CERHA HEMPEL

Open Source Software: Copyleft und Compliance

Dr. Roman Heidinger, M.A.

Rechtsanwalt

17. Österreichischer IT-Rechtstag

4. Mai 2023

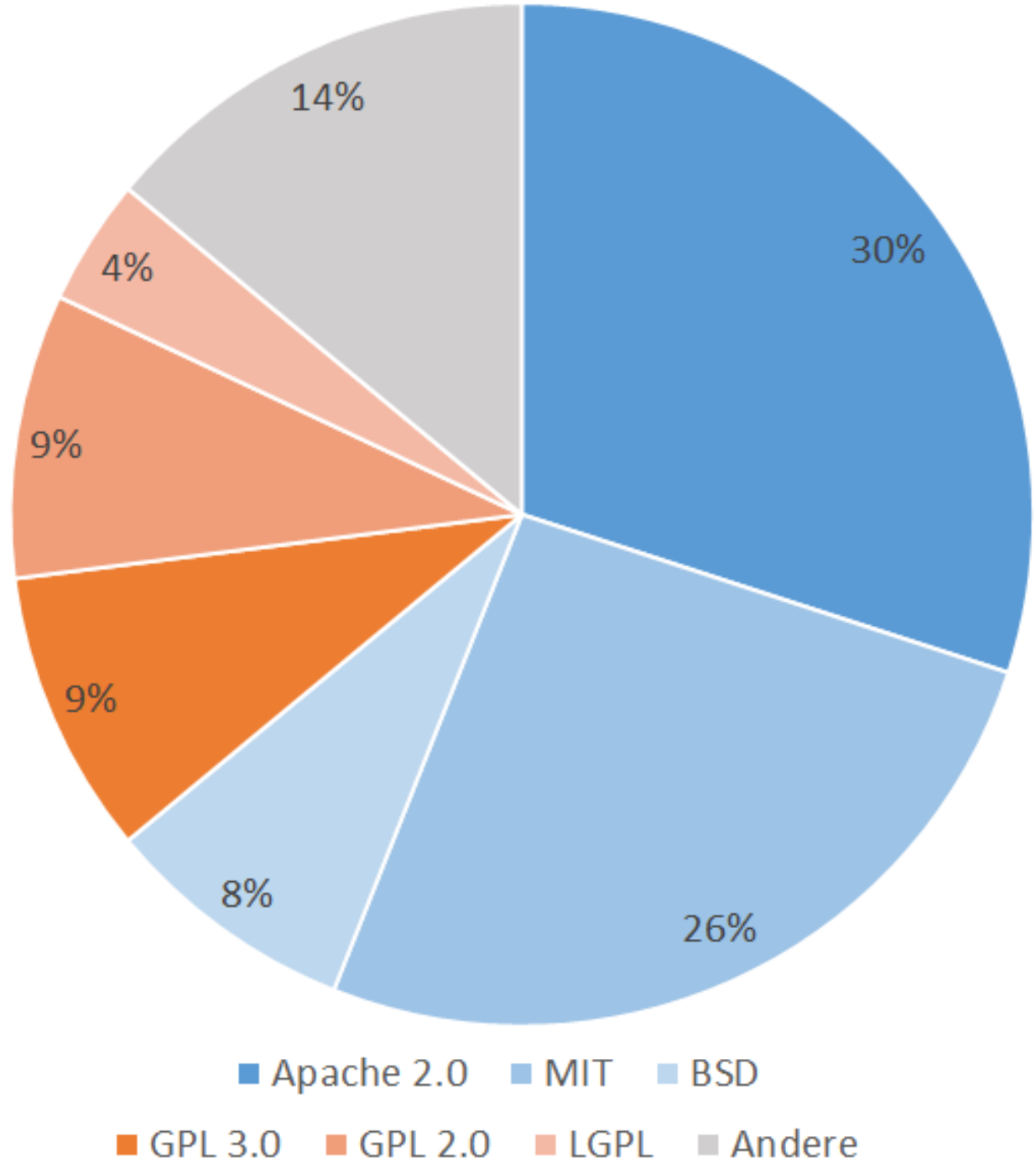
Einleitung

- Anfängliche Zweifel an der Vereinbarkeit von Open Source Lizenzen mit europäischen Rechtsordnungen sind längst beseitigt.
- Anfängliche ideologische Debatte ist überwunden.
  - Microsoft-Kampagnen / GPL als Krebsgeschwür
- Aufgrund der Modularisierung der Softwareentwicklung finden sich in fast jeder Software Open Source Komponenten.
- Normalisierung des Einsatzes von Open Source Software verleitet zu weniger sorgsamem Umgang mit Lizenzbestimmungen.
- Open Source Compliance soll die Einhaltung von Lizenzbestimmungen durch strukturelle Maßnahmen im Unternehmen sicherstellen.

