

Praxishandbuch Open Source

Technische und rechtliche Rahmenbedingungen
für einen lizenzkonformen Einsatz von FOSS
im Unternehmen

Herausgegeben von

Christian Galetzka, LL.M.

Rechtsanwalt und Fachanwalt für IT-Recht, Würzburg

Chan-jo Jun

Rechtsanwalt und Fachanwalt für IT-Recht, Würzburg

und

Yvonne Roßmann

Rechtsanwältin und Fachanwältin für IT-Recht, Würzburg

Bearbeitet von

Lisa Breunung; Christian Galetzka, LL.M.; Florian Hackel;
Chan-jo Jun; Julia Kendziorra; Ulrich Kulke; Yvonne Roßmann

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.de> abrufbar.

ISBN 978-3-8005-1763-3

dfv Mediengruppe



© 2021 Deutscher Fachverlag GmbH, Fachmedien Recht und Wirtschaft,
Frankfurt am Main

www.ruw.de

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Druck: WIRmachenDRUCK GmbH, Backnang

Printed in Germany

Inhaltsverzeichnis

Vorwort	V
Inhaltsübersicht	IX
Abkürzungsverzeichnis	XXIII
Literaturverzeichnis	XXIX
Kapitel I Free and Open Source Software (FOSS) – Idee und Risiken	1
1. Wie naiver Einsatz kostenloser Software Ihr Unternehmen bedrohen kann	2
a) Am Anfang stand die Idee: Freiheit von Copyright- zwängen	3
aa) Wie sich FOSS verbreitet hat	3
bb) Wo FOSS verzichtbar ist und wo nicht	4
b) In welchen Fällen kostenlose Software teuer werden kann	7
c) Was schlimmstenfalls passieren kann	9
aa) Häufiger Irrtum: Copyleft führt automatisch zur Freigabe als FOSS	9
bb) Ungenutztes Schädigungspotenzial der Rechtsinhaber	10
(1) Rechtlicher Fokus	10
(2) Realitätsfokus	12
d) Was bisher geschah	13
aa) Harald Welte: Pionier der deutschen FOSS Recht- sprechung	13
bb) Patrick McHardy: Kommerzialisierung der Rechts- verfolgung	14
cc) Schlaglichter FOSS relevanter Verfahren im Ausland	15
e) Das Who's Who der Open Source Community	18
aa) Free Software Foundation (FSF)	18
bb) Open Source Initiative (OSI)	19
cc) Apache Software Foundation (ASF)	20
dd) Eclipse Foundation	20
ee) Linux Foundation	21
	XI

Inhaltsverzeichnis

ff) Sonstige	21
2. Was FOSS eigentlich ist	23
a) FOSS = Free Software + Open Source Software	24
b) FOSS \neq Closed Source Software	27
aa) Kommerzielle Software	27
bb) Freeware	27
cc) Shareware	28
dd) Was ist mit Public Domain?	28
Kapitel II Technische Grundlagen: Coding und Kompilierung	31
1. Welche Arten von Code gibt es?	31
a) Source Code bzw. Quellcode	32
b) Object Code	33
c) Binärcode	33
d) Executables	34
2. Von welchen Tools die Entwickler sprechen	35
a) Entwicklungsumgebungen und Build-Tools	37
b) Werkzeugkoffer zum Coden – Compiler, Parser, Linker und Interpreter	40
3. Juristen müssen diese Verlinkung verstehen	43
a) Statische vs. dynamische Verlinkung	44
b) So linken die verschiedenen Programmiersprachen	46
aa) C und C++	47
bb) Python	47
cc) Java	48
4. Diese Befehlsstrukturen müssen Juristen erst recht verstehen	49
a) System Call	50
b) System Call zum Aufruf unselbstständiger System- oder Programmfunktionen	51
c) System Call zum Aufruf selbstständiger Anwendungs- programme – Process Call	53
d) Mounten von Dateisystemen	54
e) Pipes	55
f) Sockets	57
g) Aufruf über Kommandozeile	58

