Going Open Source

Wie das proprietäre Softwareprojekt
Open Source werden kann

BITKOM

Forum Open Source

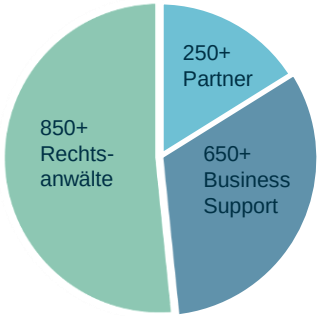
Dr. Hendrik Schöttle
Erfurt, 17. September 2019

Osborne Clarke International

mehr als

1.750

Mitarbeiter



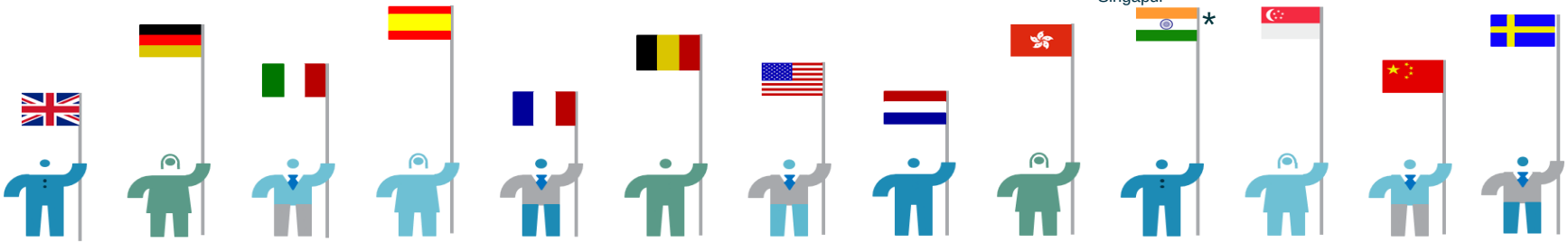
25

Internationale Standorte

Europa:
Belgien: Brüssel
Deutschland: Berlin, Hamburg, Köln, München
Frankreich: Paris
Italien: Brescia, Busto Arsizio, Mailand, Padua, Rom
Niederlande: Amsterdam
Schweden: Stockholm
Spanien: Barcelona, Madrid, Zaragoza
UK: Bristol, London, Thames Valley

USA
New York, San Francisco, Silicon Valley

Asien
China: Shanghai
Hongkong
Indien: Neu Delhi*, Mumbai*, Bangalore*
Singapur



*

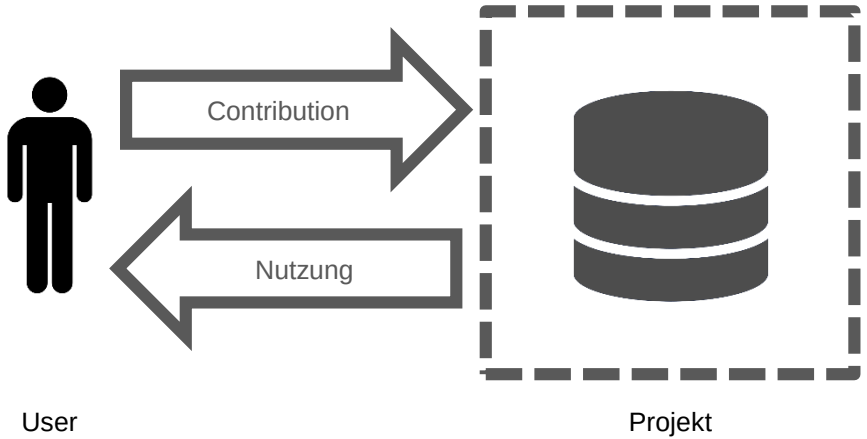
Relationship firm

Übersicht

- Projektarten
  - Übersicht
  - Beispiele
  - Offene vs. geschlossene Projekte
  - Welche Lizenz nehmen
- Compliance mit Lizenzpflichten integrierter Komponenten

