development" and "agile development".
- Modular and clearly defined products and services with improved flexibility for customization.
- Support to multiple underlying environments as a key factor to avoidance of lock-in.

5.2 Different objectives, different approaches

The Open Source approach is useful to, at least, two categories of stakeholders:

- Developers are benefitting from a carefully elaborated and fine-tuned innovation framework: tools, methods, governance, recognition, etc. This framework is fully supportive to major requirements from this community: systematic usage of source code at the centre of the development, support of agile methodologies, extremely short cycle times, to name a few.
- Some organizations (e.g. enterprises, industry associations, service providers, etc.) have quickly endorsed the innovation power of Open Source and incorporated it into their strategies. They all share the objective of creating ecosystems around innovations to rapidly test their value and shorten their time-to-market. The 'open innovation' based model prevalent as a result of Open Source enables the creation of value-added services on top of the source code.

The leading force in Open Source is the (source) code: "the code is the proof" (that the idea is sound, that it is implementable, that its works, etc.). In consequence, Open Source has some specific characteristics that make it different from standardization:

- The focus of the Open Source work is the development of an independent set of source code that can possibly be forked into another independent set of source code that will provide a different solution. This approach to multiple implementations is different than the one in standardization (which is in most case a basic assumption for the development of standards). But in any case, it should be clear that Open Source communities as well as SSOs consider multiple implementations as a key aspect of their work and are organized to support them (e.g. via a wide use of Plugfests).
- Open Source development does not necessarily rely on prior (ex-ante) specifications. In some extreme cases, a written specification and even documentation are hard to find.
- OSS is concerned with interoperability if and when it is useful and needed, for instance to enforce multi-vendors support.

On the other hand, the leading force in standards is the specification:

- The work in SSOs assume an ex-ante plurality of implementers and tries to avoid the choice of a given technology against other possible candidate ones.
- SSOs ensure neutrality vis-à-vis implementations via stable and well-controlled specifications. The major objective is to guarantee interoperability and in most of the cases to provide the means to verify it (via test specifications, validation tools, interoperability testing, etc.).

- SSOs are guaranteeing the neutrality vis-à-vis technologies through their internal processes, and a permanent search for consensus (with a result to reduce the concerns related to antitrust).

5.3 Main challenges to an efficient interaction

5.3.1 Technical challenges

Architecture

With the development of more and more complex systems, standards no longer rely only on the definition of protocol to support interoperability. They are also more and more relying on reference architectures, functional decompositions and reference points that are slowly evolving over time.

For Open Source products, the situation is comparable: to distribute the work load between various contributing programmers or code producing communities, a proper architectural and functional decomposition of the software under development is mandatory for OSS development (see [i.8]).

Incremental releases versus updates

Open Source products are largely evolving incrementally: new features are prototyped, tested and adapted very rapidly. The stability of the code is a major issue open source projects have to address by implementing proper measures for release management and versioning. Standards on the other hand are developed once, and then updated (more or less) regularly, until they become obsolete.

Standard document and source code

Standards Setting Organizations and Open Source communities produce and distribute artifacts that are different in nature:

- Standards Setting Organizations produce standards that are commonly manifested in documents that specify requirements, architecture and protocols/APIs of a system or a part of a system. The evolution of a standard is based on change requests that are examined during periodic reviews and possibly implemented via a change request in the standard. The coherent development of the standard is supported by tool environments that are essentially managing document versions associated to a list of revisions. Note that some Standards Setting Organizations guidelines include the need of having source code implementations of the standard (e.g. W3C, OGF).

- Open Source communities produce source code, a collection of computer instructions written using some human-readable programming language, usually as text. This source code evolution is guided by a permanent flow of change requests that are constantly examined by reviewers and implemented on-the-fly if deemed accurate. While source code is the main output of Open Source Projects, a solid product documentation including code documentations, architecture and functional specifications based on requirements collections or standards, and user and installation guides, is crucial for a successful OSS development. Open Source communities also often produce documentation associated with the open source code (e.g. architecture, API textual description).

Interoperability

Interoperability is an important topic to consider for standards and Open Source, though for somewhat different reasons. In the development of standards, achieving the highest degree of interoperability for any given standard is normally one of the main objectives (see [i.14] for the work done in ISO/IEC JTC1 SC38 on interoperability and portability). For Open Source projects, achieving interoperability is also important, but typically only within the technology context of the Open Source project in question.

For Cloud Computing efforts to be successful when using standards and Open Source, it is subsequently important to understand, keep track of and address all aspects of interoperability that apply in the Cloud Computing context in question. The present document will not go in-depth into suggesting concrete actions to address interoperability aspects related to standards and Open Source, but some high level recommendations specific to standards and Open Source can be made:

- Standards and Open Source are not mutually exclusive, but complementary. As a consequence, implementing Cloud Computing solutions based on standards using Open Source as the way to develop and provision the solutions is possible. Keeping focus on the particular characteristics, advantages and weaknesses of standards and Open Source in order to mitigate any problems will probably be important.

- Using Open Source to implement standards as Cloud Computing solutions is not enough. Open Source and standards by themselves do not solve and address all needs and requirements of Cloud Computing. Additionally, one needs to understand the user-specific needs and requirements, processes to be supported, information and data to be processed, security policies in place, long term goals, resource availability and limitation and much more. The complexity of creating open, transparent, secure and agile business supporting ICT solutions should not be taken lightly. However, standards and Open Source can together assist in accelerating the provisioning of Cloud Computing.

Tools and frameworks

The differences between standards and Open Source outlined in the above paragraph have their counterpart in the tools and frameworks currently used by their respective "developers":

- The tools that support standards development and distribution are typically word processors (with the internal ability to track changes) as well as document repositories (tree-based repositories, sharepoints, portals, etc.). On the other hand, there is the potential for aspects of certain standards - e.g. programming interfaces, data models or ontologies, to be described in a machine-readable fashion that can be automatically processed (e.g. into a specific library written in a specific programming language).

- An Open Source product is essentially developed around source code version management tools embedded in larger frameworks offering peer review, collaboration, etc. (such as Git and GitHub). These tools are optimized for the management of source code, less for handling paper (which is not perceived in general as a drawback by the users). Therefore, open source projects rely in addition on document management systems for product and project documentations. Moreover, tools for quality assurance (e.g. overnight testing), automated license compatibility checking, project maturity validation, software metrics, etc., are used by many open source projects.

