**OpenChain Specification**

Version 2.0

* Translation Draft Englisch / Deutsch –

Vertrauen in Open Source schaffen, aus welcher Softwarelösungen entstehen

Inhaltsverzeichnis

Copyright © 2016-2019 Linux Foundation. Dieses Dokument ist unter der Creative Commons Attribution 4.0 International (CC-BY 4.0) Lizenz lizenziert. Eine Kopie des Lizenztexts finden Sie unter https://creativecommons.org/licenses/by/4.0/.

Einleitung

|  |  |  |  |
| --- | --- | --- | --- |
| Openchain-Spec 1.2 | DRAFT OCSpec 2.0 02/2019 |  |  |
| The OpenChain Initiative began in 2013 when a group of software supply chain open source practitioners observed two emerging patterns: 1) significant process similarities existed among organizations with mature open source compliance programs; and 2) there still remained a large number of organizations exchanging software with less developed programs. The latter observation resulted in a lack of trust in the consistency and quality of the Compliance Artifacts accompanying the software being exchanged. As a consequence, at each tier of the supply chain, downstream organizations were frequently redoing the compliance work already performed by other upstream organizations.  A study group was formed to consider whether a standard program specification could be created that would: i) facilitate greater quality and consistency of open source compliance information being shared across the industry; and ii) decrease the high transaction costs associated with open source resulting from compliance rework. The study group evolved into a work group, and in April 2016, formally organized as a Linux Foundation collaborative project.  The Vision and Mission of the OpenChain Initiative are as follows:   * Vision: A software supply chain where free/open source software (FOSS) is delivered with trustworthy and consistent compliance information. * Mission: Establish requirements to achieve effective management of free/open source software (FOSS) for software supply chain participants, such that the requirements and associated collateral are developed collaboratively and openly by representatives from the software supply chain, open source community, and academia.   In accordance with the Vision and Mission, this specification defines a set of requirements that if met, would significantly increases the probability that an open source compliance program had achieved a sufficient level of quality, consistency and completeness; although a program that satisfies all the specification requirements does not guarantee full compliance. The requirements represent a base level (minimum) set of requirements a program must satisfy to be considered OpenChain Conforming. The specification focuses on the “what” and “why” qualities of a compliance program as opposed to the “how” and “when” considerations. This ensures a practical level of flexibility that enables different organizations to tailor their policies and processes to best fit their objectives.  Section 2 introduces definitions of key terms used throughout the specification. Section 3 presents the specification requirements where each one has a list of one or more Verification Materials. They represent the evidence that must exist in order for a given requirement to be considered satisfied. If all the requirements have been met for a given program, it would be considered OpenChain Conforming in accordance with version 1.2 of the specification. Verification Materials are not intended to be public, but could be provided under NDA or upon private request from the OpenChain organization to validate conformance.  Additional clarification on how to interpret the specification can be obtained by reviewing the Specification Frequently Asked Questions (FAQs) located at: <https://www.openchainproiect.org/specification-faq> | The following specification defines a core set of requirements a quality Open Source compliance program is expected to satisfy. The objective is to provide a benchmark that builds trust between organizations exchanging software with regard to the use of open source. Specification conformance ensures an organization delivers a quality set of compliance artifacts (e.g., legal notices, source code ...) required to achieve license compliance for the open source software from which their solutions are comprised. The specification focuses on the “what” and “why” qualities of a compliance program as opposed to the “how” and “when” considerations. This ensures a level of flexibility that enables different organizations to tailor their policies and processes that best fit their size, goals and scope. For instance, the scope of a compliance program may support a single product line or the entire organization.  Section 2 introduces definitions of key terms used throughout the specification. Section 3 presents the specification requirements where each includes a list of one or more Verification Material items. The Verification Materials represent the evidence that must exist in order for a given requirement to be considered satisfied. If all the specification requirements have been met, the compliance program would be considered OPENCHAINconforming. Although it is not required for the Verification Materials to be made public, an organization might choose to provide them under NDA or upon request.  Additional clarification on how to interpret the specification can be obtained by reviewing the Specification Frequently Asked Questions (FAQs) located at: https://www.openchainproject.org/specification-faq | This specification defines the key requirements of a quality open source license compliance program. The objective is to provide a benchmark that builds trust between organizations exchanging software solutions comprised of open source software. Specification conformance provides assurance that a Program has been designed to produce the required Compliance Artifacts (i.e., legal notices, source code and so forth) for each software solution. The OpenChain Specification focuses on the “what” and “why” aspects of a Program rather than the “how” and “when”. This ensures flexibility for different organizations of different sizes in different markets to choose specific policy and process content that fits their size, goals and scope. For instance, an OpenChain Conformant Program may address a single product line or the entire organization.  This introduction provides the context for all potential users. Section 2 defines key terms used throughout the Specification. Section 3 defines the requirements that a Program must satisfy to achieve conformance. A requirement consists of one or more Verification Materials (i.e., records) that must be produced to satisfy the requirement. Verification Materials are not required to be made public, though an organization may choose to provide them to others, potentially under a Non-Disclosure Agreement (NDA).  The Specification is developed as an open initiative with feedback received from over 150 contributors. Insight into its historical development can be obtained by reviewing the Specification mailing list and Frequently Asked Questions (FAQs). | Zusätzliche, klarstellende Informationen zur Auslegung der Spezifikation können dem Spezifikations-FAQ unter <https://www.openchainproject.org/specification-faq> entnommen werden. |

