Open-Source-Software im Unternehmen

Rechtliche Anforderungen und Risiken

Dr. Hendrik Schöttle

Osborne Clarke

Warum Compliance?

- Verbreitung von OSS bei Verletzung der OSS-Lizenzpflichten kann zu Unterlassungsansprüchen und zu Schadensersatz führen
- Einzelne Mitentwickler von systemnahen Linux-Komponenten bzw. Komponenten des Linux-Kernels greifen immer wieder Verletzungen bei diesen Softwareprodukten aus kommerziellen Gründen an
- Auch Organisationen überwachen die Einhaltung von OSS-Lizenzbedingungen, wie etwa die Software Freedom Conservancy
- Unternehmen verwenden OSS-Lizenzen, um kommerzielle, kostenpflichtige Lizenzen im Wege des Dual Licensing durchzusetzen

Verletzung von OSS-Lizenzbedingungen | Risiken

- Einhaltung von OSS-Lizenzen wird inzwischen von mehreren Organisationen überwacht und verfolgt
- Beispiele:
  - Software Freedom Law Center (SFLC)
  - gpl-violations.org
- Vermehrte Verfolgung derartiger Ansprüche in der jüngeren Vergangenheit auch von Privatpersonen

Best Practice

- Nach dem Compliance-Handbuch der Linux Foundation ist ein Open-Source-Compliance-Team mit **mindestens 4 Vollzeitstellen** erforderlich:
  - *„The core team, often called the Open Source Review Board (OSRB), consists of representatives from engineering and product teams, one or more legal counsel, and the Compliance Officer. The extended team consists of various individuals […] Unlike the core team, members of the extended team are only working on compliance on a part-time basis […].“*

Ibrahim Haddad, Open Source Compliance in the Enterprise, 2nd Edition 2018, p. 33

https://www.linuxfoundation.org/compliance-and-security/2018/12/open-source-compliance-in-the-enterprise/

Best Practice | Komponentenerfassung

Wissen Sie, welche Softwarekomponenten in Ihren Produkten enthalten sind?

- Policy zur Erfassung von Drittkomponenten aufsetzen
- Jede kopierte Zeile Source Code sowie auch in Binärform bezogene Komponenten sollten erfasst werden
- Eigentlich auch rekursive Erfassung von zusammengesetzten Komponenten erforderlich (teilweise sehr aufwändig)
- Jedes Produkt sollte erst nach Prüfung und Freigabe bezüglich OSS-Komponenten auf den Markt gebracht werden

Best Practice | Komponentenerfassung

Da eine rekursive Prüfung von Softwarekomponenten oftmals zu umfangreich ist, ggf. Scan-Software einsetzen:

- FOSSology + SW360 (OSS)
- ScanCode (OSS)
- SCANOSS (OSS)
- Synopsis Software Composition Analysis (ehem. Black Duck)
- Flexera Code Insight (ehem. Palamida)
- OSS Review Toolkit (OSS)
- Rapid Miner
- OpenLogic OSS Discovery/OSS Deep Discovery (OSS)
- WhiteSource
- Quartermaster

Problem: Diese Tools können manches nicht zuordnen (→ fehleranfällig!); Absolute Sicherheit ist durch ihren Einsatz deshalb nicht gewährleistet

Auch manuell ist es wirtschaftlich extrem aufwändig, alle Copyright-Klauseln auszusondern und gesammelt zu dokumentieren

Best Practice | Welche Lizenzfassung gilt?

Lizenzprüfung auf drei Ebenen:

- Abstrakte Lizenzprüfung (z.B. GPL-2.0)
- Konkrete Prüfung von Komponenten (z.B. Linux Kernel, lizenziert unter der GPL-2.0)
- Prüfung des Einsatzes einer konkreten Komponente im Hinblick auf Copyleft-Effekt

Best Practice | Informations- und Dokumentationspflichten

- Darstellen von
  - Urhebervermerk
  - Haftungsausschluss
  - Kopie des Lizenztextes
  - Zuordnung von Lizenz zur Softwarekomponente
- Mitlieferung oder Angebot von
  - Source Code
- Zahlreiche weitere Pflichten sind zu beachten!

