

# Open Source Software reference implementations for standards issued by different standards setting organisations: availability, perceptions and practices

Jonas Gamalielsson<sup>1,\*</sup>, Björn Lundell<sup>1</sup>, Christoffer Brax<sup>2</sup>, Tomas Persson<sup>3</sup>, Anders Mattsson<sup>4</sup>, Tomas Gustavsson<sup>5</sup>, Jonas Feist<sup>6</sup>

<sup>1</sup>University of Skövde

<sup>2</sup>Combitech AB

<sup>3</sup>Digitalist Sweden AB

<sup>4</sup>Husqvarna AB

<sup>5</sup>PrimeKey Solutions AB

<sup>6</sup>RedBridge AB

**Abstract:** Software reference implementations of ICT standards have an important role for verifying that a standard is implementable, supporting interoperability testing among other implementations, and providing feedback to the standard development process. Providing reference implementations and widely used implementations of a standard as Open Source Software promotes wide deployment in software systems, interoperability, longevity of systems and associated digital assets, and avoidance of different lock-in effects. In this paper results are reported on the availability of, and perceptions and practices concerning, reference implementations and widely deployed implementations provided as Open Source Software for standards issued by different standards setting organisations. Specifically, findings draw from observations and analyses related to software implementations for identified standards and policy statements, issued by ETSI, IEC, IEEE, IETF, ISO, ITU-T, OASIS, and W3C.

**Keywords:** ICT standards, Reference implementations, Open Source Software, Standards setting organisations

## Highlights:

1. Widely used and deployed Open Source Software implementations (of which a minority are referred to as reference implementations) are available for all standards identified for all the investigated standards setting organisations.
2. Implementations are maintained by a variety of organisations of different types and provided under a variety of copyleft- and permissive Open Source Software licences.
3. Standards setting organisations have to varying extent expressed perceptions and developed practices concerning collaboration with and contributions to Open Source Software communities and projects developing reference implementations for standards.
4. Different contexts of standardisation can learn from and utilise new and specific experiences related to the role and benefit of software implementations in order to improve standardisation and its associated processes

\* Corresponding author:  
jonas.gamalielsson@his.se

Submitted: 1-9-2023

Revised: 18-12-2023

Accepted: 20-2-2024

Published: 28-2-2024

Peer review: double blind

This work is licensed under a [Creative Commons Attribution 4.0 International \(CC BY 4.0\) licence](https://creativecommons.org/licenses/by/4.0/)

DOI: 10.59490/jos.2024.7140

ISSN: 2772-9249

How to cite: Gamalielsson, J. et al., Open Source Software reference implementations for standards issued by different standards setting organisations: availability, perceptions and practices, Journal of Standardisation, Vol. 3 (2), 2024, DOI: 10.59490/jos.2024.7140

©2024 the authors published by TU Delft OPEN Publishing on behalf of the authors.



## 1 Introduction

Standards are issued by a variety of standards setting organisations (SSOs) (de Vries, 2006). Many of these standards are widely adopted and implemented by software projects. This paper focuses on implementations of standards in software projects. The importance of reference implementations of ICT standards in software has been highlighted in different contexts (e.g. Blind and Boehm, 2019; Egyedi, 2007; Ghosh 2005). Further, reference implementations and widely deployed implementations of standards provided as Open Source Software<sup>1</sup> (OSS) can create conditions for interoperable software systems, avoidance of different types of lock-in effects, and promotion of system longevity (Lundell, 2012; Lundell and Gamalielsson, 2017). OSS licensed reference implementations of standards are provided under different terms and “OSS licenses are often categorized along a copyleft dimension (i.e., licenses that have a copyleft effect and those that lack such an effect, often referred to as permissive licenses)” (Lundell et al., 2022). Copyleft licences ensure that derivative work remains OSS, whereas permissive licences do not (Brock, 2013).

The importance of standards and associated implementations in software projects has been highlighted in research that addresses different aspects related to interoperability and compliance (e.g. Egyedi, 2007; Friedrich, 2011), and aspects that relate to licensing conditions for standards and associated implementations in OSS (e.g. Friedrich, 2011; Simcoe, 2006). The relationship between ICT standardisation and implementation of standards in OSS is a topic which has been addressed over a number of years in earlier studies (e.g. Blind and Boehm, 2019; Boehm and Eisape, 2021). Related to this relationship there are, for example, studies which investigate organisational influences in and between the W3C RDFa standard and its implementation in the Drupal OSS project (Gamalielsson et al., 2015), issues and experiences concerning implementation of the PDF format in OSS (Gamalielsson and Lundell, 2013), organisational involvement and collaboration in W3C standards through editorship (Gamalielsson and Lundell, 2017a), technical and licensing challenges for organisations planning software development covering IT standards (Lundell et al., 2019), issues concerning the relationship between OSS and IPR aspects including Standard Essential Patents (e.g. Blind et al., 2011; Bekkers and Updegrove, 2013; Contreras et al., 2023; Husovec, 2019; Maracke, 2019), and implementation of ISO standards in OSS (Lundell et al., 2015). Other studies analysed influences between ICT standards and their implementations in OSS projects with a specific focus on the ITU-T H.264 & H.265 multimedia standards (Gamalielsson and Lundell,

---

<sup>1</sup> Open Source Software (OSS) is software provided under terms that comply with the Open Source Definition (OSD, 2023). Widely used OSS is typically provided under a small set of OSS licences. For example, previous research that investigated 200 software projects shows that 60% provide OSS under a licence in the GPL-family of licences (Gamalielsson & Lundell, 2017b). Modern OSS licences, such as the Apache-2.0 licence (e.g. used by the Apache HTTP Server project), the EPL-2.0 license (e.g. used by the Eclipse IDE for Java), the MPL-2.0 licence (e.g. used by the Firefox web browser project), and licences in version 3 of the GPL-family of licences (e.g. used by the Nextcloud project), have “an explicit patent clause” (Lundell et al., 2022). OSS projects that use such modern OSS licences for providing OSS rely on a “classical patent grant” (Lindberg, 2019) to ensure rights for use, modification, and redistribution of OSS. Other OSS licences rely on other types of grants, such as “an express grant” (e.g. Lindberg, 2019) Moreover, version 2 of the GPL “includes the so called Liberty or Death clause, which prohibits (also) patent holders from distributing code under the GPLv2 and simultaneously claiming patent royalties from the licensees” (Haapanen, 2015, p. 20).



2021a, 2021b; Gamalielsson et al., 2021; Lundell et al., 2023) and the WebRTC standard (Gamalielsson et al., 2022a).

Even though standards and associated implementations in software projects have been highlighted in several studies, there is an observed lack of investigations that focus on specific reference implementations of standards, and that in particular address specific reference implementations provided as OSS for standards issued in the context of different SSOs. We argue that this research focus can shed light, through presentation of concrete examples, on how the relationship between ICT standardisation and implementation of standards in OSS can manifest itself. This is of relevance especially in light of an expected future scenario of standard development that can be referred to as “open hybridization” which has been described as “the well-considered marriage of open source software and open standards” (Updegrove, 2017b). In fact, it has been claimed that this “scenario of a hybrid of standardisation and OSS will emerge in the near future in addition to the three scenarios introduced by Lundell and Gamalielsson (2017)” (Blind and Boehm, 2019). To achieve this, in the view of Updegrove (2017a), it has been argued that a challenge to overcome is that OSS projects often “ignore the need for standards” and that “the prevailing opinion in the open source community is that standards are limiting, irrelevant and unnecessary”. In light of these observations, this study addresses the following two research questions: **RQ1**: “What characterises availability of OSS reference implementations for standards issued by different standards setting organisations?”<sup>2</sup>; **RQ2**: “What characterises perceptions and practices concerning OSS and OSS reference implementations in different standardisation contexts?” This paper extends results earlier reported on the availability of reference implementations in different standardisation contexts (Gamalielsson et al., 2023).

## 2 Background and previous research

Standards are provided under a range of different terms (Bekkers and Updegrove, 2013). Several policies promote provision of standards under terms which are recognised as open standards (DeNardis, 2010), and it has been argued that such standards promote interoperability and open up markets by counteracting monopolies based on proprietary technologies (Aliprandi, 2011; DeNardis, 2010; Ghosh, 2005; Lundell, 2012). Further, for a long time several European countries have provided guidance and policies concerning use of ICT standards, including the Netherlands (NOC, 2007), Portugal (Ballard, 2012), the UK (2012), and Sweden (NPS, 2016). Such approaches have been reinforced by a number of international initiatives. For example, in the Swedish context, guidance for open ICT standards is based on the European Interoperability Framework (EIF) version 1.0 (EC, 2004).

Previous research has identified several motivations for use of open standards (e.g. DeNardis, 2010; Ghosh, 2005; Krechmer, 2005; Simcoe, 2006; Lundell and Lings, 2010). For example, open standards promote interoperability and a healthy competitive market according to several studies (Ghosh, 2005; Krechmer, 2005; Simcoe, 2006; Lundell and Lings, 2010). Moreover, the existence of an open standard reduces the risk and cost of market entry, and so encourages the possible choice of multiple suppliers (DeNardis, 2010; Ghosh, 2005). Further, insistence on

---

<sup>2</sup> This includes whether and how OSS reference implementations for standards issued by different standards setting organisations are made available.



open standards reduces the risk to an organisation of being subjected to different types of lock-in effects. Open standards increase control by supporting migration, and thereby reduce an organisation's reliance on a single product or supplier (Ghosh, 2005; Krechmer, 2005; Berkman, 2005; Bhattacharya et al., 2007; Simon, 2005; West and Dedrick, 2001; Lundell, 2011). A significant current issue for many organisations is that they are restricted in their choice of software because of restrictions imposed by existing or legacy systems. Interoperability supports systems heterogeneity, thereby increasing options for organisations (Bird, 1998; Ghosh, 2005; Krechmer, 2005; Fomin et al., 2008). Further, open standards offer a basis for long-term access and archiving of digital assets, and in particular when supported by OSS reference implementations (Behlendorf, 2009). Long-term availability of software projects which provide faithful implementations of a technical specification of a digital format, for example through a reference implementation of the format provided by a software project which is recognised by an SSO, may provide very valuable feedback and clarity to the standardisation process and also to each software development project which seeks to develop a faithful software implementation of the specific format.

A reference implementation has been defined as “the implementation of a standard to be used as a definitive interpretation for the requirements in that standard”, and that reference implementations “can be used to verify that the standard is implementable, validate conformance test tools, and support interoperability testing among other implementations.” (NIST, 2023). A reference implementation may be used as an “approach for developing standards” where this implementation is defined to be correct and “all other implementations are correct if and only if they work against the reference implementation.” (Blind and Boehm, 2019).

Further, it has been claimed that “for most software standards the formal specification is insufficient and the actual standard may differ from across implementations” which “suggests the need for a reference implementation to augment – if not, perhaps, replace – the formal specification of the standard” (Ghosh, 2005). A reference implementation provided as OSS “may achieve the economic effect of an open standard” and “may act as the formal specification (especially if sufficiently well documented) and be reproduced without economic restrictions by any potential vendor of the technology” (Ghosh, 2005). An example of an OSS implementation that acts as the (*de facto* rather than formal) specification is the *Perl 5* programming language (Michaud, 2015).

The issue of ambiguity of natural language in a specification has also been raised, and that the translation of “international standards into other languages often discloses ambiguities” (Egyedi, 2007). Relatedly, for the PDF document format it was found that “there are several problematic issues related to clarity and detail in the specification” and that these issues in many cases make “the implementation of the specification unnecessarily challenging and complex” and increase “the risk that different implementations of the specification of the PDF format deviate and ultimately lead to problems related to interoperability” (Gamalielsson and Lundell, 2013). To mitigate the issue of ambiguity, research shows that use of “reference implementations helps implementers to resolve ambiguities” (Egyedi, 2007).

Further, expectations concerning availability of more than a single software implementation for a specific standard has also been expressed. One example is in the IETF internet standards process, where a draft standard is described as a “specification from which at least two



independent and interoperable implementations from different code bases have been developed” (IETF, 1996). Another example is the W3C PNG Working Group Charter, which states: “In order to advance to Proposed Recommendation, each normative specification is expected to have at least two independent implementations of every feature defined in the specification” (W3C, 2023a).

Even though there seem to be apparent benefits of reference implementations, software bugs have been raised as an issue that may affect interoperability: “Finding and fixing a bug in the reference implementation essentially changes the standard” and then the “relevance of standards’ implementation is challenged by a possible lack of interoperability” (Blind and Boehm, 2019).