Definitionen

|  |  |  |  |
| --- | --- | --- | --- |
| **Compliance Artifacts** - a collection of artifacts which represent the output of the FOSS management program for a Supplied Software release. The collection may include (but are not limited to) one or more of the following: source code, attribution notices, copyright notices, copy of licenses, modification notifications, written offers, FOSS component bill of materials, SPDX documents and so forth.  **FOSS (Free and Open Source Software)** - software subject to one or more licenses that meet the Open Source Definition published by the Open Source Initiative (OpenSource.org) or the Free Software Definition (published by the Free Software Foundation) or similar license.  **FOSS Liaison** - a designated person who is assigned to receive external FOSS inquires.  **Identified Licenses** - a set of FOSS licenses identified as a result of following an appropriate method of identifying licenses that govern the Supplied Software.  **OpenChain Conforming** **Program** - a program that satisfies all the requirements of this specification.  **Software Staff** - any employee or contractor that defines, contributes to or has responsibility for preparing Supplied Software. Depending on the organization, that may include (but is not limited to) software developers, release engineers, quality engineers, product marketing and product management.  **SPDX or Software Package Data Exchange** - the format standard created by the SPDX Working Group for exchanging license and copyright information for a given software package. A description of the SPDX specification can be found at [www.spdx.org](http://www.spdx.org).  **Supplied Software** - software that an organization delivers to third parties (e.g., other organizations or individuals).  **Verification Materials** - evidence that must exist in order for a given requirement to be considered satisfied. | **Compliance Artifacts -** a collection of artifacts which represent the output of the Open Source compliance program for a Supplied Software release. The collection may include (but are not limited to) one or more of the following: source code, attribution notices, copyright notices, copy of licenses, modification notifications, written offers, Open Source component bill of materials, SPDX documents and so forth.  **Open Source Software (Open Source) -** software subject to one or more licenses that meet the Open Source Definition published by the Open Source Initiative (OpenSource.org) or the Free Software Definition (published by the Free Software Foundation) or similar license.  **~~Open Source Liaison-~~** ~~a designated person who is assigned to receive external Open Source inquiries.~~  **Identified Licenses** - a set of Open Source Software licenses identified as a result of following an appropriate method of identifying Open Source components from which the Supplied Software is comprised.  **OpenChain Conforming Program (Program)** - a program that satisfies all the requirements of this specification.  **Software Staff** - any employee or contractor that defines, contributes to or has responsibility for preparing Supplied Software. Depending on the organization, that may include (but is not limited to) software developers, release engineers, quality engineers, product marketing and product management.  **SPDX or Software Package Data Exchange** - the format standard created by the SPDX Working Group for exchanging license and copyright information for a given software package. A description of theSPDX specification can be found at [www.spdx.org](http://www.spdx.org).  **Supplied Software** - software that an organization distributes to third parties (e.g., other organizations or individuals).  **Verification Materials** -evidence that must exist in order for a given requirement to be considered satisfied**.** | “Compliance Artifacts”-a collection of artifacts that represent the output of the Program for the Supplied Software. The collection may include (but is not limited to) one or more of the following: source code, attribution notices, copyright notices, copy of licenses, modification notifications, written offers, Open Source component bill of materials, and SPDX documents.  “Identified Licenses”-a set of Open Source Software licenses identified as a result of following an appropriate method of identifying Open Source components from which the Supplied Software is comprised.  “OpenChain Conformant” - a Program that satisfies all the requirements of this specification.  “Open Source”-software subject to one or more licenses that meet the Open Source Definition published by the Open Source Initiative (OpenSource.org) or the Free Software Definition (published by the Free Software Foundation) or similar license.  “Program” – the set of policies, processes and personnel that manage an organization’s Open Source license compliance activities.  “Software Staff” - any organization employee or contractor that defines, contributes to or has responsibility for preparing Supplied Software. Depending on the organization, that may include (but is not limited to) software developers, release engineers, quality engineers, product marketing and product management.  “SPDX” - the format standard created by the Linux Foundation’s SPDX (Software Package Data Exchange) Working Group for exchanging license and copyright information for a given software package. A description of the SPDX specification can be found at www.spdx.org.“Supplied Software”-software that an organization distributes to third parties (e.g., other organizations or individuals).  “Verification Materials” - materials that demonstrate that a given requirement is satisfied. | **Compliance-Artefakte –** Eine Zusammenstellung von Artefakten, die für eine Version Zugelieferter Software das Arbeitsergebnis des Open-Source-Compliance-Programms darstellen.  Die Zusammenstellung kann eines oder mehrere der folgenden Elemente enthalten (ist aber nicht auf diese beschränkt): Quellcode, Benennung des Autors, Urheberrechtshinweise, Kopien der Lizenzbedingungen, Bearbeitungshinweise, schriftliche Angebote, eine Open-Source-Komponenten-Stückliste („Bill of Materials“ bzw. „BoM"), SPDX-Dokumente, etc.  **Open Source Software (Open Source)** - Software, die einer oder mehreren Lizenzen unterliegt, die den Anforderungen der Open Source Definition der Open Source Initiative (OpenSource.org) oder der Free Software Definition der Free Software Foundation entsprechen.  **~~Open Source Liaison~~** ~~- eine konkrete Person, die für den Erhalt externer Open-Source-Anfragen bestimmt wird.~~  **Identifizierte Lizenzen** - eine Reihe von Open-Source-Softwarelizenzen, die aufgrund einer geeigneten Methode zur Identifizierung derjenigen Open-Source-Lizenzen, die einer Zugelieferten Software zugrunde liegen, identifiziert wurden.  **OpenChain Conforming Program (Programm)** - ein Programm, das alle Anforderungen dieser Spezifikation erfüllt.  **Software-Mitarbeiter** - jeder Mitarbeiter oder Auftragnehmer, der die Vorgaben für zugelieferte Software festlegt, zu ihr beiträgt oder für ihre Vorbereitung verantwortlich ist. Abhängig von der jeweiligen Organisation sind dies insbesondere Software-Entwickler, Release-Ingenieure, Qualitätsprüfer, Produkt-Marketing und Produkt-Management.  **SPDX oder Software Package Data Exchange** - der von der SPDX-Arbeitsgruppe erstellte Format-Standard für den Austausch von Lizenz- und Urheberrechtsinformationen für ein bestimmtes Softwarepaket. Eine Beschreibung der SPDX-Spezifikation finden Sie unter [www.spdx.org](http://www.spdx.org).  **Zugelieferte Software** - Software, die eine Organisation an Dritte weitergibt (z. B. andere Organisationen oder Einzelpersonen).  **Verifikationsmaterial** - Nachweise, die vorhanden sein müssen, damit eine bestimmte Anforderung als erfüllt angesehen werden kann. |

Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Goal 1: Know Your FOSS Responsibilities | **1.0 Program Foundation** | Part 1.0: Program Foundation | 1.0 Programm-Grundlagen |
| 1.1 A written FOSS policy exists that governs FOSS license compliance of the Supplied Software distribution. The policy must be internally communicated.  Verification Material(s):  1.1.1 A documented FOSS policy.  1.1.2 A documented procedure that makes Software Staff aware of the existence of the FOSS policy (e.g., via training, internal wiki, or other practical communication method).  Rationale:  To ensure steps are taken to create, record and make Software Staff aware of the existence of a FOSS policy. Although no requirements are provided here on what should be included in the policy, other sections may impose requirements on the policy.  ~~1.2 Mandatory FOSS training for all Software Staff exists such that:~~  ~~The training, at a minimum, covers the following topics:~~   * ~~The FOSS policy and where to find a copy;~~ * ~~Basics of Intellectual Property law pertaining to FOSS and FOSS licenses;~~ * ~~FOSS licensing concepts (including the concepts of permissive and copyleft licenses);~~ * ~~FOSS project licensing models;~~ * ~~Software Staff roles and responsibilities pertaining to FOSS compliance specifically and the FOSS policy in general; and~~ * ~~Process for identifying, recording and/or tracking of FOSS components contained in Supplied Software.~~   ~~Software Staff must have completed FOSS training within the last 24 months to be considered current (“Currently Trained“). A test may be used to allow Software Staff to satisfy the training requirement.~~  Verification Material(s):  ~~1.2.1 FOSS training materials covering the above topics exists (e.g., slide decks, online course, or other training materials).~~  ~~1.2.2 Documented method for tracking the completion of the training for the Software Staff.~~  ~~1.2.3 At least 85% of the Software Staff are Currently Trained, as per the definition above. The 85% may not necessarily refer to the entire organization, but to the totality Software Staff governed by the OpenChain Conforming program.~~  Rationale:  ~~To ensure the Software Staff have recently attended FOSS training and that a core set of relevant FOSS topics were covered in the training. The intent is to ensure a core base level set of topics are covered but a typical training program would likely be more comprehensive than what is required here.~~  1.3 A process exists for reviewing the Identified Licenses to determine the obligations, restrictions and rights granted by each license.  Verification Material(s):  1.3.1 A documented procedure to review and document the obligations, restrictions and rights granted by each Identified License ~~governing the Supplied Software.~~  Rationale:  To ensure a process exists for reviewing and identifying the license obligations for each Identified License for the various use cases. | **1.1 Policy**  **A written Open Source policy exists that governs Open Source license compliance of the Supplied Software. The policy must be internally communicated.**  **Verification Material(s)**:  1.1.1 A documented Open Source policy.  1.1.2 A documented procedure that makes Software Staff aware of the existence of the Open Source policy (e.g., via training, internal wiki, or other practical communication method).  **Rationale**:  To ensure steps are taken to create, record and make Software Staff aware of the existence of an Open Source policy. Although no requirements are provided here on what should be included in the policy, other sections may impose requirements on the policy.  **1.2 Competence**  **The organization shall:**   * **Identify the roles and the corresponding responsibilities of those roles that affects the performance and effectiveness of the Program;** * **Determine the necessary competence of person(s) fulfilling each role** * **Ensure that these persons are competent on the basis of appropriate education, training, and/or experience;** * **Where applicable, take actions to acquire the necessary competence** * **Retain appropriate documented information as evidence of competence**   **Verification Material(s):**  1.2.1 A documented list of roles with corresponding responsibilities for the different participates in the Open Source compliance program;  1.2.2 A documented that identifies the competencies for each role  1.2.3 Documented evidence of assessed competence for each program participant  **Rationale**:  To ensure that the program participants have obtain a sufficient level of competence for their respected roles and responsibilities.  1.3 Awareness  The organization shall ensure that persons doing work are aware of:  a) The Open Source policy;  b) Relevant Open Source objectives;  c) Their contribution to the effectiveness of the Open Source compliance program;  d) The implications of not conforming to the Open source compliance program requirements.  Verification Material(s):  1.3.1 Documented evidence of assessed awareness for each program participant including compliance program’s open source objectives, ones contribution within the program and implications of non-conformance.  Rationale:  To ensure program participants have obtain a sufficient level of awareness for their respected roles and responsibilities within the compliance program.  1.4 Program Scope  Different compliance programs may be governed by different levels of scope. For example, a program could govern a single product line, an entire department or an entire organization. The scope designation needs to be declared for each program seeking conformance.  Verification Material(s):  1.4.1 A written statement that clearly defines the scope of the program.  Rationale:  To provide the flexibility to construct a compliance program that best fits the scope of an organization’s needs. Some organizations could choose to maintain a compliance program for a specific product line while others could choose the program scope to govern software releases of the entire organization. Large organizations may prefer the former example while smaller organizations may prefer the latter.  1.5 License Obligations  A process exists for reviewing the Identified Licenses to determine the obligations, restrictions and rights granted by each license.  Verification Material(s):  1.5.1 A documented procedure to review and document the obligations, restrictions and rights granted by each Identified License.  Rationale:  To ensure a process exists for reviewing and identifying the license obligations for each Identified License for the various use cases an organization may encounter. | **1.1 Policy**  A written Open Source policy exists that governs Open Source license compliance of the Supplied Software. The policy must be internally communicated.  **Verification Material(s):**  1.1.1 A documented Open Source policy.  1.1.2 A documented procedure that makes Software Staff aware of the existence of the Open Source policy (e.g., via training, internal wiki, or other practical communication method).  **Rationale:**  To ensure steps are taken to create, record and make Software Staff aware of the existence of an Open Source policy. Although no requirements are provided here on what should be included in the policy, other sections may impose requirements on the policy.  1.2 Competence  The organization shall:   * Identify the roles and the corresponding responsibilities of those roles that affects the performance and effectiveness of the Program; * Determine the necessary competence of person(s) fulfilling each role * Ensure that these persons are competent on the basis of appropriate education, training, and/or experience; * Where applicable, take actions to acquire the necessary competence; and * Retain appropriate documented information as evidence of competence.   **Verification Material(s):**  1.2.1 A documented list of roles with corresponding responsibilities for the different participants in the Program.  1.2.2 A document that identifies the competencies for each role.  1.2.3 Documented evidence of assessed competence for each Program participant.  **Rationale:**  To ensure that the identified participants fulfilling Program roles have obtained a sufficient level of competence for their respective roles and responsibilities.  1.3 Awareness  The organization shall ensure that Program participants are aware of:  a) The Open Source policy;  b) Relevant Open Source objectives;  c) Their contribution to the effectiveness of the Program; and  d) The implications of not following the Program’s requirements.  Verification Material(s):  1.3.1Documented evidence of assessed awareness for each Program personnel including the Program’s objectives, ones contribution within the Program, and implications of Program non-conformance.  Rationale:  To ensure Program personnel have obtained a sufficient level of awareness for their respective roles and responsibilities within the Program.  1.4 Program Scope  Different Programs may be governed by different levels of scope. For example, a program could govern a single product line, an entire department or an entire organization. The scope designation needs to be declared for each Program.  Verification Material(s):  1.4.1 A written statement that clearly defines the scope and limits of the Program.  Rationale:  To provide the flexibility to construct a Program that best fits the scope of an organization’s needs. Some organizations could choose to maintain a Program for a specific product line while others could implement a Program to govern the Supplied Software of the entire organization.  1.5 License Obligations  A process exists for reviewing the Identified Licenses to determine the obligations, restrictions and rights granted by each license.  Verification Material(s):  1.5.1 A documented procedure to review and document the obligations, restrictions and rights granted by each Identified License.  Rationale:  To ensure a process exists for reviewing and identifying the license obligations for each Identified License for the various use cases an organization may encounter (as defined in requirement 3.2). | 1.1 Richtlinie  Es existiert eine schriftliche Open-Source-Richtlinie, in der die Anforderungen an die Open-Source-Lizenz-Compliance der Zugelieferten Software geregelt ist. Die Richtlinie muss innerhalb des Unternehmens kommuniziert werden.  Verifikationsmaterial:  1.1.1 Eine schriftlich dokumentierte Open-Source-Richtlinie.  1.1.2 Ein dokumentiertes Verfahren, welches die Software-Mitarbeiter auf die Existenz der Open-Source-Richtlinie aufmerksam macht (z. B. über Training, internes Wiki oder eine andere im Unternehmen relevante Kommunikationsmethode).  Begründung:  Es soll sichergestellt werden, dass die notwendigen Schritte unternommen wurden, um Software-Mitarbeiter auf die Existenz der Open-Source-Richtlinie hinzuweisen. Obwohl an dieser Stelle keine inhaltlichen Vorgaben an die Open-Source-Richtlinie gestellt werden, können solche inhaltlichen Vorgaben an anderer Stelle dieser Spezifikation genannt werden.  1.2 (Fach-)Kompetenz  Das Unternehmen muss:   * Rollen und zugehörigen Verantwortlichkeiten für diejenigen Rollen identifizieren, die die Performanz und Effektivität des Programms beeinflussen. * Den notwendigen Grad an Fachkompetenz der Person(en) bestimmen, welche die jeweilige Rolle ausfüllen * Sicherstellen, dass diese Personen auf Basis einschlägiger Ausbildung, Schulung und/oder Erfahrung die notwendige Fachkompetenz besitzen * Falls notwendig: Maßnahmen ergreifen, dass die hinreichende Fachkompetenz erworben wird * Eine hinreichende Dokumentation als Beleg der Fachkompetenz aufrechterhalten.   Verifikationsmaterial:  1.2.1 eine dokumentierte Liste an Rollen inklusive zugehöriger Verantwortlichkeiten für die unterschiedlichen Programmteilnehmer;  1.2.2 Ein Dokument, welches die Kompetenzanforderungen der jeweiligen Rolle festlegt.  1.2.3 Dokumentierte Nachweise der bei jedem Programm-Teilnehmer ermittelten Fachkompetenz.  Begründung:  Es soll sichergestellt werden, dass diejenigen Teilnehmer, welche Rollen im Programm ausführen, einen für ihre jeweiligen Rollen und Verantwortlichkeiten ausreichenden Grad an Fachkompetenz erreicht haben.  1.3. Sensibilisierung  Das Unternehmen muss sicherstellen, dass Programm-Teilnehmern   1. die Open-Source-Richtlinie; 2. relevante Open-Source-Ziele; 3. ihr jeweiliger Beitrag zur Effektivität des Open-Source-Compliance-Programms; 4. die Auswirkungen einer Nichterfüllung der Programm-Anforderungen   bekannt sind.  Verifikationsmaterial:  1.3.1 Dokumentierte Nachweise der bei jedem Programm-Teilnehmer ermittelten Sensibilisierung in Bezug auf die Programmziele, ihren jeweiligen Beitrag zum Programm und der Auswirkungen einer Nichtkonformität gegenüber dem Programm.  Begründung:  Es soll sichergestellt werden, dass die Teilnehmer in Bezug auf ihre jeweiligen Rollen und Verantwortlichkeiten innerhalb des Programms einen hinreichenden Awarenessgrad erreicht haben.  1.4 Programmumfang  Für unterschiedliche Programme gelten möglicherweise unterschiedliche Definitionen zu deren Umfang. Beispielsweise könnte ein Programm sich auf eine einzelne Produktlinie, einen Unternehmensbereich oder eine gesamte Organisation beziehen. Für jedes Programm muss der Umfang festgelegt werden.  Verifikationsmaterial:  1.4.1 Eine schriftliche Erklärung, die Umfang und Abgrenzung des Programms klar definiert.  Begründung:  Es soll sichergestellt werden, dass die Flexibilität besteht, ein Programm aufzusetzen, welches den Anforderungen eines Unternehmens am besten entspricht. Einige Unternehmen könnten ein Programm für eine bestimmte Produktlinie unterhalten, während andere ein Programm zur Steuerung der Zugelieferten Software des gesamten Unternehmens einrichten könnten.  1.5 Lizenzverpflichtungen  Es besteht ein Verfahren zur Überprüfung der Identifizierten Lizenzen um die jeweiligen Rechte, Einschränkungen und Verpflichtungen zu erkennen.  Verifikationsmaterial:  1.5.1 Ein dokumentiertes Verfahren zur Überprüfung und Dokumentation der Rechte, Beschränkungen und Verpflichtungen, die durch die jeweiligen Identifizierten Lizenzen bestehen.  Begründung:  Es soll sichergestellt werden, dass ein Prozess besteht, in dem die Lizenzpflichten für die verschiedenen, im Kontext des Unternehmens möglichen Anwendungsfälle geprüft und identifiziert werden (wie auch in Anforderung 3.2 gefordert). |