Some SSOs have developed very effective tools, frameworks and processes for the standards they maintain. For instance, the proof of interoperability requires a lot of procedural and technical solutions (e.g. test methods, description languages, test environments, etc.) that SSOs can potentially offer to other organizations. OSS communities may benefit from using the test or Quality Assurance services of SSOs, provided that these services have been adapted to the requirements of Open Source, e.g. by Open Source repositories, Open Source-based test development or conformance testing.

5.3.2 Organizational challenges

(Long-term) Maintenance

Maintenance (even long-term) is part of the basic operations of the SSOs, especially the SDOs that are in charge of ensuring that critical standards remain available over long period of time and can be adapted to the changes in their operating environment.

When it comes to OSS, this objective is not always taken into account when the projects are launched, though some Open Source communities in the ICT domain are willing and able to do maintenance over a long period of time. However, in some cases, this maintenance is not possible, for instance because:

- Some OSS projects have to be discontinued, e.g. for lack of resources or loss of basic experts.

- In long-lived systems, such as the ones that are deployed in telecommunications, some "de-facto" products may be deployed on the field for durations that exceed the capabilities of the originating Open Source communities.

In this case, the long-term maintenance may have to be taken over by a different organization. This can potentially be done by SSOs, provided that they are organized to provide the corresponding service to Open Source communities.

Governance

In all organizations, a clear governance model, with explicit and well-documented rules for decision-making, is a pre-condition for well-motivated and efficient participation of the actors. There are many governance schemes that fit this purpose, and none is particularly better than the other ones. In any case, clarity is a clear requirement.

In general, the governance of SSOs has been organized around the need to create and manage consensus as the basis for making widely accepted and relevant standards. The technical decisions are, in most cases, delegated to "Technical Groups" (or committees, etc.) whereas the global and strategic decisions are under the responsibility of a Board (or similar group), in general composed of elected members.

In Open Source communities, two models are well-established (with a lot of variants that share some elements of both): the so-called "benevolent dictatorship" where control is under a single entity (person or group), and the distributed control where parts of the governance is allocated to individuals that have been recognized as capable of the task.

Another difference between SSOs and Open Source communities regarding governance is its focus: while SSOs governance is directed towards achieving consensus on technical issues and addresses a relatively closed set of stakeholders, the governance of Open Source communities may address a larger collection of stakeholders. Therefore, additional requirements on the transparency of decision-making processes addressing a community with fluid borders (both in terms of contributing persons and of relevant opinions) have to be taken into account.

When addressing the interaction between Open Source communities and SSOs, the question of governance should not be underestimated. A discussion between both parties for the definition of solutions, work programs, etc., will require a clear and transparent decision point on both sides, based on well-defined processes with actors in well-defined roles.

5.3.3 Intellectual property challenges

Different issues are at stake when it comes to Intellectual Property Rights (IPR). This is a field of very active and sometimes highly controversial debates.

Some aspects are very significant for the SSOs, in particular those with an IPR policy that raise patent licensing questions or issues. Some aspects are very significant to Open Source communities that want to ensure that the use of their OSS product is not impacted by patent claims holders in particular the absence of a license or unreasonable licensing terms and conditions.

The questions of Open Source licenses and of SSOs' IPR policies are amongst those that require clarification in order to ensure an efficient interaction between SSOs and Open Source communities.

Open Source Licenses

The licenses applicable to an Open Source product or community are extremely varied (see [i.6]). Even if a vast majority of them belong to a limited set (the 6 most used licenses represent 81 % of Open Source projects, see [i.7]), the question of the license applicable to an Open Source product or within an Open Source community cannot be ignored if proposed for inclusion in a Standard or as an implementation of a Standard.

Patent and copyright policies

First, an SSO will have to ensure that the Open Source material that they want to include in their standards should not be associated with usage restrictions attached to an Open Source license, e.g. restrictions upon the distribution of commercial products and services that come in conflict with the SSO Intellectual Property Rights policy (see [i.5]).

Second, the question of whether or not the Intellectual Property Rights (IPR) policies of the SSOs are compatible with the implementation of their standards by Open Source communities is highly debated. In particular, it is often perceived (though no consensus of this point has been reached) that implementation of standards available under a FRAND license in Open Source may be difficult (see [i.3], [i.13] or [i.15]). Several examples of different approaches can be noticed:

- The ETSI Special Report on "Working in ETSI within an OSS context" [i.5] addresses the question of FRAND, provides some recommendations and lists some actual implementations of Standards by Open Source projects that comply with the ETSI IPR policy.
- The Open Standards Requirement (see [i.4] by the Open Source Initiative) defines criteria to ensure that an "open standard" will not prohibit implementations in OSS conforming to the standard.
- Some SSOs policies only support the use of "royalty-free" policy for implementation of their standards.

6 Standards and Open Source: Interaction scenarios

6.1 An overall view

The interaction between Standards Setting Organizations and Open Source communities has its origin in a reciprocal need to benefit from each other's products (e.g. standards from an SSO, or software from an Open Source community) and services (e.g. Quality Insurance or Interoperability Testing).

A few exemplary scenarios are used below to differentiate and classify some typical interactions. These scenarios are based in most cases on actual examples, sometimes with a long past experience that allows for relevant assessment. On the other hand, some are related to initial undertakings or perceived intentions of the actors but may not be yet entirely validated.

It is expected that, in case Open Source communities and SSOs want to engage a collaboration, they will have to discuss along the lines of one or more of these scenarios and address the issues that have been defined in the previous clause.

6.2 The scenarios

6.2.1 An Open Source community implements standards

6.2.1.0 Introduction

This interaction scenario (Scenario 1) includes two variants depending on whether the standards are already existing published standards (Scenario 1a) or are emerging standards (Scenario 1b).

6.2.1.1 An Open Source community implements existing standards from a Standards Setting Organization

In this scenario (Scenario 1a):

- An SSO Technical Group has developed and published a set of standards - that will be maintained and may be further evolved. This set includes detailed protocol/API standards that can be used for implementation purposes.
- An Open Source community outside the SSO wants to make a reference implementation of these standards - that will be further distributed (by the Open Source community itself or by specialized distributors) and integrated into commercial products under conditions defined by an Open Source License.
- The OSS implementation is set to be fully "compliant" with these standards or can lead to evolutions of the standards published and maintained by the SSO.