Another issue is the implementation in software of standards that are provided under FRAND (Fair, Reasonable, and Non-Discriminatory) terms (Lundell et al., 2023). Legal experts have claimed that commonly used FRAND licences are incompatible with OSS licensing due to the inability of the licensee to sub-licence to recipients that are downstream (EC, 2012). Further, the European Commission has acknowledged that such licensing terms for standards “create barriers for Open Source projects to implement the technical specification” (EC, 2013). Research has also shown that “it may be impossible to clarify conditions and obtain patent licences for standard essential patents (and all necessary rights) for use of specific ISO standards that are provided on FRAND-terms (Lundell et al., 2015).” (Lundell et al., 2019). We acknowledge that a legal analysis is beyond the scope of the study reported in this paper.

### 3 Research approach

To *address the first research question* (What characterises availability of OSS reference implementations for standards issued by different standards setting organisations?), a survey involving a systematic web search<sup>3</sup> was conducted with the goal to identify three ICT standards issued by each of eight purposely selected SSOs (ETSI, IEC, IEEE, IETF, ISO, ITU-T, OASIS, and W3C) for which there is at least one associated (preferably widely used and deployed) OSS implementation for which there are claims that it is a reference implementation of the standard (or at least claims that it supports the specific standard). Different combinations of keywords<sup>4</sup> were used to identify relevant web pages. The search was supplemented by information obtained at consortiuminfo.org (2023) which maintains links to standards issued by each of the SSOs. Further, additional analyses of public web pages and documents related to identified standards and associated OSS implementations were performed in order to address the research question. The data collection was made in February 2023. Results in section 4 (presented in Table 1 in Appendix) refer to “latest commit” and “latest release” at the point of data collection for the identified OSS implementations. Licensing conditions for implementations are referred to using the SPDX format<sup>5</sup>.

---

<sup>3</sup> through use of a search engine provided by Google (<https://www.google.se/>)

<sup>4</sup> including, but not limited to, the name of each SSO, “standard”, “specification”, “reference implementation”, “reference software”, “definitive implementation”, and “open source software”

<sup>5</sup> <https://spdx.org/licenses/>



The criteria for purposeful selection of SSOs are presented in the following. The chosen SSOs all provide standards in the ICT domain and comprise a selection that incorporates different categories of organisations. We acknowledge that “there is no universally acknowledged taxonomy for distinguishing one type of standards development organization from another” (Bekkers and Updegrove, 2013), and we therefore had the goal to include well-known SSOs. All chosen SSOs are international organisations except one<sup>6</sup>. Four of the eight organisations are, in accordance with de Vries (2006), formal SSOs (ETSI, IEC, ISO, and ITU-T). The remaining organisations (IEEE, IETF, OASIS, and W3C) are, according to the same source, not formal organisations and constitute four of the “main Internet Standardization Organizations” (Wenning and Bos, 2007). The goal to identify *three* ICT standards was set to be able to show that *several* standards with associated OSS implementations are available for each SSO.

To *address the second research question* (What characterises perceptions and practices concerning OSS and OSS reference implementations in different standardisation contexts?), a survey involving a systematic web search<sup>7</sup> was conducted with the goal to identify perceptions and practices for each of the eight purposely selected SSOs. Different combinations of keywords<sup>8</sup> were used to identify relevant web pages. The data collection was made in August 2023.

## 4 On availability of OSS reference implementations in different standardisation contexts

In relation to the first research question, this section accounts for an identified selection of standards issued by the selected SSOs and associated widely used and deployed implementations in OSS (which in some cases are referred to as reference implementations) for each of these standards.

Table 1 (in Appendix) provides an overview of the identified 24 standards and associated implementations with respect to maintainer, licence, latest release, and latest commit.

### 4.1 European Telecommunications Standards Institute (ETSI)

**ETSI** standardises a wide variety of standards across various industries and fields, including telecommunications and information technology (ETSI, 2023a). In this standardisation context there are several standards which have OSS implementations that claim to implement those standards. One example is *Network Functions Virtualisation (NFV)* which “decouples software implementations of” network functions “from the physical resources offered by the hardware they use, such as computation, storage, and networking” (ETSI, 2021). For this standard there is a reference implementation *Open Source MANO* that “is an ETSI-hosted project to develop an Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV” (ETSI, 2023b). It has also been stated that “OpenMANO is the

---

<sup>6</sup> ETSI, which is a regional SSO (de Vries, 2006), but which clearly has international impact.

<sup>7</sup> through use of a search engine provided by Google (<https://www.google.se/>)

<sup>8</sup> including, but not limited to, the name and web site domain of each SSO, “reference implementation”, “reference software”, “definitive implementation”, “open source” and “open source software”



reference implementation of an NFV-O (Network Functions Virtualisation Orchestrator)” (Rao, 2016).

A second example of standard is within the context of Internet of Things (IoT) and specifically the ETSI standards developed in the context of the “One machine-to-machine partnership project” (*oneM2M*) which is a “global standards initiative that covers requirements, architecture, API specifications, security solutions and interoperability for Machine-to-Machine and IoT technologies” (ETSI, 2023c). The *Eclipse OM2M architecture* claims to adhere to the ETSI M2M standards (eclipse.org, 2023a).

A third example is the Digital Video Broadcasting - Terrestrial (*DVB-T*) standard which specifies “frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system” (ETSI, 2015). *VLC Media Player* (videolan.org, 2023a) is a widely used multimedia player provided as OSS that claims to implement the DVB-T standard (videolan.org, 2023b).

## 4.2 International Electrotechnical Commission (IEC)

**IEC** is engaged in the standardisation of electrical and electronic technologies (IEC, 2023a). The IEC does not maintain open-source reference implementations, but there are standards that have associated open-source implementations with claimed standard support that are developed by other organisations or communities. One example is *IEC 61131-3* (IEC, 2013), which is a standard for programming industrial automation systems. One widely used OSS implementation that claims to be compliant with this standard is *OpenPLC* (openplcproject.com, 2023).

A second example of standard is *IEC 61850* (IEC, 2023b), which defines the communication protocols and data models used in power system automation. For this standard there is the OSS implementation *IEC61850bean* (beanit.com, 2023), formerly known as OpenIEC61850, which is a java implementation with claimed support for the standard. The project was “started by Fraunhofer ISE, energy & meteo Systems GmbH, and OFFIS as part of the eTelligenz research project funded by Germany’s Federal Ministry of Economics and Technology” (openmuc.org, 2023).

A third example of standard is *IEC 61499* (IEC, 2012), which defines the function block architecture for distributed control systems. A widely deployed implementation of the standard in OSS is *Eclipse 4diac*, which “provides an open source infrastructure for distributed industrial process measurement and control systems based on the IEC 61499 standard” (eclipse.org, 2023b).

## 4.3 Institute of Electrical and Electronics Engineers (IEEE)

**IEEE** develops a wide range of standards across different industries and fields, including electrical engineering, computer science, and telecommunications (IEEE, 2023a). In this context of standardisation there are a number of standards which have OSS implementations that are claimed to implement those standards. One example is *IEEE 802.3* (IEEE, 2022a), which specifies ethernet local area network operation. For this standard there is a widely deployed implementation in the *Linux kernel* (kernel.org, 2023a) which in particular claims to



implement different physical interface modes and protocols related to the IEEE 802.3 suite of standards (kernel.org, 2023b).

Relatedly, a second example of a standard is **IEEE 802.11** (IEEE, 2020), which specifies wireless local area networks. For this standard there are several implementations provided as OSS, of which one is *hostapd* which “is a user space daemon for access point and authentication servers” that “implements IEEE 802.11 access point management” (w1.fi, 2023).

A third example is the **IEEE 1588** (IEEE, 2019), which specifies a precision clock synchronization protocol for networked measurement and control systems. OSS and Open Source Hardware (OSH) developed in the *White Rabbit project* (CERN, 2023) had a key role during standardisation of the latest edition of the standard (IEEE 1588-2019), where White Rabbit software and technologies allowed for vastly improved accuracy and precision. White Rabbit at a top level is maintained by CERN (a research organisation based in Switzerland) and its research has assessed its ecosystem to be “diversified and long-term sustainable” (Gamalielsson et al., 2022b). The software stack in the White Rabbit project that specifically implements IEEE 1588-2019 is called *ppsi* (ohwr.org, 2023).

#### 4.4 Internet Engineering Task Force (IETF)

**IETF** standardises a wide range of standards related to the Internet and web technologies (IETF, 2023). In this standardisation context there are several standards which have OSS implementations that are claimed to be reference implementations of those standards. One example is the standard **RFC 6101** (Secure Sockets Layer, SSL), which is “a security protocol that provides communications privacy over the Internet” (IETF, 2011a). The *OpenSSL* project (openssl.org, 2023a) provides an implementation of the standard in OSS “and is often used as the reference implementation for any new feature” (openssl.org, 2023b).

A second example of a standard is **RFC 7540** (Hypertext Transfer Protocol, HTTP/2), which specifies and “describes an optimized expression of the semantics of the Hypertext Transfer Protocol” (IETF, 2015). One widely deployed OSS-implementation that supports HTTP/2, but not yet version HTTP/3 (IETF, 2022), of the standard is the *Apache HTTP Server Project* (apache.org, 2023a) which “exists to provide a robust and commercial-grade reference implementation of the HTTP protocol” (apache.org, 2023b).

A third example of standard is **RFC 1035** (Domain names - implementation and specification), which specifies and “describes the details of the domain system and protocol” (IETF, 1987). A widely deployed implementation in OSS of this standard is *BIND 9* (isc.org, 2023). In the documentation for the software it is claimed that “BIND 9 strives for strict compliance with IETF standards. To the best of our knowledge, BIND 9 complies with” RFC 1035 amongst other protocol specifications (readthedocs.io, 2023). It has also been claimed that BIND 9 “is an implementation of the Domain Name System (DNS) protocols and provides an openly redistributable reference implementation of the major components of the Domain Name System” (nlnet.nl, 2023).

#### 4.5 International Organization for Standardization (ISO)

**ISO** standardises and adopts a wide range of standards in different fields (ISO, 2023a). In this standardisation context there are several standards which have OSS implementations claiming





to implement those standards. One example is *ISO/IEC 26300-1* (Open Document Format for Office Applications, ODF) (ISO, 2015), which is an XML-based file format for office documents (e.g. text documents, spreadsheets, and presentations), which was originally developed as an OASIS standard (OASIS, 2011). *LibreOffice* (libreoffice.org, 2023) is a widely used OSS implementation of the ODF standard that “saves documents in OpenDocument Format by default” (libreoffice.org, 2016) and claims support for ODF version 1.2.

A second example of a standard is *ISO/IEC 14882* (Programming Languages — C++), which “specifies requirements for implementations of the C++ programming language” (ISO, 2020). The *GNU Compiler Collection (GCC)* (gnu.org, 2023a) is an OSS implementation of the standard that “supports different dialects of C++, corresponding to the multiple published ISO standards” and it “has full support for the 2014 C++ standard” and earlier versions, and almost full support or experimental support for later versions of the standard (gnu.org, 2023b).

A third example of a standard is *ISO/IEC 13818-3* (Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio), which specifies MPEG-2 Audio Layer III, also known as MP3 (ISO, 1998). The actual development of the standard was performed by contributors at Fraunhofer Society in Germany. The *LAME* project provides a widely deployed OSS implementation of the standard that was initially based on the original demonstration source code (referred to as *dist10*) developed at Fraunhofer, and the goal for LAME was initially “only to speed up the *dist10* sources, and leave its quality untouched.” (sourceforge.io, 2023)

#### 4.6 International Telecommunication Union Telecommunication Standardization Sector (ITU-T)

ITU-T is responsible for developing and publishing standards in the telecommunications domain for global use (ITU-T, 2023). In this context of standardisation there are several standards for which ITU-T itself provides reference software and for these standards there are also external OSS implementations claiming to comply with those standards. One example is *ITU-T H.264* (Advanced video coding for generic audiovisual services), which is a widely used format for recording, compression, and distribution of video content (ITU-T, 2021a). ITU-T provides reference software for the standard (ITU-T, 2016a), however this software is not provided under an OSS licence. One widely deployed OSS implementation of the standard is the *x264* project, which “is a free software library and application for encoding video streams into the H.264/MPEG-4 AVC compression format” (videolan.org, 2023c). Further, the ITU-T Telecommunication Standardization Advisory Group (TSAG) has in a proposal exclusively listed *x264* as an external OSS project that is related to the ITU-T reference implementation of H-264 (ITU-T, 2017).

