

Die rechtliche Relevanz von Open Source Lizenzen unter besonderer Berücksichtigung praktischer Problemstellungen

von

Mag. Dr. Kai Erenli

kai.erenli@informationsrecht.at

(Dissertation, eingereicht im WS 2007/2008 bei

Univ. Prof. Dr. Alfred Schramm

Institut für Rechtsphilosophie, Rechtssoziologie und Rechtsinformatik

und:

Univ. Prof. Dr. Gunter Nitsche

Institut für Österreichisches und Internationales Unternehmens- und
Wirtschaftsrecht)

INHALTSVERZEICHNIS

1.EINLEITUNG	1
2.ENTSTEHUNG UND ENTWICKLUNG VON OPEN SOURCE UND DIE DEFINITION UND ABGRENZUNG VERSCHIEDENER SOFTWAREVARIANTEN	6
2.1.Die Entstehung von Softwarelizenzen	7
2.1.1. Die Geschichte der Softwareindustrie	8
2.1.2. Die ersten Open Source Lizenzen	10
2.2.Die Geburtsstunde der „Free Software Foundation“	11
2.3.„Freie Software und “Open Source Software“	14
2.3.1. Die Abspaltung von Richard Stallman.....	15
2.3.2. “The Cathedral and the Bazaar”	16
2.3.3. Bill Gates und die Halloweendokumente.....	17
2.3.4. Linux	17
2.3.5. Aus Netscape wird Mozilla	19
2.3.6. Der Begriff Open Source wird geprägt	20
2.3.7. Die „Open Source Definition“	21
2.4.Abgrenzung verschiedener Softwarevarianten.....	25
2.4.1. Freie Software.....	26
2.4.2. Public Domain	27
2.4.3. „Copyleft“ und „Non-Copyleft“-Software	28
2.4.4. Microsoft Shared Source.....	30
2.4.5. Freeware, Donationware und Cardware	31
2.4.6. Shareware und Crippleware.....	32
2.4.7. Kommerzielle und Proprietäre Software.....	32
2.5.Ergebnis	34
3.URHEBERRECHTLICHE ASPEKTE	36
3.1.Allgemeines zum Schutz nach dem österreichischen Urheberrechtsgesetz	37
3.2.Der Schutz von Computerprogrammen im österreichischen UrhG.....	39
3.2.1. Schutzgegenstand	40
3.2.1.1. Einordnung und Werkhöhe.....	40

3.2.1.2.	Begriff.....	42
3.2.2.	Der Urheber als Rechteinhaber und die verschiedenen Arten der Urheberschaft an Open Source Software.....	44
3.2.2.1.	Der oder die Alleinurheber	45
3.2.2.2.	Miturheber	48
3.2.2.3.	Gehilfe.....	50
3.2.2.4.	Teilurheberschaft.....	51
3.2.2.4.1.	Gruppenarbeitsverträge im Open Source Bereich	53
3.2.2.4.2.	Gelegenheitsgesellschaften im Open Source Bereich.....	54
3.2.2.4.3.	Kooperationsverträge im Open Source Bereich.....	54
3.2.2.4.4.	Die Genossenschaft im Open Source Bereich.....	55
3.2.2.5.	Arbeitnehmerurheberrecht	57
3.2.2.5.1.	Neuentwicklung durch einen einzelnen Dienstnehmer.....	58
3.2.2.5.2.	Neuentwicklung durch mehrere Dienstnehmer	59
3.2.2.5.3.	Weiterentwicklung durch einen einzelnen oder mehrere Arbeitnehmer	60
3.2.2.5.4.	Neu- oder Weiterentwicklung auf Grund eines freien Dienstvertrags oder durch arbeitnehmerähnliche Personen	61
3.2.2.5.5.	Abgrenzung zwischen in „ <i>Erfüllung der Arbeitspflicht</i> “ entwickelte Open Source Software, bei „ <i>Gelegenheit des Arbeitsverhältnis</i> “ entwickelte Open Source Software und „ <i>außerhalb des Arbeitsverhältnis</i> “ entwickelte Open Source Software.	62
3.2.3.	Der zivilrechtliche Schutz und die Auswirkung der verschiedenen Arten der Urheberschaft auf die Aktivlegitimation	65
3.2.3.1.	Unterlassungsanspruch.....	66
3.2.3.2.	Beseitigungsanspruch.....	69
3.2.3.3.	Anspruch auf angemessenes Entgelt und der Anspruch auf Schadenersatz und die Herausgabe des Gewinns	70
3.3.	Das Urhebervertragsrecht und die Open Source Lizenzen	74
3.3.1.	Die Open Source Lizenzen – Werknutzungsbewilligung oder Werknutzungsrecht?	76
3.3.2.	Schutzrechtsumfang und -wirkungen	77
3.3.2.1.	Die Problematik des Erschöpfungsgrundsatzes.....	78

3.3.2.1.1.	Download.....	79
3.3.2.1.2.	Eigentumsübertragung ohne Download.....	81
3.3.2.2.	Zugänglichmachung im Internet	82
3.3.2.3.	Die Problematik der Vermietung.....	84
3.3.2.4.	Die Problematik des Application Service Providing und Software as a Service.....	86
3.3.2.5.	Der Einfluss des GebG auf Open Source Lizenzverträge.....	88
3.4.	Ergebnis	91
4.	DIE ANNAHME DES LIZENZVERTRAGS	95
4.1.	Die Anwendbarkeit von österreichischem Recht auf Open Source Lizenzverträge	95
4.2.	Die Annahme des Lizenzvertrags.....	97
4.2.1.	Abschlussvarianten	98
4.2.2.	Überblick über die Notwendigkeit zum Abschluss eines Open Source Lizenzvertrags	99
4.2.3.	Download und Kaufvertrag.....	100
4.2.4.	OEM und Embedded-Systems.....	102
4.3.	AGB.....	103
4.4.	Lizenzsprache	104
4.5.	Erwerb der Lizenzrechte von einem unberechtigten Dritten	106
4.6.	Ergebnis	107
5.	DIE OPEN SOURCE LIZENZEN – RECHTE UND PFLICHTEN	109
5.1.	Die Lizenzarten	111
5.2.	Lizenzen ohne Copyleft-Effekt – BSDartige Lizenzen	112
5.2.1.	Die BSD-Lizenz.....	113
5.2.1.1.	Hintergrund	114
5.2.1.2.	Lizenzmerkmale	115
5.2.1.3.	Kompatibilität.....	116
5.2.2.	Die Apache Software License	116
5.2.2.1.	Hintergrund	117
5.2.2.2.	Lizenzmerkmale der Version 1.0 und 1.1	117
5.2.2.3.	Lizenzmerkmale der Version 2.0.....	118
5.2.2.4.	Kompatibilität.....	120
5.2.2.4.1.	Sichtweise der FSF.....	120
5.2.2.4.2.	Sichtweise der ASF.....	121

5.2.2.4.3.	Kommentar zu den Sichtweisen.....	122
5.2.3.	Die OpenLDAP Public License	123
5.2.3.1.	Hintergrund	123
5.2.3.2.	Lizenzmerkmale	123
5.2.3.3.	Kompatibilität.....	124
5.2.4.	W3C Software Notice und License	124
5.2.4.1.	Hintergrund	125
5.2.4.2.	Lizenzmerkmale	125
5.2.4.3.	Kompatibilität.....	126
5.2.5.	„X11“ oder auch „MIT License“	126
5.2.5.1.	Hintergrund	126
5.2.5.2.	Lizenzmerkmale	127
5.2.5.3.	Kompatibilität.....	128
5.2.6.	Zope Public License.....	129
5.2.6.1.	Hintergrund	129
5.2.6.2.	Lizenzmerkmale der Versionen 1.0 und 1.1	129
5.2.6.3.	Lizenzmerkmale der Versionen 2.0 und 2.1	131
5.2.6.4.	Kompatibilität.....	132
5.3.	Lizenzen ohne Copyleft-Effekt – Sonstige Lizenzen	133
5.3.1.	Die “Open Group Test Suite License”	133
5.3.1.1.	Hintergrund	134
5.3.1.2.	Lizenzmerkmale	134
5.3.1.3.	Kompatibilität.....	136
5.4.	Lizenzen mit strengem Copyleft-Effekt – GPL-artige Lizenzen ...	137
5.4.1.	Die GNU General Public License 2.0.....	137
5.4.1.1.	Hintergrund	138
5.4.1.2.	Lizenzmerkmale	139
5.4.1.2.1.	Veröffentlichung der Software.....	143
5.4.1.2.2.	Der Begriff „ <i>derived work</i> “	145
5.4.1.3.	Kompatibilität.....	155
5.4.1.3.1.	Kompatibilität mit anderen Open Source Lizenzen	155
5.4.1.3.2.	Kompatibilität mit proprietären Lizenzen	156
5.4.2.	Die GNU General Public License 3.0.....	157
5.4.2.1.	Hintergrund	157

5.4.2.2.	Lizenzmerkmale	158
5.4.2.3.	Kompatibilität.....	166
5.4.2.3.1.	Kompatibilität mit anderen Open Source Lizenzen	166
5.4.2.3.2.	Kompatibilität mit proprietären Lizenzen	168
5.5.	Lizenzen mit strengem Copyleft-Effekt – Sonstige Lizenzen.....	168
5.5.1.	Die European Union Public License.....	169
5.5.1.1.	Hintergrund	169
5.5.1.2.	Lizenzmerkmale	170
5.5.1.3.	Kompatibilität.....	177
5.5.1.3.1.	Die Kompatibilität hinsichtlich der aufgeführten Open Source Lizenzen	178
5.5.1.3.2.	Die Kompatibilität hinsichtlich der nicht aufgeführten Lizenzen.....	179
5.5.1.3.3.	Die Kompatibilität mit proprietären Lizenzen....	179
5.5.2.	Die Common Public License	180
5.5.2.1.	Hintergrund	180
5.5.2.2.	Lizenzmerkmale	181
5.5.2.3.	Kompatibilität.....	185
5.5.3.	Die Nethack General Public License.....	185
5.5.3.1.	Hintergrund	186
5.5.3.2.	Lizenzmerkmale	186
5.5.3.3.	Kompatibilität.....	187
5.6.	Lizenzen mit beschränktem Copyleft-Effekt – MPL-artige Lizenzen	187
5.6.1.	Die Mozilla Public License	188
5.6.1.1.	Hintergrund	189
5.6.1.2.	Lizenzmerkmale	189
5.6.1.3.	Kompatibilität.....	195
5.7.	Sonstige Lizenzen mit beschränktem Copyleft-Effekt	196
5.7.1.	Die GNU Library General Public License	197
5.7.1.1.	Hintergrund	197
5.7.1.2.	Lizenzmerkmale	198
5.7.1.3.	Kompatibilität.....	202
5.7.2.	Die Apple Public Source License	202
5.7.2.1.	Hintergrund	202