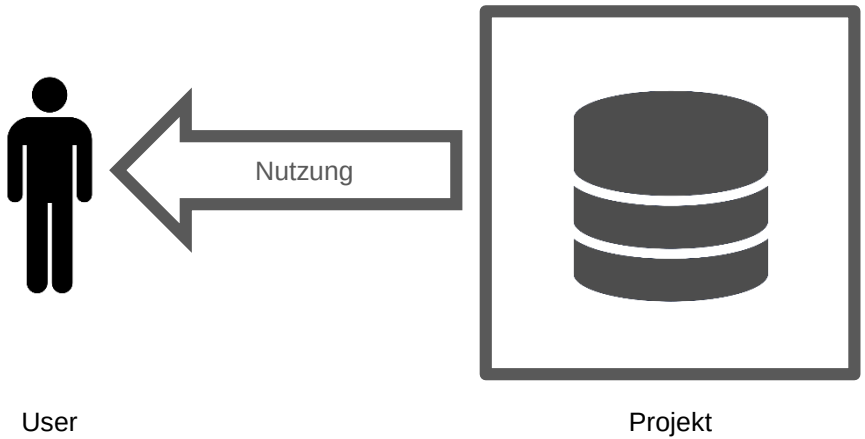
Projektarten | Übersicht

- Folgende unterschiedliche Projektarten gibt es:
  - Geschlossenes Projekt
  - Offenes Projekt
    - ohne und
    - mit Contributor License Agreement (CLA)



Projektarten | Geschlossenes Projekt

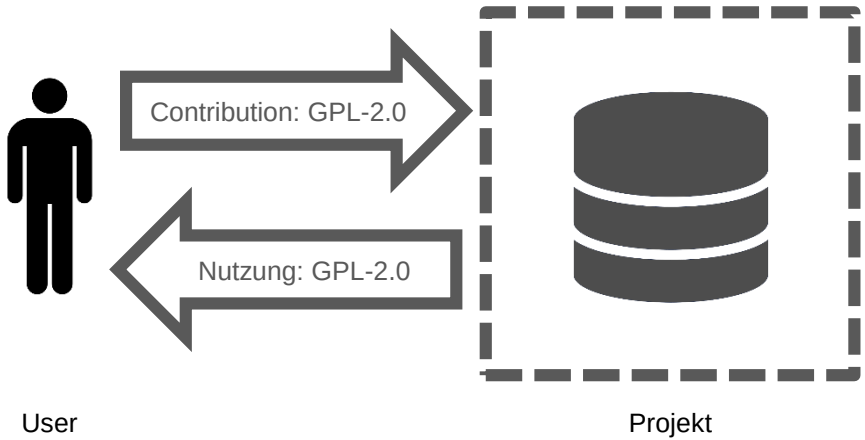
- Keine Contribution möglich (allenfalls Feature Requests und Bug Reports akzeptiert)
- Nur Nutzung möglich
- Aber: eigene Verwendung oder Forking möglich. Forking: „Neugründung“ eines Projekts auf Basis eines bestehenden



Projektarten | Offenes Projekt ohne CLA

- Contribution und Nutzung möglich, sie erfolgen unter derselben Lizenz
- Vorteile:
  - einheitliches Rechtsregime für alle
  - Kein zentrales Rechtsmanagement erforderlich
- Nachteil:
- weder Umlizenzierung noch duale Lizenzierung für Projektinhaber möglich
- Lizenzwahl gilt, falls es keine Upgrade-Klauseln gibt, für die Ewigkeit