A second example of standard is *ITU-T H.265* (High efficiency video coding), which is a format for video coding that has evolved from the earlier ITU-T standards (including H.264) in response to a need for higher compression of video content (ITU-T, 2021b). Like for the H.264 standard, ITU-T provides reference software for the H.265 standard (ITU-T, 2016b), but this software is not provided under an OSS licence (Lundell et al., 2023). A widely deployed implementation in OSS of the H.265 standard is the *x265* project, which “is a free software library and application for encoding video streams into the H.265/MPEG-H HEVC compression format” (x265.org, 2023). Like for the H.264 standard the TSAG has exclusively



listed x265 as an external OSS project that is related to the ITU-T reference implementation of H-265 (ITU-T, 2017).

A third example of a standard is **ITU-T G.722** (7 kHz audio-coding within 64 kbit/s), which specifies and “describes the characteristics of an audio wideband (WB, 50 to 7 000 Hz) coding system which may be used for a variety of higher quality speech applications” (ITU-T, 2012). It has been stated that reference software provided by ITU-T “for the algorithm in the main body of ITU-T G.722 is found in the ITU-T G722 module of the ITU-T G.191 Software Tools Library” (ITU-T, 2012), however this software is not provided as OSS. A widely used and deployed OSS implementation of the standard is *FFmpeg* (ffmpeg.org, 2023a). It has been claimed that the G.722 ADPCM audio codec in FFmpeg is “a bit-exact implementation of the ITU G.722 specification for all three specified bitrates - 64000bps, 56000bps and 48000bps” and that “it passes the ITU tests” (ffmpeg.org, 2023b).

#### 4.7 Organization for the Advancement of Structured Information Standards (OASIS)

**OASIS** develops and maintains standards within “cybersecurity, blockchain, IoT, emergency management, cloud computing, legal data exchange, and much more” (OASIS, 2023a). In this standardisation context there are several standards which have OSS implementations claiming to implement those standards. One example is **CMIS** (Content Management Interoperability Services), which specifies and “defines a domain model and Web Services, Restful AtomPub and browser (JSON) bindings that can be used by applications to work with one or more Content Management repositories/systems” (OASIS, 2015). A widely used and deployed OSS project that implements the standard is the community edition of *Alfresco* (alfresco.com, 2023a), which is an information management system where its content services “fully implement both the CMIS 1.0 and 1.1 standards to allow your application to manage content and metadata in a repository.” (alfresco.com, 2023b).

A second example is the standard for **OData** (Open Data Protocol), which “enables the creation and consumption of REST-based data services which allow resources, identified using Uniform Resource Locators (URLs) and defined in a data model, to be published and edited by Web clients using simple HTTP messages” (OASIS, 2020). A widely used and deployed OSS project that implements the standard is *Apache Olingo* that provides “a Java library that implements the Open Data Protocol (OData)” (apache.org, 2023c). Further, in September 2015 the project announced the release of “Apache Olingo 4.0 with support for building OData V4-compliant RESTful services and clients.” (Pizzo, 2015).

A third example of a standard is **SAML** (Security Assertion Markup Language), which specifies and “defines the syntax and processing semantics of assertions made about a subject by a system entity” (OASIS, 2005). A widely used and deployed OSS project that implements SAML is *OpenSAML*, which “is a set of open source Java libraries used in support of the Shibboleth Project’s implementation of the Security Assertion Markup Language (SAML)” as specified by OASIS (atlassian.net, 2023). Further, it has been claimed that “OpenSAML 4, the current Java library version, is based on Java 11, and supports SAML 1.0, 1.1, and 2.0.” (atlassian.net, 2023).



#### 4.8 World Wide Web Consortium (W3C)

**W3C** is the primary international organisation that develops and maintains standards for the World Wide Web (W3C, 2023b). In this context of standardisation there are several standards which have OSS implementations for which there are claims that they implement those standards. One example is *XML* (Extensible Markup Language), which specifies and “describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them” (W3C, 2006). A widely used OSS implementation of the standard is the *Apache Xerces project* (apache.org, 2023d). In this project, the Xerces C++ processor is claimed to be “faithful to the XML 1.0 and 1.1 recommendations and many associated standards” and the Xerces java processor is claimed to provide support for XML 1.0 (4<sup>th</sup> Edition) and XML 1.1 (2<sup>nd</sup> Edition) (apache.org, 2023d).

A second example of standard is *SVG* (Scalable Vector Graphics), which specifies and “defines the features and syntax for Scalable Vector Graphics (SVG) Version 2” where “SVG is a language based on XML for describing two-dimensional vector and mixed vector/raster graphics” (W3C, 2018). A widely used and deployed OSS project that implements this standard is *Inkscape* (inkscape.org, 2023a), which “prides itself on being a fully standard-compliant SVG editor”. As of March 2015, the Inkscape project “even supports rendering of SVG properties which are not yet officially part of the standard, but have reached a stable state in the SVG Working Group’s draft for SVG 2.0”. Further, the project “also takes part in the further development and refinement of SVG features by delegating a representative to the W3C SVG Working Group.” (inkscape.org, 2023b).

A third example of standard is *PNG* (Portable Network Graphics), which specifies and “describes PNG (Portable Network Graphics), an extensible file format for the lossless, portable, well-compressed storage of static and animated raster images” (W3C, 2022a). A widely deployed software project that implements this standard is *libpng*<sup>9</sup>, which claims to be “the official PNG reference library” (libpng.org, 2023). The specification for PNG refers to libpng as a “sample implementation in portable C” and refers to a webpage (libpng.org, 2022) under the libpng domain as “a central location for current information about PNG”, which indicates a close relation between W3C standardisation of PNG and the libpng implementation of the standard.

### 5 On perceptions and practices concerning OSS and OSS reference implementations in different standardisation contexts

In relation to the second research question, this section accounts for identified perceptions and practices concerning OSS and OSS reference implementations as expressed by the selected SSOs.

---

<sup>9</sup> This software project has since 1998 provided many releases under permissive conditions using slightly adapted versions of the OSI-approved Zlib license (<https://spdx.org/licenses/Zlib.html>).



## 5.1 ETSI

The complementary roles of standards and OSS have been elaborated in the ETSI context: “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.” (ETSI, 2016) Further, the importance of standards to promote technology neutrality and avoidance of lock-in has been highlighted: “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” and “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.” (ETSI, 2016)

More than ten years ago, ETSI elaborated on the usage of OSS in standardisation processes and that “Open Source Software is a different approach, which raises the question of its use in conjunction with standards” (ETSI, 2012). Further, three relationships between ETSI standards and OSS were identified: (1) usage of OSS in the ETSI organisation, (2) adoption and usage of elements of OSS in the elaboration of ETSI standards, and (3) adoption of ETSI standards within OSS communities (ETSI, 2012). The main challenges concerning the second relationship is perceived to be to “work in a collaborative mode with the OSS community” and to “ensure that the OSS license applying to the material (for example interfaces) and to any derivative work is not going to prevent licensing of essential patents on FRAND terms or otherwise be incompatible with the ETSI IPR policy”. (ETSI, 2012)

Further, ETSI recognises the role of OSS reference implementations and that “it is necessary to engage the relevant industry and open-source forums to jointly proceed.” (ETSI, 2023d) It has also been claimed that “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.” (ETSI, 2016)

The use of OSS to promote interoperability in the context of ETSI Network Functions Virtualisation (NFV) has been highlighted: “Open Source software can facilitate the implementation of an ETSI aligned NFV architecture, provide practical and essential feedback to the ETSI ISG NFV and increase the likelihood of interoperability among NFV implementations.” (ETSI, 2023e) It has also been claimed that ETSI Open Source MANO “provides an opportunity to capitalize on the synergy between standardization and open source approaches by accessing a greater and more diverse set of contributors and developers than would normally be possible” and that “this approach maximizes innovation, efficiency and time to market and ensures a continuing series of true (conformant) reference implementations.” (ETSI, 2023e)

## 5.2 IEC

In the IEC context, the IEC Manager of International Standards Development at the Standards Council of Canada, who is also a member of the IEC Standardization Management Board, has highlighted the role and impact of OSS in standardisation in the following way: “The era of digital transformation will impact the nature of standards development, which is traditionally grounded in text-based documents. The shift towards digital content and virtual work has the potential to change text-based documents beyond recognition. The role of open-source software



in standards will expand and could alter the very definition of a standard. It may affect the nature and scope of IEC standardization. Some of our members will welcome and master this transition, and others will not.” (IEC, 2022) In order to explore the role of OSS, one of the tasks for IEC WG1 was to monitor trends and “analyse new business and development models (e.g. Open Source, DevOps) related to communication technologies and assess their impact on IEC activities.” (IEC, 2023c)

The Standardization Management Board (SMB) of IEC was concerned about how OSS can affect copyright and licensing on standardisation, and decided therefore to set up the ad hoc group *ahG 92* that “will investigate the consequences of new developments such as machine-readable standards and open source on IEC copyright policy as well as the distribution and licensing of code components and other digital artifacts. It will also identify opportunities and existing gaps within IEC committees and best practices from other organizations.” (IEC, 2023d) An expectation is that the group can provide support in “developing guidance on the current application of the IEC copyright policy, licensing, and processes for handling code components and other digital artifacts in standardization work.” (IEC, 2023d) The IEC has also perceived that OSS can lead to avoidance of formal standardisation: “There exists some risk that proponents of particular technologies may prefer that they be developed in an open source community and eschew formal standardization” and in order to “minimize this risk, SC 32’s WGs have been working more closely with open source communities.” (IEC, 2020a)

Relatedly, the possibility to have OSS (e.g. LibreOffice) for drafting standards (instead of the proprietary software MS Word) has been raised in the context of an IEC Academy webinar, even if the answer was that IEC at the time were unable to provide support for other software than MS Word for the purpose (IEC, 2020b).

### 5.3 IEEE

In the context of IEEE, it has been recognised that “standardization continues to evolve to a point where open source code development will increasingly play a complementary and sometimes a substitute for open standards development in the SDO sense.” (Parsons, 2017)

Further, to promote open source code development in IEEE scenarios, the established collaboration platform “IEEE SA OPEN is intended to bridge the gap between standards developers and other open technical communities to enable nimble and creative technical solutions”, and the platform aims to “address common challenges faced by the open-source development community, such as lack of relevant engagement in projects, solution incompatibility, and complexity around intellectual property (IP) licensing.” (IEEE, 2023b)

Recently, in the context of IEEE SA OPEN, the “Global Initiative on Digital Transformation of Pandemic Surveillance” was launched (IEEE, 2023c), and “the purpose of this activity is to initiate a global effort to enable the digital transformation of pandemic surveillance” (IEEE, 2022b). It is claimed that “This activity will fully embrace the concept of open-source and open-access, minimize or eliminate the paid-use of patents, and employ a non-profit business model for public good in order to serve the broadest diversity of mankind.” (IEEE, 2023c). One of the specific outcomes of the activity is “Open-source reference implementations for the new systems and applications needed in the digital transformation of Pandemic Surveillance.” (IEEE 2022b).



## 5.4 IETF

In the IETF context, the expectation that more than a single software implementation for a specific standard is available to promote interoperability was originally expressed in RFC 1310 (IETF, 1992) more than 30 years ago where it is stated that a ‘specification from which at least two independent and interoperable implementations have been developed, and for which adequate operational experience has been obtained, may be elevated to the “Draft Standard” level.’ Concerning the independence of implementations, it has also been stated that “independent implementations should be written by different people at different organizations using different code and protocol libraries” (IETF, 2009). The expectation of multiple software implementations of a standard is also expressed later in RFC 2026 (IETF, 1996), RFC 6410 (IETF, 2011b), and by Bradner (1999). The latter RFC also stresses the importance of “measuring interoperability through widespread deployment of multiple implementations from different code bases” (IETF, 2011b).