Examples of such implementations can be found in the ETSI 2012 report [i.5]. For Cloud Computing, an example of this scenario includes the OpenStack implementation of DMTF CADF specification (see Annex C for further information).

NOTE: "The OpenStack Word Mark, the OpenStack Logos and all OpenStack trademarks (hereinafter referred to individually as an "OpenStack Mark" or collectively as "OpenStack Marks") are trademarks of the OpenStack Foundation."

6.2.1.2 An Open Source community implements emerging standards from an SSO

In this scenario (Scenario 1b):

- An SSO Technical Group is developing a set of standards that is not yet stable and published. This set includes standards at various stages of the standards development chain (e.g. standards on requirements, architecture, protocols/APIs).
- An Open Source community outside the SSO wants to undertake an implementation of this set of standards.

- The OSS implementation may be only "inspired by" the on-going work of the SSO and can:
  - Significantly diverge from it if the progress of the Open Source implementation is not fed back to the SSO. In some case, the result of the Open Source community work is a product implementing a subset of the standards under preparation in the SSO.
  - Provide early feedback on the standards under elaboration in the SSO specification by rapidly prototyping some aspects of it, in order to come more rapidly to a stable version of the relevant standards.

Scenario 1b is a variant of Scenario 1a, with potentially significant impacts on the emerging standards under preparation in the SSO.

An example of Scenario 1b is the interaction between the Industry Specification Group "Network Function Virtualization" in ETSI (ISG NFV) and the Open Platform for NFV (OPNFV) (see more details in Annex C).

6.2.2 An SSO develops an Open Source reference implementation

In this scenario (Scenario 2):

- An SSO Technical Group has developed and published a set of standards - that will be maintained and may be further evolved.
- To speed-up the market adoption, the Technical Group decides to develop a reference implementation of these standards or of a subset of them, using an Open Source methodology and environment (including for testing purpose).

The result of the SSO work is a bundle of the Standards and the reference implementation source code. The reference implementation:

- Is one of many implementations in line with the published set (or subset) of standards.
- Can be used by an Open Source community for inclusion in its product and distribution or directly included in commercial products (e.g. some vendors/integrators) under conditions defined by an Open Source License.

To make this happens, the SSO should have implemented internally an Open Source hosting framework.

Examples of this scenario include the Open Source implementation by OMA of the RCS specification, which is integrated into commercially available products. Similar initiatives are starting in oneM2M and 3GPP partnership projects. For Cloud Computing, one example is the DMTF Standards Setting Organization, which is developing an OpenStack implementation of the CIMI specification (see Annex C for further information).

6.2.3 An SSO develops standards based on the results of an Open Source community

In this scenario (Scenario 3):

- An Open Source community is designing and developing a software implementation that fulfils the needs of an SSO, e.g. providing an implementation covering the functional and architectural requirements expressed in standards published or under development by that Standards Organization.
- The Standard Organization decides to endorse the results of the Open Source community and develops standards based on the documented APIs developed by the Open Source community.
- The Open Source community has opted for an Open Source license.

The resulting standard is a set of "tried and tested" APIs acting as a reference in its industry segment.

An example in Cloud Computing is the DMTF specification of an OpenStack profile for CADF (See Annex C for further information).

6.2.4 A collaboration ("joint project") is established between a Standard Organization and an Open Source community

In this scenario (Scenario 4):

- A joint collaboration ("joint project") between a Standards Organization Technical Group and an Open Source community is established with the objectives of developing together a set of standards and an Open Source implementation of these standards.
- The set of standards includes standards at various stages of the standards development chain (e.g. standards on requirements, architecture, protocols/APIs) while the Open Source implementation provides a reference implementation of these standards.
- This collaboration includes the establishment of a joint steering Technical Committee whose tasks is to coordinate the development of standards by the Standard Organization and the development of the Open Source implementation. This Technical Committee will drive the roadmap in terms of use cases, requirements and architecture that should be supported by the Open Source implementation.

Scenario 4 can be viewed as a combination of Scenario 1b and Scenario 3 with the addition of a formal "joint project" between a Standard Organization and an Open Source community to help fostering and coordinating efforts in a coherent and agile manner.

It has to be noted though that an equivalent scenario exist in the field of standardization where collaboration between Standard Organizations are possible, e.g. partnership projects between regional Standard Organizations such as 3GPP or OneM2M or collaborative teams between ISO/IEC JTC 1 sub-committees and ITU-T study groups.

6.3 Current and future situation

Some of the above scenarios are well established whereas some are more in their early discussion phase within the concerned organizations (Open Source communities as well as SSOs). With the expected clarification of the interactions between SSOs and Open Source communities, more examples of the above scenarios as well as new scenarios are likely to emerge in the coming years.

7 Better aligning the standards and OSS communities

7.1 Alignment: if and when needed

Regarding alignment, it is not intended here to find a way to come up with similar ways of working and expected outcome (which would be impossible considering the major differences in purpose identified in clause 5). Actually, there is a difference between the expectations of SSOs and Open Source communities from this standpoint: SSOs are realizing that, in order to respond better to the needs of the standardization stakeholders, they need to find ways to incorporate Open Source in their very genes. This is not the case for Open Source organizations that do not feel the same pressing need for cooperation. This asymmetry between the view of SSOs and OSS is in particular visible in the scenarios identified in clause 6, where a majority stems from the needs of SSOs rather than the opposite.

Alignment will come from the realization by both SSOs and Open Source communities that they have good opportunities to improve the effectiveness of their respective results (more speed, better quality, more testing, improved maintainability, etc.). This is especially true in the Cloud Computing industry segment.

The progress will come with the identification of mutually satisfactory solutions to specific problems. It is not expected that each SSO will set up a complete Open Source strategy implementing all the scenarios of clause 6 (though some may want to do it).

7.2 Strategies

As a result of the different expectations pointed out above, the maturity of strategies to address Standards and Open Source interaction is different for SSOs and Open Source organizations.

The SSOs strategies are regarding Open Source are currently at different stages:

- Many are just starting to discuss the issue and identify their objectives.

- Some have already a previous experience and have clearly outlined objectives.
- A few have started to analyse how their assets may be impacted by this new approach (e.g. technical organization, membership policy, Intellectual Property policy).
- Only a few have a very complete and detailed approach.