5.7.2.2.	Lizenzmerkmale	203
5.7.2.3.	Kompatibilität.....	206
5.7.3.	Die Sybase Open Watcom Public License.....	207
5.7.3.1.	Hintergrund	207
5.7.3.2.	Lizenzmerkmale	208
5.7.3.3.	Kompatibilität.....	211
5.7.4.	Die wxWindows Library License	212
5.7.4.1.	Hintergrund	212
5.7.4.2.	Lizenzmerkmale	212
5.7.4.3.	Kompatibilität.....	214
5.8.	Lizenzen mit Wahlmöglichkeiten.....	214
5.8.1.	Die Artistic License.....	215
5.8.1.1.	Hintergrund	215
5.8.1.2.	Lizenzmerkmale	216
5.8.1.3.	Kompatibilität.....	218
5.8.2.	Die Sleepycat License.....	218
5.8.2.1.	Hintergrund	219
5.8.2.2.	Lizenzmerkmale	219
5.8.2.3.	Kompatibilität.....	220
5.9.	Lizenzen mit Sonderrechten	221
5.9.1.	Die Q Public License.....	221
5.9.1.1.	Hintergrund	222
5.9.1.2.	Lizenzmerkmale	222
5.9.1.3.	Kompatibilität.....	224
5.10.	Das Dual Licensing	225
5.10.1.	Dual Licensing: Proprietäre Lizenz und Open Source Lizenz mit Copyleft Effekt	226
5.10.2.	Dual Licensing: Proprietäre Lizenz und Open Source Lizenz ohne Copyleft Effekt	227
5.11.	Ergebnis	228
6.	HAFTUNG UND GEWÄHRLEISTUNG	235
6.1.	Die einzelnen Vertragskonstellationen	236
6.1.1.	Kostenloser Erwerb durch das Herunterladen der Software im Internet direkt vom Lizenzgeber	236
6.1.1.1.	Schenkung	237

6.1.1.2.	Unentgeltlicher Vertrag sui generis	240
6.1.1.3.	Rechtsfolgen	240
6.1.1.3.1.	Gewährleistung	240
6.1.1.3.2.	Haftung	242
6.1.1.3.3.	Vertragshaftung	242
6.1.1.3.3.1.	Produkthaftung	244
6.1.2.	Erwerb der Software gegen Entgelt direkt vom Lizenzgeber	247
6.1.2.1.	Schenkung	247
6.1.2.2.	Kauf.....	248
6.1.2.2.1.	Vertrieb auf Datenträger	248
6.1.2.2.2.	Vertrieb über Download	249
6.1.2.2.3.	Gesamtbeurteilung des Rechtsgeschäfts	250
6.1.2.3.	Rechtsfolgen	252
6.1.2.3.1.	Gewährleistung	252
6.1.2.3.1.1.	Der Erwerbsvorgang.....	253
6.1.2.3.1.2.	Der Mängelbegriff	253
6.1.2.3.1.3.	Die Gewährleistungsbehelfe	256
6.1.2.3.2.	Haftung	257
6.1.3.	Erwerb der Software gegen Entgelt im Einzelhandel oder vom Distributor	258
6.1.3.1.	Schenkung und Kauf	259
6.1.3.2.	Rechtsfolgen	260
6.1.3.2.1.	Gewährleistung	261
6.1.3.2.2.	Haftung	262
6.2.	Ergebnis	263
7.	MARKENRECHT	266
7.1.	Die Rechtsquellen des Markenrechts	266
7.2.	Was wird durch das Markenrecht geschützt?	268
7.3.	Die Entstehung des Markenschutzes	269
7.3.1.	Die Registrierung von Marken von Open Source Software...	270
7.3.2.	Die Priorität der Anmeldung	271
7.3.3.	Der Markeninhaber im Open Source Bereich	271
7.3.4.	Die Grenzen der Registrierbarkeit von Open Source Software an Hand konkreter Beispiele	272

7.3.4.1.	Fehlende Unterscheidungskraft als Registrierungshindernis am Beispiel „ <i>kllustrator</i> “ und „ <i>samba</i> “	274
7.3.4.2.	Beschreibende Zeichen als Registrierungshindernis im Open Source Bereich am Beispiel „ <i>Konsole</i> “	278
7.3.4.3.	Gattungsbezeichnungen als Registrierungshindernis im Open Source Bereich am Beispiel des Pinguins „ <i>Tux</i> “	281
7.4. Die Lizenzierbarkeit von Open Source Software-Marken.....		284
7.4.1.	Die Einordnung der Lizenzbestimmungen unter § 14 MSchG	285
7.4.1.1.	Die Einschränkung des Lizenzgebietes für Open Source Software-Marken	286
7.4.1.2.	Die sachliche Beschränkung der Lizenz für Open Source Software-Marken	286
7.4.1.3.	Die Rechteeinräumung für Open Source Software-Marken am Beispiel von „ <i>Linux</i> “ und dem „ <i>Linux Sublicense Agreement</i> “	287
7.4.2.	Mögliche Spannungsfelder zwischen § 14 MSchG und der OSD	292
7.4.3.	Die OSD, die Marke „ <i>Open Source</i> “ und der Fall „ <i>Sugar CRM</i> “	295
7.5. Der zivilrechtliche Schutz von Open Source Software-Marken ...		298
7.5.1.	Die Aktivlegitimation bei Open Source Marken	298
7.5.1.1.	Die Gebrauchsüberlassung einer Open Source Marke	299
7.5.1.2.	Die Einräumung einer Open Source Markenlizenz mit absoluter Wirkung.....	300
7.5.2.	Der Unterlassungsanspruch von Open Source Software-Marken am Bsp der Marke „Open Source“	301
7.5.3.	Der Beseitigungsanspruch von Open Source Software-Marken am Bsp der Marke „Open Source“	305
7.6. Die Situation von Open Source Gemeinschaftsmarken in Bezug auf die EU-Erweiterung.....		306
7.7. Ergebnis		309
8. VERGABERECHT.....		313
8.1. Gemeinschaftsrechtliche Grundlagen		314
8.2. Grundlagen des österreichischen Rechts.....		316

8.2.1.	Das Problem des persönlichen und sachlichen Geltungsbereichs nach dem BVergG in Bezug auf Open Source Software.....	317
8.2.2.	Die Einordnung von Open Source Software unter den sachlichen Geltungsbereich	318
8.2.3.	Die Einordnung der Entgeltlichkeit im Vergaberecht in Bezug auf Open Source Software	320
8.3.	Die Grundsätze des Vergabeverfahrens in Bezug auf Open Source Software	322
8.3.1.	Open Source Software und freier, lauterer und fairer Wettbewerb	323
8.3.2.	Open Source Software und die Gleichbehandlung aller Bewerber und Bieter	324
8.3.3.	Open Source Software und die Vergabe an befugte und leistungsfähige Bieter	326
8.3.4.	Open Source Software und die Vergabe zu einem angemessenen Preis	327
8.4.	Die Leistungsbeschreibung und die technische Spezifikation in Bezug auf Open Source Software	328
8.5.	Ergebnis	331
9.	PATENTRECHT	335
9.1.	Einführung	336
9.1.1.	Die vier Grundprinzipien des Patentrechts.....	337
9.1.2.	Die vier Grundprinzipien des Patentrechts und Open Source.....	338
9.2.	Open Source – die technischen und patentrechtlichen Grundlagen, Bedenken und Schutzmöglichkeiten	341
9.2.1.	Die „Doppelnatur“ von Software.....	342
9.2.2.	Das PatG und die Problematik der Technizität	343
9.2.3.	Die patentrechtlichen Bedenken der Open Source Gemeinschaft.....	346
9.2.3.1.	RTLinux – Der Open Source Patentrechtsfall	347
9.2.3.2.	Aktuelle Bedenken - Anmerkungen zum EPLA	349
9.2.4.	Open Source Software und die Schutzmöglichkeiten bei einer erfolgten Anmeldung	351
9.3.	Ergebnis	352
10.	ENDERGEBNIS	356

11. LITERATURVERZEICHNIS	365
KOMMENTARE	365
MONOGRAPHIEN, SAMMELBÄNDE UND LEHRBÜCHER	366
AUFSÄTZE UND BEITRÄGE	368
WEBSEITENVERZEICHNIS	374
JUDIKATURVERZEICHNIS	380
ABBILDUNGSVERZEICHNIS	384

ABKÜRZUNGSVERZEICHNIS

AA	Anderer Ansicht
aaO	am angegebenen Ort
Abb	Abbildung
ABGB	Allgemeines Bürgerliches Gesetzbuch JGS 946
Abs	Absatz
AG	Arbeitgeber
AGB	Allgemeine Geschäftsbedingungen
AN	Arbeitnehmer
Art	Artikel
ASF	Apache Software Foundation
ASGG	Bundesgesetz über die Arbeits- und Sozialgerichtsbarkeit BGBl 104/1985
ASP	Application Service Providing
Bd	Band
BGBI	Bundesgesetzblatt
BGH	Bundesgerichtshof
BlgNR	Beilage zu den Stenografischen Protokollen des Nationalrates
BM	Bundesministerium
BSD	Berkley Software Distribution
bspw	beispielsweise
BVergG	Bundesgesetz über die Vergabe von Aufträgen

	BGBI I 17/2006
bzw	beziehungsweise
CR	Computer und Recht
DEC	Digital Equipment Corporation
dh	das heißt
E	Entscheidung
ECG	Bundesgesetz, mit dem bestimmte rechtliche Aspekte des elektronischen Geschäfts- und Rechtsverkehrs geregelt und das Signaturgesetz sowie die Zivilprozessordnung geändert werden, BGBI I 152/2001
EB RV	Erläuternde Bemerkungen zur Regierungsvorlage
EG	Vertrag zur Gründung der Europäischen Gemeinschaft (vor 1999: EGV bzw vor 1993 EWGV); Europäische Gemeinschaft(en)
EGBGB	Einführungsgesetz zum Bürgerlichen Gesetzbuch
EO	Gesetz über das Exekutions- und Sicherungsverfahren RGGI 79/1896
EU	Vertrag über die Europäische Union (vor 1999: EUV); Europäische Union
EuG	Europäischer Gericht erster Instanz
EuGH	Europäischer Gerichtshof
ERP	Enterprise Resource Planning
EvBI	Evidenzblatt
EVÜ	Europäisches Schuldvertragsübereinkommen
EWG	Europäische Wirtschaftsgemeinschaft
EWR	Europäischer Wirtschaftsraum

f	und der/die folgende
ff	und die folgenden
FAZ	Frankfurter Allgemeine Zeitung
Fn	Fußnote
FS	Festschrift
FSF	Free Software Foundation
FSFE	Free Software Foundation Europe
GebG	Gebührengesetz BGBl 267/1957
gem	gemäß
GesbR	Gesellschaft bürgerlichen Rechts
GMG	Bundesgesetz über den Schutz von Gebrauchsmustern BGBl 211/1994
GMV	Gemeinschaftsmarkenverordnung
GNU	GNU's not Unix
GP	Gesetzgebungsperiode
GPL	General Public License
GRUR	Gewerblicher Rechtsschutz und Urheberrecht
GRUR Int	Gewerblicher Rechtsschutz und Urheberrecht International
HABM	Harmonisierungsamt für den Binnenmarkt
hM	herrschende Meinung
Hrsg	Herausgeber
http	Hyper Text Transfer Protocol
IBM	International Business Machines
idF	in der Fassung
IFROSS	Institut für Rechtsfragen der Open Source Software

IHR	Internationales Handelsrecht
IP	Internet Protocol
IPRG	Bundesgesetz über das internationale Privatrecht BGBl 304/1978
iVm	in Verbindung mit
iSd	im Sinne des
JBl	Juristische Blätter
Kap	Kapitel
KMU	Klein- und mittelständische Unternehmen
KSchG	Bundesgesetz mit dem Bestimmungen zum Schutz der Verbraucher getroffen werden BGBl 140/1979
LDAP	Lightweight Directory Access Protocol
LG	Landesgericht/Landgericht
lit	litera
mE	meines Erachtens
MIT	Massachusetts Institute of Technology
MSchG	Markenschutzgesetz BGBl 260/1970
MR	Medien und Recht
MuSchG	Bundesgesetz über den Schutz von Mustern BGBl 497/1990
mwA	mit weiteren Anmerkungen
NJW	Neue Juristische Woche
ÖBl	Österreichische Blätter
OEM	Original Equipment Manufacturer
OGH	Oberster Gerichtshof