More specifically, the importance of “Open Code Availability” for software implementations of standards has also been elaborated in the IETF context: “Protocols with freely available implementation code have a greater chance of success than protocols without. Often, this is more important than any technical consideration” (IETF, 2008). Further, it has been claimed that the IETF community needs to support and promote a symbiotic relationship between Free and Open Source Software (FOSS<sup>10</sup>) and open standards: “There is strong agreement within the IETF that we as a community need to be a part of the change by helping FOSS and open standards work together, combining their respective strengths in a way that creates value for the entire network engineering community. Open source and open standards have a natural and symbiotic relationship, and instantiation of open standards in open source projects strengthens the standards and the community at large.” (IETF, 2016a) Further, the importance of “Hackathons” has been highlighted as a means to promote this relationship: “Work at IETF Hackathons has identified issues with standards still under development by IETF working groups, providing invaluable information about what worked and what didn’t work in real operating network environments.” (IETF, 2016b)

## 5.5 ISO

In the context of ISO, the chair of ISO/TC 68 has acknowledged the role of open source tools in the context of information security: “Primary drivers of the expanded use of PKI include the maturing of commercial certification authority (CA) products and open source tools, increased automation capabilities of commercial CA software products, and improved ability of applications to handle digital certificates.” (Lundin, 2007)

Further, ISO TC211 has recently acknowledged OSS and encouraged participation in an announcement for “the 2023 Open Standards and Open Source Software Code Sprint” which “will cover multiple open source projects and related OGC & ISO/TC 211 Standards” (ISO, 2023b)

Statements concerning reference implementations, but not necessarily provided as OSS, have been identified in a number of ISO/IEC standards specifying reference software for different formats in the multimedia field, e.g. ISO/IEC 15444-5:2021 (containing reference software for

---

<sup>10</sup> <https://www.gnu.org/philosophy/floss-and-foss.html>



the JPEG 2000 format) for which it is stated that “this Recommendation | International Standard provides three independently created software reference implementations of Rec. ITU-T T.800 | ISO/IEC 15444-1, in order to assist implementers of Rec. ITU-T T.800 | ISO/IEC 15444-1 in testing and understanding its content. The packages are JASPER, JJ2000 and OPENJPEG.” (ISO, 2021)

## 5.6 ITU-T

In the context of ITU-T, a resolution (“Resolution 90”) was released in 2016 on the role, engagement and use of OSS that resolves the Telecommunication Standardization Advisory Group (TSAG) “continue to work on the benefits and disadvantages of the implementation of open-source projects in relation with the work of the ITU Telecommunication Standardization Sector (ITU-T).” (ITU-T, 2016c) Further, applicable study groups in ITU-T are instructed to: “provide inputs to TSAG enquiries on open source”, “consider output from TSAG on open source, in order to study the value of using open source to develop reference implementations of ITU-T Recommendations, as appropriate”, “continue using open source as appropriate”, “to support the use of open-source projects in their work, as appropriate”, and “to continue engaging with open-source projects” (ITU-T, 2016c).

In the wake of “Resolution 90” (ITU-T, 2016c), several documents containing different suggestions and comments have been provided by ITU-T contributors. It has been suggested, in order to foster successful collaboration with the open source community, to “study successful examples of open source reference implementations of ITU-T Recommendations to identify what factors contributed to successful collaborations and develop a list of lessons learned”, to raise “awareness of ITU-T recommendations to open source communities so those communities can identify any standards that might lend themselves to open source software implementations”, to “review the relationship between the Open Source licensing types, licensing mechanisms, and the ITU’s own software copyright guidelines”, and that “the process already being followed to develop reference implementations be continued.” (ITU-T, 2017a) Further, it has been stated that “as the goal of ITU-T study groups is to develop interoperable standards and data exchange across varied products and services from multiple vendors, study group recommendations are made public and are open to parties interested in developing a reference implementation” and that “several study groups including SG11, SG12 and SG16 have developed reference implementations of ITU-T recommendations.” (ITU-T, 2017a)

It has also been claimed that in an OSS development process the “mechanisms, tools, and working methods are very different to the consensus approach used to develop an ITU-T recommendation” and that OSS working practices and tools “can be used to develop source code in a standard that is then privately held.” (ITU-T, 2017b). Further, it is stated that “the result is not open source, but would provide a collaborative environment for ITU-T members to collaborate on the development of software” which “could be useful when working on reference implementations” as described in “Resolution 90” (ITU-T, 2017b). In addition, it is claimed that “the ITU-T needs to be active in the standards ecosystem to determine how best to leverage the power of open source while maintaining the architectural constancy and interoperability that traditional standards bring.” (ITU-T, 2017b)

Concerning the role of ITU-T reference implementations (e.g. for the ITU-T H.264 and H.265 standards) in relation to external OSS implementations of the standards, it has been claimed



that “these open source implementations started after ITU releasing its reference implementations” and that “the existence of ITU-T reference implementations triggered and sped-up the development of external open source implementations by easing the test of functionalities against the reference implementation.” (ITU-T, 2017c) As a consequence, it is suggested that “ITU-T reference implementations participate heavily in an easier market adoption of ITU-T recommendations.” and, in general, that “ITU-T groups consider developing reference implementations for their Recommendations and provide the resources needed for such development.” (ITU-T, 2017c)

## 5.7 OASIS

In the context of OASIS, it is claimed that the organisation “offers projects - including open source projects - a path to standardization and de jure approval for reference in international policy and procurement” and the mission is “to advance the fair, transparent development of open source software and standards through the power of global collaboration and community” (OASIS, 2023a). One of the key programs in OASIS is “Open Projects” where work is conducted “in an environment of cross-organizational sharing and collaboration, where you can develop open source code and standards” (OASIS, 2023a). Further, the program “Open Projects gives communities the power to develop what they choose - APIs, code, specifications, reference implementations, guidelines– in one place, under open source licenses, with a path to recognition in global policy and procurement.” (OASIS, 2019)

It has also been claimed that the OASIS ‘technical committees have co-developed their standards with open source reference implementations and proofs of concept for over 20 years’ and that creation of the Open Projects program ‘encouraged future projects to operate on a “FOSS-first” basis, adding open source license defaults to our other routine process protections.’ (OASIS, 2021) Further, the members of the Open Projects Advisory Council at OASIS are “working to create a more transparent and collaborative future for industry, standards consortia, and open source” in order to “influence the direction of Open Projects program by providing strategic insight on the needs of open source projects and the industries they support” (OASIS, 2023b)

## 5.8 W3C

In the context of W3C, the expectation that more than a single software implementation for a specific standard is available to promote interoperability is not only expressed for specific standards like W3C PNG (W3C, 2023a), but also for W3C processes in general where it is stated that it is required to “ensure that independent interoperable implementations of each feature of the specification” are available (W3C, 2023c). This is also mentioned in the charters of specific working groups, e.g. the HTML working group stating that a “specification is expected to have at least two independent implementations of every feature defined in the specification.” (W3C, 2022b) Further, it has been claimed that “the W3C technical report development process is designed to” amongst other things, “facilitate royalty-free, interoperable implementations of Web Standards” (W3C, 2023c).

Concerning reference implementations, more than ten years ago the “Implementation and Test Cases Task Force” at W3C proposed to “define implementation interoperability requirements” and to “coordinate development of documentation on reference implementations.” (W3C,





2011) Further, two decades ago it was claimed that “The natural complement to W3C specifications is running code. Implementation and testing is an essential part of specification development and releasing the code promotes exchange of ideas in the developer community” and that “all W3C software is Open Source/ Free Software, and GPL compatible.” (W3C, 2003) The role and impact of OSS specifically has been further elaborated in the W3C context, and it has been claimed that the “web grew through Open Source” and that “Open Source grew through the Web” (Hazaël-Massieux, 2021). It was also claimed that “key implementations are open source” and that “W3C has benefited enormously from adopting OSS practices & infrastructure” (Hazaël-Massieux, 2021).

## 6 Analysis

Concerning availability of OSS reference implementations in different standardisation contexts (RQ1), results indicate that widely used and deployed OSS implementations are available for all standards identified for all the investigated SSOs. For five of the 24 identified implementations there are identified claims stating that they are reference implementations of a standard (Open Source MANO for ETSI NFV, OpenSSL for IETF RFC 6101, Apache HTTP Server for IETF RFC 7540, BIND 9 for IETF RFC 1035, and libpng for W3C PNG). For the rest of the identified OSS implementations, merely claims stating that they comply with or have support for a specific standard have been identified. In some cases, SSOs themselves provide reference implementations for identified standards as OSS (ETSI) and under other conditions (ITU-T). In other cases, reference implementations are provided by other organisations and communities, which in some cases (e.g. the libpng reference implementation for the W3C PNG standard) have a close connection to and collaboration with the SSO.

There is a wide variety of maintaining organisations and individuals for the identified OSS projects implementing standards. Nine of the (in total 24) OSS projects are maintained by foundations, five by an OSS community and its contributors, four by not-for-profit organisations, two by larger US-based enterprises, and the remaining four projects are maintained by a research organisation, an SME, an SSO, and an independent developer. Concerning location of headquarters, 10 of the 18 OSS projects that are maintained by organisations are maintained by a US-based organisation, three by an organisation based in France, two by an organisation based in Belgium, two by an organisation based in Germany, and one by an organisation based in Switzerland.

Regarding licensing, there is a range of different copyleft- and permissive OSS licences under which the identified OSS projects are claimed to be provided. The majority of the OSS projects are provided under copyleft conditions. Moreover, almost half (11 of 24) of the OSS projects are provided under a licence in the GPL-family of licences. In particular, the majority (15) of the (in total 24) OSS projects are provided under copyleft conditions, of which eight projects are licensed under strong copyleft conditions (different versions of the GPL and CERN-OHL) and seven under weak copyleft conditions (different versions of the LGPL, EPL-2.0, and MPL-2.0). The remaining nine projects are claimed to be provided under permissive OSS licences (Apache-2.0, BSD-3-Clause, and libpng-2.0). Further, findings show that software projects that implement standards from all investigated SSOs provide OSS under one (or several) modern OSS licences. Specifically, for all of the investigated SSOs, we find that several software



projects for each SSO provide OSS under one of the following OSS licences: the Apache-2-0 licence and a licence in the GPL-family of licences.

Concerning the activity in the identified OSS projects, a clear majority of the projects (21 out of 24) have either provided new software releases or have received commits to their source code repositories during the latest 12 months, and 17 out of 24 during the latest month. Overall, this indicates a high recent activity amongst the identified OSS projects. The OSS project with least recent activity (OpenPLC which implements the IEC 61131-3 standard) released its latest version in April 2019. Adoption and deployment of an OSS project with high recent activity may promote longevity of software systems. We acknowledge that the level of maturity of OSS projects and the release schedule used (time- or feature based) may have impact on the observed activity in the projects.

Regarding perceptions and practices related to OSS and OSS reference implementations in different standardisation contexts (RQ2), results suggest that all SSOs have perceived that software implementations and OSS impact on standardisation. Further, it is evident that SSOs have to varying extent considered and developed practices concerning use of OSS, and collaboration with OSS communities and projects developing implementations of standards. Results indicate that IETF, OASIS and W3C have to larger extent (compared to the other five SSOs) developed such practices. More specifically, results indicate that *IETF* practices the use of multiple software implementations of standards to promote interoperability. Moreover, the value of freely available software implementations (even if not referred to as reference implementations) of standards and the need to promote a symbiotic relationship between FOSS and open standards is emphasised. IETF hackathon initiatives promote this relationship. Further, results suggest that *OASIS* uses OSS and OSS reference implementations in standardisation processes since many years, and has developed both an initiative (“Open Projects”) and an organisational structure (the Open Projects Advisory Council) to support the work. Similar to the IETF, results indicate that *W3C* practices the use of multiple software implementations of standards to promote interoperability and highlight the importance and impact of W3C standards being royalty free. Further, the W3C uses OSS, OSS work practices and infrastructure, and an organisational structure (the Implementation and Test Cases Task Force) supported work on interoperability and reference implementations.

These findings concerning IETF, OASIS and W3C, especially from the perspective of interoperable implementations and royalty free conditions, are in line with earlier studies that suggested that a “major benefit that has emerged from the use of RF licensing for technical standards is the ease with which such standards may be implemented in open source software (OSS)” (Contreras et al., 2023), and that there is a “prevalence of standards organizations focusing on software (IETF, W3C, OASIS) that discourage patent royalties” (Blind & Kahin, 2019). Further, it has earlier been shown that “the primary SDOs responsible for Internet standards, the IETF and W3C, have evolved strong policies and norms favouring RF standards” and that the “preference for RF standards at the IETF and W3C can be traced, in part, to the historical origins of these groups in academia and government and their ties to the open source movement.” (Contreras, 2016). Regarding W3C specifically and the need for royalty free conditions and OSS to evolve the web, it has been claimed that “open source software became increasingly commonly used to provide the software ‘stack’ supporting the servers that enable the Web’s existence. The result was the adoption by W3C in 2003 of an extremely license fee



intolerant Patent Policy” (Bekkers & Updegrave, 2013) and that the “primary goal of the W3C Patent Policy is to enable W3C Recommendations to be implemented on a royalty-free basis” (Weitzner, 2005).