|  |  |  |  |
| --- | --- | --- | --- |
| Goal 2: Assign Responsibility for Achieving Compliance | 2.0 Relevant Tasks Defined and Supported | 2.0 Relevant Tasks Defined and Supported | Ziel 2: Relevante Rollen und Verantwortlichkeiten sind definiert und werden unterstützt |
| 2.1 Identify External FOSS Liaison Function ("FOSS Liaison").  Assign individual(s) responsible for receiving external FOSS inquiries;  FOSS Liaison must make commercially reasonable efforts to respond to FOSS compliance inquiries as appropriate; and  Publicly identify a means by which one can contact the FOSS Liaison.  Verification Material(s):  2.1.1 Publicly visible identification of FOSS Liaison (e.g., via a published contact email address, or the Linux Foundation's Open Compliance Directory).  2.1.2 An internal documented procedure that assigns responsibility for receiving FOSS compliance inquiries.  Rationale:  To ensure there is a reasonable way for third parties to contact the organization with regard to FOSS compliance inquiries and that this responsibility has been effectively assigned.  2.2 Identify Internal FOSS Compliance Role(s).  Assign individual(s) responsible for managing internal FOSS compliance. The FOSS Compliance role and the FOSS Liaison may be the same individual.  FOSS compliance management activity is sufficiently resourced:   * Time to perform the role has been allocated; and * Commercially reasonable budget has been allocated.     Assign responsibilities to develop and maintain FOSS compliance policy and processes;  Legal expertise pertaining to FOSS compliance is accessible to the FOSS Compliance role (e.g., could be internal or external); and  A process exists for the resolution of FOSS compliance issues  Verification Material(s):  2.2.1 Name of persons, group or function in FOSS Compliance role(s) internally identified.  2.2.2 Identification of legal expertise available to FOSS Compliance role(s) which could be internal or external.  2.2.3 A documented procedure that assigns internal responsibilities for FOSS compliance.  2.2.4 A documented procedure for handling review and remediation of non-compliant cases.  Rationale:  To ensure certain FOSS responsibilities have been effectively assigned. | 2.1 Maintain a process to effectively respond to external Open Source inquiries. Publicly identify a means by which a third party can make an Open Source compliance inquiry.  Verification Material(s):  2.1.1Publicly visible method any third party make an Open Source compliance inquiry(e.g., via a published contact email address, or the Linux Foundation's Open Compliance Directory).  2.1.2 An internal documented procedure for responding to third party Open Source compliance inquiries.  Rationale:  To ensure there is a reasonable way for third parties to contact the organization with regard to Open Source compliance inquiries and that the organization is prepared to effectively respond.  2.2 Identify and Resource Open Source Compliance Task(s).   * Assign accountability to ensure the successful execution of Open Source compliance tasks. * Open Source compliance tasks are sufficiently resourced:   + Time to perform the tasks have been allocated; and   + Commercially reasonable budget has been allocated. * A process exists for reviewing and updating the policy and supporting tasks; * Legal expertise pertaining to Open Source compliance is accessible to those who may need such guidance; and * A process exists for the resolution of Open Source compliance issues.   Verification Material(s):  2.2.1 Document with name of persons, group or function in Open Source Compliance role(s)identified.  2.2.2 The identified roles have been properly staffed and adequate funding provided.  2.2.2 Identification of legal expertise available to address Open Source Compliance matters which could be internal or external.  2.2.3 A documented procedure that assigns internal responsibilities for Open Source compliance.  2.2.4 A documented procedure for handling the review and remediation of non-compliant cases.  Rationale:  To ensure: i) Open Source compliance responsibilities are effectively supported and resourced and ii) policies and supporting processes are regularly updated to accommodate changes in Open Source compliance best practices. | 2.1 Access  Maintain a process to effectively respond to external Open Source inquiries. Publicly identify a means by which a third party can make an Open Source compliance inquiry.  Verification Material(s):  2.1.1Publicly visible method that allows any third party to make an Open Source license compliance inquiry (e.g., via a published contact email address, or the Linux Foundation's Open Compliance Directory).  2.1.2An internal documented procedure for responding to third party Open Source license compliance inquiries.  Rationale:  To ensure there is a reasonable way for third parties to contact the organization with regard to Open Source compliance inquiries and that the organization is prepared to effectively respond.  2.2 Effectively Resourced  Identify and Resource Program Task(s):   * Assign accountability to ensure the successful execution of Program tasks. * Program tasks are sufficiently resourced:   + Time to perform the tasks have been allocated; and   + Adequate funding has been allocated. * A process exists for reviewing and updating the policy and supporting tasks; * Legal expertise pertaining to Open Source license compliance is accessible to those who may need such guidance; and * A process exists for the resolution of Open Source license compliance issues.   Verification Material(s):  2.2.1 Document with name of persons, group or function in Program role(s)identified.  2.2.2 The identified Program roles have been properly staffed and adequate funding provided.  2.2.3Identification of legal expertise available to address Open Source license compliance matters which could be internal or external.  2.2.4 A documented procedure that assigns internal responsibilities for Open Source compliance.  2.2.5 A documented procedure for handling the review and remediation of non-compliant cases.  Rationale:  To ensure: i)Program responsibilities are effectively supported and resourced and ii) policies and supporting processes are regularly updated to accommodate changes in Open Source compliance best practices. | 2.1 Erstellung und Aufrechterhaltung eines Prozesses, um auf Open-Source-Anfragen von außerhalb des Unternehmens zu reagieren. Veröffentlichung einer Schnittstelle, über die Dritte Open-Source-Compliance-Anfragen an das Unternehmen absetzen können.  Verifikationsmaterial:  2.1.1 Eine öffentlich sichtbare Bekanntgabe einer Schnittstelle, über welche Dritte eine Open-Source-Compliance-Anfrage stellen können (z. B. durch Veröffentlichen einer Kontakt-E-Mail-Adresse oder Aufnahme in das Open Compliance-Verzeichnis der Linux Foundation).  2.1.2 Ein intern dokumentiertes Verfahren, das die Verantwortung für den Empfang und die Bearbeitung von Open-Source-Compliance-Anfragen zuweist.  Begründung:  Es soll sichergestellt werden, dass Dritte eine angemessene Möglichkeit besitzen, sich mit der Organisation in Bezug auf Open-Source-Compliance-Anfragen in Verbindung zu setzen – als auch dass die Organisation darauf vorbereitet ist, tatsächlich auf dieselben zu reagieren.  2.2 Identifikation der internen Open-Source-Compliance Rolle(n) und Ausstattung dieser Rollen mit den notwendigen Ressourcen.   * Zuweisen der Verantwortlichkeiten für die erfolgreiche Bearbeitung von Open-Source-Compliance-Aufgaben. * Open-Source-Compliance-Aufgaben verfügen über ausreichende Ressourcen: * für die Ausführung der Aufgaben wurde ausreichend Zeit zur Verfügung gestellt; und * es wurde ein angemessenes finanzielles Budget zugewiesen. * Es existiert ein Prozess, für die Entwicklung und Pflege der Open-Source-Compliance-Richtlinie sowie für hierbei unterstützende Aufgaben * Juristische Expertise in Bezug auf Open-Source-Compliance ist vorhanden und für diejenigen Personen verfügbar, welche hierzu Unterstützung benötigen; und * es existiert ein Prozess für die Lösung von Open-Source-Compliance-Problemen.   Verifikationsmaterial:  2.2.1 Ein Dokument mit den Personennamen, Gruppenzugehörigkeiten oder Funktionen, denenOpen-Source-Compliance-Rolle(n) zugeordnet sind.  2.2.2 Die identifizierten Rollen sind mit ausreichenden personellen und finanziellen Ressourcen ausgestattet  2.2.3 Benennung der juristischen Expertise, die sowohl intern als auch extern zur Adressierung von Open-Source-Compliance-Themen zu Verfügung steht.  2.2.4 Ein dokumentiertes Verfahren, das interne Verantwortlichkeiten für die Open-Source-Compliance zuweist.  2.2.5 Ein dokumentiertes Verfahren zur Prüfung und Behebung von Fällen der Nichterfüllung von Open-Source-Compliance-Anforderungen.  Begründung:  Es soll sichergestellt sein, dass i) Open-Source-Compliance-Verantwortlichkeiten tatsächlich unterstützt und mit ausreichenden Ressourcen ausgestattet sind und ii) Richtlinien und unterstützende Prozesse regelmäßig aktualisiert werden, um Änderungen in den Best Practices für Open Source-Compliance zu berücksichtigen. |