The Open Source organizations strategies regarding standards are largely conditioned by the answer to a first question: how can they benefit from collaborating with the SSOs?

Annex A of the present document is outlining the role of some SSOs involved in Cloud Computing standardization (e.g. those who have been identified during Cloud Standards Coordination Phase 1) and the status of their Open Source strategies.

Annex B has a similar objective for some Open Source organizations involved in Cloud Computing.

7.3 Solutions

Some possible solutions have been highlighted in the previous clauses but a first analysis shows that there are not so many in place at this stage. However, the following approaches can be highlighted.

1) Change the setting of SSOs to accommodate Open Source projects within the existing organization. This can take several forms, possibly complementary:
   - Definition of specific technical organizations within the SSO, different from their regular technical committees. For instance, the creation of Technical Committees with special rules within the SSO (e.g. specific membership rules, different IPR policies).
   - Definition of specific membership conditions meant to attract regular participant of Open Source projects within an SSO context (e.g. special conditions to academics, to micro-enterprise, etc.)
   - Investigation of the benefits, risks and required changes associated to the adaptation of the IPR policies when it becomes necessary to accommodate some of the expected licensing schemes in the Open Source community (provided there is a real - in particular business - advantage).
2) Define OSS-oriented services in SSOs. To offer attractive services, some elements should be defined:
   - List of services in support of Open Source communities:
     - Hosting of OSS project, including the availability of an OSS platform.
     - Testing support, in particular interoperability testing, Plugfests, etc.
     - Quality Assurance.
     - Maintenance.
     - etc.
   - Conditions for use: legal framework, Service Level Agreement, etc.

8 Conclusions and Recommendations

The goals of standards are different from those of Open Source Software (OSS). Standardization aims at producing specifications that can be implemented in any appropriate technology. This is essential to avoid vendor lock-in situations as well as for promoting innovative implementations. Open Source projects aim to favour the rapid development of high quality software or reference implementations allowing for the discovery and validation of concepts or providing solutions that respond to given use cases and derived functional and architectural requirements. Open Source is also potentially an important vector of growth and innovation in the Cloud Computing space.

Standards and Open source approaches have an important role to play in complementing each other, and in fact, to some extent, more and more ICT projects do combine the two approaches. However, only standards provide the stability and technology neutrality, in particular required for public policies that seek to improve interoperability while reducing lock-in to any particular technology solution.

In its role of supporting Standards, Open Source can:

- Help overcome limitations in the development and implementation of Standards.
- Speed the development and improve the quality of Standards.
- Facilitate the understanding of standards for implementers.
- Improve the Standard interoperability by using Open Source reference test-bed implementation and testing software.

Therefore in order for standards and Open Source to adequately support each other, the following recommendations are proposed.

Collaboration

- Encourage collaboration between OSS communities and SSOs working on similar or closely related topics, e.g. NFV and OPNFV, possibly through joint events like workshops, plugtests.
- Encourage the creation of "joint projects" between the SSOs where the standards are developed and Open Source communities in order to push for close relationship, interaction, exchange and cooperation.

Roadmaps

- Make sure that collaboration between SSOs and Open Source communities address the known Cloud Computing (standards) gaps, e.g. in Service Level Agreement, Security, Privacy and Integrity.
- Encourage Open Source initiatives to standardize their specifications that are important for interoperability (e.g. APIs: Data Model, Protocol, Format).

Organization

- Facilitate the implementation of Open Source solutions based on Standards (developed or under development in a SSO), in particular by narrowing the gap between different approaches of Patent and Copyright policies.
- Ensure that pre-standardization activities (e.g. those emanating from research projects) can be sustained over a longer period in order to allow for a smooth transition of results within Cloud Computing standardization.

Education, dissemination, promotion

- Encourage SSOs for early and increased effort in the dissemination of plans for/work on new Cloud related specifications towards the Open Source communities in the Cloud area.
- Engage industrial users of Cloud Computing Open Source communities.

9 Areas for further study

Some areas for further study have been outlined in the previous clauses of the present document:

- Clarification of the Standards and Open Source interaction scenarios:
  - Content.
  - Completeness.
  - Operational working procedures.
  - Methods and tools.
- Clarification of the impact on SSO organizations of greater compatibility of FRAND and Open Source licenses.

Annex A:
Standards Related Organizations Approaches

This annex is presenting some initiatives of SSOs and a few related organizations in the field of Cloud Computing that relate to their interactions with Open Source communities.

Table A.1: Strategies of SSOs towards Open Source communities

Organization	Type	Scope in CC	Strategy, position, initiatives with Open Source
3GPP	SDO	Transparent file sharing over mobile networks.	There are a number of Open Source implementations of 3GPP specification.
ATIS	SDO	ATIS has developed a number of Cloud Computing related standards; some have been retained in the list of Standards of CSC (phase 1).	ATIS has developed a complete framework, in particular legal, to allow for standard developers and Open Source developers to work together, taking into account different models for business, licensing or IP protection.
CSMIC	SSO	The CSMIC aims to contribute to the solution of the cloud-based service measurement problem. Including development of a Service Measurement Index (SMI) and a framework for organizing and classifying service measures. The goal is a standard way of describing and documenting service measures.	
DMTF	SSO	DMTF developed the Open Virtualization Format (OVF) that is broadly used to describe virtual machine images in a portable way. DMTF also specified the Cloud Infrastructure Management Interface (CIMI) which is a self-service IaaS management interface and the Cloud Audit Data Federation (CADF) specification defines a normative event data model along with a compatible set of interfaces for federating events, logs and reports between cloud providers and cloud customers.	DMTF has a long history supporting Open Source implementations of its standards. DMTF DSP2038 defines a CADF representation for use with the OpenStack Cloud Management Platform.
ENISA	European Agency	ENISA as part of the activities under the EU cloud strategy developed a list of different certification schemes that could be relevant for potential Cloud Computing customers. The creation of this list is explicitly mentioned as a key action in the European Cloud Strategy. This list was developed by ENISA in tight collaboration with the European Commission and the private sector.	The certification schemes list can be found at: https://resilience.enisa.europa.eu/cloud-computing-certification. (Interested readers were referred to a paper ENISA published last year that gives an overview of a range of information security certification schemes, used in different sectors.)
ETSI	SDO	ETSI is the home of the Network Function Virtualization (NFV) Industry Specification Group (ISG).	The ETSI Board has developed a first framework for Standards and Open source in 2012. They are currently in the process of expanding it to take into account more scenarios.