Überblick: Open Source Lizenzen

- Breites Spektrum an teils sehr unterschiedlichen Open Source Lizenzen.
- Wesentliche Unterscheidung in Copyleft und Non-Copyleft Lizenzen.
  - Aber: fließender Übergang
- Bedeutung von Non-Copyleft-Lizenzen im Steigen begriffen.
  - 2012: 41 %
  - 2021: 78 %

Quelle: The Complete Guide for Open Source Licenses (Mend.io, 2022)



Copyleft Lizenzen

- Das Copyleft-Prinzip stellt darauf ab, die Nutzungsarten über die gesamte Lizenzkette hinweg offenzuhalten.
- Weiterentwicklungen müssen daher immer unter den identischen Bedingungen zur ursprünglichen Lizenz verbreitet werden.
  - Dies gilt auch bei Nutzung von Komponenten, ohne dass Quellcode direkt übernommen wird.
  - Vgl die weite Definition der „Corresponding Source“ in Ziffer 1 GPL 3.0.
- Bei schwachem Copyleft (zB LGPL oder Mozilla Public License) ist es – unter engen Voraussetzungen – zulässig, eigene oder abweichende Lizenzbestimmungen zu verwenden.
  - Libraries können ohne Copyleft Effekt genutzt werden.
  - Details der konkreten technischen Ausgestaltung sind von großer Bedeutung!

Durchsetzung: Wo kein Kläger, da kein Richter?

- Jeder Miturheber kann auf Unterlassung klagen.
  - UWG/Rechtsbruch bei Immaterialgüterrechten grds nicht einschlägig
    - Vgl aber OGH 4 Ob 62/14t – *Sportlerbilder*
- Durchsetzung von Schadenersatzansprüchen schwieriger
- Strafrechtliche Sanktionen bei Vorsatz (Privatanklagedelikt)
- Community Plattformen zur Koordination der Rechtsdurchsetzung
  - Teilweise auch mit (umstrittenem) kommerziellem Hintergrund
  - Beispiel: Quellcode des österreichischen E-Card Systems wurde nach Interventionen aus der OSS-Community zugänglich gemacht.
- Wirtschaftlicher Schaden (zB durch Vertriebsstopp des Softwareproduktes, Schadenersatz- und Gewährleistungsansprüche)
  - Vernichtung jahrelanger Entwicklungsarbeit, dh potentiell existenzgefährdend!

Wahlmöglichkeiten bei Lizenzverstoß

- Ein Verstoß gegen die Copyleft-Bestimmung einer OSS-Lizenz führt nicht automatisch dazu, dass die gesamte Software unter die OSS-Lizenz gestellt werde muss.
- Rechtsfolge: „nur“ Wegfall der Nutzungsrechte ex-nunc (hM)
- Nach dem Rechtewegfall stellt der weitere Vertrieb eine Urheberrechtsverletzung dar.
- Wiederherstellung der Nutzungsrechte bei lizenzkonformer Nutzung (vgl Punkt 8 der GPL 3.0)

Plan B: Rettung eines Softwareprojekts

- Offenlegung des Quellcodes
  - Muss zum Geschäftsmodell passen
  - Beispiel: E-Card
- Austausch der betroffenen Komponenten
  - Durch andere Komponenten mit Non-Copyleft-Lizenzen, proprietären Lizenzen oder selbstgeschriebenem Code
  - Over-the-air updates?
- Technische Anpassungen/Gestaltungsmöglichkeiten
  - Isolation bzw getrennter Vertrieb der Copyleft-Komponenten
  - Detaillierte technische und rechtliche Prüfung erforderlich
  - Rechtliches Risiko verbleibt (va mangels Rechtsprechung)

Non-Copyleft Lizenzen

- Auch diese Lizenzen enthalten bestimmte Pflichten des Anwenders, wie zB hinsichtlich des Lizenztexts und der Urheberrechtsvermerke.
- Die Pflichten sind aber nicht durch einen Copyleft geschützt, was bedeutet, dass Weiterentwicklungen nicht zwingend unter die gleiche Lizenz gestellt werden müssen.
- Der unveränderte Teil der Software steht weiterhin unter der ursprünglichen Open Source Lizenz.
  - Einhaltung der Lizenzbestimmungen muss trotzdem sichergestellt werden.
- Einhaltung der Lizenzbestimmungen kann gerade bei einer Vielzahl an Softwarekomponenten mit großem Aufwand verbunden sein.