Amongst the remaining five SSOs, results suggest that ETSI and ITU-T share similarities. *ETSI* has elaborated on the complementary roles of OSS and standards to promote interoperability and avoidance of lock-in effects, and also elaborated use of OSS and OSS community collaboration to facilitate and support standardisation processes. ETSI has also recognised benefits of OSS reference implementations and specifically provided ETSI Open Source MANO as a reference implementation for ETSI NFV. Similar to ETSI, *ITU-T* has elaborated on and scrutinised how OSS may be utilised in standardisation processes and the value of OSS reference implementations for promoting interoperability. ITU-T has also scrutinised issues concerning collaboration with OSS communities and OSS licensing in relation to the SSO’s IPR guidelines. Further, ITU-T has recognised the value of adopting OSS work practices for its own software development, and use of own reference implementations to support and speed up the adoption and implementation of ITU-T specifications by third party OSS projects.

Results indicate that IEC and ISO share similarities and have mostly perceived that software implementations and OSS impact on standardisation. Specifically, *IEC* has perceived and acknowledged the role and potential impact of OSS on IEC standardisation processes. Further, an organisational structure (the *ahG 92* group) was appointed to monitor and assess how OSS licensing relates to IEC copyright policy. Similarly, *ISO* has perceived and acknowledged the role of OSS in standardisation processes, and statements concerning reference implementations (not necessarily in OSS) have been introduced in a number of ISO/IEC standards. Similar to IEC and ISO, *IEEE* has perceived and acknowledged the role and potential impact of OSS on IEEE standardisation processes. However, in addition, IEEE has established the collaboration platform “IEEE SA OPEN” to address challenges concerning the relation between OSS development and standards development, and in the context of this platform recently launched an initiative where one expected outcome is OSS reference implementations. We note that IEEE has changed its way of working concerning standardisation and it has been claimed that IEEE SA OPEN “is based on the notion that open source does not necessitate a lack of rules or structure. On the contrary, adhering to a small set of predefined governance rules and practices means that open-source communities can attract and keep supporters by minimizing the overhead that building an open-source community entails.” (Fish, 2020)

Amongst investigated SSOs which engage with software projects that implement standards we find that the extent to which established norms and work practices for OSS are being adhered to vary between different SSOs and software projects. For example, a report from ETSI refers to OSS with a definition that does not comply with the Open Source Definition (ETSI, 2016). Further, previous research shows that reference software developed by the Joint Collaborative Team on Video Coding (through a collaboration between the ITU-T and ISO/IEC JTC1) is provided “under a software licence that does not qualify as an OSS licence as its terms do not fulfil the Open Source Definition” (Lundell et al., 2023). Specifically, the “reference software is provided under a software licence (based on the BSD 3-Clause licence) which explicitly excludes patent rights” (Lundell et al., 2023).



## 7 Implications for practice

In this study (and specifically RQ1) we have focused on claimed support for standards in OSS implementations, and we acknowledge the distinction and difference between claimed- and actual support for a standard in a software implementation. Implementations may implement a subset of a standard and may also include features which go beyond the standard (Gamalielsson and Lundell, 2013). Moreover, a software implementation may deviate from a specification (Lundell et al., 2019). In our experience, this is often the case for standard implementations in the area of information security, where implementations (e.g. the OpenSSL project) may include different “tweaks” and extensions in order to be interoperable with other widely deployed and “de facto” implementations of a standard. Further, we have experienced that there may be disagreements between different stakeholders in a business scenario concerning the coverage of a specific standard. For example, in an information security scenario, a product developer may claim support for a certain standard but the customer may disagree since some features of a standard (such as access control and authentication) may be difficult to interpret in the standard. To further enhance the correctness of standards and their implementations, there are specific approaches employed in different fields. Moreover, for cryptography standards, it is in our experience common to arrange contests amongst researchers to identify weaknesses. Further, for most cryptography standards, test vectors are provided for implementers to use in order to verify a that a certain input generates the correct output.

Reference implementations may primarily be used for verifying that a standard is implementable, supporting interoperability testing among other implementations, and providing feedback to the standard development process. However, providing reference implementations and widely used implementations of a standard as OSS also promotes wide deployment in software systems and products, and may also affect the speed of diffusion and acceptance of standard specifications. In our experience, companies (especially startups and smaller companies) may benefit from the availability of OSS licensed implementations of standards that lower the barrier for adoption and deployment of standards in new products. Utilisation of existing reference (and widely deployed) implementations of standards in new products may be beneficial in order to avoid “reinventing the wheel”, and it may reduce the risk for introducing additional ambiguities related to the standard. Relatedly, in our experience, there is in some contexts a need for more efficient OSS implementations that are related to existing (OSS) reference implementations of standards, for deployment in software systems and products. One example is the OSS project libspng that implements the W3C PNG standard for which “the goal is to provide a fast PNG library with a simpler API than libpng” (libspng.org, 2023). Similarly, there are two acknowledged reference implementations for the JPEG image standard: “the fast and widely deployed libjpeg-turbo code, along with a complete implementation of JPEG coming from the Committee itself” (jpeg.org, 2019). It has been claimed that under some circumstances “libjpeg-turbo is generally 2-6x as fast as libjpeg” developed by the committee (Independent JPEG Group), and in “many cases, the performance of libjpeg-turbo rivals that of proprietary high-speed JPEG codecs” (libjpeg-turbo.org, 2023). It should also be mentioned that our experience is that some of the identified standards in this study have been deployed very widely in software systems and products, and that OSS implementations have been instrumental to achieve such deployment. For example, an implementation of the IETF 802.11 standard for wireless internet is an essential part of many



wireless products. Further, for the construction of a secure internet, the IETF SSL and OASIS SAML standards<sup>11</sup> are, in our view, of fundamental importance.

For effective deployment of OSS reference implementations of standards, it is, in our experience, important to consider the programming language of implementations. Some programming languages have broader support (for example, in terms of available compilers) in different operating systems and hardware platforms (e.g. the C programming language) and are therefore more portable. However, languages with broad support (such as C) may have other disadvantages such as low-level memory management. Formal and strongly typed programming languages, which are often used in the field of information security, may be beneficial for implementing standards to avoid ambiguities. However, such languages may not be broadly supported in operating systems and hardware platforms. Further, the code architecture for an OSS project may be important to consider, so that integration of code into other code bases can be done in an appropriate manner. Highly modularised architectures may promote use of OSS licensed reference implementation plugins. Moreover, the readability of software source code and documentation of code for OSS reference implementations may be crucial to consider to avoid further bugs and unwanted side effects. Further, in our experience, OSS implementations (and particularly OSS licensed compilers) for programming languages and associated standards are of great importance since they are widely used to build fundamental lower level IT infrastructure such as operating- and database systems. For some implementations of programming languages (e.g. the OpenJDK reference implementation of Java) there are different variants that are optimised with respect to different aspects such as memory- and processor use, and different companies may provide tailored forks for specific contexts (e.g. the RedHat build of OpenJDK).

In our experience, prior to adoption and deployment, it is also important to consider the community dynamics of OSS projects implementing standards, since some projects have higher barriers of entry (e.g. by requiring contributor agreements or other conditions for contributing) than other projects. Further, by observing how pull requests, issues, and bugs are handled in an OSS project, the responsiveness and social climate in the community may be assessed.

## 8 Conclusions

The study contributes a characterisation of availability of, and perceptions and practices concerning, reference implementations provided as Open Source Software for standards issued by different standards setting organisations. In conclusion, findings show that widely used and deployed Open Source Software implementations (of which a minority are referred to as reference implementations) are available for all standards identified for all the investigated standards setting organisations. These implementations are maintained by a variety of organisations of different types and provided under a variety of copyleft- and permissive Open Source Software licences. Further, findings show that standards setting organisations have to varying extent expressed perceptions and developed practices concerning collaboration with and contributions to Open Source Software communities and projects developing reference implementations for standards. Reference implementations and widely deployed

---

<sup>11</sup> Further, the IETF TLS (Transport Layer Security) standard (which is also implemented in OpenSSL, but which was not included in the results) is important, and the use of a mutual authentication (mTLS) process.



implementations of standards provided as Open Source Software create conditions for avoidance of different lock-in effects, and promotion of interoperable and long-term sustainable software systems.

It should be emphasised that reference implementations provided by standards setting organisations, such as the ITU-T reference implementations for the H.264 and H.265 standards, even though not provided as Open Source Software, have an important role and contribute to the evolution of the standards they implement and help other stakeholders that implement a standard in software provided both as Open Source Software licences and under proprietary conditions in order to validate different features of standards. Further, the role of the software implementation can be different in other standardisation contexts. In ETSI, for example, an effort where standardisation has promoted utilisation of OSS in their work is the ETSI-hosted Open Source MANO project that develops an Open Source NFV Management and Orchestration (MANO) software stack that is aligned with ETSI NFV. This OSS project has provided different types of feedback to aid formulating and finalising the ETSI MANO standards. Hence, different contexts of standardisation can learn from and utilise new and specific experiences related to the role and benefit of software implementations in order to improve standardisation and its associated processes in a rapidly evolving world.

## References

- alfresco.com (2023a). Alfresco. <https://www.alfresco.com/>
- alfresco.com (2023b). CMIS API reference. <https://docs.alfresco.com/content-services/6.0/develop/reference/cmisis-ref/>
- Aliprandi, S. (2011). Interoperability And Open Standards: The Key To True Openness And Innovation. *International Free and Open Source Software Law Review*, 3(1), pp. 5-24.
- apache.org (2023a). Apache HTTP Server Project. <https://httpd.apache.org/>
- apache.org (2023b). The Apache Software Foundation. [https://svn.apache.org/repos/asf/httpd/site/branches/ASF/docs/ABOUT\\_APACHE.html](https://svn.apache.org/repos/asf/httpd/site/branches/ASF/docs/ABOUT_APACHE.html)
- apache.org (2023c). Apache Olingo. <https://olingo.apache.org/>
- apache.org (2023d). The Apache Xerces Project. <https://xerces.apache.org/>
- atlassian.net (2023). OpenSAML. <https://shibboleth.atlassian.net/wiki/spaces/OSAML/overview>
- Ballard, M. (2012). Portugal's prescribed open standards - full list. *ComputerWeekly.com*, 22 November. <http://www.computerweekly.com/blogs/public-sector/2012/11/-tablemytable-border-collapse.html>
- beanit.com (2023). IEC61850bean. <https://www.beanit.com/iec-61850/>
- Behlendorf, B. (2009). How Open Source Can Still Save the World. Keynote Presentation, In 5th IFIP WG 2.13 International Conference on Open Source Systems, OSS 2009, Skövde, Sweden, 5 June.
- Bekkers, R. & Updegrave, A. (2013). IPR Policies and Practices of a Representative Group of Standards-Setting Organizations Worldwide, Commissioned by the Committee on Intellectual Property Management in Standard-Setting Processes, National Research Council, Washington, May.
- Berkman (2005). Roadmap for Open ICT Ecosystems. Berkman Centre for Internet & Society at Harvard Law School.
- Bhattacharya, J., Ilavarasan, P. V. & Gupta, S. (2007). Open standards and accessibility to information: a critical analysis of OOXML in India. In *Proceedings of the 1st international Conference on theory and Practice of Electronic Governance, ICEGOV'07*, Vol. 232, ACM, New York, pp. 151-154.
- Bird, G. B. (1998). The Business Benefit of Standards. *StandardsView*, 6(2), pp. 76-80.