Kapitel III Rechtliche Grundlagen: insbesondere Lizenzvorgaben und Copyleft	61
1. FOSS Lizenzen sind AGB – es gilt Schenkungsrecht	61
a) Wie FOSS Lizenzen einbezogen werden	62
aa) Wirksame Einbeziehung	63
bb) Wirksamkeit	65
b) Wie Vertragsgrundlagen bei Ansprüchen helfen können	67
aa) Lizenzvertrag mit schenkungsvertragsrechtlichen Elementen	68
bb) Vertragsabschluss	70
cc) Schriftform	73
2. Was FOSS Lizenzen generell ausmacht	74
a) Kein Problem bei der reinen Nutzung	75
b) Probleme fangen erst bei der Weitergabe an	75
c) Das Leitmotiv: Freigabe von Source Code und Copyleft	76
3. Die wirklich entscheidenden juristischen Kriterien	77
a) Hinter „Weitergabe“ stehen verschiedene Verwertungshandlungen	77
b) EuGH: Erschöpfung schlägt jede Lizenz	82
c) Für eine Veränderung muss man programmieren	86
aa) Vergleiche U. S. und deutsche Rechts-terminologie	86
bb) Wann genau die Grenze zur Veränderung überschritten wird	88
d) Wann ein Code Bestandteil über die Hürde der Schutzfähigkeit springt	89
aa) Allgemeine Kriterien für die Schutzfähigkeit	89
bb) Kriterien der Rechtsprechung	90
cc) Begriff des „Elements“	91
dd) Das Problem mit der Nachweisbarkeit	93
4. Am Ende ein Scheinproblem: Rechtswahl- und Gerichtsstandsklauseln	95
5. Definiere Copyleft	98
6. Auslegung von derivative work entscheidet über das Copyleft	99
a) Orientierung an U. S. Copyright Act	100

Inhaltsverzeichnis

b)	Maßgeblich ist Veränderung bzw. Bearbeitung nach UrhG.	101
c)	Auslegung der GPL-2.0: Wann liegt ein derivative work vor?	101
aa)	Abgrenzungskriterien helfen für eine Annäherung	103
bb)	Formal: Vertreibe Copyleft Software getrennt	104
cc)	Inhaltlich-funktionale Einheit kann derivative work sein	105
dd)	Besser: Trenne zusätzlich technisch	105
d)	Was das alles bezogen auf den konkreten Einzelfall bedeutet.	106
aa)	Bearbeitung eines einzelnen Werks	106
bb)	Interaktion von proprietären Software-Komponenten mit GPL Software	108
e)	Derivative work führt zu Copyleft, Kurzformel.	108
7.	Taxonomie der Lizenzen	108
a)	FOSS Lizenzen mit strengem Copyleft	109
aa)	Die GPL-2.0	109
bb)	Die GPL-3.0	110
cc)	Die AGPL-3.0	111
dd)	Weitere Lizenzen mit strengem Copyleft	111
b)	FOSS Lizenzen mit beschränktem Copyleft	112
aa)	Die LGPL	112
bb)	Die MPL	113
cc)	Die EPL-2.0	114
c)	FOSS Lizenzen ohne Copyleft	115
d)	Sonstige Software-Klassen	115
8.	Das Who's Who der FOSS Lizenzen	116
a)	Die GPL-2.0 und GPL-3.0	117
b)	Die AGPL-3.0	118
c)	Die LGPL	119
d)	Die MPL-2.0	123
e)	Die EPL	124
f)	Die BSD Lizenzen	127
g)	Apache Lizenzen	128
h)	MIT	130
9.	Nicht jeder Lizenzverstoß erzeugt Strafbarkeit	130