Best Practice | Informations- und Dokumentationspflichten

- Bill of Materials als zentrales Dokument mit sämtlichen Lizenzinformationen und Hinweisen
- Idealerweise rekursiv und auf Datei- bzw. Verzeichnisebene
- Am besten in strukturierter Form speichern (SPDX) – wird von einigen Tools automatisch erzeugt. Daraus lassen sich dann über Parser menschenlesbare Formate erzeugen (HTML, PDF, etc.)
- Zudem ein Source Code Repository, also zentrale Speicherung sämtliche Source Codes

Best Practice | Lizenzmatrix

- Warum eine Lizenzmatrix bei der Umsetzung von Open Source Compliance?
  - Übliches Vorgehen: Raussuchen von Lizenztext und Urhebervermerken, Mitliefern dieser Texte, fertig.
  - Das reicht nicht!
  - In vielen Fällen kann OSS nur unter Verstoß gegen die zugrunde liegenden Lizenzbedingungen eingesetzt werden!
  - Warum? Entwickler nehmen einfach „die neueste Lizenz“, ohne deren Pflichten zu prüfen und ohne sich über die Folgen im Klaren zu sein.
  - Beispiel: Lizenzierung von iOS-Apps unter der GPL-3.0
  - DRM-Verbot der GPL-3.0 in Ziffer 3: Derjenige, der ein GPL-Programm vertreibt, muss auf ein Verbot zur Umgehung technischer Schutzmaßnahmen für dieses Programm verzichten.

Best Practice | Lizenzmatrix

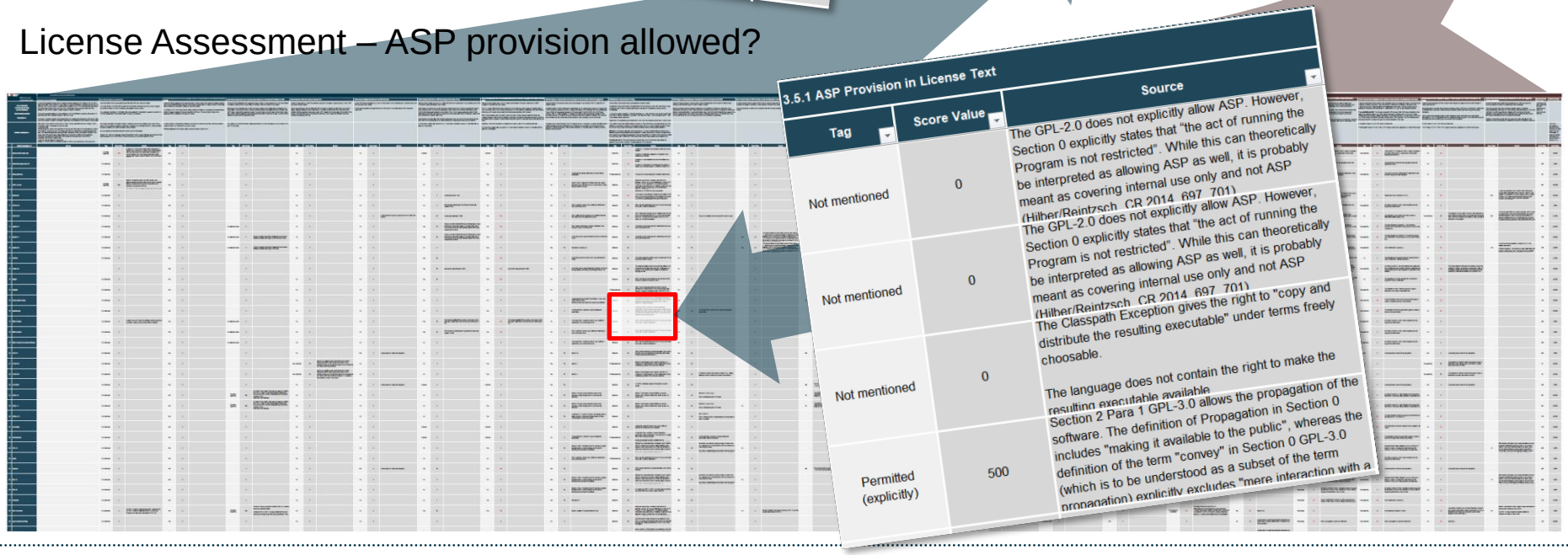
- Warum eine Lizenzmatrix bei der Umsetzung von Open Source Compliance?
  - Übliche Scanning Tools stellen oftmals lediglich Lizenztexte und Copyright-Klauseln zusammen, bieten aber keinen detaillierten und nachvollziehbaren Überblick über übrige Lizenzpflichten
  - Sämtliche Pflichten einer Lizenz müssen
    - erfasst, bewertet und dann
    - gegen eigene Verwendung abgeglichen werden.
  - Die Interpretation einzelner Lizenzen und deren Pflichten ist häufig umstritten
  - allein die binäre Darstellung eines Ergebnisses hilft bei umstrittenen Interpretationen nicht weiter