Organization	Type	Scope in CC	Strategy, position, initiatives with Open Source
IEEE	SSO	P2301 is a working group for creating a Guide for Cloud Portability and Interoperability Profiles. P2302 - Standard for Intercloud Interoperability and Federation (SIIF) is the first Cloud standardization activity of IEEE.	IEEE expects these newest standards will not only follow the consensus-based process championed by IEEE, but will also leverage the latest in technology development best practices, such as live global test beds and Open Source references.
IETF	SSO	The actual technical work of the IETF is done in its working groups (WGs), which are organized by topic into several areas (e.g. routing, transport, security, etc.). IETF is specifying protocols and data models for: • SDN (NETCONF/Yang data models, PCEP; Service Function Chaining, etc.); • RTCWeb; • HTTP.	IETF has started to discuss a framework to support Open Source. During its meetings, IETF is holding a Hackathon to encourage developers to discuss, collaborate and develop utilities, ideas, sample code and solutions that show practical implementations of IETF standards. IETF individuals start using GitHub for editing IETF drafts, prototyping, test suites (e.g. https://github.com/http2).
OASIS	SSO	Topology and Orchestration Specification for Cloud Applications (TOSCA).	Leading Open Source organizations have also embraced TOSCA with numerous projects already active, e.g. integration with OpenStack HEAT, OpenNebula.
ODCA	SSO		
OGF	SSO	OGF has developed the Open Cloud Computing Interface (OCCI) specification and other specifications that are useful in Cloud environments though not specifically developed for Clouds, e.g. WS-Agreement for Cloud Service Level Agreements.	There are a number of Open Source implementations of OGF specifications. In general OGF encourages the use of Open Source implementations for evaluating specifications and for testing interoperability. Open Source implementations may be used to develop extensions to specifications. OGF regularly organizes Plugfests to support this.
OMA	SSO	The Open Mobile Alliance (OMA) delivered the technical specification for Unified Cloud Disk (UCD) Enabler V1.0. The UCD Enabler provides unified cloud storage system in mobile cloud computing environments.	OMA has been addressing the question on how SSOs can adapt/evolve such that they better enable the application developer to take advantage of the standard specifications they produce. Some OMA efforts in the area of Open Source are on-going such as the adoption of specific tools for specifications, the usage of Github repository, etc.
SNIA	SSO	The SNIA Cloud Data Management Interface (CDMI) is an ISO/IEC standard that enables cloud solution vendors to meet the growing need of interoperability for data stored in the cloud. The CDMI standard is applicable to all types of clouds - private, public and hybrid.	SNIA's Cloud Storage Initiative (CSI) promotes cloud storage adoption with open standards that provide vendors and end users with choice, interoperability, and portability. CSI leads as an industry neutral authority on cloud storage environments and is committed to educating vendor and end user communities on cloud storage & industry standardization benefits. SNIA also supports Open Source projects in storage.
TMF	SSO	The primary objective of TM Forum's Cloud Services Initiative is to help the industry overcome today's barriers around Cloud services and assist in the growth of a commercial marketplace for cloud based services. The centrepiece of this initiative is an ecosystem of major buyers and sellers who will collaborate to define a range of common approaches, processes, metrics and other key service enablers.	As part of its ZOOM (Zero-touch Orchestration, Operations and Management) project, TMF has produced a ZOOM position on Open source.

Organization	Type	Scope in CC	Strategy, position, initiatives with Open Source
W3C	SSO	W3C itself has not yet developed a specific standard for Cloud computing. However, as most (if not all) of the WS-* specifications are targeting web-based.	W3C considers Open sources need Open standards and Cloud services need Open Standards. W3C is happy to contribute and collaborate: • Open Platform Capabilities. • Data and Semantic Support. • Advanced Service Management.

Annex B:
Open Source Communities Approaches

B.1 Open Source Cloud middleware projects

In this clause the major Open Source Cloud middleware projects regarding their support for Cloud standards, their relation to Standards Development Organizations and their possible involvement in the development of Cloud standards have been analysed.

Table B.1: Strategies of Open Source organizations towards SSOs

Organization	What they do in CC	Strategy, position, initiatives with SSOs
Ceph	Open source software storage platform designed to deliver object, block, and file storage from a single distributed unified system.	
CloudStack	CloudStack is developing Cloud computing software for creating, managing, and deploying infrastructure cloud services.	CloudStack supports the OCCI Cloud Computing standard. CloudStack as a community is not participating in the development of standards. However, individual members of the Cloudstack community may be active in SSOs, in particular Verizone Terremark claims to be, may be active in the development of open standards.
CompatibleOne Broker	CompatibleOne is a cloud broker based on open standards.	CompatibleOne supports the OCCI and WS-Agreement standards. CompatibleOne as a community is not participating in the development of standards. However, individual members of the CompatibleOne community may be active in SSOs.
Contrail	The CONTRAIL project delivered is an Open Source system in which resources that belong to different operators are integrated into a single homogeneous Federated Cloud. A follow-up activity is OpenContrail.	Contrail supports a number of Cloud Computing standards: OCCI, OVF, CDMI and WS-Agreement. Members of the Contrail project participated in the development of OCCI during the lifetime of the project.
Eucalyptus	Eucalyptus is a software for building Amazon Web Services (AWS)-compatible private and hybrid Cloud Computing environments.	Eucalyptus supports the OCCI Cloud Computing standard. Eucalyptus as a community is not participating in the development of standards. However, individual members of the Eucalyptus community may be active in SSOs.
KVM	Open source software virtualization solution for Linux® on x86 hardware containing virtualization extensions.	
Nimbus	The Nimbus Platform is an integrated set of tools that deliver IaaS Clouds to scientific users; the Nimbus Infrastructure is an Open Source EC2/S3-compatible Infrastructure-as-a-Service implementation also with a focus on scientific users.	Nimbus supports the OCCI and standard. Nimbus as a community is not participating in the development of standards. However, individual members of the Nimbus community may be active in SSOs.
OpenDaylight	Open source SDN controller platform for network programmability to enable SDN and create a solid foundation for NFV for networks at any size and scale.	The SDN controller supports multiple south-bound interfaces and protocols developed in the IETF (OVSDB, Netconf/Yang, PCEP, LISP, BGP, Opflex, CoAP, etc.), ONF (OpenFlow).
OpenNebula	OpenNebula is a cloud computing platform for managing heterogeneous distributed data centre infrastructures.	OpenNebula supports a number of Cloud Computing standards: OVF, CDMI and OCCI. OpenNebula as a community is not participating in the development of standards. However, individual members of the OpenNebula community may be active in SSOs.