a) Objektiver Tatbestand: Zunächst jede unerlaubte Verwertung	132
aa) Heilungsmöglichkeiten in FOSS Lizenzen schützen nicht zuverlässig	133
bb) Allein auf den Zeitpunkt der Verwertungshandlung kommt es an.	135
b) Subjektiver Tatbestand: Vorsatz erfordert konkrete Kenntnis	136
c) Keine Strafbarkeit bei nachträglicher Kenntnis	140
d) Keine Verfolgung ohne Strafantrag	142
10. Exkurs: Recht ist relativ – Wie viel Rechtssicherheit darf es heute sein?	142
a) Der bequeme Weg.	143
b) Scannen alleine schafft nur Scheinsicherheit	145
Kapitel IV Die elementaren Prozessschritte: Ermittlung – Bewertung – Umsetzung	149
1. Der Dreiklang des Compliance-Prozesses.	149
2. Exkurs: FOSS Risiken als „Goldgrube“ in der Due Diligence	150
a) Irgendein FOSS Risiko findet sich immer	151
b) Effektive Klauselinhalte für den Kaufvertrag.	153
Kapitel V Ermittlung der eingesetzten FOSS.	155
1. Prüftiefe abhängig von Herkunft der FOSS.	155
a) Warum Eigenentwicklungen voll geprüft werden sollten	157
b) Die Wahl bei zugelieferten Software-Produkten	161
aa) Vertrauen ist gut, Kontrolle ist besser?	162
bb) Vollprüfung vs. Stichproben	165
c) Exkurs: Was im Vertrag bzgl. FOSS geregelt sein sollte	168
2. Prüftiefe abhängig vom Einsatz der Software.	169
a) Keine Weitergabe: wie intern es dafür sein muss.	171
aa) „Dual Use“ bei Tooling Software	172
bb) Weitergabe von Werkzeugen wie Compiler und Parser.	173
b) Wo die Weitergabe nicht offenkundig, aber trotzdem zu bejahen ist.	173

Inhaltsverzeichnis

aa)	Konzerninterne Weitergabe oder Weitergabe „im kleinen Kreis“	174
bb)	Aufruf von FOSS über einen Browser	176
cc)	Weitergabe bei der Nutzung von Clouddiensten	178
(1)	Weitergabe durch Übermittlung der FOSS an eine Cloud	179
(2)	Weitergabe aufgrund von Zugriffsmöglichkeit durch Dritte	180
c)	Was Cleared Code Base bedeutet und warum sie sinnvoll ist.	180
3.	Bill-of-Material: Was ist drin und woher nehmen?	182
a)	Notwendige und optionale Inhalte	183
b)	Was ist wirklich drin: File Lizenzen und Dependencies	186
aa)	File Lizenzen aus den Header Informationen der einzelnen Files.	186
bb)	Automatisch eingebundene Abhängigkeiten, sogenannte Dependencies	188
c)	Manuelle Bestandslisten selten vollständig	189
d)	Ziel: Automatisierte Erfassung aus Entwicklungs-umgebungen	190
e)	Code Scans als essenzieller Prozessbaustein	192
aa)	„Header“ Scan vs. „Snippet“ Scan vs. „Cloud“ Scan	193
bb)	„Daily“/„Nightly“ Scan vs. Code Audit	196
cc)	Details zu gängigen Scantools aus der Praxis	197
Kapitel VI Rechtliche Bewertung der eingesetzten FOSS 201		
1.	Vorspann: Clustering nach Compliance-Leveln	201
a)	Kategorie A: Panisch	202
b)	Kategorie B: Konservativ	203
c)	Kategorie C: Liberal	203
d)	Kategorie D: Leichtfertig	203
e)	Wenn der CEO das Risiko kalkulieren will	203
aa)	Eintrittswahrscheinlichkeit	204
bb)	Schadenspotenziale	205
(1)	Gesetzliche Ansprüche	205
(2)	Ansprüche aus Vertrag	206