OLG	Oberlandesgericht
OSD	Open Source Definition
OSI	Open Source Initiative
ÖJZ	Österreichische Juristenzeitung
ÖSGRUM	Österreichische Schriftenreihe zum gewerblichen Rechtsschutz, Urheber- und Medienrecht
ÖZW	Österreichische Zeitschrift für Wirtschaftsrecht
PatG	Patentgesetz BGBl 259/1970
PDA	Personal Digital Assistant
PHG	Bundesgesetz über die Haftung für ein fehlerhaftes Produkt BGBl 99/1988
RdW	Recht der Wirtschaft
RGBI	Reichsgesetzblatt
RL	Richtlinie
Rsp	Rechtsprechung
Rz	Randziffer/Randzahl
StGB	Strafgesetzbuch BGBl 60/1974
SZ	Entscheidungen des österreichischen Obersten Gerichtshofes in Zivil- (und Verwaltungssachen) (Band, Nummer)
TCP	Transfer Control Protocol
TRIPS-Abk	Abkommen über handelsbezogene Aspekte der Rechte des Geistigen Eigentums
u	und
ua	unter anderem
UFITA	Archiv für Urheber- und Medienrecht

UGB	Bundesgesetz über besondere zivilrechtliche Vorschriften für Unternehmen RGBI 219/1897
UrhG	Urheberrechtsgesetz BGBl 111/1936
UWG	Bundesgesetz gegen den unlauteren Wettbewerb BGBl 448/1984
Vgl	Vergleiche
VfGH	Verfassungsgerichtshof
VO	Verordnung
WBI	Wirtschaftsrechtliche Blätter
XML	Extensible Markup Language
Z	Zahl/Ziffer
ZfRV	Zeitschrift für Rechtsvergleichung
zit	zitiert
ZPO	Gesetz über das gerichtliche Verfahren in bürgerlichen Rechtsstreitigkeiten RGBI 113/1895
ZUM-RD	Zeitschrift für Urheber- und Medienrecht Rechtssprechungsdienst

1. EINLEITUNG

„Die einzige Freiheit, die diesen Namen verdient, ist das Recht, unser Wohlergehen auf unserem eigenen Wege zu verfolgen, solange wir nicht anderen das ihrige verkümmern oder ihre darauf gerichteten Bemühungen durchkreuzen.“

- **John Stuart Mill**

Die vorliegende Arbeit beschäftigt sich mit der rechtlichen Relevanz von Open Source Lizenzen unter besonderer Berücksichtigung ausgewählter Problemstellungen. Open Source Software hat eine längere Tradition als manch einer vermuten mag und viel mit „*free software*“ gemein. Setzt man sich mit dem Begriff Open Source Software näher auseinander, stellt man fest, dass Open Source Software mehr als nur eine Softwarevariante ist und ihre eigene Philosophie verfolgt. Beide Begriffe – Open Source Software und „*free software*“ – sind zwar beinahe deckungsgleich, dahinter stehen jedoch unterschiedliche Intentionen. Während „Free Software“ vom Protagonisten der Szene *Richard M Stallman* angeführt wird, vertritt die *Open Source Initiative* (OSI) die Interessen der Open Source Software. Beider Hauptanliegen ist dabei der Freiheitsgedanke, der in den Open Source Lizenzen zum Ausdruck kommt.

Interessant in diesem Zusammenhang ist die Bedeutung des Wortes „frei“. Das Wort "frei" stammt von „fri“ ab, was zugleich Frau (=ursprünglich Herrin) bedeutet¹. Die Bedeutung "frei" entwickelte sich aus "eigen", vermutlich aus Wendungen wie "die eigenen Kinder". So kennzeichnete Freiheit einen Aspekt von Bindung und ist weit von der heute mit Freiheit assoziierten Bindungslosigkeit entfernt! Freiheit nach heutiger Definition bezeichnet die Möglichkeit, unter mehreren Optionen wählen zu können, ohne Sanktionen befürchten zu müssen. Ethisch ist Freiheit das Recht, die Möglichkeit und die Verpflichtung des Menschen zur Selbstbestimmung und zum Ausdruck seines freien Willens. Philosophisch gesehen ist Freiheit die Fähigkeit zur Entscheidung.

¹ <http://www.lexikon-definition.de/Freiheit.html> und <http://de.wikipedia.org/wiki/Freiheit>.

*Mill*² plädiert bspw für das Recht jedes Menschen, seine eigenen Fehler machen zu dürfen. Die Idee verliert ihren Schrecken, wenn man wie *Mill* der Meinung ist, dass Menschen aus ihren Fehlern lernen können und weil Menschen ihre Meinung ändern können, besteht er auf der Freiheit des Gedankens und der Diskussion. Es gibt keinen einzigen Grund, die Meinung Andersdenkender zu unterdrücken: Erstens, weil sie richtig sein könnte, zweitens weil sie falsch sein könnte (aber dann verhindert man, dass die richtige Meinung als richtig bestätigt wird) und drittens, weil sie weder falsch noch richtig ist und man nur im ständigen Austausch der Meinungen Irrtümer korrigieren kann. Meinungen und „Wahrheiten“ müssen ständig bewegt werden, sie brauchen Widerspruch, sonst erstarren sie zu toten Dogmen.

So gesehen greift *Mill* auf aktuelle Themen des 21. Jahrhunderts vor, welche sich auch in der eingangs erwähnten Philosophie der Open Source Gemeinde wieder finden. Es ist an dieser Stelle daher angebracht, einen kurzen Einblick in die Methodik der Softwareentwicklung im Open Source Bereich zu geben, um die Bedeutung der Freiheit der Open Source Software auch aus wirtschaftlicher Sicht besser beschreiben zu können:

Exkurs: Die Methodik der Open Source Entwicklung

„Of the first ten people to download Linux, five sent back bug fixes, code improvements, and new features.“

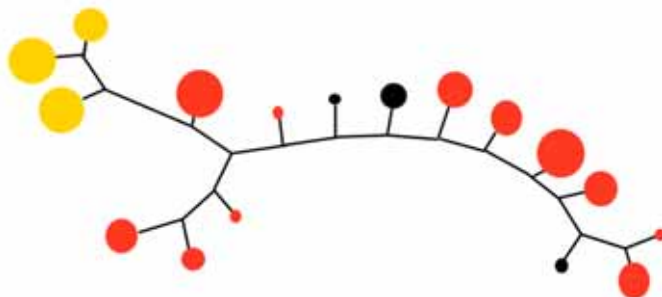
-Aus einer Linux-Geschichte³

Open Source wird häufig auch als ein „*way of life*“ verstanden. Viele Programmierer arbeiten neben ihrem Hauptberuf in ihrer Freizeit an Open Source Projekten mit. Sie sehen diese Art des Programmierens als Hobby, bei denen sie Ideen verwirklichen können, zu denen ihnen in der Arbeit der dienstliche Auftrag fehlt oder die sie alleine nicht verwirklichen könnten. Der Großteil der Open

² *John Stuart Mill* verdanken wir das Recht auf die Freiheit der Person, welches bspw auch in Art 2 des deutschen Grundgesetzes verankert ist.

³ http://www.firstmonday.org/issues/issue5_3/kuwabara/.

Source Projekte weist eine modulare Architektur auf. Hierdurch wird gewährleistet, dass sich Einzelne oder auch eine Programmiergruppe um verschiedene Teile kümmern können, um die Funktionalität des Systems zu erweitern ohne die existierenden Kernfunktionen anzugreifen. Voraussetzung für die Teilnahme an einem Projekt ist sinnvoller Weise ein Internetzugang. Erfolg oder Misserfolg eines Projekts hängen von der Motivation der Teilnehmer ab. Erlahmt ihr Interesse, wird das Projekt nicht weiter verfolgt. Das Interesse der „Außenwelt“ resultiert aus der hohen Innovation der Open Source Produkte. Die Innovation wiederum erklärt sich aus der Struktur der Open Source Projekte der von *Raymond* beschriebenen Basarmethode. *Klein* vergleicht sie mit einer Pflanze, die sich beim Heranwachsen ständig verändert und stellt dieses Bild gegenüber der festen und klaren Struktur eines Gebäudes. Als entscheidend für die hohe Innovation sieht er die Häufigkeit an, mit der die Programmierer Informationen über ein System einholen, um es zu optimieren und zu stabilisieren⁴. Zur Darstellung sollen folgende Grafiken dienen, die den Entwicklungsprozess des Open Source Projektes „Apache“⁵ darstellen⁶



Arbeitsgruppen Stand 1.1.1999

Abb 1

Die verschiedenfarbigen Punkte stellen einzelne Projektgruppen dar, welche an unterschiedlichen Modulen gearbeitet haben. Rot ist bspw ein HTTP-, Orange eine modperl- und Schwarz eine Administrationstool-Arbeitsgruppe.

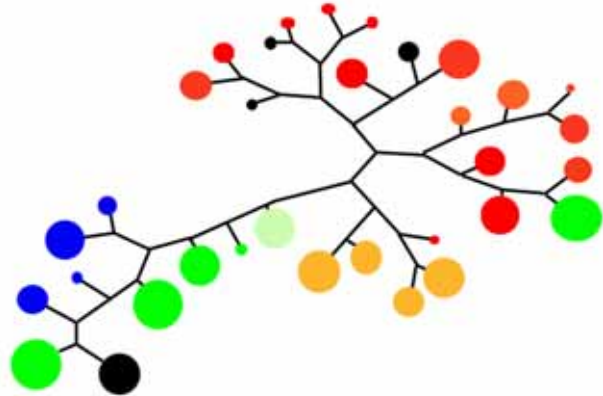
⁴ *Klein*, Alles Zufall – Die Kraft die unser Leben bestimmt (2004) 283.

⁵ Vgl Kap 5.2.2.

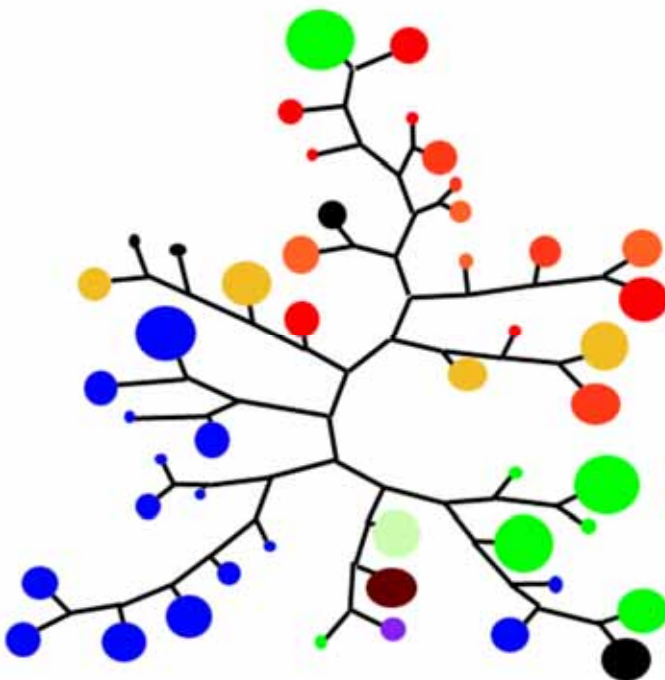
⁶ <http://libresoft.dat.escet.urjc.es/html/downloads/woss-icse-2004.pdf>.