|  |  |  |  |
| --- | --- | --- | --- |
| Goal 3: Review and Approve FOSS Content | 3.0 Open Source Content Review and Approval | 3.0 Open Source Content Review and Approval | 3.0: Überprüfung und Genehmigung von Open-Source-Inhalten |
| 3.1 A process exists for creating and managing a FOSS component bill of materials which includes each component (and its Identified Licenses) from which the Supplied Software is comprised.  Verification Material(s):  3.1.1 A documented procedure for identifying, tracking and archiving information about the collection of FOSS components from which a Supplied Software release is comprised.  3.1.2 FOSS component records for each Supplied Software release which demonstrates the documented procedure was properly followed.  Rationale:  To ensure a process exists for creating and managing a FOSS component bill of materials used to construct the Supplied Software. A bill of materials is needed to support the systematic review of each component’s license terms to understand the obligations and restrictions as it applies to the distribution of the Supplied Software.  3.2 The FOSS program must be capable of handling common FOSS license use cases encountered by Software Staff for Supplied Software, which may include the following use cases (note that the list is neither exhaustive, nor may all of the use cases apply):   * distributed in binary form; * distributed in source form; * integrated with other FOSS such that it may trigger copyleft obligations; * contains modified FOSS; * contains FOSS or other software under an incompatible license interacting with other components within the Supplied Software; and/or * contains FOSS with attribution requirements.   Verification Material(s):  3.2.1 A documented procedure for handling the common FOSS license use cases for the FOSS components of the Supplied Software.  Rationale:  To ensure the program is sufficiently robust to handle an organization’s common FOSS license use cases. That a procedure exists to support this activity and that the procedure is followed. | 3.1 A process exists for creating and managing a bill of materials which includes each Open Source component (and its Identified Licenses) from which the Supplied Software is comprised.  Verification Material(s):  3.1.1 A documented procedure for identifying, tracking and archiving information about the collection of Open Source components from which the Supplied Software is comprised.  3.1.2 Open Source component records for the Supplied Software which demonstrates the documented procedure was properly followed.  Rationale:  To ensure a process exists for creating and managing an Open Source component bill of materials used to construct the Supplied Software. A bill of materials is needed to support the systematic review of each component’s license terms to understand the obligations and restrictions as it applies to the distribution of the Supplied Software.  3.2 The Open Source compliance program must be capable of handling common Open Source license use cases encountered by Software Staff for Supplied Software, which may include the following use cases (note that the list is neither exhaustive, nor may all of the use cases apply):   * distributed in binary form; * distributed in source form; * integrated with other Open Source such that it may trigger copyleft obligations; * contains modified Open Source; * contains Open Source or other software under an incompatible license interacting with other components within the Supplied Software; and/or * contains Open Source with attribution requirements.   Verification Material(s):  3.2.1A documented procedure for handling the common Open Source license use cases for the Open Source components of the Supplied Software.  Rationale:  To ensure the program is sufficiently robust to handle an organization’s common Open Source license use cases. That a procedure exists to support this activity and that the procedure is followed. | 3.1Bill of Materials  A process exists for creating and managing a bill of materials that includes each Open Source component (and its Identified Licenses) from which the Supplied Software is comprised.  Verification Material(s):  3.1.1 A documented procedure for identifying, tracking, reviewing, approving, and archiving information about the collection of Open Source components from which the Supplied Software is comprised.  3.1.2 Open Source component records for the Supplied Software that demonstrates the documented procedure was properly followed.  Rationale:  To ensure a process exists for creating and managing an Open Source component bill of materials used to construct the Supplied Software. A bill of materials is needed to support the systematic review and approval of each component’s license terms to understand the obligations and restrictions as it applies to the distribution of the Supplied Software.  3.2 License Compliance  The Program must be capable of managing common Open Source license use cases encountered by Software Staff for Supplied Software, which may include the following use cases (note that the list is neither exhaustive, nor may all of the use cases apply):   * distributed in binary form; * distributed in source form; * integrated with other Open Source such that it may trigger copyleft obligations; * contains modified Open Source; * contains Open Source or other software under an incompatible license interacting with other components within the Supplied Software; and/or * contains Open Source with attribution requirements.   Verification Material(s):  3.2.1 A documented procedure for handling the common Open Source license use cases for the Open Source components of the Supplied Software.  Rationale:  To ensure the program is sufficiently robust to handle an organization’s common Open Source license use cases. That a procedure exists to support this activity and that the procedure is followed. | 3.1 Es existiert ein Prozess zum Erstellen und Verwalten einer Bill of Materials, die jede Open-Source-Komponente (und ihre Identifizierten Lizenzen) enthält, aus der sich sie Zugelieferte Software zusammensetzt.  Verifikationsmaterial:  3.1.1 Ein dokumentiertes Verfahren zur Identifizierung, Nachverfolgung und Archivierung von Informationen über die Zusammensetzung von Open-Source-Komponenten, aus denen eine Version Zugelieferter Software besteht.  3.1.2 Eine Aufzeichnung der Open-Source-Komponenten für jede Version Zugelieferter Software, welche nachweist, dass die dokumentierte Prozedur ordnungsgemäß befolgt wurde.  Begründung:  Es soll sichergestellt werden, dass ein Prozess zum Erstellen und Verwalten einer Bill of Materials der Open-Source-Komponenten existiert, anhand dessen die Zugelieferte Software erstellt wird. Die Bill of Materials ist erforderlich, um systematisch die Lizenzbedingungen jeder Komponente mit dem Ziel zu überprüfen, die Lizenzpflichten und -bedingungen mit Blick auf die Verbreitung der Zugelieferten Software zu ermitteln.  3.2 Das Open-Source-Managementprogramm muss es ermöglichen, die üblichen Anwendungsfälle von Open-Source-Lizenzen in Zugelieferter Software abzudecken. Zu den üblichen Fällen zählen dabei insbesondere (beachten Sie allerdings, dass die Liste weder erschöpfend ist, noch alle Anwendungsfälle auf Sie Anwendung finden müssen):   * Verbreitung in Binärform; * Verbreitung in Sourcecode-Form; * Integration mit anderer Open-Source-Software, so dass die Voraussetzungen des Copyleft vorliegen können; * Enthält bearbeitete Open-Source-Software; * Enthält Open-Source-Software oder andere Software unter einer inkompatiblen Lizenz, die mit anderen Komponenten innerhalb der Zugelieferten Software interagiert; und / oder * Enthält Open-Source-Software mit Verpflichtungen hinsichtlich einer Nennung der Urheberschaft.   Verifikationsmaterial:  3.2.1 Ein dokumentiertes Verfahren, welches es ermöglicht, die üblichen Anwendungsfälle von Open-Source-Lizenzen für die Open-Source-Komponenten von Zugelieferter Software abzudecken.  Begründung:  Es soll sichergestellt werden, dass das Programm ausreichend robust ist, um die üblichen Anwendungsfälle von Open-Source-Lizenzen einer Organisation zu behandeln. Es muss gewährleistet sein, dass ein Verfahren zur Unterstützung dieser Tätigkeit besteht und dass die vorgesehene Prozedur befolgt wird. |