2.	Shopping list für die rechtliche Bewertung	207
	a) Bill-of-Material (BOM)	208
	b) Schicksal des Endprodukts	209
	c) Sonderproblem: Strukturen und Deadlines in der Projektverwaltung.	210
3.	Whitelist/Blacklist-Ansätze: Begrenzter Nutzen	211
	a) Sortierung nach Gefährlichkeit von FOSS Klauseln naheliegend.	213
	b) Listen führen zu pauschalem Verbot von zulässigem Einsatz	214
4.	Legitime Shortcuts bei der rechtlichen Prüfung	215
	a) Vereinfachtes Prüfungsschema: Grundfrage Weitergabe	216
	b) Warum auch interne Software erfasst werden sollte.	217
	c) Schnelle Bewertung von FOSS unter Lizenzen ohne Copyleft	218
	aa) Beifügen von Informationen genügt	218
	bb) Umgehung der Pflichtangaben durch Relizen- zierung?	220
5.	Schutz vor Copyleft Infektion	224
	a) Einhaltung der Vorgaben zum beschränkten Copyleft als „letzte Ausfahrt“	226
	aa) Ausweg Verlinkung für LGPL.	227
	bb) Differenzierte Betrachtung für EPL	229
	b) Hinreichende Abgrenzung der Software-Bestandteile	233
	aa) Grundannahme/„Daumenregel“	233
	bb) System Calls/Process Calls zur Verhinderung von Copyleft bei GPL.	234
	c) License Exceptions: gut gemeint, nur halb durch- dacht?	237
	d) Copyleft detailliert und am Fall	239
	aa) Grundfall: WordPress und „Forking“	240
	(1) Kein Lizenzwechsel erlaubt	241
	(2) Verstoß gegen Copyleft der GPL.	242
	bb) Spezielle Anwendungsfälle der Interaktion.	242
	(1) Einheitliche Installationsroutine des CMS mit proprietären Software-Komponenten und Vertrieb auf einem Datenträger	242

Inhaltsverzeichnis

(2) Verwendung/Aufruf von proprietären Standardkomponenten über WordPress-eigene Schnittstellen	244
(3) Hinzufügen eigener proprietärer Funktionskomponenten in WordPress.	245
(a) Ergänzungen/Veränderungen von WordPress-eigenem Code	246
(b) Änderung durch Austausch von Funktionskomponenten	246
(4) Verlinkung von proprietären Programm-bibliotheken mit WordPress	247
e) LG Berlin – Irrläufer-Urteil zur Reichweite des Copyleft	249
aa) Grundsätze der Entscheidung LG Berlin – Surfsitter	250
bb) Gegenpol: LG Mannheim und OLG Karlsruhe.	252
cc) In Berlin hätte es besser laufen können	255
f) Linux kernel, user space und UAPI: die „Büchse der Pandora“	256
6. Gefahr durch Ablauf-/Signaturprüfungen („TiVo“).	258
a) TiVo-Maßnahmen verhindern Veränderung installierter Software	261
b) Ausdrückliche Beschränkungen der Tivoisierung in FOSS Lizenzen.	263
aa) GPL/AGPL-3.0	263
bb) LGPL-3.0.	265
cc) Regelungen zu Tivoisierung in der GPL-2.0.	265
c) Faktische Beschränkungen der Tivoisierung in LGPL.	268
aa) LGPL-2.0.	269
bb) LGPL-2.1.	269
cc) Funktionserhaltende Austauschbarkeit	270
d) Exceptions als Befreiung von Lizenzvorgaben, auch bzgl. Tivoisierung.	271
e) Wie das Problem mit Tivoisierungsbeschränkungen gelöst werden kann.	273
7. Oft übersehen: Wenn Lizenzen allergisch aufeinander reagieren	274
a) Was Lizenzinkompatibilität bedeutet	275