Abb 2

Es ist deutlich zu sehen, dass die bestehenden Module und Arbeitsgruppen eine neue Struktur erhalten haben. Durch die Erweiterung um XML-Module (grün) ist ein neuer Zweig entstanden.



Arbeitsgruppen Stand 1.1.2000



Arbeitsgruppen Stand 1.9.2000

Abb 3

Im September 2000 ist die Struktur der Arbeitsgruppen schon sehr weit gefasst. Arbeitsgruppen an Modulen, die 1999 entwickelt wurden, sind kleiner geworden und befassen sich nur noch mit Fehlerbehebung, etc. Neue eigenständige Strukturen haben sich dezentral entwickelt.

Surowiecki hat dieses Phänomen näher untersucht und vergleicht⁷ die Innovation mit einer Bienenkolonie. Windows gehört Microsoft und wird nur von Microsoftmitarbeitern bearbeitet. Linux gehört niemandem. Taucht bei Linux ein Problem auf, wird dieses nur behoben, wenn jemand aus eigener Initiative heraus eine Lösung liefert. Die dezentrale Organisation erreicht bei Linux - wie bei anderen Open Source Projekten - die für sie wichtige Diversifikation.

⁷ *Surowiecki*, *The Wisdom of Crowds – Why the Many Are Smarter Than the Few* (2004) 72 f.

Traditionelle Firmenmodelle arbeiten nach dem Prinzip, die besten Mitarbeiter zu beschäftigen und ihnen Anweisungen zu geben, an welchen Problemfeldern sie arbeiten sollen. Diese Methode begrenzt die Summe möglicher Lösungen für ein Problem, da eine Firma nur über eine bestimmte Anzahl von Mitarbeitern verfügt, die nur ein bestimmtes Arbeitspensum erfüllen können. Im Falle von Open Source Projekten sieht dies anders aus: Hier ist eine nicht bestimmbare Menge an „Mitarbeitern“ auf der Suche nach Lösungswegen oder wie es *Raymond* ausdrückt: „*Given enough eyeballs, all bugs are shallow*“ – Wenn jemand genug Augen hat, sind alle Fehler unerheblich. Diese „Mitarbeiter“ verhalten sich ähnlich einem Bienenschwarm. Die Open Source Gemeinschaft sendet auf der Suche nach Lösungswegen eine Vielzahl an „Kundschaftern“ aus in der Hoffnung, dass zumindest einer unter ihnen die beste Route zum Blumenbeet finden wird. Diese Methode mag einem Ökonomen wenig effizient erscheinen, werden doch viele auf die Suche nach einem Weg geschickt, den vielleicht nur einer findet, aber genau davon lebt Open Source.

Der Exkurs hat gezeigt, dass die Freiheit, mit einem Programm möglichst restriktionslos umgehen zu können, ein wesentlicher Bestandteil von Open Source Software ist. Um ein besseres Verständnis für diese Problematik zu erhalten, wird am Anfang der Arbeit die Entstehung und Entwicklung von Open Source und die damit verbundene Notwendigkeit der Definition des Begriffs sowie eine Abgrenzung gegenüber anderen Softwarevarianten vorgestellt.

2. ENTSTEHUNG UND ENTWICKLUNG VON OPEN SOURCE UND DIE DEFINITION UND ABGRENZUNG VERSCHIEDENER SOFTWAREVARIANTEN

„Gesellschaftliche Veränderung fängt immer mit Außenseitern an, die spüren, was notwendig ist.“

-Robert Jungk⁸

In der breiten Öffentlichkeit wird mit dem Begriff Open Source und insbesondere Open Source Software meist das Betriebssystem Linux und die damit verbundene *General Public License* (GPL) in Verbindung gebracht. Linux ist die derzeit wohl bekannteste Open Source Software und die GPL⁹ die bekannteste Lizenz. Dennoch sollte man die Entstehung und den wirtschaftlichen Durchbruch von Open Source Software nicht auf die Entwicklung und das Medienecho von Linux¹⁰ und der GPL alleine zurückführen, da sich parallel dazu eine breite Masse an Anwendungssoftware etabliert hat, die in der Regel ihre eigenen Softwarelizenzen benutzt.

Die Software, die wir heute als Open Source Software bezeichnen, hat eine lange Geschichte, auf die zu Beginn der Arbeit kurz eingegangen wird. Gleichzeitig werden die Begrifflichkeiten, die mit der Thematik verbunden sind, erläutert und eine Abgrenzung verschiedener Softwarevarianten vorgenommen

⁸ Österreichischer Schriftsteller und Zukunftsforscher (1913-1994).

⁹ Die GPL liegt aktuell in Version 3 vor. Die bis dahin bekanntere Version ist die Version 2. Im Verlauf der Arbeit werden beide Lizenzen behandelt und mit „GPL 2“ bzw „GPL 3“ gekennzeichnet. Wird die Versionsnummer weggelassen, sind beide Lizenzen gemeint.

¹⁰ Vgl Kapitel 2.3.4.

2.1. Die Entstehung von Softwarelizenzen

Entstehung und Entwicklung von Open Source ist nicht denkbar ohne die damit Hand in Hand gehende Entstehung von Lizenzen. Ohne Lizenzen gäbe es keine Open Source Software, jedenfalls nicht in ihrer heutigen Definition. Eine Lizenz (v lat: *licere* = erlauben) ist eine Erlaubnis oder Genehmigung zur Nutzung eines Rechts durch den Urheber oder Inhaber dieses Rechts¹¹. Im juristischen Bereich stellt die Lizenz einen Vertrag dar, durch welchen einfache oder ausschließliche Rechte eingeräumt werden können.

Hinter der Lizenzierung im Softwarebereich standen und stehen in erster Linie wirtschaftliche Interessen der „Monopolisten“ wie IBM und Microsoft. Gleichzeitig dienten und dienen die mit IBM und Microsoft abgeschlossenen Lizenzen auch den Lizenznehmern, da diesen Wartung, Gewährleistung, Kompatibilität und Weiterentwicklung zugesichert wurden und wird.

Die Lizenz ist Schutzinstrument neben den gesetzlichen Bestimmungen¹² und versucht oftmals dem Benutzer Pflichten aufzubürden, die keinen Rückhalt in den gesetzlichen Bestimmungen finden. Die Lizenz bürdet Lizenzgeber wie -nehmer Pflichten auf, die den Lizenzgeber vor allem bei der Gewährleistung treffen. Das Besondere an Open Source Lizenzen ist, dass einfache Nutzungsrechte pauschal an jedermann eingeräumt werden.

Die Entstehung von Softwarelizenzen ist eng mit der Entwicklung der Softwareindustrie verbunden und wird im Folgenden erläutert.

¹¹ <http://www.langenscheidt.de/?fremdwb=lizenz>.

¹² Siehe Kap 3.

2.1.1. Die Geschichte der Softwareindustrie

Ende der 50er Jahre im letzten Jahrhundert setzten immer mehr Firmen EDV ein und Programmierer sahen erstmals ihre Chance, mit dem Anbieten ihrer Leistungen Geld zu verdienen¹³. Bereits Anfang der 60er Jahre konnte ein Kunde, der Hardware bei einem der damaligen Anbieter erwarb, darauf vertrauen, dass er die Software sowie Betriebssystem und gewünschte Applikationssoftware schon im Preis der Hardware inkludiert mitbekam.¹⁴ Bei der Software handelte es sich meist um reine Individualsoftware, die auf die Bedürfnisse des Kunden hin programmiert wurde und die im Quellcode vorlag.¹⁵

Ende 1960 beherrschte weitgehend IBM den Großrechnermarkt, da es das erste Unternehmen war, das den größten Marktanteil an kompatiblen Maschinen aufwies und damit für diese Computer die höchste Softwareverfügbarkeit reklamieren konnte. Ähnlich verstand es DEC damals, als erstes Unternehmen diese Segmente im Minicomputerbereich zu bedienen¹⁶.

IBM wollte seine Produkte nicht verkaufen, sondern vermieten bzw. verleasen. Was heute als getrennte Ware bzw. Dienstleistung wahrgenommen wird – Hardware auf der einen Seite, Software und Support auf der anderen – wurde als einheitliche Leistung vertrieben. Auf diese Weise gelang es IBM Kunden an sich zu binden. Wer einen IBM-Rechner hatte, der benutzte IBM-Software¹⁷.

¹³ http://www.softwarehistory.org/history/d_50s.html.

¹⁴ http://www.softwarehistory.org/history/d_60s.html.

¹⁵ Die Hersteller förderten sogar die Selbstorganisation ihrer programmierkompetenten Nutzer in Usergroups, wie bspw dem 1955 gegründeten Share von IBM, welches auch heute noch unter <http://www.share.org> aktiv ist.

¹⁶ <http://www.wi.wu-wien.ac.at/Publikationen/Janko/paper/main3/node5.html>.

¹⁷ *Mühlbauer*, Die Resozialisierung des Giganten, <http://www.heise.de/tp/r4/artikel/8/8778/1.html>.

Ein weiterer wichtiger Faktor für den Erhalt des IBM-Monopols war das System des "eingesperrten" Kunden. "Eingesperrt" waren IBM-Kunden, die beim Erwerb bereits erheblich in IBM-Produkte investiert hatten. Wer viel Geld und Arbeitszeit für die Beherrschung und Anpassung der IBM-Software aufgewendet hatte, der wechselte nicht leichtfertig zu einer anderen Firma - auch wenn deren Rechner mitunter leistungsfähiger und billiger waren. Den "eingesperrten" Kunden wurden höhere Preise berechnet, neue Kunden dagegen wurden mit Dumpingpreisen angelockt. Konsequenterweise angewendet wurde diese Form der Preispolitik aber nicht nur von IBM; bis heute werden die Einstiegsprodukte von Softwarefirmen häufig wie „Einstiegsdrogen“ verschenkt. Kopien werden erst großzügig geduldet, sobald eine dominante Marktposition gesichert ist, werden die Preise angezogen.

Die Methoden, das damals neue System OS/360 durchzusetzen, führten schließlich dazu, dass neben verschiedenen kleineren Computerherstellern auch die *Johnson-Administration* im Jahre 1969 gegen den Monopolisten klagte. Das amerikanische Justizministerium wollte IBM, das zum damaligen Zeitpunkt einen Marktanteil von 70%¹⁸ innehatte und 7 Milliarden Dollar Umsatz im Jahr generierte, in sieben Unternehmen aufteilen. IBM entkam der Zerschlagung nur durch die "Entbündelung" seiner Preise, was zu einer weiteren Verbreiterung der Softwareindustrie beitrug. Am 30. Juni 1969 gab IBM bekannt, dass ab dem 1. Januar 1970 Hardware und Software getrennt voneinander bezahlt werden musste¹⁹. Erst durch diese Trennung von Hardware und Software konnte ein lukrativer Markt für Software entstehen. Gleichzeitig entstand die Praxis, den Quellcode der Software von nun an geheim zu halten und ihn so vor Mitbewerbern zu schützen. Erst jetzt konnte sich Open Source Software als Alternative zu proprietärer Software entwickeln!

¹⁸ <http://www.cs.mun.ca/~ulf/csh/commcomp.html>.

¹⁹ http://www.softwarehistory.org/history/d_60s.html.