|  |  |  |  |
| --- | --- | --- | --- |
| Goal 4: Deliver FOSS Content Documentation and Artifacts | 4.0 Compliance Artifact Creation and Delivery | 4.0 Compliance Artifact Creation and Delivery | Ziel 4: Erzeugung und Bereitstellung von Compliance-Artefakten |
| 4.1 A process exists for creating the set of Compliance Artifacts for each Supplied Software release.  Verification Material(s):  4.1.1 A documented procedure that ensures the Compliance Artifacts are prepared and distributed with Supplied Software release as required by the Identified Licenses.  4.1.2 Copies of the Compliance Artifacts of the Supplied Software release are archived and easily retrievable, and the archive is planned to exist for at least as long as the Supplied Software is offered or as required by the Identified Licenses (whichever is longer).  Rationale:  To ensure the complete collection of Compliance Artifacts accompany the Supplied Software as required by the Identified Licenses ~~that govern the Supplied Software~~ along with other reports created as part of the FOSS review process. | 4.1 A process exists for creating the set of Compliance Artifacts for the Supplied Software.  Verification Material(s):  4.1.1 A documented procedure that ensures the Compliance Artifacts are prepared and distributed with the Supplied Software as required by the Identified Licenses.  4.1.2 A documented procedure for archiving copies of the Compliance Artifacts of the Supplied Software -where the archive is planned to exist for at least 12 months since the last offer of the Supplied Software or as required by the Identified Licenses (whichever is longer). Records exist that demonstrate the procedure has been properly followed.  Rationale:  To ensure the complete collection of Compliance Artifacts accompany the Supplied Software as required by the Identified Licenses along with other reports created as part of the Open Source review process. | 4.1 Compliance Artifacts  A process exists for creating the set of Compliance Artifacts for the Supplied Software.  Verification Material(s):  4.1.1 A documented procedure that documents the process under which the Compliance Artifacts are prepared and distributed with the Supplied Software as required by the Identified Licenses.  4.1.2 A documented procedure for archiving copies of the Compliance Artifacts of the Supplied Software -where the archive is planned to exist for a reasonable period of time since the last offer of the Supplied Software; or as required by the Identified Licenses (whichever is longer). Records exist that demonstrate the procedure has been properly followed.  Rationale:  To ensure reasonable commercial efforts have been instituted in the preparation of the Compliance Artifacts that accompanies the Supplied Software, as required by the Identified Licenses. | 4.1 Es existiert ein Prozess, um die Compliance-Artefakte für jede Version einer Zugelieferten Software zu erstellen.  Verifikationsmaterial:  4.1.1 Ein dokumentiertes Verfahren, welches sicherstellt, dass die Compliance-Artefakte mit jeder Version Zugelieferter Software entsprechend den Anforderungen der Identifizierten Lizenzen zusammengestellt und verteilt werden.  4.1.2 Ein dokumentiertes Verfahren, um Kopien der Compliance-Artefakte Zugelieferten Software zu archivieren – wobei geplant sein muss, dass das Archiv mindestens 12 Monate länger besteht, wie die Zugelieferte Software angeboten wird bzw. mindestens so lange, wie es die Identifizierten Lizenzen verlangen (je nachdem, welcher Zeitraum länger ist).  Begründung:  Es soll sichergestellt werden, dass die vollständigen Compliance-Artefakte entsprechend den Anforderungen der Identifizierten Lizenzen, sowie sonstige Berichte, die während der Open-Source-Überprüfung erstellt wurden, mit jeder Version der Zugelieferten Software ausgeliefert werden. |