Organization	What they do in CC	Strategy, position, initiatives with SSOs
OpenStack	OpenStack is developing a cloud-computing software platform. Most often used for IaaS.	OpenStack supports a number of Cloud Computing standards: OVF, CDMI, OCCI. OpenStack as an organization (represented by the OpenStack Foundation) is not participating in the development of standards. However, individual members of the OpenStack community, especially industrial members, may be active in SSOs.
Open vSwitch (OvS)	Open Source Software switch designed to be used as a vswitch in virtualized server environments. Open vSwitch has also been integrated into various cloud computing software platforms and virtualization management systems, including OpenStack, OpenNebula, etc.	Open vSwitch supports standard management interfaces (e.g. sFlow, NetFlow, IPFIX, etc.), and is open to programmatic extension and control using OpenFlow and the OVSDB management protocol.
OPNFV	Open source project focused on accelerating NFV's evolution through an integrated, open platform.	The scope of OPNFV's initial release is focused on building NFV Infrastructure (NFVI) and Virtualized Infrastructure Management (VIM) of the ETSI NFV architectural framework (ETSI GS NFV 002 [i.16]).
OPTIMIS	A toolkit for managing IaaS Clouds especially supporting hybrid Clouds, Cloud federation and Cloud bursting.	OPTIMIS supports two Cloud Computing standards: OVF and WS-Agreement. Members of the OPTIMIS project participated in the development of WS-Agreement Negotiation during the lifetime of the project.
OW2	The OW2 community is engaged in several cloud computing projects such as CompatibleOne cloud broker, OpenCloudware multi-IaaS PaaS, XLCloud HPC cloud platform, and OCCIware, a formal framework for the management of any digital resource in the cloud.	OW2 facilitates the development of Open Source Software with a strong focus on infrastructure software and cloud computing. OW2 is not a standard organization but encourages its members to take part in standard workgroups. OW2 encourages its projects to support open standards and is starting to have some experience with OCCI.
WSO2	Has developed software for Cloud environments, e.g. the Cloud Gateway for publishing services and data to the Cloud from inside the enterprise. Has a commercial offering for a Cloud environment.	WSO2 supports the OCCI and WS-Agreement standards. WSO2 as a community is not participating in the development of standards. However, individual members of the WSO2 community may be active in SSOs.

B.2 Standards usage summary table

Table B.2 lists the support to standards for the above selected Open Source projects.

Table B.2: Open source products adherence to standards

Organization	OVF	CIMI	CDMI	OCCI	WS-Agreement
CompatibleOne Broker	N/A	No	N/A	Yes	Yes
CloudStack	N/A	Yes	N/A	Yes	No
Eucalyptus	N/A	No	N/A	Yes	No
Nimbus	N/A	No	N/A	Yes	No
OpenContrail	Yes	No	(Yes)	Yes	Yes
OpenNebula	Yes	No	Yes	Yes	No
OpenStack	Yes	(Yes)[1]	(Yes)[2]	(Yes)[3]	No
OPTIMIS	Yes	No	No	No	Yes
WSO2	N/A	No	N/A	(Yes)	Yes

NOTE 1: (Yes) indicates that there is no full implementation but a rudimentary interface only.
NOTE 2: OpenStack supports a number of Cloud Computing standards including OVF, CDMI, OCCI, but this support is mostly via independent open-source add-on projects. The core OpenStack community has clarified that if there was sufficient community desire, further standards support could be incorporated into the core OpenStack projects. Given limited development resources, it is essentially a matter of community priority. (see https://github.com/osaddon).
NOTE 3: See https://wiki.openstack.org/wiki/Occi and https://github.com/stackforge/occi-os.

Annex C:
Interaction scenarios in practice in Cloud Computing

C.1 Case Studies

The description of the scenarios in the previous clause has raised some questions that this clause wishes to address. To do this, a few case studies that serve as a more concrete illustration of the opportunities and issues raised by the collaboration of Open Source communities and SSOs have been selected.

C.2 Sharing specifications: NFV and OPNFV

C.2.1 Introduction

Some of the scenarios above are related to the implementation by an Open Source community of one or more specifications (already existing or under development) coming from an SSO. The question of the adoption of an emerging standard by an Open Source community addressed in scenario 2 can be illustrated by the relationship between NFV (an Industry Specification Group within ETSI) and OPNFV (an Open Source Organization).

C.2.2 The actors

NFV (Network Function Virtualization)

The **NFV ISG (Industry Specification Group)** was created end of 2012 under the auspices of the European Telecommunications Standards Institute (ETSI), in response to a call for action from a group of major operators. Network Functions Virtualization aims to transform the way that network operators architect networks by evolving standard IT virtualization technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage, which could be located in Datacenters, Network Nodes and in the end user premises.

The first NFV ISG outputs were published in October 2013, including a use case specification ETSI GS NFV 001 [i.17], and an NFV Architectural Framework specification ETSI GS NFV 002 [i.16] that identifies NFV system components and the interfaces between them. Then these first outputs were complemented with new deliverables published by ETSI in January 2015. Altogether this first specification phase (known as **NFV Phase 1**) sets the foundational concepts and creates a common vocabulary and architectural framework widely shared in the industry.