2.1.2. Die ersten Open Source Lizenzen

1969 entwickelten die Programmierer *Ken Thompson* und *Dennis Ritchie* das Betriebssystem UNIX in den Bell Telephone Laboratories²⁰. Der 1956 abgeschlossene „*Consent Decree*“ verbot aber AT&T, der Muttergesellschaft der Bell Telephone Laboratories, das Betreten neuer Märkte – wie dem des Computermarkts – und beschränkte AT&T auf das Telekommunikationsgeschäft. Aus diesem Grund wurde Unix (in der 1975 aktuellen Version 6) lediglich zum Preis der Datenträger verschiedenen Universitäten zur Verfügung gestellt. Inkludiert war der vollständige Quellcode²¹. Die Lieferung erfolgte nur gegen Vorkasse, von AT&T gab es keinerlei Support oder Bugfixes²². Da also von AT&T keinerlei Hilfe zu erwarten war, begann im universitären Umfeld eine rege Entwicklertätigkeit²³, die sich vor allem des damals schon im Gebrauch befindlichen Usenet²⁴ als Support-Netzwerk bediente. Koordiniert wurde diese Arbeit von der University of Berkley in Kalifornien, die in Folge auch eine eigene UNIX-Distribution herausbrachte: Die Berkley Software Distribution (BSD), auf die später noch genauer eingegangen wird.

²⁰ <http://ig.cs.tu-berlin.de/oldstatic/w2000/ir1/referate1/k-1a/>.

²¹ <http://www.linux-praxis.de/linux1/geschichte.html>.

²² Sogenanntes „Bugfixing“ beschreibt die Behebung von Mängeln am Quellcode.

²³ Siehe auch *Gulbins/Obermayer*, Unix System V.4. Begriffe, Konzepte, Kommandos, Schnittstellen (1995) 7.

²⁴ <http://www.usenet-abc.de/whatis.htm>.

2.2. Die Geburtsstunde der „Free Software Foundation“

„Free software“ is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech”, not as in “free beer.”

- aus der „Free Software Definition“²⁵ -

Die Entwicklungen im amerikanischen Telekommunikationsmarkt brachten es mit sich, dass AT&T zu Beginn der 80er Jahre schließlich doch im Softwaregeschäft tätig sein durfte und UNIX-Lizenzen zu höheren Preisen verkaufen konnte. Nun wurde der Quellcode auch von AT&T geheim gehalten und nicht wie bisher anderen zur freien Verfügung gestellt. Verärgert über diese Wendung und vor allem über die Kommerzialisierung von UNIX begann 1983 *Richard Stallman*²⁶ mit den Arbeiten an einem auf UNIX basierenden Betriebssystem, das er mit dem rekursiven Akronym²⁷ GNU – für GNU’s not UNIX – bezeichnete.

Richard Stallman ist eine der schillerndsten Persönlichkeiten in der Open Source Geschichte. Er wurde 1953 in Manhattan, NYC (USA), geboren und begann 1971 am Massachusetts Institute of Technology in der Abteilung für künstliche Intelligenz als Experte für Assemblersprachen, Betriebssysteme und Texteditoren eng mit einer Gruppe von Programmierern zusammenzuarbeiten, die sich selbst als „Hacker“ bezeichneten. Die Mitglieder dieser „Hackergemeinschaft“ waren schon damals rigorose Vertreter des unbegrenzten Informationsflusses²⁸. Viele Firmen begannen, wie bereits erwähnt, Software nicht mehr in der bis dahin weitgehend üblichen Form von Quelltexten auszuliefern, sondern in Form eines rein maschinenlesbaren Formats. Auch

²⁵ <http://www.gnu.org/philosophy/free-sw.html>.

²⁶ <http://www.stallman.org/#serious>.

²⁷ Als rekursives Akronym bezeichnet man eine Abkürzung, die in der Erklärung ihrer Bedeutung auf sich selbst verweist. Rekursive Akronyme sind normalerweise Initialwörter. Der Begriff wird häufig auch für Abkürzungen gebraucht, die eigentlich keine Akronyme sind. Die meisten rekursiven Akronyme treten in der Open Source-Szene auf, wie bspw. *PHP: Hypertext Preprocessor*, *Anna is Not aN Acronym* oder auch *LAME Ain't an Mp3 Encoder*.

²⁸ <http://www.heise.de/tp/r4/artikel/6/6475/1.html>.

statteten nun einige Firmen ihre Software mit Lizenzen aus, die es den Anwendern verboten, die Programme weiterzuverteilen oder die Programme selbst zu verändern.

Exkurs: „Open Letter to Hobbyists“ von William Gates III

“As the majority of hobbyists must be aware, most of you steal your software.”

Aus dem “*Open Letter to Hobbyists*” von **Bill Gates** vom 03.02.1976

Am 3. Februar 1976 schrieb *Bill Gates* einen offenen Brief an Hobbyentwickler von Software²⁹. Darin beklagte er sich, dass viele von ihnen seine Software stehlen würden, um sie weiterzuentwickeln. Er propagierte für sich, dass gute, qualitative Software nur bestehen kann, wenn dafür gezahlt wird. Hintergrund war, dass *Bill Gates* seine wirtschaftlichen Interessen bedroht sah als er erfuhr, dass sein Programm Basic, das er für den Altair mit *Paul Allen* ua entwickelt hatte und das die Firma MITS in Lizenz vertrieb, von vielen „Hobbyentwicklern“ kopiert und damit kostenlos benutzt wurde. *Gates* forderte sie auf, dies zu unterlassen.

Stallman empfand diesen Kontrollverlust der Benutzer über ihre eingesetzte Software als erhebliche Einschränkung ihrer Rechte. Um diesem Trend entgegenzusteuern, schuf er eine Lizenz, die unter dem Namen General Public License (GPL)³⁰ bekannt werden sollte. Er kündigte 1984 seine Stelle am MIT und entwickelte in Folge unter anderem die erste Version von GNU Emacs³¹, den GNU Symbolic Debugger (GDB), den ersten plattformübergreifenden C-Compiler (gcc) sowie verschiedene für eine Unix-Umgebung benötigte Utilities.

²⁹ <http://www.blinkenlights.com/classiccmp/gateswhine.html>.

³⁰ Vgl Kap 5.41.

³¹ GNU Emacs ist ein komplexer, programmierbarer Texteditor, zu dessen Originalversion er bereits beigetragen hatte.

1985 gründete *Stallman* die *Free Software Foundation (FSF)*³², eine gemeinnützige Organisation, die sich der Förderung und Produktion Freier Software verschrieben hat und versuchte, Kapital für die Arbeit am GNU-Projekt zusammenzutragen. Die FSF hat bis heute starken Einfluss auf die Lizenzentwicklungen im Open Source Bereich und darf neben der OSI als der Hauptakteur im „Lobbying“-Bereich für die Entwicklung und den Umgang mit Freier Software angesehen werden.

In Europa gibt es die am 10. März 2001 gegründete *Free Software Foundation Europe*, die sich den europäischen Belangen im Bereich Freie Software widmet. Als offizielle Schwesterorganisation der FSF in den USA konzentriert sie ihre Aktivitäten im Umkreis des GNU-Projekts, beschränkt sich jedoch nicht allein darauf, sondern betrachtet es als ihre Hauptaufgabe, Initiativen Freier Software in Europa zu koordinieren sowie ein Kompetenzzentrum für Politiker, Anwälte und Journalisten anzubieten und die Infrastruktur für Freie-Software-Projekte und speziell das GNU-Projekt zur Verfügung zu stellen.

Im Februar 2003 hat die FSF Europe das „*Fiduciary Licence Agreement*“ (FLA, dt.: Treuhänderische Lizenzvereinbarung) vorgestellt, das europäischen Software-Entwicklern die Übertragung zeitlich unbegrenzter exklusiver Nutzungsrechte und juristischer Vertretungsrechte an die FSF ermöglicht. Hierauf wird in dieser Arbeit noch näher eingegangen³³.

Richard Stallman ist trotz seiner zahlreichen Beiträge zur Freien Software eher umstritten in der Open Source Gemeinschaft, denn er polarisiert mit seiner Meinungen, indem er die Open Source Bewegung deutlich von der der Freien Software-Bewegung abgrenzt³⁴. Unter anderem unterstellt er der Open Source

³² <http://www.fsf.org>.

³³ Vgl Kap 5.4.1.2. zu Ziffer 10 GPL.

³⁴ <http://www.slackware.com/book/index.php?source=x68.html> und <http://www.fsf.org/licensing/essays/free-software-for-freedom.html/view?searchterm=freedom>.

Bewegung, zum Zwecke größerer Akzeptanz in der Wirtschaft, die Freiheit als argumentative Grundlage zu vernachlässigen und sich nur auf Vorteile im Entwicklungsmodell oder die technische Überlegenheit der einzelnen Programme zu beschränken. Daher wird er in der Open Source Bewegung des Öfteren als zu radikal kritisiert. Im Endeffekt arbeiten beide Bewegungen in dieselbe Richtung.

2.3. „Freie Software und “Open Source Software“

“There's a difference between wanting to write RMS [Stallman] out of history (which nobody wants to do) and wanting to shut out his rhetoric and tactics in the present (which many people do desire). Many of us count him as a personal friend (I personally have known and valued him for longer than the GNU project has existed) but find that his public behaviour continually exasperates us.”

- *Eric S. Raymond*³⁵

„Open Source“ ist der englische Ausdruck für „quelloffen“. Einerseits kann das bedeuten, dass der Quelltext eines Programms frei erhältlich ist andererseits kann „offene Quelle“ auch bedeuten, dass ein Werk frei zur Verfügung steht. Software gilt als Open Source, wenn sie bestimmte Kriterien erfüllt³⁶, die in ihrer jeweiligen Open Source Lizenz geregelt sind.

Eng verbunden mit der Geschichte von Open Source ist – wie erwähnt³⁷ – die Geschichte von Freier Software und oft führt die falsche Verwendung beider Begriffe zu Missverständnissen³⁸.

³⁵ *Raymond* in *Leonard*, The Richard Stallman Saga, <http://archive.salon.com/21st/feature/1998/09/11feature2.html>.

³⁶ Vgl Kap 2.3.7.

³⁷ Vgl Kap 2.1.1.

³⁸ <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

2.3.1. Die Abspaltung von Richard Stallman

Ursprünglich unterschied sich der Begriff Open Source Software nicht von jenem Begriff, der mit Freier Software verbunden wurde. Doch, wie bereits erwähnt³⁹, richtete sich die Gruppierung um *Stallman* völlig gegen die wirtschaftliche Ausrichtung der gewinnorientierten Softwareunternehmen und sah in ihnen geldgierige, kapitalistische Herrscher, während die Softwareindustrie die Anhänger der Freien Software wiederum als fanatische und kommunistische Idealisten darstellte. Somit sorgte die Entstehung der Freien Software für erheblichen Zündstoff.

Viele Anhänger von Freier Software, denen *Stallmans* Ansichten entweder zu weit oder in die falsche Richtung gingen, distanzierten sich. Für sie war *Stallman* „eine Peinlichkeit und ein Hindernis. Er muss um jeden Preis in ein Hinterzimmer verbannt werden, bevor er Investoren verschreckt“⁴⁰. Um mit den Investoren einen Waffenstillstand zu schließen und um Gerüchte und Missverständnisse, die um Freie Software kursierten, aus der Welt zu schaffen⁴¹, begann die Suche nach einem neuen Terminus. Diese Suche nach einer neuen Sprachpolitik kann durchaus als „Vatermord“⁴² an der Person *Richard Stallmans* durch die neue Generation gesehen werden.⁴³

³⁹ Vgl Kap 2.2.