|  |  |  |  |
| --- | --- | --- | --- |
| Goal 5: Understand FOSS Community Engagement | 5.0 Understand Open Source Community Engagement | 5.0 Understanding Open Source Community Engagement | Ziel 5: Verstehen Sie Ihr Engagement gegenüber der Open Source Community |
| 5.1 A written policy exists that governs contributions to FOSS projects by the organization. The policy must be internally communicated.  Verification Material(s):  5.1.1 A documented FOSS contribution policy;  5.1.2 A documented procedure that makes all Software Staff aware of the existence of the FOSS contribution policy (e.g., via training, internal wiki, or other practical communication method).  Rationale:  To ensure an organization has given reasonable consideration to developing a policy with respect to publicly contributing to FOSS. The FOSS contribution policy can be made a part of the overall FOSS policy of an organization or be its own separate policy. In the situation where contributions are not permitted at all, a policy should exist making that position clear.  5.2 If an organization permits contributions to FOSS projects then a process must exist that implements the FOSS contribution policy outlined in Section 5.1.  Verification Material(s):  5.2.1 Provided the FOSS contribution policy permits contributions, a documented procedure exists that governs FOSS contributions.  Rationale:  To ensure an organization has a documented process for how the organization publicly contributes FOSS. A policy may exist such that contributions are not permitted at all. In that situation it is understood that no procedure may exist and this requirement would nevertheless be met. | 5.1 A written policy exists that governs contributions to Open Source projects by the organization. The policy must be internally communicated.  Verification Material(s):  5.1.1 A documented Open Source contribution policy.  5.1.2 A documented procedure that makes all Software Staff aware of the existence of the Open Source contribution policy (e.g., via training, internal wiki, or other practical communication method).  Rationale:  To ensure an organization has given reasonable consideration to developing a policy with respect to publicly contributing to Open Source. The Open Source contribution policy can be made a part of the overall Open Source policy of an organization or be its own separate policy. In the situation where contributions are limited or not permitted at all, a policy should exist making that position clear.  5.2 If an organization permits contributions to Open Source projects then a process exists that implements the Open Source contribution policy outlined in Section 5.1.  Verification Material(s):  5.2.1Provided the Open Source contribution policy permits contributions, a documented procedure that governs Open Source contributions.  Rationale:  To ensure an organization has a documented process for how the organization publicly contributes Open Source. A policy may exist such that contributions are not permitted at all. In that situation it is understood that no procedure may exist and this requirement would nevertheless be met. | 5.1Contributions  If an organization permits contributions to Open Source projects then   * A written policy exists that governs contributions to Open Source projects; * the policy must be internally communicated; and * a process exists that implements the policy   Verification Material(s):  If an organization permits contributions to Open Source projects then the following mustexist:  5.1.1 a documented Open Source contribution policy;  5.1.2 a documented procedure that governs Open Source contributions; and  5.1.3 a documented procedure that makes all Software Staff aware of the existence of the Open Source contribution policy (e.g., via training, internal wiki, or other practical communication method).  Rationale:  When an organization permits Open Source contributions, we want to ensure the organization has given reasonable consideration to developing and implementing a contribution policy. The Open Source contribution policy can be made a part of the overall Open Source policy or be its own separate policy. | 5.1 Es gibt eine schriftliche Richtlinie, die die Beiträge zu Open-Source-Projekten durch die Organisation regelt. Die Richtlinie muss intern kommuniziert werden.  Verifikationsmaterial:  5.1.1 Eine dokumentierte Richtlinie für Beiträge zu Open Source;  5.1.2 Ein dokumentiertes Verfahren, welches alle Software-Mitarbeiter auf die Existenz der Richtlinie für Beiträge zu Open Source aufmerksam macht (z. B. mittels Training, ein internes Wiki oder andere praktische Kommunikationsmethode).  Begründung:  Es soll sichergestellt werden, dass die Organisation der Entwicklung einer Richtlinie für öffentliche Beiträge zu Open Source eine ausreichende Beachtung geschenkt hat. Die Richtlinie für Beiträge zu Open Source kann Teil einer übergreifenden Open-Source-Richtlinie oder eine eigene separate Richtlinie sein. In dem Fall, dass Beiträge zu Open Source überhaupt nicht erlaubt sind, sollte es eine Richtlinie geben, die diese Haltung klarstellt.  5.2 Wenn eine Organisation Beiträge zu Open-Source-Projekten zulässt, muss ein Prozess existieren, der die in Abschnitt 5.1 skizzierte Richtlinie für Beiträge zu Open Source umsetzt.  Verifikationsmaterial:  5.2.1 Wenn die Richtlinie Beiträge zu Open Source zulässt, muss ein dokumentiertes Verfahren existieren, anhand dessen Beiträge zu Open Source erfolgen.  Begründung:  Es soll sichergestellt werden, dass eine Organisation einen dokumentierten Prozess hat, wie sie öffentlich zu Open Source beiträgt. Es kann eine Richtlinie dergestalt bestehen, dass Beiträge gar nicht gestattet sind. Aus dieser Situation folgt zwingend, dass kein Verfahren existieren kann und, dass diese Anforderung auch ohne Verfahren erfüllt werden würde. |