- Blind, K., Bekkers, R. N. A., Dietrich, Y., Iversen, E. J., Köhler, F., Müller, B., Pohlmann, T., Smeets, S. J. J., & Verweijen, G. J. H. (2011). Study on the interplay between standards and intellectual property rights (IPRs). Office for Official Publications of the EC, ISBN 978-92-79-20654-2
- Blind, K. & Boehm, M. (2019). The Relationship Between Open Source Software and Standard Setting. Thumm, N. (Ed.) EUR 29867 EN, JRC (Joint Research Centre) Science for Policy Report, Publications Office of the European Union, Luxembourg, 2019, ISBN 978-92-76-11593-9.
- Blind, K. & Kahin, B. (2019). Standards and the Global Economy. In Contreras, J. L. (Ed.) The Cambridge Handbook of Technical Standardization Law: Competition, Antitrust, and Patents, Cambridge University Press, Cambridge, ISBN 978-1-107-57013-9, pp. 7-16.  
<https://doi.org/10.1017/9781316416723.003>
- Boehm, M., & Eisape, D. (2021). Standard setting organizations and open source communities: Partners or competitors?. *First Monday*, 26(7).
- Bradner, S. (1999). The Internet Engineering Task Force. In DiBona, C. et al. (Eds.) *Opensources: Voices from the Open Source Revolution*, O'Reilly & Associates, ISBN: 1-56592-582-3, pp. 43-47.
- Brock, A. (2013). Understanding Commercial Agreements With Open Source Companies. In Coughlan, S. (Eds.) *Thoughts on Open Innovation - Essays on Open Innovation from leading thinkers in the field*, OpenForum Europe LTD for OpenForum Academy, Brussels
- CERN (2023). The White Rabbit Project. <http://white-rabbit.web.cern.ch/consortiuminfo.org> (2023). ConsortiumInfo.org. <https://www.consortiuminfo.org/>
- Contreras, J. L. (2016). Patents and Internet Standards. Centre for International Governance Innovation and Chatham House, Paper Series No. 29, April.  
[https://www.cigionline.org/sites/default/files/gcig\\_no29\\_web.pdf](https://www.cigionline.org/sites/default/files/gcig_no29_web.pdf)
- Contreras, J. L., Bekkers, R. N. A., Biddle, C. B., Bonadio, E., Carrier, M. A., Chao, B., Duan, C., Gilbert, R., Henkel, J., Hovenkamp, E., Husovec, M., Jakobs, K., Kim, D., Lemley, M. A., Love, B. J., McDonagh, L., Scott Morton, F., Schultz, J., Simcoe, T., Urban, J. M., & Xiang, J. (2023). Preserving the Royalty-Free Standards Ecosystem. *European Intellectual Property Review*, 45(7), 371-375.
- DeNardis, L. (2010). E-Governance Policies for Interoperability and Open Standards. *Policy & Internet*, 2(3), pp. 129-164.
- EC (2004). European Interoperability Framework for pan-European eGovernment Services, Version 1.0, European Commission, ISBN 92-894-8389-X.
- EC (2012). Implementing FRAND standards in Open Source: Business as usual or mission impossible?. Report summary from a workshop organised by the European Commission (EC) and the European Patent Office, Brussels, 22 November.
- EC (2013). Commission staff working document: Guide for the procurement of standards-based ICT — Elements of Good Practice, SWD(2013) 224 final. 25 June, European Commission, Brussels.
- eclipse.org (2023a). Eclipse OM2M. <https://projects.eclipse.org/projects/iot.om2m>
- eclipse.org (2023b). Eclipse 4diac. <https://www.eclipse.org/4diac/>
- Egyedi, T. M. (2007). Standard-compliant, but incompatible?!. *Computer Standards & Interfaces*, 29(6), pp. 605–613.
- ETSI (2012). 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. ETSI SR 002 960 V1.0.1.  
[https://www.etsi.org/deliver/etsi\\_sr/002900\\_002999/002960/01.00.01\\_60/sr\\_002960v010001p.pdf](https://www.etsi.org/deliver/etsi_sr/002900_002999/002960/01.00.01_60/sr_002960v010001p.pdf)
- ETSI (2015). ETSI EN 302 755 V1.4.1. Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2).  
[https://www.etsi.org/deliver/etsi\\_en/302700\\_302799/302755/01.04.01\\_20/en\\_302755v010401a.pdf](https://www.etsi.org/deliver/etsi_en/302700_302799/302755/01.04.01_20/en_302755v010401a.pdf)



- ETSI (2016). Cloud Standards Coordination Phase 2; Cloud Computing Standards and Open Source; Optimizing the relationship between standards and Open Source in Cloud Computing. ETSI SR 003 382 V2.1.1. [https://www.etsi.org/deliver/etsi\\_sr/003300\\_003399/003382/02.01.01\\_60/sr\\_003382v020101p.pdf](https://www.etsi.org/deliver/etsi_sr/003300_003399/003382/02.01.01_60/sr_003382v020101p.pdf)
- ETSI (2021). ETSI GS NFV 006 V2.1.1. Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Architectural Framework Specification. [https://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/006/02.01.01\\_60/gs\\_nfv006v020101p.pdf](https://www.etsi.org/deliver/etsi_gs/NFV/001_099/006/02.01.01_60/gs_nfv006v020101p.pdf)
- ETSI (2022). OSM Release Thirteen Release Notes. December, [https://osm-download.etsi.org/ftp/osm-13.0-thirteen/OSM\\_Release\\_THIRTEEN\\_Release\\_Notes.pdf](https://osm-download.etsi.org/ftp/osm-13.0-thirteen/OSM_Release_THIRTEEN_Release_Notes.pdf)
- ETSI (2023a). ETSI. <https://www.etsi.org/>
- ETSI (2023b). Open Source Mano. <https://osm.etsi.org/>
- ETSI (2023c). ONE MACHINE-TO-MACHINE PARTNERSHIP PROJECT (ONEM2M). <https://www.etsi.org/committee/1419-onem2m>
- ETSI (2023d). In the Light of Ten Years from the NFV Introductory Whitepaper. ETSI White Paper No. 53. <https://www.etsi.org/images/files/ETSIWhitePapers/ETSI-WP-53-In-the-Light-of-Ten-years-from-the-NFV-Introductory-Whitepaper.pdf>
- ETSI (2023e). Open Source MANO (OSM). <https://www.etsi.org/technologies/open-source-mano>
- ffmpeg.org (2023a). FFmpeg. <https://ffmpeg.org/>
- ffmpeg.org (2023b). libavcodec/g722.c File Reference. [https://ffmpeg.org/doxygen/1.0/g722\\_8c.html](https://ffmpeg.org/doxygen/1.0/g722_8c.html)
- ffmpeg.org (2023c). FFmpeg License and Legal Considerations. <https://www.ffmpeg.org/legal.html>
- Fish, R. S. (2020). IEEE Open: A New Option for Open Source. Commentary, IEEE Communications Standards Magazine, 4(1), pp. 4-4.
- Fomin, V. V., Kühn Pedersen, M., & de Vries, H. J. (2008). Open Standards and Government Policy: Results of a Delphi Survey. Communication of the AIS, 22(25), pp. 459-484.
- Friedrich, J. (2011). Making innovation happen: The role of standards and openness in an innovation-friendly ecosystem, In 2011 7th International Conference on Standardization and Innovation in Information Technology, IEEE, pp. 1-8.
- Gamalielsson, J. & Lundell, B. (2013). Experiences from implementing PDF in open source: challenges and opportunities for standardisation processes. In Jakobs, K. (Ed.) Proceedings of the 8th IEEE Conference on Standardization and Innovation in Information Technology (SIIT 2013), ISBN 3-86130-802-9, IEEE, Piscataway, pp. 39-49.
- Gamalielsson, J. & Lundell, B. (2017a). On organisational involvement and collaboration in W3C standards through editorship. Journal of Internet Services and Applications, 8(5).
- Gamalielsson, J. & Lundell, B. (2017b). On licensing and other conditions for contributing to widely used open source projects: an exploratory analysis. In Proceedings of the 13th International Symposium on Open Collaboration (OpenSym '17). ACM, New York, NY, USA, Article 9, 14p.
- Gamalielsson, J. & Lundell, B. (2021a). On Engagement With ICT Standards and Their Implementations in Open Source Software Projects: Experiences and Insights From the Multimedia Field. International Journal of Standardization Research (IJSR), 19(1), pp. 1-28.
- Gamalielsson, J. & Lundell, B. (2021b). Exploring Influences Between ICT Standards and Their Implementations in Open Source Software Projects: The Case of H.264. In: Jakobs, K. (Ed.) Joint Proceedings of the 25th EURAS Annual Standardisation Conference - Standardisation and Innovation & 11th International Conference on Standardisation and Innovation in Information Technology (SIIT) - The Past, Present and FUTURE of ICT Standardisation, EURAS contributions to standardisation research, Vol. 16, Mainz Publishers, ISBN 978-3-95886-421-4, pp. 211-226.





- Gamalielsson, J., Lundell, B. & Butler, S. (2021). On Engagement with ICT Standards and Their Implementations in Open Source Software Projects: The Case of H.265. In: Jakobs, K. (Ed.) Joint Proceedings of the 25th EURAS Annual Standardisation Conference - Standardisation and Innovation & 11th International Conference on Standardisation and Innovation in Information Technology (SIIT) - The Past, Present and FUTURE of ICT Standardisation, EURAS contributions to standardisation research, Vol. 16, Mainz Publishers, ISBN 978-3-95886-421-4, pp. 191-210.
- Gamalielsson, J., Lundell, B., Brax, C., Persson, T., Mattsson, A., Gustavsson, T., & Feist, J. (2023). On availability of Open Source Software reference implementations for standards issued by different standards setting organisations, In: Jakobs, K. (Ed.) Joint Proceedings of the 27th EURAS Annual Standardisation Conference & 12th International Conference on Standardisation and Innovation in Information Technology (SIIT) - (Responsible) Standardisation for Smart Systems, EURAS contributions to standardisation research, Vol. 18, Mainz Publishers, ISBN 978-3-95886-491-7, pp. 97-116.
- Gamalielsson, J., Lundell, B., Butler, S., Brax, C., Persson, T., Mattsson, A., Gustavsson, T., Feist, J., & Öberg, J. (2022a). On engagement with ICT standards and their implementations in open source software projects: the case of WebRTC. In: Jakobs, K. & Kim, D. H. (Eds.): Proceedings of the 26th EURAS Annual Standardisation Conference – Standardisation and Open Source, EURAS contributions to standardisation research, Vol. 17, Mainz Publishers, ISBN 978-3-95886-446-7, pp. 143-162.
- Gamalielsson, J., Lundell, B., Butler, S., Brax, C., Persson, T., Mattsson, A., Gustavsson, T., Feist, J. & Öberg, J. (2022b) On Engagement with Open Source Software, Open Source Hardware, and Standard Setting: The Case of White Rabbit, In The 18th International Symposium on Open Collaboration (OpenSym 2022), ACM, New York.
- Gamalielsson, J., Lundell, B., Feist, J., Gustavsson, T. & Landqvist, F. (2015). On organisational influences in software standards and their open source implementations, *Information and Software Technology*, 67, pp. 30-43.
- Ghosh, R. A. (2005). Open Standards and Interoperability Report: An Economic Basis for Open Standards. FLOSSPOLs, Deliverable D4, 12 December.
- gnu.org (2023a). GCC, the GNU Compiler Collection. <https://gcc.gnu.org/>
- gnu.org (2023b). C++ Standards Support in GCC. <https://gcc.gnu.org/projects/cxx-status.html#cxx11>
- gnu.org (2023c). GCC steering committee. <https://gcc.gnu.org/steering.html>
- Haapanen, A. (2015). Free and Open Source Software & the Mystery of Software Patent Licenses, *International Free and Open Source Software Law Review*, 7(1), pp. 19-28.
- Hazaël-Massieux, D. (2021). W3C & Open Source. 11 March. <https://www.w3.org/2021/Talks/dhm-oss-ofe/>
- Husovec, M. (2019). Standardization, open source and innovation: sketching the effect of IPR policies. In Contreras, J. (Ed.) *The Cambridge Handbook of Technical Standardization Law: Further Intersections of Public and Private Law*, Volume 2, ISBN 978-1-107-12971-9, pp. 177-197.
- IEC (2012). IEC 61499-1:2012. Function blocks - Part 1: Architecture. <https://webstore.iec.ch/publication/5506>
- IEC (2013). IEC 61131-3:2013. Programmable controllers - Part 3: Programming languages. <https://webstore.iec.ch/publication/4552>
- IEC (2020a). SC 32 Business Plan 2020. 12 September. <https://www.iec.ch/public/miscfiles/sbp/jtc1-sc32.pdf>
- IEC (2020b). IEC ACADEMY WEBINAR Q&A - Good practice in Drafting and Editing IEC Standards. 9 June. [https://www.iec.ch/system/files/2020-07/Q%26A\\_IEC\\_Good\\_practice\\_DraftingEditing\\_Standards\\_0.pdf](https://www.iec.ch/system/files/2020-07/Q%26A_IEC_Good_practice_DraftingEditing_Standards_0.pdf)
- IEC (2022). Meeting diversity and sustainability challenges head on. 7 February. <https://www.iec.ch/blog/meeting-diversity-and-sustainability-challenges-head>
- IEC (2023a). International Electrotechnical Commission. <https://iec.ch/>
- IEC (2023b). IEC 61850:2023 SER Series. Communication networks and systems for power utility automation - ALL PARTS. <https://webstore.iec.ch/publication/6028>