⁴⁰ *Leonard*, The Saint of the Free Software, http://archive.salon.com/21st/feature/1998/08/cov_31feature2.html.

⁴¹ *Sieckmann*, Bravehack (2001) 6.

⁴² *Grassmuck*, Freie Software – Zwischen Privat- und Gemeineigentum (2002) 231.

⁴³ Siehe auch <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

2.3.2. “The Cathedral and the Bazaar”

“The fact that this bazaar style seemed to work, and work well, came as a distinct shock.”

- *Eric S. Raymond*⁴⁴

Das Werk, das im Zusammenhang mit der Suche nach einer neuen Definition am häufigsten genannt wird, ist der Aufsatz von *Eric S. Raymond*⁴⁵ „*The Cathedral and the Bazaar*“ aus dem Jahr 1998. *Raymond*, der auch unter dem Kürzel „*ESR*“ in der Hacker- und Open Source Szene bekannt ist, stellte in diesem Aufsatz ein zentral gelenktes Projekt – wie es in den meisten Softwarefirmen der Fall ist – als Kathedralenbau dar, dem gegenüber die kreativen Selbstorganisationen und die Interaktionen der Freien Softwaregemeinde in Form eines Basars stehen. Während beim „Kathedralenbau“ kleine Projektgruppen zum Einsatz kommen, zeichnet sich das „Basarmodell“ für *Raymond* durch kurze Intervallzeiten sowie einen hochwertigen „peer-review“-Prozess und konstruktives Feedback aus⁴⁶. *Raymond* war vor allem von dem 1991 erstmals vorgestellten Betriebssystem Linux und der Entwicklungsweise des Linux-Erfinders *Linus Torvalds* fasziniert. Dieser veröffentlichte seine neuen Entwicklungen sooft wie möglich – teilweise mehrmals am Tag – und delegierte dabei nötige Arbeitsschritte an andere Mitglieder der „Linux-Community“. *Raymond* testete nun selbst diese Vorgangsweise, indem er andere User einlud, mit ihm zusammen an einem Programm⁴⁷ zu schreiben. Dieses Projekt überzeugte: *Raymond* bekam meist nicht nur Nachrichten, in denen ihm Fehler im Programm mitgeteilt wurden, oft wurde ihm auch schon die Lösung für diese Fehler mitgeschickt.

⁴⁴ *Raymond*, *The Cathedral and the Bazaar*,
http://www.firstmonday.org/issues/issue3_3/raymond/index.html.

⁴⁵ *Raymond*, *The Cathedral and the Bazaar*,
http://www.firstmonday.org/issues/issue3_3/raymond/index.html.

⁴⁶ Aus der Filmdokumentation “*Revolution OS*” (©Copyright 2002 Wonderview Productions, LLC All Rights Reserved).

⁴⁷ Dieses Programm ist unter dem Namen „*fetchmail*“ bekannt und auch heute noch ein beliebtes email-Weiterleitungsprogramm.

2.3.3. Bill Gates und die Halloweendokumente

Im Jahr 1998⁴⁸ wurden die Befürchtungen von *Bill Gates* erneut deutlich, als *Eric S. Raymond* über eine unbekannte Quelle interne Strategiepapiere von Microsoft erhielt, die sich mit der Gefährdung befassen, die Open Source Software auf Microsoft-Produkte ausüben könnten. Raymond veröffentlichte diese Dokumente – mit Kommentaren versehen – am Halloweenwochenende 1998, weswegen sie heute als „Halloweendokumente“ bezeichnet werden. Diese Dokumente beschreiben, dass Open Source Software - wie bspw Linux - eine immer stärkere Position auf dem Softwaremarkt einnimmt und das Microsoft aufpassen müsse, wenn es seine Vormachtstellung hier nicht verlieren wolle. Auch werden Taktiken und Mittel diskutiert, um dieser Strömung Einhalt zu gebieten. Und obwohl nicht weiter darauf eingegangen werden soll, zeigt es doch, dass Open Source Software immer ernster genommen wurde.

2.3.4. Linux

“I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones.”

*Aus einer email von **Linus Torvalds** vom 25.08.1991⁴⁹*

Wer Linux sagt, meint oft Open Source und umgekehrt. Das Betriebssystem, das – nach seinem Kernel⁵⁰ Linux benannt – weltweit bekannt wurde, steht nicht nur für die populärste Open Source Software, sondern auch für die Etablierung freier Software in der Welt der Computernutzer. Deswegen wird

⁴⁸ <http://www.opensource.org/halloween/>.

<http://www.linux-magazin.de/Artikel/ausgabe/1999/01/Halloween/halloween.html>

⁴⁹ http://groups.google.de/groups?as_umsgid=1991Aug25.205708.9541@klaava.Helsinki.FI.

⁵⁰ Ein „Kernel“ – oder auf Deutsch auch Kern – ist die wichtigste Komponente eines Betriebssystems. In ihm ist die Prozess- und Datenorganisation festgelegt, auf der alle weiteren Softwarebestandteile des Betriebssystems aufbauen.

an dieser Stelle kurz verdeutlicht, dass Linux tatsächlich nur ein Teil von Open Source ist.

Linux wurde am 17. September 1991⁵¹ in seiner ersten öffentlichen Version 0.01 von dem damals 21 Jahre alten finnischen Studenten Linus Benedict Torvalds freigegeben, der – aufbauend auf dem Betriebssystem Minix – versuchte, ein neues Betriebssystem zu entwickeln. Linux hatte bis dahin ca. 10.000 Zeilen an Code und einen Benutzer: Linus Torvalds. 1993 war der Code auf 100.000 Zeilen angewachsen und hatte schon über 10.000 Benutzer. 1997 waren es 800.000 Zeilen Code und über 3.5 Millionen Benutzer. 1999 benutzten ca. 12 Millionen Menschen Linux. Heute wird die Zahl auf etwa 18 Millionen geschätzt⁵².

Dabei ist es wichtig zu wissen, dass Linux im Gegensatz zu Windows nicht in einer Distribution vorliegt, sondern in vielen verschiedenen. Die beliebtesten darunter sind Mandrakelinux, Ubuntu, Fedora, SUSE, Debian, Gentoo, Slackware, Knoppix, MEPIS und Xandros Desktop⁵³. Diese Vielzahl erklärt sich aus der Freiheit, das Betriebssystem zu verändern. Dennoch ist Linux nur ein Betriebssystem, das neben den unzählbaren anderen Open Source Anwendungen – wie beispielsweise Serverapplikationen und Anwendungsprogrammen – steht. *Linus Torvalds* selbst beschreibt es treffend, wenn er meint, dass niemand wirklich mit einem Betriebssystem arbeitet. Benutzer arbeiten mit Programmen, die auf einem Betriebssystem laufen und für die das Betriebssystem die nötigen Ressourcen zur Verfügung stellt. Das Betriebssystem selbst nutzt aber kaum jemand.

⁵¹ <http://www.li.org/linuxhistory.php>.

⁵² <http://counter.li.org/>.

⁵³ <http://distrowatch.com/>.

2.3.5. Aus Netscape wird Mozilla

“Software is like sex: it's better when it's free.”

- Linus Torvalds

1995 begann der erste „Browser-Krieg“, der bis 1998 andauern sollte⁵⁴. Da Microsoft die Entwicklung des Internets unterschätzt hatte, beherrschte zunächst die Firma Netscape mit ihrem Netscape Navigator den Markt für Internetbrowser. Microsoft versuchte daraufhin diesen Markt zu erobern, indem es 1995 die erste Version des Internet Explorers auf den Markt brachte. Dabei hatte Microsoft einerseits den Vorteil, über wesentlich mehr Geld zu verfügen und andererseits den Browser gebündelt mit seinem eigenen Betriebssystem Windows ausliefern zu können, verbunden mit der Hoffnung, was einmal installiert ist, wird benutzt werden. Netscape wurde in Folge von Microsofts Internet Explorer verdrängt und verlor seine Marktvorherrschaft⁵⁵. 1998 kapitulierte Netscape schließlich und gab, inspiriert von *Eric S. Raymonds* „Cathedral and the Bazaar“, den Quellcode des Netscape Navigators frei⁵⁶. Aus dem Netscape-Navigator wurde die Mozilla Foundation⁵⁷, die heute mit dem Mozilla Firefox versucht, als Open Source Projekt die momentane Marktherrschaft des Internet Explorers anzugreifen.

Die Freigabe des Quellcodes war juristisch nicht ganz einfach. Dies zeigt die Tatsache, dass dafür zwei neue Lizenzen, die Netscape Public License (NPL) und die Mozilla Public License (MPL) entwickelt wurden, auf die im weiteren Verlauf der Arbeit eingegangen wird.

⁵⁴ http://news.zdnet.com/2100-3513_22-996866.html.

⁵⁵ “*Memoirs from the browserwars*”, http://software.ericssink.com/Browser_Wars.html.

⁵⁶ Epilog: Netscape Embraces the Bazaar, <http://www.catb.org/%7Eesr/writings/cathedral-bazaar/cathedral-bazaar/ar01s13.html>; siehe auch: <http://wp.netscape.com/newsref/pr/newsrelease558.html>.

⁵⁷ <http://www.mozilla.org/about/>; siehe auch Kap 5.6.1.

2.3.6. Der Begriff Open Source wird geprägt

„Open Source promotes software reliability and quality by supporting independent peer review and rapid evolution of Source code.“

„The one-sentence version“, der **OSI**⁵⁸

Angespornt von dieser Entwicklung beschloss die neue Generation⁵⁹, eine neue Marketingstrategie für die Freie Software-Gemeinde zu suchen und einen Ersatz für den 1984 geprägten Begriff „Free Software“ zu finden. Einerseits, um sich von *Stallman* loszusagen und den Begriff im Wettbewerb mit proprietärer Software als geschäftsfreundlich und weniger ideologisch belastet darstellen zu können andererseits, um den Begriff näher zu definieren, denn für viele Menschen war der Begriff Freie Software neu und für nicht wenige hatte „frei“ die Bedeutung „kostenlos“.⁶⁰

Im Februar 1998 gründete *Bruce Perens*⁶¹ zusammen mit *Eric S. Raymond* die Open Source Initiative (OSI). Auf dem Gründungstreffen am 3. Februar 1998 in Palo Alto, Kalifornien, an dem neben *Raymond* und *Perens* auch *Todd Anderson*, *Chris Peterson*⁶², *John "maddog" Hall*, *Larry Augustin* (beide für Linux International) und *Sam Ockman* (von der Silicon Valley Linux User's Group) teilnahmen, wurde der Begriff Open Source erdacht⁶³ und in Folge von *Bruce Perens* als Warenzeichen⁶⁴ angemeldet. Der Begriff wurde zum Erfolg und dankbar von Medien und Wirtschaft aufgenommen.

⁵⁸ <http://www.opensource.org/advocacy/faq.php>.

⁵⁹ Siehe auch Kapitel 2.3.

⁶⁰ Das Online-Englishwörterbuch Leo verzeichnet 30 verschiedene Bedeutungen für „free“ (bzw Übersetzungen von „free“). Nur zwei davon sind „gratis bzw umsonst“. Der Rest bezieht sich auf Freiheit und das Fehlen von Zwängen.

⁶¹ *Bruce Perens* war damals Leiter des Debian-Projektes, ein Betriebssystem, welches auf dem Linuxkern aufbaut.

⁶² <http://www.foresight.org/FI/Peterson.html>.

⁶³ *Christine Peterson* soll die Urheberin des Begriffes sein.

⁶⁴ Zur markenrechtlichen Behandlung von „Open Source“ siehe Kap 7.1.3.