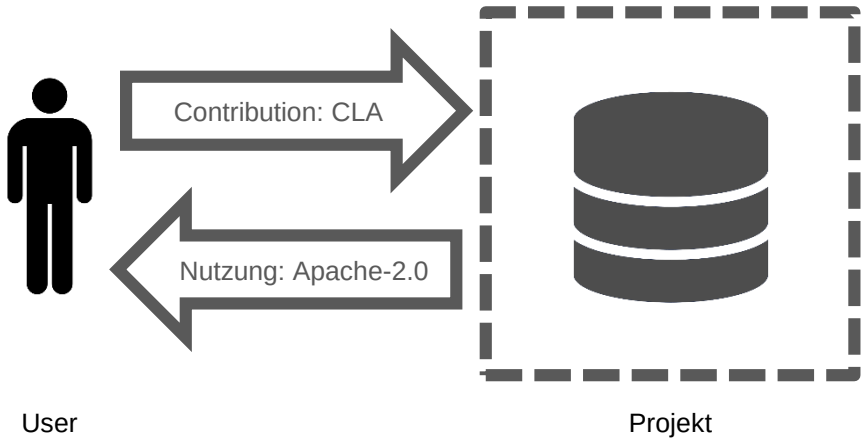
Beispiel: Linux Kernel



Projektarten | Offenes Projekt mit CLA

- Contribution und Nutzung möglich, sie erfolgen unter unterschiedlichen Lizenzen.
- In der Regel weitere Rechtseinräumung bei Contributions als umgekehrt
- Vorteile:
  - Umlizenzierung und kommerzielle Nutzung bleiben dem Projektinhaber möglich
- Nachteil:
  - zentrales Rechtsmanagement erforderlich
  - Aber: Forking ebenfalls möglich, dann aber nur unter der Nutzungs-Lizenz

Beispiel: Apache Foundation



Übersicht

- Projektarten
  - Übersicht
  - Beispiele
  - Offene vs. geschlossene Projekte
  - Welche Lizenz nehmen
- Compliance mit Lizenzpflichten integrierter Komponenten

Projektarten | Beispiele

Beispiel: Apache Software Foundation

Ziffer 1 des CLA der ASF:

Subject to the terms and conditions of this Agreement, You hereby grant to the Foundation and to recipients of software distributed by the Foundation a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

Ziffer 2 der Apache-2.0 ist ähnlich wie links.

Aber: Ziffer 4 a. Apache-2.0:

You must give any other recipients […] a copy of this License; and

Ziffer 4 Abs. 2 Apache-2.0:

You may […] provide additional or different license terms […], provided Your […] distribution […] otherwise complies with the conditions stated in this License.

Projektarten | Beispiele

Beispiel Apache Software Foundation

- Ziffer 2 CLA der ASF enthält Patent License Grant:

  Subject to the terms and conditions of this Agreement, You hereby grant to the Foundation and to recipients of software distributed by the Foundation a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s)alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) were submitted.

Projektarten | Beispiele

Aufpassen bei exklusiver Rechtseinräumung!

Beispiel: ISA Contributor Agreement v1.1, Ziffer 2.1. (b):

> *[…] You grant to Us a perpetual, worldwide, exclusive, royalty-free, transferable, irrevocable licence […]*

https://joinup.ec.europa.eu/document/isa-contributor-agreement-v11

Aber: Gleichzeitige Rücklizenzierung, siehe Ziffer 2.1 (d):

> *Upon such transfer of rights to Us, to the maximum extent possible, We immediately grant to You a perpetual, worldwide, non-exclusive, royalty-free, transferable, irrevocable licence*

Übersicht

- Projektarten
  - Übersicht
  - Beispiele
  - Offene vs. geschlossene Projekte
  - Welche Lizenz nehmen
- Compliance mit Lizenzpflichten integrierter Komponenten

Projektarten | Offene vs. geschlossene Projekte

- Offenes Projekt: Dann sinnvoll, wenn Software endgültig als OSS auslizenziert werden soll.
- Ohne CLA dann, wenn zentrales CLA-Management vermieden werden soll
- Ohne CLA bedeutet aber auch, dass ein „Zurückholen“ von Contributions aus der Community in den proprietären Teil der Software nicht mehr möglich ist.
- Mit CLA: Verwaltungsaufwand. Aber: Einlizenzierung unter weitergehenden Rechten. Umlizenzierung und Dual Licensing bleibt möglich.
- Bei klassischem Dual-Licensing-Modell aufpassen, dass offene und proprietäre Teile sauber getrennt sind, wenn kein CLA vorhanden

Projektarten | Offene vs. geschlossene Projekte

- Geschlossenes Projekt: Dann sinnvoll, wenn Software Contributions verzichtbar sind.
- Achtung: geschlossenes Projekt bietet keinen Schutz vor Forking!
- Proprietäre Umlizenzierung jederzeit möglich.