b)	Wie Lizenzinkompatibilität durch Copyleft entstehen kann	276
c)	Welche FOSS Lizenzen konkret allergisch aufeinander reagieren	278
aa)	Apache-2.0 vs GPL	278
bb)	LGPL vs. GPL vs. GPL	280
cc)	Copyleft vs. Copyleft with Exceptions	281
d)	Lizenzkompatibilität: Wie prüfen	282
8.	Technische Anpassungen zur Rettung des legalen Einsatzes	283
a)	Austausch oft die leichteste Lösung	285
b)	Zwischenschichten als Schutzwall gegen Copyleft?	285
aa)	„Ränder“-Lösung über GPL mit Exception	286
bb)	Legitime Ausgestaltung separierender Zwischenlayer	287
c)	Techniker können die Einhaltung von Exception-Vorgaben bestätigen	289
d)	Innovative Ansätze, LGPL Austauschbarkeit zu erreichen	290
aa)	Verwendung von individuellen Authentifizierungsschlüsseln	292
bb)	Support-Lösung	292
cc)	Docker/Snap-Lösung	293
dd)	Aufteilungslösung	294
e)	Irrelevanter Link bei mitgelieferter Standard Library?	294
f)	Zwei Verlinkungen, kein Copyleft: technisches Argument bei Standard Library	299
9.	Rechtliche Argumentation als ultima ratio	302
a)	Wo kein Schutz, da kein Problem	303
b)	Bearbeiterurheber: Rechtsverfolgung aussichtslos?	305
c)	Wie viel toxisches Copyleft kann ein Werk schadlos wegstecken?	307
d)	Möglicher Einsatz trotz fehlender File Lizenz	311
e)	Wie man Erschöpfung bei der FOSS Verbreitung nutzen kann	314

Inhaltsverzeichnis

Kapitel VII Umsetzung des lizenzgerechten Einsatzes von FOSS in Produkten	319
1. Copyright, Lizenzen & Co.: Leicht zu erstellen, aber zeitaufwendig.	319
a) Welche verpflichtenden Angaben FOSS Lizenzen verlangen	321
b) „Written Offer“ als Arbeitserleichterung	325
c) Wie diese Pflichtangaben aussehen sollten	327
d) Und noch wichtiger: Was nicht vergessen werden sollte	329
e) Wann ein Link auf die Pflichtangaben nicht ausreicht. . .	330
2. Source Code Bereitstellung: Was zu tun ist.	332
a) Was genau bereitzustellen ist	333
b) Es muss nicht immer ein Datenträger sein	334
c) Das betrifft nicht immer nur Source Code	335
3. Verzicht auf Pflichtangaben oder wann man sich den „Kram“ sparen kann	336
Kapitel VIII Herausforderung: Weitergabe des Linux kernel.	341
1. Pflichtangaben	341
2. Verlinkungen auf Kernel Funktionen.	342
3. Binär-Blobs	342
4. Anwendungscode und kurze Textdateien	343
5. Dateien ohne Copyrightvermerke	344
6. Hardware-Treiber.	344
7. Lizenzkompatibilität	345
8. Bereinigung der Code Base	345
Kapitel IX Anhang: Compliance-Material und Checklisten	347
1. Beispiel Chart zur Kategorisierung der Lizenzen	347
2. Beispiel One Pager zur Einordnung der Lizenzen und Lizenzvorgaben	348
3. Beispiel Whitelist/Blacklist	349
4. Beispiel FOSS Compliance-Musterprozess.	349
5. Beispiel Übersicht zur Information über Vor- und Nachteile des FOSS Einsatzes.	350
6. Beispiel Darstellung Pflichtangaben	350

7. Übersicht der Rechtsprechung und Rechtsverfolgung bei FOSS	355
8. Lizenzkompatibilitätsmatrix	369
9. Praxisfall: Vertragsverhältnisse und Auswirkungen bei FOSS in Produkten	370
10. Pool von FOSS Vertragsklauseln.....	377
a) Klauselvorschlag zur Definition von FOSS im Vertrag	378
b) Ökonomisch effiziente Klausel zur Verwendung von FOSS.....	379
c) Umfassende Klausel zur Verpflichtung des Auftragnehmers.....	380
Glossar	385
Stichwortverzeichnis	401
Zusatzcontent zum Download	415