The ETSI NFV architectural framework enables Virtualized Network Functions (VNF) to be deployed and executed on a **distributed carrier-grade cloud infrastructure** known as the **NFV Infrastructure (NFVI)**, which consists of pools of commodity hardware resources (computing, storage and network) wrapped with a software layer that abstracts and logically partitions them. In hypervisor-based deployments, a VNF is typically mapped to one Virtual Machine (VM) in the NFVI but may also be split into multiple VNF components (VNFC) loaded on separate virtual machines (e.g. with different scaling requirements). The deployment, execution and operation of VNFs on an NFVI are steered by a Management & Orchestration (M&O) system, whose behaviour is driven by a set of metadata (a.k.a. NFV descriptors) describing the characteristics of the network services and their constituent VNFs. The M&O system includes an NFV Orchestrator (NFVO) in charge of the lifecycle of network services, a set of VNF managers in charge of the lifecycle of the VNFs (including VNF scaling out/in) and a Virtualized Infrastructure Manager (VIM), which can be viewed as an extended Cloud Management System responsible for controlling and managing NFVI resources.

A major challenge remains to achieve **interoperability** for the key interfaces identified in the NFV architectural framework. The **NFV Phase 1** specifications are not sufficient to meet this objective as they only provide a functional description of these interfaces, while interoperability usually requires the specification of a protocol and/or an API and often a data model. One of the most important goals of the **NFV Phase 2** work program is to close the aforementioned interoperability gap by providing the right level of specifications for the key interfaces. A pre-requisite to any protocol or API specification work is to complete the functional description of these interfaces identified during NFV Phase 1. Once this step is done, a follow-up specification phase should lead to a protocol specification, presumably in the form of a profile of - or an extension to - an existing base standard. One key task will be to select the most appropriate base standard per interface.

Closely related to interoperability are the VNF **portability** and VNF migration topics. Portability refers to the ability to deploy a VNF on different types of servers, when live migration refers to the ability to move an active VNF instance from one server to another while ensuring service continuity. One of the work items for NFV Phase 2 will be the specification of a set of interfaces enabling the VNFC code to access acceleration services provided by the infrastructure, in an implementation-independent manner.

Another prospective work item for NFV Phase 2 is a report that studies the internal architectural structure/physical components of an NFVI Node and provide a set of guidelines to support an NFV environment. The goal is to facilitate the availability of these components in a multi-vendor environment. Besides, the publication of a report on the role of software-defined networking (SDN) in the NFV architectural framework is expected.

OPNFV (Open Platform for NFV)

The Open Source Open Platform for NFV (OPNFV) initiative, led by the Linux® Foundation, was launched on the 30th September 2014 with the participation of several IT and telecom vendors and telecommunications service providers. The objective of OPNFV is to provide a reference infrastructure platform for Network Functions Virtualization (NFV). Therefore, large parts of the OPNFV architecture are directly related to the architecture outlined in the documents provided by ETSI ISG NFV. To start with, OPNFV addresses an integrated solution for NFVI and VIM components of the ETSI NFV architecture that together build the infrastructure layer of the NFV framework.

To achieve this goal, OPNFV will work in close collaboration with a number of "upstream" Open Source initiatives (e.g. OpenStack, OpenDayLight, KVM, OVS, etc.). In addition to code development, OPNFV includes a number of requirements projects but also integration and testing projects. OPNFV will execute on a release cadence of a release approximately every six months. OPNFV Release 1 'ARNO' was delivered in June 2015. This first OPNFV release includes the following existing Open Source components (see [i.9]):

- OpenStack Release Juno;
- OpenDaylight Release Helium for Network control;
- Ceph object storage orchestrated by Cinder;
- OVS for virtual switch;
- KVM hypervisor for Virtual Machine.

C.2.3 Working together: opportunities, issues

At the formal level, the current type of partnership engagement is called "Letter of Intent" (LoI) allowing the cooperation between ETSI and OPNFV. This establishes a formal contact between ETSI with OPNFV and serves to exchange operational/promotional information and to identify common roadmaps. The intention is to co-operate in the standardization efforts in the area of NFV and to work together for promoting and encouraging further engagement between the respective communities. For this purpose, the parties may seek to encourage and develop collaborative activities in various ways, including the exchange of ideas and expertise.

In practice, it is clear that keeping each organization abreast of developments of joint interest is challenging, given the quite different governance processes of ETSI NFV ISG and OPNFV. As a matter of fact, the development of many requirements projects in OPNFV overlaps with ETSI NFV. One of the important goals of the ETSI NFV phase 2 work program is to provide functional specifications for the key interfaces of the NFV architectural framework. OPNFV could be considered as providing de-facto 'specifications' for the NFV Infrastructure and its manager (VIM) while more conventional standards are likely to be developed by ETSI NFV and/or other SDOs for other NFV management and orchestration interfaces. However, it is likely that OPNFV will develop the "de-facto" 'specifications' for the NFV Infrastructure and its manager from the *deliverables of its own requirements projects* rather than from ETSI NFV specifications. Moreover, some recently agreed OPNFV requirements projects are addressing other management and orchestration interfaces as well, thereby increasing the risk of seeing a fragmentation of the requirements specifications in the industry.

C.3 Open Source and Standards: OpenStack

C.3.1 Introduction

Open Source Organizations develop their projects with their own dynamics. Their relationship with Standards Setting Organizations may or may not exist (depending on the scenario). The following clause investigates the case of OpenStack as an example of Scenario 5 ("An OSS organization implements Reference APIs").

C.3.2 The actors

OpenStack

OpenStack (http://www.openstack.org) is an Open Source Cloud Computing software that provides infrastructure as a service (IaaS) Cloud deployment for public and private Clouds. OpenStack was first introduced in June 2010, born with its initial code from NASA's Nebula platform and Rackspace's Cloud Files platform. Today, the OpenStack project is managed by the OpenStack Foundation established in September 2012. The OpenStack Foundation is an independent body providing shared resources to help achieve OpenStack mission by empowering, protecting and promoting OpenStack software and the community around it including users, developers and the entire ecosystem. OpenStack community with more than 500 supporting companies is having around six-month release cycle and till now OpenStack has released eleven major OpenStack releases from Austin in October 2010 to Kilo in April 2015.

OpenStack is organized around three main modules i.e. compute, storage and networking. Along with these three, dashboard is another important component providing interface to administrators and users for provisioning and release of resources. These components interact with user's application and underlying hardware over which other OpenStack services run. OpenStack Compute (Nova) has an abstraction layer for compute drivers. This is what allows choosing which hypervisor(s) to use for any OpenStack compute deployment.