Best Practice | Lizenzmatrix

- Beispiel: Use Case „ASP-Nutzung“: Ist ein Zugänglichmachen von Software in Form von Application Service Provision (ASP/SaaS) zulässig?
  - Viele Lizenzen enthalten hierzu keine klaren Regelungen
  - Übliche Lösung: Viele Memos zu einzelnen Lizenzen. Unübersichtlich und keine Hilfe bei schnellem Überblick über Einhaltung der Pflichten abhängig von konkretem Use Case
- Unsere Lösung:
  - Aufspalten der Frage in Teilaspekte und -argumente, die dafür oder dagegen sprechen
  - Gewichtung der Aspekte mit unterschiedlichen Score-Werten und Bewertungslogiken
  - Berechnung der Score-Werte
  - Übersichtliche Darstellung
  - Vergleich mit Anwendungsszenarien des jeweiligen Unternehmens

Use Cases

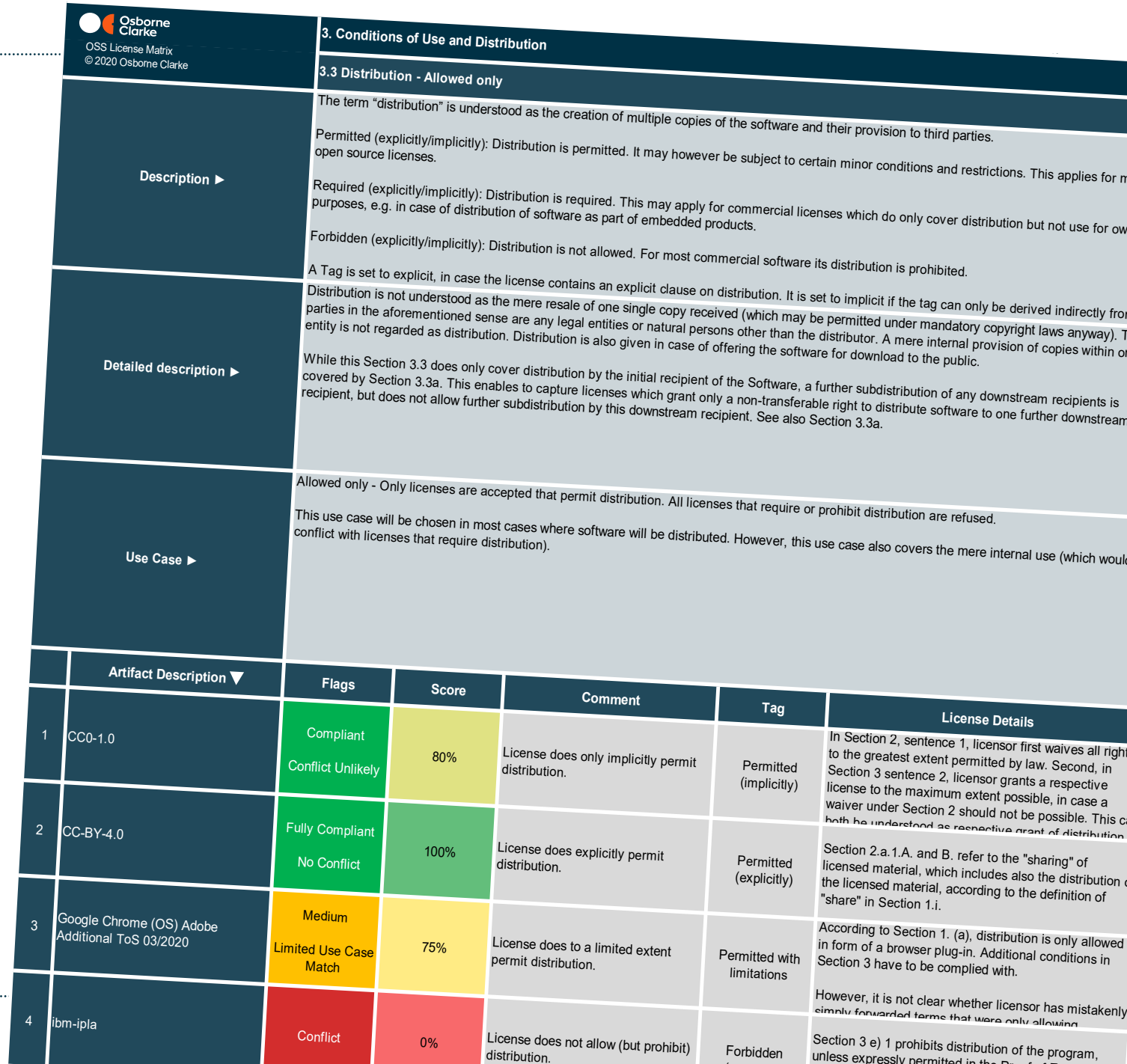
License Assessment – ASP provision allowed?