Übersicht

- Projektarten
  - Übersicht
  - Beispiele
  - Offene vs. geschlossene Projekte
  - Welche Lizenz nehmen
- Compliance mit Lizenzpflichten integrierter Komponenten

Projektarten | Welche OSS-Lizenz nehmen?

- Welche Lizenz soll verwendet werden?
- Verschiedene Aspekte müssen beachtet werden:
  - Starker **Copyleft-Effekt** (z.B GPL) oder nicht?
  - **Permissive** (z.B. MIT oder BSD) oder nicht?
  - **Kompatibilität** der Lizenzen (z.B. Integration in vorhandene GPL-Komponenten)
- Die Auswahl der Lizenz hängt besonders davon ab, welche Ziele verfolgt werden

Projektarten | Welche OSS-Lizenz nehmen?

- Was will ich erreichen? Beispiele für Ziele und entsprechend passende Lizenzen:
  - Verbesserungen und Updates durch die OSS-Community = GPL-2.0?
    Nicht zwingend! Gut gemanagtes Projekt unter der BSD/MIT ist mehr wert, als unkoordiniertes Einsammeln von GPL Contributions
  - Schutz vor SaaS = AGPL, SSPL
  - Schutz vor Patentverletzungsklagen = GPL-3.0 oder andere Lizenz mit Patent Retaliation Clause
    - Aber: viele Unternehmen vermeiden GPL-3.0
    - Gefahr des Patentleft-Effekts
    - GPL-3.0 in vielen Szenarien nicht einsetzbar (z.B. DRM-geschützte „User Products“ wie iPhone Apps)