- IEC (2023c). WG1 – Trend monitoring.  
[https://www.iec.ch/ords/f?p=103:14:608715761021717:::FSP\\_ORG\\_ID,FSP\\_LANG\\_ID:21085,25#](https://www.iec.ch/ords/f?p=103:14:608715761021717:::FSP_ORG_ID,FSP_LANG_ID:21085,25#)
- IEC (2023d). SMB Communiqué – Issue #2. <https://www.iec.ch/smb-communication/issue2>
- IEEE (2019). IEEE 1588-2019. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. <https://standards.ieee.org/ieee/1588/6825/>
- IEEE (2020). IEEE 802.11-2020. IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. <https://standards.ieee.org/ieee/802.11/7028/>
- IEEE (2022a). IEEE 802.3-2022. IEEE Standard for Ethernet. <https://standards.ieee.org/ieee/802.3/10422/>
- IEEE (2022b). Global Initiative on Digital Transformation of Pandemic Surveillance - Industry Connections Activity Initiation Document (ICAID). [https://standards.ieee.org/wp-content/uploads/2022/07/IC20-005-Global-Initiative-on-Digital-Transformation-of-Pandemic\\_Surveillance.pdf](https://standards.ieee.org/wp-content/uploads/2022/07/IC20-005-Global-Initiative-on-Digital-Transformation-of-Pandemic_Surveillance.pdf)
- IEEE (2023a). IEEE. <https://www.ieee.org/>
- IEEE (2023b). IEEE SA OPEN. <https://saopen.ieee.org/>
- IEEE (2023c). Global Initiative on Digital Transformation of Pandemic Surveillance.  
<https://standards.ieee.org/industry-connections/digital-transformation-pandemic-surveillance/>
- IETF (1987). Domain names - implementation and specification. RFC 1035.  
<https://datatracker.ietf.org/doc/rfc1035/>
- IETF (1992). The Internet Standards Process. RFC 1310. <https://datatracker.ietf.org/doc/html/rfc1310/>
- IETF (1996). The Internet Standards Process -- Revision 3. RFC 2026. <https://datatracker.ietf.org/doc/rfc2026/>
- IETF (2008). What Makes for a Successful Protocol?. RFC 5218. <https://www.rfc-editor.org/rfc/rfc5218>
- IETF (2009). Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard. RFC 5657. <https://datatracker.ietf.org/doc/html/rfc5657/>
- IETF (2011a). The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101.  
<https://datatracker.ietf.org/doc/rfc6101/>
- IETF (2011b). Reducing the Standards Track to Two Maturity Levels. RFC 6410.  
<https://datatracker.ietf.org/doc/html/rfc6410>
- IETF (2015). Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540.  
<https://datatracker.ietf.org/doc/rfc7540/>
- IETF (2016a). IETF Trends and Observations. 29 February. <https://datatracker.ietf.org/doc/draft-arkko-ietf-trends-and-observations/>
- IETF (2016b). Bringing together open standards and open source in 2016. 31 March.  
<https://www.ietf.org/blog/bringing-together-open-standards-and-open-source-2016/>
- IETF (2022). HTTP/3. RFC 9114. <https://datatracker.ietf.org/doc/rfc9114/>
- IETF (2023). IETF. <https://www.ietf.org/>
- inkscape.org (2023a). Inkscape. <https://inkscape.org/>
- inkscape.org (2023b). About SVG. <https://inkscape.org/develop/about-svg/>
- isc.org (2023). BIND 9. <https://www.isc.org/bind/>
- ISO (1998). ISO/IEC 13818-3:1998. Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio. <https://www.iso.org/standard/26797.html>
- ISO (2015). ISO/IEC 26300-1:2015. Information technology — Open Document Format for Office Applications (OpenDocument) v1.2 — Part 1: OpenDocument Schema. <https://www.iso.org/standard/66363.html>
- ISO (2020). ISO/IEC 14882:2020. Programming languages — C++. <https://www.iso.org/standard/79358.html>
- ISO (2021). ISO/IEC 15444-5:2021. Information technology — JPEG 2000 image coding system — Part 5: Reference software. <https://www.iso.org/standard/81575.html>
- ISO (2023a). ISO. <https://www.iso.org/>



- ISO (2023b). The 2023 Open Standards and Open Source Software Code Sprint. 23 February.  
<https://committee.iso.org/sites/tc211/home/standards-in-action/news/2023-02-23-the-2023-open-standar.html>
- ITU-T (2012). ITU-T G.722 (09/2012). 7 kHz audio-coding within 64 kbit/s. <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11673>
- ITU-T (2016a). H.264.2: Reference software for ITU-T H.264 advanced video coding.  
<https://www.itu.int/rec/TREC-H.264.2-201602-I/en>
- ITU-T (2016b). H.265.2: Reference software for ITU-T H.265 high efficiency video coding.  
<https://www.itu.int/rec/T-REC-H.265.2-201612-I/en>
- ITU-T (2016c). Resolution 90 – Open source in the ITU Telecommunication Standardization Sector. 25 October.  
<https://www.itu.int/pub/publications.aspx?lang=en&parent=T-RES-T.90-2022>
- ITU-T (2017a). Open Source and its Relationship to ITU Standardization Study Groups. TSAG-C.13.  
[https://www.itu.int/dms\\_pub/itu-t/md/17/tsag/c/T17-TSAG-C-0013!!MSW-E.docx](https://www.itu.int/dms_pub/itu-t/md/17/tsag/c/T17-TSAG-C-0013!!MSW-E.docx)
- ITU-T (2017b). Open Source and the ITU-T. TSAG-C.3. [https://www.itu.int/dms\\_pub/itu-t/md/17/tsag/c/T17-TSAG-C-0003!!MSW-E.docx](https://www.itu.int/dms_pub/itu-t/md/17/tsag/c/T17-TSAG-C-0003!!MSW-E.docx)
- ITU-T (2017c). Open Source and Reference Implementation. TSAG-C.4. <https://www.itu.int/md/T17-TSAG-C-0004/en>
- ITU-T (2021a). ITU-T H.264 (V14) (08/2021). Advanced video coding for generic audiovisual services.  
<https://www.itu.int/ITU-T/recommendations/rec.aspx?id=14659>
- ITU-T (2021b). ITU-T H.265 (V8) (08/2021). High efficiency video coding. <https://www.itu.int/ITU-T/recommendations/rec.aspx?id=14660>
- ITU-T (2023). ITU Telecommunication Standardization Sector. <https://www.itu.int/en/ITU-T/Pages/default.aspx>
- jpeg.org (2019). JPEG develops an XLM verification model for its Next Generation Image Coding Standard (JPEG XL). [https://jpeg.org/items/20190204\\_press.html](https://jpeg.org/items/20190204_press.html)
- kernel.org (2023a). The Linux Kernel Archives. <https://kernel.org/>
- kernel.org (2023b). PHY Abstraction Layer. <https://www.kernel.org/doc/html/next/networking/phy.html>
- Krechmer, K. (2005). The Meaning of Open Standards, In Proceedings of the 38th Hawaii International Conference on System Sciences – 2005, IEEE Computer Society, Los Alamitos, 10p.
- libjpeg-turbo.org (2023). libjpeg-turbo. <https://libjpeg-turbo.org/>
- libpng.org (2022). Portable Network Graphics. <http://www.libpng.org/pub/png/>
- libpng.org (2023). libpng. <http://www.libpng.org/pub/png/libpng.html>
- libreoffice.org (2023). LibreOffice. <https://www.libreoffice.org/>
- libreoffice.org (2016). Getting Started Guide. Appendix B. Open Source, Open Standards, OpenDocument.  
<https://documentation.libreoffice.org/assets/Uploads/Documentation/en/GS5.1/HTML/GS5116-OpenSourceStandardsDoc.html>
- libspng.org (2023). libspng. <https://libspng.org/>
- Lindberg, V. (2019). OSS and FRAND: Complementary Models for Innovation and Development, The Columbia Science & Technology Law Review, 20(2), pp. 251-270.
- Lundell, B. (2011). e-Governance in public sector ICT procurement: what is shaping practice in Sweden?. European Journal of ePractice, 12(4), March/April, pp. 66-78.  
[https://joinup.ec.europa.eu/sites/default/files/76/a7/05/ePractice%20Journal-%20Vol.%202012-March\\_April%202011.pdf](https://joinup.ec.europa.eu/sites/default/files/76/a7/05/ePractice%20Journal-%20Vol.%202012-March_April%202011.pdf)
- Lundell, B. (2012). Why do we need Open Standards?, In Orviska, M. and Jakobs, K. (Eds.) Proceedings 17th EURAS Annual Standardisation Conference ‘Standards and Innovation’, The EURAS Board Series, Aachen, ISBN: 978-3-86130-337-4, pp. 227-240.



- Lundell, B. & Gamalielsson, J. (2017). On the potential for improved standardisation through use of open source work practices in different standardisation organisations: How can open source-projects contribute to development of IT-standards?. In Blind, K. and Jakobs, K. (Eds.) *Digitalisation: Challenge and Opportunity for Standardisation*. Proceedings of the 22nd EURAS Annual Standardisation Conference, EURAS Contributions to Standardisation Research, Vol. 12. Verlag Mainz, Aachen, pp. 137-155.
- Lundell, B. & Lings, B. (2010). How open are local government documents in Sweden? A case for open standards. In Agerfalk, P. et al. (Eds.) *Open Source Software: New Horizons*, Springer, Berlin, pp. 177-187.
- Lundell, B., Butler, S., Fischer, T., Gamalielsson, J., Brax, C., Feist, J., Gustavsson, T., Katz, A., Kvarnström, B., Lönroth, E. & Mattsson, A. (2022) *Effective Strategies for Using Open Source Software and Open Standards in Organizational Contexts – Experiences From the Primary and Secondary Software Sectors*, IEEE Software, 39(1), pp. 84-92.
- Lundell, B., Gamalielsson, J. & Katz, A. (2015). On implementation of Open Standards in software: To what extent can ISO standards be implemented in open source software?. *International Journal of Standardization Research*, 13(1), pp. 47-73.
- Lundell, B., Gamalielsson, J. & Katz, A. (2019). Implementing IT Standards in Software: challenges and recommendations for organisations planning software development covering IT standards. *European Journal of Law and Technology*, 10(2).  
<https://ejlt.org/index.php/ejlt/article/view/709/>
- Lundell, B., Gamalielsson, J. & Katz, A. (2023). Implementing the HEVC standard in software: Challenges and Recommendations for organisations planning development and deployment of software. *Journal of Standardisation*, 2, <https://doi.org/10.18757/jos.2022.6695>
- Lundin, M. A. (2007). ISO standard offers iron-clad protection for financial services. *ISO Focus*, 4 (7/8), pp. 28-29. [https://www.iso.org/files/live/sites/isoorg/files/news/magazine/ISO%20Focus%20\(2004-2009\)/2007/ISO%20Focus,%20July-August%202007.pdf](https://www.iso.org/files/live/sites/isoorg/files/news/magazine/ISO%20Focus%20(2004-2009)/2007/ISO%20Focus,%20July-August%202007.pdf)
- Maracke, C. (2019). Free and Open Source Software and FRAND-based patent licenses: How to mediate between Standard Essential Patent and Free and Open Source Software. *Journal of World Intellectual Property*, 22 (3-4), pp. 78-102.
- Michaud, P. R. (2015). How (not) to create a language specification for Perl 6 – Lessons learned. FOSDEM 2015, Brussels, January 31.  
[https://archive.fosdem.org/2015/schedule/event/perl6\\_lang\\_spec\\_lessons\\_learned/attachments/slides/735/export/events/attachments/perl6\\_lang\\_spec\\_lessons\\_learned/slides/735/fosdem\\_specs\\_1.pdf](https://archive.fosdem.org/2015/schedule/event/perl6_lang_spec_lessons_learned/attachments/slides/735/export/events/attachments/perl6_lang_spec_lessons_learned/slides/735/fosdem_specs_1.pdf)
- NIST (2023). Reference Implementation. [https://csrc.nist.gov/glossary/term/Reference\\_Implementation](https://csrc.nist.gov/glossary/term/Reference_Implementation)
- nlnet.nl (2023). ISC BIND 9. NLnet foundation. <https://nlnet.nl/project/iscbind9/>
- NOC (2007). *The Netherlands in Open Connection: An action plan for the use of Open Standards and Open Source Software in the public and semi-public sector*. The Ministry of Economic Affairs, The Hague, November.
- NPS (2016). *Open IT-standards*. National Procurement Services, Kammarkollegiet, Dnr 96-38-2014, 7 March.  
<https://www.avropa.se/globalassets/dokument/open-it-standards.pdf>
- OASIS (2005). SAML V2.0 Standard. [https://wiki.oasis-open.org/security/FrontPage#SAML\\_V2.0\\_Standard](https://wiki.oasis-open.org/security/FrontPage#SAML_V2.0_Standard)
- OASIS (2011). *Open Document Format for Office Applications (OpenDocument) Version 1.2*. <http://docs.oasis-open.org/office/v1.2/OpenDocument-v1.2.pdf>
- OASIS (2015). *Content Management Interoperability Services (CMIS) Version 1.1*. <http://docs.oasis-open.org/cmisis/CMIS/v1.1/CMIS-v1.1.html>
- OASIS (2019). *OASIS Introduces Open Projects Program to Bridge Open Source and Standards Development*. 8 May. <https://www.oasis-open.org/2019/05/08/introducing-open-projects/>
- OASIS (2020). *OData Version 4.01*. <https://www.oasis-open.org/standard/odata-v4-01-os/>