Zum näheren Verständnis des Begriffs schuf *Perens* die „Open Source Definition“ (OSD), eine Richtlinie, die zur Bewertung von Lizenzen aus dem Bereich der Open-Source Software und der freien Software dienen soll⁶⁵. *Perens* hatte schon in den „Debian Free Software Guidelines (DFSG)“⁶⁶ versucht, die Langtextversion der FSF über die Definition von Freier Software in eine praktikablere Fassung zu bringen. *Perens* sah sich angesichts der Vielzahl an Nachbarlizenzen herausgefordert, genauer zu definieren, was denn die Freiheit sei, die Debian meinte. Die OSD entstand direkt aus dem Geist der DFSG und wird im Folgenden kurz vorgestellt.

2.3.7. Die „Open Source Definition“

Die Open Source Definition (OSD) ist keine Lizenz, sondern ein Standard, an dem sich Lizenzen messen lassen. Allerdings ist sie in der fast wortgleichen Fassung der DFSG die Lizenz von Debian GNU/Linux. Die OSD soll herangezogen werden, um die Kriterien aufzuzeigen, die erfüllt sein müssen, damit eine Software als Open Source iSd OSD bezeichnet werden darf⁶⁷:

1. Freie Weiterverbreitung

Die Lizenz muss die uneingeschränkte Weitergabe der Software erlauben. Deshalb darf für die Nutzung der Software auch keinerlei Lizenz- oder andersartige Gebühr verlangt werden.

⁶⁵ <http://www.opensource.org/docs/definition.php>.

⁶⁶ http://www.debian.org/social_contract#guidelines.

⁶⁷ http://www.opensource.org/docs/definition_plain.php.

2. Quellcode

Das Programm muss im Quellcode vorliegen und sowohl die Verbreitung als Quellcode als auch in kompilierter Form gestatten. Wird ein Teil des Produkts nicht mit Quellcode verbreitet, muss darauf hingewiesen werden, dass der Quellcode im Internet kostenlos heruntergeladen werden kann. Ein Programmierer soll den Quellcode verändern können. Deshalb ist ein absichtlich verwirrend geschriebener Quellcode nicht erlaubt. Ebenso sind Zwischenformen wie die Ausgabe eines Präprozessors oder eines Übersetzers verboten. Diese Vorgabe ist grundlegend, weil man Programme ohne den Zugriff auf einen verständlichen Quellcode nicht weiterentwickeln und verändern kann. Da es Ziel der OSI ist, die Entwicklung so einfach wie möglich zu machen, muss auch die Veränderung des Codes einfach sein.

3. Auf dem Programm basierende Werke

Die Lizenz muss die Weiterentwicklung bzw Veränderung des Programms erlauben. Ebenso muss die Lizenz die Weitergabe und Verbreitung unter gleichen Lizenzbedingungen gestatten. Die Weitergabe und Verbreitung unter gleichen Lizenzbestimmungen ist eine „Kann-Bestimmung“ wie schon *Jaeger/Metzger* richtig bemerken⁶⁸, da diese nicht verpflichtend ist. Auch reicht allein die Möglichkeit den Quellcode zu lesen nicht aus, um eine von der OSI gewünschte unabhängige Prüfung und evolutionsähnliche Auslese zu erreichen.

4. Die Unversehrtheit des Originalcodes

Eine Einschränkung der Verbreitung von modifiziertem Quellcode durch die Lizenz ist nur erlaubt, wenn sie die Verbreitung von so genannten „Patch-dateien“ in Verbindung mit dem Originalcode gestattet, damit das Programm vor der Benutzung verändert werden kann. Die Lizenz muss ausdrücklich die

⁶⁸ *Jaeger/Metzger*, Open Source Software – Rechtliche Rahmenbedingungen der Freien Software (2002) 2.

Verbreitung von Software erlauben, die mit verändertem Quellcode erstellt wurde. Auch darf sie von auf dem Programm basierenden Werken verlangen, einen von der Originalsoftware unterschiedlichen Namen oder eine andere Versionsnummer zu tragen. Die Benutzer der Software sollen wissen, wer für das Produkt verantwortlich ist. Genau wie die Autoren das Recht haben darüber in Kenntnis zu sein, welche Software sie unterstützen sollen, um dadurch ihren guten Ruf zu schützen.

5. Keine Diskriminierung von einzelnen Personen oder Gruppen

Die Lizenz darf keinerlei Personen oder Personengruppen diskriminieren. Aus diesem Grund verbietet die OSI jeder Open Source Lizenz, jemanden aus dem Verfahren auszuschließen.

6. Keine Einschränkungen für bestimmte Anwendungsbereiche

Die Lizenz darf niemanden in der Benutzung des Programms in einem bestimmten Einsatzgebiet einschränken. So darf sie beispielsweise nicht die kommerzielle Nutzung oder die Benutzung in der Genforschung verbieten. Während sich also Z 5 der OSD gegen die Diskriminierung von Personengruppen richtet, liegt das Hauptaugenmerk von Z 6 darauf, zu verhindern, dass Klauseln einer Lizenz die kommerzielle Nutzung des Programms verbieten.

7. Verbreitung der Lizenz

Die zum Programm gehörigen Rechte müssen für jeden gelten, der das Programm erhalten hat. Die Verknüpfung der Lizenz mit einer anderen ist unzulässig. Damit soll verhindert werden, dass durch indirekte Mittel - wie das Verlangen eines Einverständnisses, die Software nicht offen weiterzugeben - die Lizenz umgangen wird.

8. Die Lizenz darf nicht für ein bestimmtes Produkt gelten

Die Rechte, welche die Lizenz gewährt, dürfen nicht davon abhängen, dass das Programm Teil einer bestimmten Softwaredistribution ist. Wird das Programm außerhalb einer solchen Distribution genutzt oder verbreitet, gelten für den Benutzer dieselben Rechte, die in der Originaldistribution gewährt werden.

9. Die Lizenz darf andere Software nicht beeinträchtigen

Die Lizenz darf keine andere Software einschränken, die zusammen mit der lizenzierten Software verbreitet wird. Die Lizenz darf beispielsweise nicht verlangen, dass jegliche Software, die auf demselben Datenträger verbreitet wird, Open Source Software sein muss. Damit will die OSI klarstellen, dass jeder, der Open Source Software benutzen oder weiterverbreiten möchte, das Recht hat, sich selbst seine eigene Software auszusuchen.

10. Die Lizenz muss technologieneutral sein

Keine Bestimmung der Lizenz darf den Einsatz der Software auf irgendeine einzelne Technologie oder auf eine Art der Schnittstelle beschränken.

Obwohl die Auseinandersetzung über die Begriffe Open Source und Freie Software zu einer Spaltung im Lager der Freien Software-Gemeinde geführt hat, muss betont werden, dass beide Lager in den Grundsätzen von Freier und Open Source Software übereinstimmen. Auch in dieser Arbeit sollen daher beide Begriffe gleichbedeutend verwendet werden. Nachfolgend wird eine Abgrenzung gegenüber verschiedenen anderen Softwarevarianten erläutert.

2.4. Abgrenzung verschiedener Softwarevarianten

Heute ist Open Source ein anerkannter und oft benutzter Begriff, der (fast immer) für kostenlose, qualitativ hochwertige, frei verfügbare, modifizierbare und weiterverbreitbare Software steht. Am besten einordnen lässt sich der Begriff, wenn man ihn gegenüber anderen Softwarevarianten abgrenzt⁶⁹, wobei die Grenzen oft nicht genau gezogen werden können, da sie fließend sind. Wichtig ist auch, dass sich einzelne Varianten nicht unbedingt ausschließen müssen. Folgende Abbildung der FSF soll dies verdeutlichen.

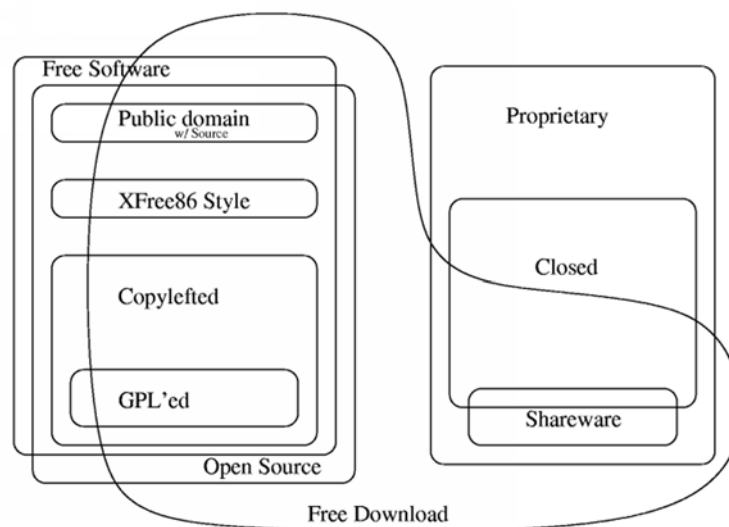


Abb 4⁷⁰

Die Abbildung zeigt, dass eine genaue Abgrenzung nicht immer möglich ist und oftmals im Einzelfall zu beurteilen ist. Interessant ist auch die Überschneidung von Open Source und Free Software.

⁶⁹ Siehe auch <http://www.fsf.org/licensing/essays/categories.html>.

⁷⁰ <http://www.fsf.org/licensing/essays/category.png>.

2.4.1. Freie Software

„Jim Prendergast's recent article mistakenly called me a "leader in the open Source community". While I appreciate the praise that might be read into that expression, it is not the case: I do not advocate "Open Source" and never did.“

aus **Richard Stallmans** Blog⁷¹

Nach *Stallmans* Definition lässt sich Freie Software in verschiedene Stufen aufteilen⁷²:

- Stufe 0** Die Freiheit, das Programm zu jedem Zwecke auszuführen.
- Stufe 1** Die Freiheit, die Funktionsweise des Programms zu studieren, Voraussetzung dafür ist der Zugang zum Quellcode.
- Stufe 2** Die Freiheit, Kopien des Programms anzufertigen und zu verbreiten.
- Stufe 3** Die Freiheit, das Programm an eigene Bedürfnisse anzupassen und/oder zu verbessern und dies dann zu veröffentlichen.

Freie Software kann verkauft werden, muss also nicht zwangsweise kostenlos sein⁷³. Jeder kann die erhaltene Software kopieren und weitergeben. Eine etwaige Restriktion gegen diese Verfahrensweise ist mit dem Begriff der Freien Software nicht vereinbar. Diese Stufen, die im Langtext vorliegen, bildeten die Grundlage für die Open Source Definition⁷⁴ und in weiterer Folge für die wohl wichtigste und am weitesten verbreitete Lizenz im Open Source Bereich, die GNU GPL.

⁷¹ <http://www.fsf.org/blogs/rms/entry-20050130>.

⁷² <http://www.fsf.org/licensing/essays/free-sw.html>.

⁷³ <http://www.fsf.org/licensing/essays/free-sw.html>.

⁷⁴ Vgl Kap 2.3.7.

2.4.2. Public Domain

Die rechtliche Bedeutung von Public Domain (engl. für „gemeinfrei“) beruht auf der Tatsache, dass in den USA viele Softwareprojekte an Universitäten mit Hilfe von öffentlichen Geldern gefördert wurden und diese Projekte danach nicht mehr verkauft werden durften, sondern zum Allgemeingut wurden⁷⁵. In Österreich gibt es keine "echte" Public Domain wie in den USA, wo der Begriff seinen Ursprung hat und wo auf alle Rechte verzichtet werden kann. Hat sich der Urheber eines Werkes, das nicht unbedingt Software sein muss, dazu entschieden, sein Werk in die Public Domain zu stellen, tritt er alle seine Rechte und Ansprüche an die Allgemeinheit ab. Das Werk unterliegt dann keinem Copyright oder anderen rechtlich bindenden (Lizenz-)Bedingungen mehr. Dies ist in Österreich nicht möglich⁷⁶.