Übersicht

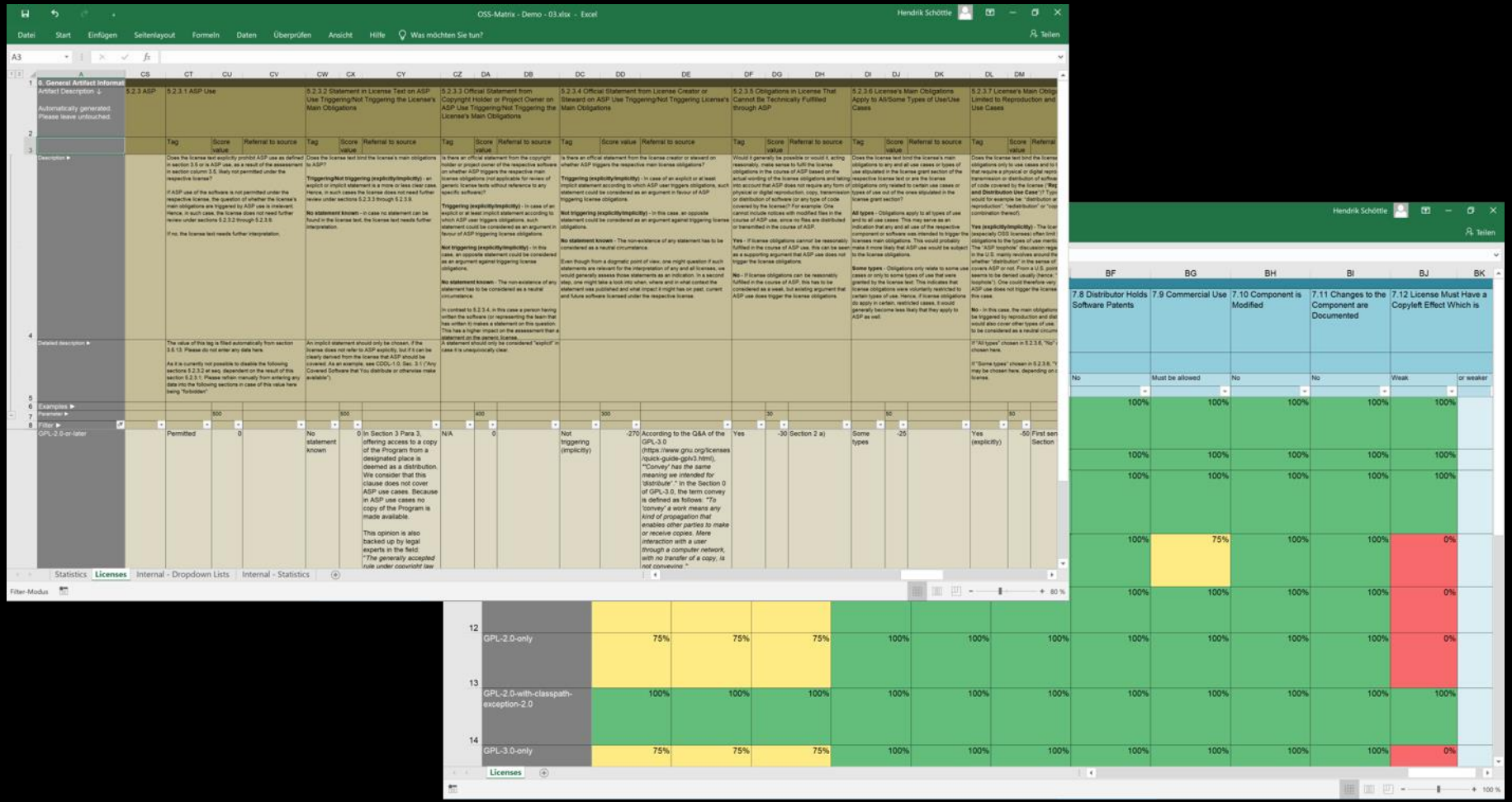
- Projektarten
  - Übersicht
  - Beispiele
  - Offene vs. geschlossene Projekte
  - Welche Lizenz nehmen
- Compliance mit Lizenzpflichten integrierter Komponenten

Compliance mit Lizenzpflichten integrierter Komponenten

- Was ist zu tun? Sämtliche Pflichten einer Lizenz erfassen, bewerten und gegen eigene Verwendung abgleichen. Welche Lizenzen kann ich akzeptieren, welche nicht?
- Beispiele:
  - Verbot des Einsatzes von OSS in DRM-geschützten Umgebungen.
  - Besondere Pflichten beim Einsatz von OSS als ASP/SaaS.
  - Hinweispflichten bei Bewerbung des die OSS enthaltenden Produktes
- Übliche Scanning Tools sind gut bei BOM & Co…
- …bieten aber keinen detaillierten und nachvollziehbaren Überblick über übrige Lizenzpflichten
- Zudem ist die Interpretation einzelner Lizenzen und deren Pflichten umstritten