Unser Ansatz: Lizenzmatrix

Standardisierte Bewertung einzelner Lizenzpflichten

- **Auswertung von Lizenzen**, standardisiert, vollständig dokumentiert und parametrisierbar mit Prozentangaben zur automatischen Weiterverarbeitung
- Derzeit ~ 200 Lizenzen, klassifiziert nach 54 Attributen (insgesamt mit Stammdaten 72 Attribute).
- Use Case Mapping gegen die jeweiligen Lizenzen mit automatischer Konfliktprüfung



Osborne Clarke
OSS License Matrix
© 2020 Osborne Clarke

	3. Conditions of Use and Distribution
	3.3 Distribution - Allowed only
Description ▶	The term "distribution" is understood as the creation of multiple copies of the software and their provision to third parties. Permitted (explicitly/implicitly): Distribution is permitted. It may however be subject to certain minor conditions and restrictions. This applies for m open source licenses. Required (explicitly/implicitly): Distribution is required. This may apply for commercial licenses which do only cover distribution but not use for ow purposes, e.g. in case of distribution of software as part of embedded products. Forbidden (explicitly/implicitly): Distribution is not allowed. For most commercial software its distribution is prohibited. A Tag is set to explicit, in case the license contains an explicit clause on distribution. It is set to implicit if the tag can only be derived indirectly fro
Detailed description ▶	Distribution is not understood as the mere resale of one single copy received (which may be permitted under mandatory copyright laws anyway). T parties in the aforementioned sense are any legal entities or natural persons other than the distributor. A mere internal provision of copies within or entity is not regarded as distribution. Distribution is also given in case of offering the software for download to the public. While this Section 3.3 does only cover distribution by the initial recipient of the Software, a further subdistribution of any downstream recipients is covered by Section 3.3a. This enables to capture licenses which grant only a non-transferable right to distribute software to one further downstream recipient, but does not allow further subdistribution by this downstream recipient. See also Section 3.3a.
Use Case ▶	Allowed only - Only licenses are accepted that permit distribution. All licenses that require or prohibit distribution are refused. This use case will be chosen in most cases where software will be distributed. However, this use case also covers the mere internal use (which woul conflict with licenses that require distribution).