OpenStack provides an IaaS solution through a set of interrelated services. Each service offers an Application Programming Interface (API) (see [i.11]) that facilitates this integration. These RESTful APIs and OpenStack currently consist of several different service code projects to make it modular, each having its different code name for the project. This code name describes the different modules of OpenStack and their configuration files respectively. Each OpenStack project also provides a command-line client, which enables accessing the project API through easy-to-use commands (see http://docs.openstack.org/cli-reference/content/).

Table C.1 describes the OpenStack services that provide the OpenStack architecture.

Table C.1: OpenStack services in support of OpenStack architecture

Service	Code name of project	Description
Dashboard	Horizon	Provides a web-based self-service portal to interact with underlying OpenStack services.
Compute	Nova	Manages the lifecycle of compute instances.
Networking	Neutron	Enables Network-Connectivity-as-a-Service.
Object Storage	Swift	Stores and retrieves arbitrary unstructured data objects via a RESTful, HTTP based API.
Block Storage	Cinder	Provides persistent block storage to running instances.
Shared services		
Identity service	Keystone	Provides an authentication and authorization service for other OpenStack services.
Image service	Glance	Stores and retrieves virtual machine disk images.
Telemetry	Ceilometer	Monitors and meters the OpenStack cloud for billing, benchmarking, scalability, and statistical purposes.
Higher-level services		
Orchestration	Heat	Orchestrates multiple composite cloud applications by using templates.
Database service	Trove	Provides scalable and reliable Cloud Database-as-a-Service functionality for both relational and non-relational database engines.
Data processing service	Sahara	Provides capabilities to provision and scale Hadoop clusters in OpenStack.

Distributions

There are many ways to install and deploy OpenStack through software distributions. The OpenStack Marketplace includes a list of commercial software distributions powered by OpenStack. In addition to commercial offerings, OpenStack is also included with several non-commercial Linux® distributions.

C.3.3 Support of standards

The development of the OpenStack software through its thigh release cycles is a key element of the output of the OpenStack community. Amongst the decisions that have to be taken for the product, the choice of which Cloud Computing standard will be supported is important. More information regarding the support of Cloud Computing standards by OpenStack and other important Cloud Computing Open Source projects can be found in Annex B.

C.4 Distributed Management Task Force (DMTF)

C.4.1 DMTF Standards

DMTF (www.dmtf.org) develops standards that enable the management of diverse traditional and emerging technologies including Cloud Computing, virtualization, network and infrastructure. Regarding Cloud Computing, DMTF has produced several specifications including:

- DSP0243 [i.20], Open Virtualization format (OVF). OVF is a common packaging format to package and securely distribute virtual appliances. This enables portability of virtual appliances across multiple virtualization platforms and products. OVF is a packaging standard and not a runtime standard. An OVF package contains one or more image files, an .ovf XML metadata file that contains information about the virtual machine, and possibly other files as well. OVF does not dictate any particular disk format (e.g. VHD, VMDK, VDI, QCOW2, etc.) to be used. An OVF package can be distributed in different manners. For example, it can be distributed as a set of discrete files, or as a tar archive file with an .ova (open virtual appliance/application) extension.

- DSP0263 [i.22], Cloud Infrastructure Management Interface (CIMI). CIMI is a self-service IaaS management interface, allowing Cloud customers to dynamically provision, configure and administer their Cloud usage using a high level interface that abstracts away much of the complexity of systems management. The interface uses the Hyper Text Transfer Protocol (HTTP) to send and receive messages that are formatted using either Java Script Object Notation (JSON) or the eXtensible Markup Language (XML).

- DSP0262 [i.21], Cloud Audit Data Federation (CADF). The Cloud Audit Data Federation specification defines a normative event data model along with a compatible set of interfaces for federating events, logs and reports between Cloud providers and Cloud customers. More than a format, the CADF standard defines a full event model anyone can use to fill in the essential data needed to certify, self-manage and self-audit application security in Cloud environments.

OVF and CIMI are adopted as International Standards, respectively ISO/IEC 17203 [i.18] and ISO/IEC 19831 [i.19].

C.4.2 DMTF standards and OpenStack

The objective of this clause is to look at how the OVF, CIMI and CADF standards developed by DMTF have been adopted in major Open Source projects, i.e. OpenStack and CloudStack. Note that DMTF has recently entered into an Alliance Partner relationship with the OpenStack Foundation (http://www.dmtf.org/sites/default/files/OpenStack-DMTF-WR-1_1.pdf). Both DMTF and OpenStack are committed to cross-body collaboration, integrating existing standards to enhance interoperability for the good of the industry. This relationship initially focuses on standards critical to Cloud security, improving cloud auditability to accelerate enterprise adoption.

OVF: The OpenStack Image Service provides discovery, registration and delivery services for disk and server images. When adding an image to OpenStack Glance, the virtual machine image's disk format and container format need to be specified. The disk format of a virtual machine image is the format of the underlying disk image. The container format refers to whether the virtual machine image is in a file format that also contains metadata about the actual virtual machine. Both OVF and OVA can be specified as values for the container format.

CIMI: OpenStack does not support the DMTF CIMI specification. CIMI on OpenStack Nova project in Github (https://github.com/osaddon/cimi) was started with the goal of adding the support of CIMI to OpenStack. However this has seen no activity since 2012 and would need to be updated to the latest version of OpenStack. Apart from OpenStack, CIMI has had several open source projects implementing parts of it such as DeltaCloud and OW2 Sirocco projects providing a proxy system with CIMI as the top API and support for multiple backend-clouds.

CADF: CADF is currently implemented in pyCADF (https://github.com/openstack/pycadf): A Python-based CADF Library, used by OpenStack. DMTF DSP2038 defines a CADF representation for use with the OpenStack Cloud Management Platform.

References

- DMTF DSP0243 [i.20]: Open Virtualization Format Specification.
- DMTF DSP0262 [i.21]: Cloud Auditing Data Federation (CADF) - Data Format and Interface Definitions Specification.
- DMTF DSP0263 [i.22]: Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol.
- DMTF DSP2038 [i.23]: Cloud Audit Data Federation - OpenStack Profile (CADF-OpenStack).

Annex D:
Change History

Date	Version	Information about changes
July 2015	1.0.0	First publication of the SR for comments
October 2015	2.0.0	Final publication based on the changes provided by: - Comments from the NTECH Technical Committee review - Comments from the public review gathered on http://csc.etsi.org - Additional changes proposed during the final review workshop

History

Document history		
V2.1.1	February 2016	Publication