Beispiel: Pflichten der Apache 2.0 Lizenz

Die Verbreitung von Werken unter der Apache 2.0 Lizenz setzt die Erfüllung folgender Pflichten voraus:

- Eine Kopie der Apache-Lizenz muss beigefügt werden.
- Bei modifizierten Dateien muss an auffälliger Stelle angeben werden, dass sie modifiziert sind.
- Falls das Originalwerk eine Textdatei namens „NOTICE“ enthält, müssen die dort enthaltenen Urhebervermerke auf eine in der Lizenz genauer vorgeschriebene Art und Weise mitverbreitet werden.
- (Bei der Verbreitung in der Quellform müssen alle Original-Urheberrechtsvermerke beibehalten werden.)

Beispiel: BSD-Lizenz

- *(3) All advertising materials mentioning features or use of this software must display the following acknowledgement:*

  “This product includes software developed by the University of California, Berkeley and its contributors.”

- Trotz fehlendem Copyleft können schwierige Abgrenzungsfragen entstehen.
  - Beispiel: Eine BSD-Verschlüsselungskomponente wird in einem Softwareprodukt verwendet. Findet die Klausel bei Bewerbung des Produktes „mit höchsten Sicherheitsstandards bei der Verschlüsselung“ Anwendung?
- Umgehung durch Umlizenzierung?

Lizenz(in)kompatibilitäten

- Durch die (typischerweise) umfassende Wiederverwendung von Code und Bibliotheken bestehen Softwareprojekte aus einer Vielzahl an Beiträgen, die unterschiedlichen Lizenzen unterliegen.
  - Lizenzbedingungen von Komponenten müssen ebenfalls eingehalten werden.
- Lizenzkompatibilität muss im Einzelfall geprüft werden.
  - Rechtliche Ebene: Sind einander widersprechende Lizenzbestimmungen vorgesehen?
  - Technischer Aspekt: Ob ein abgeleitetes Werk vorliegt, ist (auch) anhand der Implementierung zu beurteilen (bei LGPL und MPL relevant).
- „Überraschende“ Inkompatibilitäten: BSD Code kann wegen Werbeklausel nicht in GPL Projekten eingesetzt werden.

Open Source Compliance - Zweck

- Sicherstellung des rechtskonformen Einsatzes von Open Source Software durch strukturelle Maßnahmen
- Schadensprävention
  - Vermeidung von Sanktionen nach dem UrhG
  - Verhinderung von Kosten für die Herstellung des rechtskonformen Zustandes
  - Im Urheberrecht gibt es keinen gutgläubigen Erwerb von Nutzungsrechten
- Erfüllung vertraglicher Verpflichtungen gegenüber Kunden
- Kann auch ein Marketingargument sein

Compliance-System

- Risikoanalyse
  - In welchem Umfang wird Open Source Software eingesetzt?
  - Welche Schäden drohen bei Verstoß gegen Lizenzbedingungen?
- Schriftliche Open Source Policy, die Prozesse und Verantwortlichkeiten regelt.
- Bestellung eines Open Source Compliance Officer
- Schulung der Mitarbeiter
- Zertifizierungen (zB OpenChain)
- Open Source Compliance als Teil der IT-Compliance

- Open Source Audits bei Due Dilligences bzw beim Unternehmenskauf

Risikoanalyse

- Eintrittswahrscheinlichkeit:
  - Aus welchen Quellen werden Open Source Komponenten bezogen?
  - Umfang/Anzahl der Fremdkomponenten
  - Welche Lizenzen werden eingesetzt (= je mehr Pflichten, desto eher besteht das Risiko von Verletzungen)?
  - Klagsrisiko (BSD vs GPL?)
- Schadensausmaß
  - Sanktionen (vertraglich / deliktisch)
  - Stellenwert der betroffenen Software im Gesamtunternehmen
  - Kosten der Beseitigung der Folgen von Lizenzverstößen
    - Einfaches over-the-air update vs Produktrückruf von Hardware
    - Ist die Offenlegung des Quellcodes im Notfall denkbar? (Beispiel: E-Card System)
  - Möglicher Reputationsverlust