- OASIS (2021). Managed Open Projects. 8 September. <https://www.oasis-open.org/2021/09/08/managed-open-projects/>
- OASIS (2023a). OASIS Open. <https://www.oasis-open.org/>
- OASIS (2023b). Open Projects Advisory Council. <https://www.oasis-open.org/open-projects-advisory-council/>
- ohwr.org (2023). PPSi. <https://ohwr.org/project/ppsi>
- openmuc.org (2023). OpenIEC61850 Overview. <https://www.openmuc.org/iec-61850/>
- openplcproject.org (2023). OPENPLC. <https://openplcproject.com/>
- openssl.org (2023a). OpenSSL. Cryptography and SSL/TLS Toolkit. <https://www.openssl.org/>
- openssl.org (2023b). SSL and TLS Protocols. [https://wiki.openssl.org/index.php/SSL\\_and\\_TLS\\_Protocols](https://wiki.openssl.org/index.php/SSL_and_TLS_Protocols)
- OSD (2023). The Open Source Definition. Open Source Initiative, <https://opensource.org/osd/>
- Parsons, G. (2017). IEEE open source offering. 15 June 2017. <https://mentor.ieee.org/802-ec/dcn/17/ec-17-0103-00-00SA-ieee-open-source-offering.pdf>
- Pizzo, M. (2015). Olingo releases OData V4 Support for Java Developers. 29 September, <https://www.odata.org/blog/olingo-releases-odata-v4-support-for-java/>
- Rao, S. (2016). Open Source NFV Part Four: Open Source MANO. The New Stack, 23 July. <https://thenewstack.io/opensource-nfv-part-4-opensource-mano>
- readthedocs.io (2023). General DNS Reference Information. [https://bind9.readthedocs.io/en/v9\\_18\\_9/general.html](https://bind9.readthedocs.io/en/v9_18_9/general.html)
- shibboleth.net (2023). Shibboleth Consortium. <https://www.shibboleth.net/>
- Simcoe, T. S. (2006). Open Standards and intellectual property rights. In Chesbrough, H., Vanhaverbeke, W. and West, J. (Eds.) Open Innovation researching a new paradigm, Oxford University Press, Oxford.
- Simon, K. D. (2005). The value of open standards and open-source software in government environments. IBM Systems Journal, 44(2), pp. 227-238.
- sourceforge.io (2023). The LAME Project. <https://lame.sourceforge.io/>
- UK (2012). Open Standards Principles: For software interoperability, data and document formats in government IT specifications. HM Government, 7 September. [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/459074/Open-Standards-Principles-2012.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/459074/Open-Standards-Principles-2012.pdf)
- Updegrove, A. (2017a). Open Source Stacks: Jumping the Shark or Poised for Dominance?. ConsortiumInfo.org, 16 June, <https://www.consortiuminfo.org/open-source-open-standards/open-source-stacks-jumping-the-shark-or-poised-for-dominance/>
- Updegrove, A. (2017b). Open Source or Open Standards? (Yes!) The Future has Arrived. ConsortiumInfo.org, 15 November, <https://www.consortiuminfo.org/open-source-open-standards/open-source-or-open-standards-yes-the-future-has-arrived/>
- videolan.org (2023a). VLC media player. <https://www.videolan.org/vlc/>
- videolan.org (2023b). VLC Features. <https://www.videolan.org/vlc/features.html>
- videolan.org (2023c). x264. <https://www.videolan.org/developers/x264.html>
- de Vries, H. J. (2006). IT Standards Typology. In Jakobs, K. (Ed.) Advanced Topics in Information Technology Standards and Standardization Research, IGI Global, Vol. 1, pp. 1-26.
- w1.fi (2023). hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator. <http://w1.fi/hostapd/>
- W3C (2003). W3C Open Source Software. 29 May. <https://www.w3.org/QA/2003/06/Status.html>
- W3C (2006). Extensible Markup Language (XML) 1.1 (Second Edition). W3C Recommendation, <https://www.w3.org/TR/xml11/>
- W3C (2011). Implementation and Test Cases Task Force. 1 August. [https://www.w3.org/2011/prov/wiki/Implementation\\_and\\_Test\\_Cases\\_Task\\_Force](https://www.w3.org/2011/prov/wiki/Implementation_and_Test_Cases_Task_Force)
- W3C (2018). Scalable Vector Graphics (SVG) 2. W3C Candidate Recommendation, <https://www.w3.org/TR/SVG2/>



- W3C (2022a). Portable Network Graphics (PNG) Specification (Third Edition). W3C First Public Working Draft, <https://www.w3.org/TR/png/>
- W3C (2022b). HTML Working Group Charter. 6 June. <https://www.w3.org/2022/06/html-wg-charter.html>
- W3C (2023a). PNG Working Group Charter. <https://www.w3.org/Graphics/PNG/png-2021.html>
- W3C (2023b). W3C. <https://www.w3.org/>
- W3C (2023c). W3C Process Document. 12 June. <https://www.w3.org/2023/Process-20230612/>
- Weitzner, D. J. (2005). Standards, Patents, and the Dynamics of Innovation on the World Wide Web. In Bolin, S. (Ed.) *The Standards Edge: Open Season*, The Bolin Group, Ann Arbor, ISBN 0-9748648-2-X, pp. 187-198.
- Wenning, R. & Bos, B. (2007). A list of specification developing organizations (SDO). 10 April. <https://www.w3.org/2004/copras/docu/sdo-list.html>
- West, J. & Dedrick, J. (2001). Proprietary vs. Open Standards in the Network Era: An Examination of the Linux Phenomenon. In *Proceedings of the 34th Hawaii International Conference on System Sciences - 2001*, IEEE Computer Society, Los Alamitos, 10p.
- x265.org (2023). x265 HEVC Encoder / H.265 Video Codec. <https://www.x265.org/>



## Appendix

**Table 1.** An overview of the identified 24 standards and associated implementations with respect to maintainer, licence, latest release, and latest commit. For the “Maintainer” column, the organisation type and location of headquarters are shown in parenthesis. Country codes for headquarters are stated in accordance with the ISO 3166-1 alpha 2 standard. Organisation types, in order of appearance, comprise Standard Setting Organisation (SSO), Foundation (FND), Nonprofit Organisation (NPO), Small and medium-sized enterprise (SME), self-employed, Research Organisation (RO), and Larger Enterprise (LE). For the column “Licence”, licensing conditions for implementations are referred to using the SPDX format (<https://spdx.org/licenses/>).

SSO	Standard	OSS-implementation	Maintainer	Licence	Latest release [YYYY-MM]	Latest commit [YYYY-MM]
ETSI	NFV	Open Source MANO	ETSI (SSO, FR)	Apache-2.0	2022-12	N/A
ETSI	oneM2M	Eclipse OM2M architecture	Eclipse Foundation (FND, BE)	EPL-2.0	2021-02	2023-02
ETSI	DVB-T	VLC Media Player	VideoLAN (NPO, FR)	GPL-2.0-or-later (with some of the libraries provided under LGPL-2.1-or-later)	2022-11	2023-02
IEC	IEC 61131-3	OpenPLC	Project community and contributors (N/A, N/A)	GPL-3.0-only	N/A	2019-04
IEC	IEC 61850	IEC61850bean	Beanit GmbH (SME, DE)	Apache-2.0	N/A	2021-05
IEC	IEC 61499	Eclipse 4diac	Eclipse Foundation (FND, BE)	EPL-2.0	2021-12	2023-02
IEEE	IEEE 802.3	Linux kernel	Linux Foundation (FND, US)	GPL-2.0-only (with Linux-syscall-note)	2023-02	2023-02
IEEE	IEEE 802.11	hostapd	Independent developer (self-employed, FI)	BSD-3-Clause	2022-01	2023-02
IEEE	IEEE 1588	White Rabbit	CERN (RO, CH)	Different OSS- and OSH licences (including GPL-2.0-only, LGPL-2.1-only, CERN-OHL-1.1, and CERN-OHL-1.2). More specifically, ppsi implements IEEE 1588-2019 and claims to be licensed under GPL-2.0-only or LGPL-2.1 (depending on configuration).	N/A	2022-11 (ppsi)
IETF	RFC 6101	OpenSSL	OpenSSL Software Foundation (FND, US)	Apache-2.0	2023-02	2023-02
IETF	RFC 7540	Apache HTTP Server	Apache Software Foundation (FND, US)	Apache-2.0	2023-02	2023-02
IETF	RFC 1035	BIND 9	Internet Systems Consortium (NPO, US)	MPL-2.0	2023-02	2023-02
ISO	ISO/IEC 26300-1	LibreOffice	The Document Foundation (FND, DE)	MPL-2.0	2023-02	2023-02
ISO	ISO/IEC 14882	GNU Compiler Collection	Free Software Foundation (FND, US)	GPL-3.0-with-GCC-exception	2022-08	2023-02
ISO	ISO/IEC 13818-3	LAME	Project community and contributors (N/A, N/A)	LGPL-2.0-only	2017-10	2021-06
ITU-T	ITU-T H.264	x264	VideoLAN (NPO, FR)	GPL-2.0-or-later (and proprietary licensing conditions can also be offered)	2023-01	2023-01
ITU-T	ITU-T H.265	x265	MulticoreWare Inc. (LE, US)	GPL-2.0-or-later (and proprietary licensing conditions can also be offered)	2021-03	2023-02



ITU-T	ITU-T G.722	FFmpeg	Project community and contributors (N/A, N/A)	LGPL-2.1-or-later. However, FFmpeg incorporates optional parts and optimizations that are provided under GPL-2.0-or-later, and “if those parts get used the GPL applies to all of FFmpeg”	2023-02	2023-02
OASIS	CMIS	Alfresco	Alfresco Software Inc. (LE, US)	LGPL-3.0-only	2022-12	2023-02
OASIS	OData	Apache Olingo	Apache Software Foundation (FND, US)	Apache-2.0	2023-02	2023-02
OASIS	SAML	OpenSAML	Shibboleth consortium (NPO, US)	Apache-2.0	2023-01	2023-02
W3C	XML	Apache Xerces	Apache Software Foundation (FND, US)	Apache-2.0	2022-01	2022-07
W3C	SVG	Inkscape	Project community and contributors (N/A, N/A)	GPL-2.0-or-later	2022-01	2023-02
W3C	PNG	libpng	Project community and contributors (N/A, N/A)	libpng-2.0	2022-11	2023-02





### **Funding**

This research has been financially supported by the Swedish Knowledge Foundation (KK-stiftelsen) and participating partner organisations in the SUDO project.

### **Data/Software Access Statement**

The data is available in accordance with reference citations.

### **Contributor Statement**

Conceptualization: J.G., Data curation: J.G., Analysis: J.G., B.L., C.B., T.P., A.M., T.G., J.F., Investigation J.G., Methodology: J.G., Project administration: J.G., Writing – original draft: J.G., Writing – review & editing: J.G., B.L.

### **Acknowledgements**

The authors are grateful for the stimulating collaboration and support from colleagues and partner organisations in the SUDO project.

### **Conflict Of Interest (COI)**

The authors declare no conflict of interest.