	Artifact Description ▼	Flags	Score	Comment	Tag	License Details
1	CC0-1.0	Compliant Conflict Unlikely	80%	License does only implicitly permit distribution.	Permitted (implicitly)	In Section 2, sentence 1, licensor first waives all right to the greatest extent permitted by law. Second, in Section 3 sentence 2, licensor grants a respective license to the maximum extent possible, in case a waiver under Section 2 should not be possible. This c both be understood as respective grant of distribution
2	CC-BY-4.0	Fully Compliant No Conflict	100%	License does explicitly permit distribution.	Permitted (explicitly)	Section 2.a.1.A. and B. refer to the "sharing" of licensed material, which includes also the distribution the licensed material, according to the definition of "share" in Section 1.i.
3	Google Chrome (OS) Adobe Additional ToS 03/2020	Medium Limited Use Case Match	75%	License does to a limited extent permit distribution.	Permitted with limitations	According to Section 1. (a), distribution is only allowed in form of a browser plug-in. Additional conditions in Section 3 have to be complied with. However, it is not clear whether licensor has mistakenly simply forwarded terms that were only allowing
4	ibm-ipla	Conflict	0%	License does not allow (but prohibit) distribution.	Forbidden	Section 3 e) 1 prohibits distribution of the program, unless expressly permitted in the Pi of Section

Fazit

- Compliance erfordert Scanning von Software und Erstellung von Bill of Materials. Dies geht in der Regel nur mit Tool-Unterstützung
- Eine Prüfung der übrigen Anforderungen der Lizenzen ist essenziell. Diese sollten dann mit Use Cases bezüglich des konkret geplanten Einsatzes der OSS-Komponenten abgeglichen werden. Tools können hierbei ebenfalls helfen

Kontakt

Dr. Hendrik Schöttle
Partner, Fachanwalt für IT-Recht
Germany

+49 89 5434 8046
hendrik.schoettle@osborneclarke.com

„Im Bereich Open Source ein Spitzenname“

Wettbewerber, JUVE-Handbuch 2021/2022

Dr. Hendrik Schöttle berät im IT- und Datenschutzrecht.

Hendrik Schöttle wurde in den letzten Jahren wiederholt sowohl vom Handelsblatt und von Best Lawyers als auch von der Wirtschaftswoche und vom Kanzleimonitor als einer der besten Anwälte bzw. als mehrfach empfohlener Anwalt im IT-Recht genannt. Laut JUVE-Handbuch 2021/2022 ist er „im Bereich Open Source ein Spitzenname“. Das Kanzleihandbuch Legal 500 Deutschland empfiehlt ihn, weil er durch „sehr gute IT-Kenntnisse besticht, auch wenn es sich um exotische Fragen handelt“ und durch ein „sehr schnelles Verständnis technischer Details“.

Er hat langjährige Erfahrung bei der Beratung, Vertragsgestaltung und Verhandlung von komplexen IT-Projekten. Seine Schwerpunkte sind IoT, Digitalisierung und Cloud Computing. Er berät zu Software-Lizenzmodellen, insbesondere zu Open-Source-Software, und im Datenschutzrecht. Zu seinen Mandanten gehören international tätige Technologiekonzerne sowie namhafte IT- und E-Business-Unternehmen.

Hendrik Schöttle arbeitet seit 2005 als Rechtsanwalt, seit 2007 im Münchner Büro von Osborne Clarke. Er war mehrfach im Rahmen von Secondments in Rechtsabteilungen von IT-Unternehmen tätig. Zudem hat er mehrere Jahre als Software-Entwickler am Institut für Rechtsinformatik der Universität des Saarlandes gearbeitet. Seine praktische Erfahrung und sein technisches Know-how kommen seinen Mandanten bei der technologienahen Beratung zugute.

Er ist Autor zahlreicher Veröffentlichungen, Mitautor mehrerer Handbücher und Kommentare, unter anderem des Beck’schen Handbuchs IT- und Datenschutzrecht und des juris Praxiskommentars zum BGB.

Hendrik Schöttle ist Dozent der Deutschen Anwaltakademie für den Fachanwaltslehrgang IT-Recht und hält regelmäßig Vorträge zu Themen des IT-Rechts.

Er ist Mitglied im Vorstand des Arbeitskreises Open Source des BITKOM, Mitglied des Ausschusses Datenschutzrecht der Bundesrechtsanwaltskammer (BRAK), der Arbeitsgemeinschaft Informationstechnologie im Deutschen Anwaltverein (DAV) und der Deutschen Gesellschaft für Recht und Informatik (DGRI).