Compliance mit Lizenzpflichten integrierter Komponenten

- Zum Beispiel der 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


OSS-Matrix - Demo - 03.xlsx - Excel
Hendrik Schöttle
Datei Start Einfügen Seitenlayout Formeln Daten Überprüfen Ansicht Hilfe Was möchten Sie tun?
Teilen
A3
0. General Artifact Informat
Artifact Description
Automatically generated. Please leave untouched.
5.2.3 ASP
5.2.3.1 ASP Use
5.2.3.2 Statement in License Text on ASP Use Triggering/Not Triggering the License's Main Obligations
5.2.3.3 Official Statement from Copyright Holder or Project Owner on ASP Use Triggering/Not Triggering the License's Main Obligations
5.2.3.4 Official Statement from License Creator or Steward on ASP Use Triggering/Not Triggering License's Main Obligations
5.2.3.5 Obligations in License That Cannot Be Technically Fulfilled through ASP
5.2.3.6 License's Main Obligations Apply to All/Some Types of Use/Use Cases
5.2.3.7 License's Main Obligation Limited to Reproduction and Use Cases
Tag
Score value
Referral to source
Description
Examples
Parameter
Filter
GPL-2.0-or-later
Permitted
0
No statement known
0
In Section 3 Para 3, (offering access to a copy of the Program from a designated place is deemed as a distribution. We consider that this clause does not cover ASP use cases. Because in ASP use cases no copy of the Program is made available.
This opinion is also backed up by legal experts in the field: "The generally accepted rule under copyright law
N/A
0
Not triggering (implicitly)
-270
According to the Q&A of the GPL-3.0 (https://www.gnu.org/licenses/quick-guide-gplv3.html), "'Convey' has the same meaning we intended for 'distribute'." In the Section 0 of GPL-3.0, the term convey is defined as follows. "To 'convey' a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying."
Yes
-30
Section 2 a)
Some types
-25
Yes (explicitly)
-50
First sen
Section
Statistics
Licenses
Internal - Dropdown Lists
Internal - Statistics
Filter-Modus
80 %
Hendrik Schöttle
Teilen
BF
BG
BH
BI
BJ
BK
7.8 Distributor Holds Software Patents
7.9 Commercial Use
7.10 Component is Modified
7.11 Changes to the Component are Documented
7.12 License Must Have a Copyleft Effect Which is
No
Must be allowed
No
No
Weak
or weaker
100%
75%
0%
GPL-2.0-only
GPL-2.0-with-classpath-exception-2.0
GPL-3.0-only
Licenses
100 %

Compliance mit Lizenzpflichten integrierter Komponenten

- Legal-(Low-)Tech-Lösung zur Lizenzbewertung und Use Case Matching mit folgenden Features:
  - **Standardisierte Bewertung** einzelner Lizenzpflichten
  - **Volle Dokumentation** der einzelnen Schritte zum gefundenen Ergebnis
  - **Parametrisierbarkeit** einzelner Faktoren, daher unterschiedliche Wertungen möglich (konservativer Ansatz vs. risikofreudiger Ansatz)
  - **Genaue Risikoabschätzung** durch Scorewerte möglich (nicht nur ja/nein, sondern Abstufungen im Prozentbereich)
  - Übersichtliche Darstellung
  - **Matching** einzelner Pflichten gegen die jeweiligen Lizenzen mit übersichtlicher Angabe von Risiko-Scorewerten und Verweis auf einzelne Prüfungsabschnitte

Fazit

- Vor Auslizenzierung von eigener Software OSS unbedingt Rahmenbedingungen abstecken
- Falsche Lizenzwahl in Verbindung mit einlizenzierten Contributions der Community lässt sich später oft nicht mehr korrigieren
- Abgleich mit Pflichten bestehender Komponenten ist unverzichtbar
- Schematisierte Darstellung/Visualisierung/Kompatibilitätsprüfung bestehender Komponenten kann dabei helfen

Kontakt

Dr. Hendrik Schöttle
Partner/Fachanwalt für IT-Recht
T +49 89 5434 8058
hendrik.schoettle@osborneclarke.com

Dr. Hendrik Schöttle berät im IT- und Datenschutzrecht.

Hendrik Schöttle wurde 2019 und 2018 sowohl vom Handelsblatt und von Best Lawyers als auch von der Wirtschaftswoche als einer der besten Anwälte im IT-Recht genannt. Das JUVE-Handbuch 2018/2019 empfiehlt ihn als „kooperativ und stets pragmatisch“. Er wird im Kanzleimonitor 2018/2019 und 2017/2018 als mehrfach empfohlener Anwalt im IT-Recht geführt. 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“. 2015 wurde er mit dem Client Choice Award von Lexology und dem International Law Office (ILO) in der Kategorie IT- und Internetrecht ausgezeichnet.

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 Arbeitskreis 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).