|  |  |  |  |
| --- | --- | --- | --- |
| Goal 6: Certify Adherence to OpenChain Requirements | 6.0 Verify Adherence to Open Chain Requirements | 6.0 Adherence to the Specification Requirements | 6.0: Verifikation der Erfüllung der OpenChain-Anforderungen |
| 6.1 In order for an organization to be OpenChain Certified, it must affirm that it has a FOSS program that meets the criteria described in this OpenChain Specification version 1.2.  Verification Material(s):  6.1.1 An affirmation of the existence of a FOSS management program that meets all the requirements of this OpenChain Specification version 1.2.  Rationale:  To ensure that if an organization declares that it has a program that is OpenChain Conforming, that such program has met all the requirements of this specification. The mere meeting of a subset of these requirements would not be considered sufficient.  6.2 Conformance with this version of the specification will last 18 months from the date conformance validation was achieved. Conformance validation requirements can be found on the OpenChain project’s website.  Verification Material(s):  6.2.1 The organization affirms that a FOSS compliance program exists that meets all the requirements of this OpenChain Specification version 1.2 within the past 18 months of achieving conformance validation.  Rationale:  It is important for the organization to remain current with the specification if that organization wants to assert conformance over time. This requirement ensures that the program’s supporting processes and controls do not erode if the conforming organization continues to assert conformance over time. | 6.1 In order for a compliance program to be deemed Open Chain Conforming, the organization must affirm that the program meets the criteria described in this OpenChain Specification version 2.0.  Verification Material(s):  6.1.1 A document affirming the program meets all the requirements of this OpenChain Specification version 2.0.  Rationale:  To ensure that if an organization declares that it has a program that is OpenChain Conforming, that such program has met all the requirements of this specification. The mere meeting of a subset of these requirements would not beconsidered sufficient.  6.2 Conformance with this version of the specification will last 18months from the date conformance validation was obtained. The conformance validation registration procedure can be found on the OpenChain project’s website.  Verification Material(s):  6.2.1 The organization affirms the existence of a program that meets all the requirements of this OpenChain Specification version 2.0 within the past 18 months of obtaining conformance validation.  Rationale:  It is important for the organization to remain current with the specification if that organization wants to assert program conformance overtime. This requirement ensures that the program’s supporting processes and controls do not erode if an organization continues to assert program conformance over time. | 6.1Conformance  In order for a Program to be deemed OpenChain Conformant, the organization must affirm that the program satisfies the requirements presented in this specification.  Verification Material(s):  6.1.1 A document affirming the Program specified in requirement 1.4 satisfies all the requirements of this specification.  Rationale:  To ensure that if an organization declares that it has a program that is OpenChain Conforming, that such program has met all the requirements of this specification. The mere meeting of a subset of these requirements would not be considered sufficient.  6.2Duration  A Program that is OpenChain Conformant with this version of the specification will last 18 months from the date conformance validation was obtained. The conformance validation registration procedure can be found on the OpenChain project’s website.  Verification Material(s):  6.2.1 A document affirming the Program meets all the requirements of this version of the specification (version 2.0), within the past 18 months of obtaining conformance validation.  Rationale:  It is important for the organization to remain current with the specification if that organization wants to assert program conformance overtime. This requirement ensures that the program’s supporting processes and controls do not erode if an organization continues to assert program conformance overtime. | 6.1 Damit einer Organisation ein OpenChain-konformes Programm bescheinigt werden kann, muss diese bestätigen, dass ihr Programm die in dieser OpenChain-Spezifikation Version 2.0 beschriebenen Kriterien erfüllt.  Verifikationsmaterial:  6.1.1 Ein Dokument, welches das Vorhandensein eines Programms, welches alle Anforderungen dieser OpenChain Spezifikation Version 2.0 erfüllt, bestätigt  Begründung:  Es soll sichergestellt werden, dass ein Open-Source-Programm alle Anforderungen dieser Spezifikation erfüllt, wenn eine Organisation angibt, dass ihr Programm OpenChain-konform sei. Lediglich Teile der Anforderungen zu erfüllen, wird nicht als ausreichend angesehen werden.  6.2 Die Übereinstimmung mit dieser Version der Spezifikation ist ab dem Datum der Validierung der Konformität für 18 Monate gültig. Die Anforderungen der Validierung der Konformität finden Sie auf der Website des OpenChain-Projekts.  Verifikationsmaterial:  6.2.1 Die Organisation bestätigt, dass ein Open-Source-Compliance-Programm existiert, das alle Anforderungen dieser OpenChain Spezifikation Version 1.3 während der vergangenen 18 Monate seit Erreichen der Konformitäts-Validierung erfüllt.  Begründung:  Es ist wichtig, dass die Organisation auf einem aktuellen Stand bezüglich der Spezifikation bleibt, wenn sie als Organisation die Programmkonformität auf Dauer behaupten will. Diese Anforderung stellt sicher, dass die die Konformität unterstützenden Prozesse und Kontrollen des Programms nicht abgeschwächt werden, wenn eine Organisation ihre Konformität auf Dauer behaupten möchte. |

|  |  |  |  |
| --- | --- | --- | --- |
| Appendix I: Language Translations | Appendix I: Language Translations | Appendix I: Language Translations | Anhang I: Sprachübersetzungen |
| To facilitate global adoption, we welcome efforts to translate the specification into multiple languages. Because OpenChain functions as an open source project, translations are driven by those willing to contribute their time and expertise to perform translations under the terms of the CC-BY 4.0 license and the project’s translation policy. The details of the policy and available translations can be found on the OpenChain project specification webpage. | To facilitate global adoption, we welcome efforts to translate the specification into multiple languages. Because OpenChain functions as an open source project, translations are driven by those willing to contribute their time and expertise to perform translations under the terms of the CC-BY 4.0 license and the project’s translation policy. The details of the policy and available translations can be found on the OpenChain project specification webpage. | To facilitate global adoption, we welcome efforts to translate the specification into different languages. Because OpenChain functions as an open source project, translations are driven by those willing to contribute their time and expertise to perform translations under the terms of the CC-BY 4.0 license and the project’s translation policy. The details of the policy and available translations can be found on the OpenChain project specification webpage. | Um die globale Adoption zu erleichtern, begrüßen wir die Bemühungen, die Spezifikation in mehrere Sprachen zu übersetzen. Da auch die OpenChain Initiative wie ein Open Source Projekt aufgesetzt ist, werden Übersetzungen durch diejenigen gesteuert, die bereit sind, ihre Zeit und ihr Fachwissen zu Übersetzungen unter den Bedingungen der CC-BY 4.0-Lizenz und der Richtlinie des Projekts für Übersetzungen beizutragen. Die Details der Richtlinien und der verfügbaren Übersetzungen finden Sie auf der Spezifikations-Webseite des OpenChain-Projekts. |