Erhebung des Ist-Standes

- Erstellung einer Komponentenliste/Bill of Material (BOM)
  - Automatischer Export aus der Entwicklungsumgebung möglich
  - Einsatz von Lizenzmanagement Tools
- Ermittlung aller verwendeten Open Source Komponenten samt dazugehöriger Lizenzen
- Ermittlung der Lizenzpflichten
- Sichtbarmachen von Abhängigkeiten der Komponenten
  - Art der Verbindung der Komponenten wichtig

Open Source Scanner

- Durch Abgleich mit Datenbanken können die Open Source Bestandteile identifiziert und nach Lizenzen klassifiziert werden
  - Als Nebeneffekt können auch Sicherheitsmängel bei veralteten Versionen identifiziert werden.
- Scan bei bloßem Vorliegen des Objektcodes nicht möglich bzw fehleranfällig
- Automatische Verwaltung von Lizenztexten und Urhebervermerken bei Softwarekomponenten
- Schulung der Programmierer notwendig

Schematische Bewertung des Risikos beim Einsatz von Open Source Software

- Bloßer Einsatz von Open Source Software im Unternehmen (zB Tools)
- Weiterentwicklungen/Anpassungen für den unternehmensinternen Bereich

- Einsatz von Open Source Software bei SaaS-Angeboten
- Softwaredistributionen, die nicht unternehmenskritisch sind (zB App für bloße Marketingzwecke)

- Einsatz in Softwaredistributionen, die zum Kerngeschäft gehören
- Einsatz von Open Source Software in Hardware(komponenten)

Inhalt der Open Source Policy

- Festlegen von Prozessen zur Ermittlung von Programmbestandteilen, die an Dritte vertrieben werden und damit Lizenzpflichten auslösen
- Unternehmensinterne Klassifikation von Lizenzen
- Garantieerklärungen, die von Lieferanten verlangt werden
- Use Cases für bestimmte Nutzungsarten von Open Source Software
  - zB SaaS, Vertrieb im Objektcode, Beteiligung an Open-Source Projekten
- Kontrollsystem zur Einhaltung der Richtlinien
  - Überprüfung der Mitlieferung der Lizenztexte und Urheberrechtsvermerke
  - Wird das Copyleft eingehalten bzw der Quellcode mitgeliefert?

Beispiel: https://opensource.ebay.com/

Blacklist/Whitelist bzw andere Klassifikationen

- Einstufung der Lizenzen nach rechtlichem Risiko:
  - Lizenzen, bei denen die Einhaltung der Lizenzbedingungen allenfalls zeitaufwändig, aber sonst leicht umsetzbar ist
  - Einhaltung der Lizenzbedingungen bei Weitergabe potentiell problematisch und von der technischen Ausgestaltung abhängig
  - Lizenzen, deren Einsatz im Regelfall nicht möglich ist
- Besonderes unternehmensinternes Genehmigungsverfahren bei bestimmten Lizenzen
- Unterschiedliche Klassifikationen für unterschiedliche Anwendungsszenarien denkbar bzw notwendig
  - zB Unterscheidung nach SaaS und Softwaredistributionen
  - ASP löst bei der GPL den Copyleft-Effekt nicht aus, sofern kein Code (zB Javascript) an den Client übermittelt wird.

Mögliche Klassifikation der Lizenzen

• MIT Lizenz • BSD Lizenzen • Apache Lizenzen	• Einsatz freigegeben • Maßnahmen/Anleitung zur Einhaltung der Lizenzbedingungen
• Mozilla Public License (MPL) • Eclipse Public License (EPL)	• Verwendung nur nach Prüfung und Genehmigung zulässig • Technische Analyse notwendig
• LGPL (2.0/3.0) • GPL (2.0/3.0) • Affereo General Public License	• Verbot des Einsatzes • Allenfalls Ausnahmen mit Sondergenehmigung

Fazit

- Einhaltung der Lizenzpflichten von Open Source Software sollte durch Maßnahmen wie Open Source Policies sichergestellt werden.
  - Insbesondere Risiken für das geistige Eigentum/Know How des Unternehmens
- Das Open Source Compliance System muss an die individuellen Anforderungen und das Geschäftsmodell des Unternehmens angepasst werden.
  - Individuelles Eingehen auf besondere, unternehmensspezifische Gefahren
- Auch beim Einsatz proprietärer Softwarekomponenten bestehen die gleichen bzw ähnliche Risiken.
  - Gerade bei inkludiertem Programmcode aus Non-Copyleft Lizenzen von Dritten

CERHA HEMPEL

Vielen Dank für *Ihre Aufmerksamkeit!*