Die Berner Konvention⁷⁷ schützt jedes niedergeschriebene Werk urheberrechtlich. Software wird dabei zu den literarischen Werken gerechnet, was zur Folge hat, dass in den 160 Staaten⁷⁸, die die Konvention unterzeichnet haben, besondere rechtliche Schritte nötig sind, um ein Werk in die Public Domain zu stellen, wenn es nicht von der jeweiligen nationalen Rechtsordnung sowieso ausgeschlossen ist. Will jemand sein Werk in Österreich der Allgemeinheit übergeben, so kann er aber gem § 24 UrhG Dritten eine Werknutzungsbewilligung oder ein Werknutzungsrecht einräumen. Die Gemeinfreiheit bezieht sich also immer auf die jeweilige nationale Rechtsordnung. So sind beispielsweise Fotos von US-Regierungsbehörden, die

⁷⁵ *Siepmann*, Lizenz- und haftungsrechtliche Fragen bei der kommerziellen Nutzung freier Software, <http://jura.uni-sb.de/jurpc/aufsatz/19990163.htm>.

⁷⁶ § 19 Abs 2 UrhG bestimmt, dass ein solcher Verzicht unwirksam ist. Auf diesen Punkt wird in weiterer Folge der Arbeit im Rahmen der Urheberpersönlichkeitsrechte noch genau eingegangen.

⁷⁷ http://www.wipo.int/treaties/en/ip/berne/trtdocs_wo001.html#P82_10336.

⁷⁸ Stand: 20.03.2006.

in den USA keinem Copyright unterliegen, in Österreich urheberrechtlich geschützt.⁷⁹

Relevant für Open Source Lizenzen ist, das was einmal in die Public Domain gegeben wurde, nicht verpflichtend dort verbleiben muss. Es ist sogar möglich, das Werk aus der Public Domain zu nehmen und es bspw in nicht-freie Software umzuwandeln. Deshalb steuern die Open Source Lizenzen diesem Effekt entgegen, indem sie bestimmen, dass das, was einmal frei ist auch frei bleiben muss. Diese Vorgangsweise beschreiben die Begriffe copyleft und non-copyleft, die im Folgenden besprochen werden.

2.4.3. „Copyleft“ und „Non-Copyleft“-Software

„Copyleft“⁸⁰ ist ein Schutzverfahren in bestimmten Lizenzen freier Software, das einen bestimmten Aspekt des Urheberrechts in sein Gegenteil zu verkehren versucht. Das „Copyleft“ erzwingt die Freiheit von Weiterbearbeitungen und Fortentwicklungen eines freien (Ur-)Werkes, um dadurch die (unfreie) Vereinnahmung freier Werke – wie bspw Public Domain-Werke – zu verhindern.⁸¹



Abb. 5

Bearbeitet jemand mit Erlaubnis das Werk eines anderen Urhebers, erhält er gem. § 11 UrhG als Miturheber ein Mitspracherecht, darüber zu entscheiden, wie die Bearbeitung verwendet werden darf. War das ursprüngliche Werk bspw noch für jeden frei kopierbar,

Abb 5: Das Logo zeigt deutlich die absicht-liche Verbundenheit mit dem Copyrightzeichen.

⁷⁹ <http://www.unc.edu/~uncclng/public-d.htm>.

⁸⁰ <http://www.fsf.org/licensing/essays/copyleft.html>.

⁸¹ Siehe auch *Jaeger*, Copyright oder Copyleft, Computerwoche Spezial 4/2000, 36 = http://www.ifross.de/ifross_html/art6.html.

verteilbar oder veränderbar, so übertragen sich diese Freiheiten nicht automatisch auf die Bearbeitung. Diese Auswirkung versuchen die Copyleft-Lizenzen umzukehren, indem der ursprüngliche Autor nur dann die Weitergabe der Bearbeitungen erlaubt, wenn sie auch zu den gleichen Bedingungen an jeden Dritten lizenziert werden.

Non-Copyleft Software dagegen kommt vom Urheber mit der Erlaubnis, sie weiterzuverbreiten und zu verändern und zusätzliche Einschränkungen hinzuzufügen.⁸² Bei Non-Copyleft Software ist es möglich, dass Kopien oder veränderte Versionen nicht mehr frei sind, so dass bspw eine Softwarefirma das Programm mit oder ohne Veränderungen kompilieren und die ausführbare Datei als proprietäres Softwareprodukt vertreiben kann. Das X Window System⁸³ ist bspw solch ein Fall. Das X-Konsortium gibt seine Software X11 unter Lizenzbedingungen⁸⁴ frei, die nicht unter Copyleft fallen. Prinzipiell ist jeder berechtigt, kostenlos die Software herunterzuladen und zu bearbeiten. Es gibt auch unfreie Versionen der X11-Software und für viele gängige Workstations und PC Grafikkarten ist eine der unfreien Versionen die einzig funktionsfähige. Damit ist X11 also - insgesamt gesehen - keine freie Software mehr.

Durch das Copyleft der GPL bspw wurden Menschen veranlasst, freie Software zu schreiben, die es wahrscheinlich sonst nie getan hätten. Doch auch wenn das Copyleft-Verfahren bei Open Source Software gute Dienste leistet, findet es nicht in allen Open Source Lizenzen Deckung, da unterschiedliche Copyleft-Lizenzen meist inkompatibel zueinander sind. Warum dies so ist, welche Probleme noch im Zusammenhang mit Copyleft auftreten können und wie eine mögliche Lösung aussehen kann, wird in Kapitel 5 erklärt.

⁸² <http://www.gnu.org/philosophy/categories.de.html#ProprietarySoftware>.

⁸³ <http://www.x.org/>.

2.4.4. Microsoft Shared Source⁸⁵

„In my opinion, the GPL is optimized to build a strong software community at the expense of a strong commercial software business model.“

- *Craig Mundie, Microsoft CTO*⁸⁶

Diesen Abschnitt könnte man auch mit dem Titel überschreiben: „*The Empire strikes back*“. Der von vielen Open Source Anhängern ausgemachte „Feind“ der Bewegung, Microsoft, hat im Mai 2001 selbst erste Schritte in Richtung Offenlegung des Quelltextes unternommen. Aber auch in der Shared Source Initiative wird Open Source als Behinderung des Fortschritts und der Entwicklung angesehen⁸⁷. Das Microsoft Shared Source Programm bezog sich in seinem Ursprung vor allem auf das Betriebssystem Windows CE. Microsoft stellte dieses unter die Microsoft Shared License⁸⁸. Damit wurde dem Benutzer gestattet, das Programm zu verändern und in der veränderten Version zum nicht-kommerziellen Gebrauch weiterzugeben. *Eric S. Raymond* hat diese Idee folgendermaßen zusammengefasst: *"Man muss für das Privileg bezahlen, die Quelltexte der Software zu sehen. Wer Verbesserungen einbringt, wird dafür weder bezahlt noch erhält er irgendwelche Rechte an seinen Verbesserungen. Und wenn jemand versucht, zu verwenden, was er beim Fixen Microsoft'scher Bugs gelernt hat, verstößt er gegen Microsofts Rechte an dessen geistigen Eigentum."* Kurzum: *"Was uns gehört, bleibt unser. Und wenn du bei Shared Source kooperierst, wird dein Beitrag auch uns gehören"*⁸⁹. Shared Source gehört damit in keiner Form zu einem Ableger von Open Source. Ebenso wenig sollte es als eine Art Kompromisslösung, eher als gescheiterter Angriffsversuch auf die

⁸⁴ http://www.x.org/Downloads_terms.html.

⁸⁵ <http://www.microsoft.com/resources/sharedsource/default.aspx>.

⁸⁶ http://www.microsoft.com/resources/sharedsource/Initiative/speeches/mundie_model.aspx.

⁸⁷ <http://www.heise.de/newsticker/meldung/17553>.

⁸⁸ <http://msdn.microsoft.com/embedded/usewinemb/ce/sharedsrccode/eula/default.aspx>.

⁸⁹ <http://www.heise.de/newsticker/meldung/17857>.

Open Source Bewegung gesehen werden. Dennoch hat Microsoft seit 2001 einige weitere Programme mit Shared Source Lizenzen ausgestattet.⁹⁰

2.4.5. Freeware, Donationware und Cardware

„Freeware is... free, just as you'd imagine it!“

- Aus der *Freewaredefinition* von *tucows*⁹¹

Auch wenn es keine eindeutige rechtsgültige Definition für Freeware gibt, darf Freeware keinesfalls mit Free Software oder Freier Software verwechselt werden. Freeware (engl: "free ware": "freie Ware") bezeichnet im allgemeinen Sprachgebrauch kommerzielle Software, die vom Urheber zwar zur kostenlosen Nutzung zur Verfügung gestellt wird und meistens weitgehend ohne Einschränkung verbreitet werden darf, bei der aber anhand der Lizenzbedingungen zu prüfen ist, welche konkreten Rechte der Urheber den Anwendern gewährt. Freeware liegt fast nie im Quellcode vor, sodass der einzige Unterschied zu kommerzieller Software darin besteht, dass man sie kostenlos beziehen kann. Beispiel für Freeware ist der Internet Explorer, auch wenn dieser eher selten als Freeware bezeichnet wird, der aber im Gegensatz zu seinem Konkurrenten Mozilla Firefox zwar auch kostenlos bezogen werden kann, bei dem der Benutzer aber alleine das Recht zur Benutzung und Weitergabe, nicht aber zur Veränderung hat. Meist sind die Urheber von Freeware auch Soloprogrammierer oder kleine Programmiergemeinschaften, die aus Eigeninitiative so genannte „Tools“ schreiben und diese bspw über Tucows anbieten. Gefällt dem Benutzer das Programm, steht ihm meist die Möglichkeit frei - wie bspw bei Bittorent⁹² - einen selbst zu bemessenden Betrag an den oder die Urheber zu spenden (im englischen: „to donate“). In diesem Fall spricht man auch von Donationware.

⁹⁰ <http://www.microsoft.com/resources/sharedsource/Licensing/default.msp>.

⁹¹ <http://www.tucows.com>.

⁹² <http://www.bittorent.com>.

Bei der Cardware handelt es sich ebenfalls um Freeware. Der oder die Urheber erwarten sich bei Benutzung lediglich eine Ansicht(post)karte. Diese dient ihnen als Feedback und Motivation, das Programm weiterzuentwickeln. Außerdem zeigt die Ansichtskarte, wo überall auf der Welt das Programm zum Einsatz kommt.

2.4.6. Shareware und Crippleware

Shareware ist kommerzielle Software, die innerhalb eines bestimmten Zeitraums oder nur für eine bestimmte Anzahl an Programmaufrufen kostenlos getestet werden kann und dann entweder gelöscht oder registriert werden muss. Das Shareware-Prinzip baut hier auf Fairness des Urhebers und des Nutzers auf. Nach der Testphase („trial-time“) darf der Benutzer das Programm nicht mehr verwenden, will er das weiterhin, muss er eine Lizenz erwerben. Darüber hinaus gibt es auch Sharewareversionen, die dem Benutzer erlauben, eine „abgespeckte“ Version des Programms zu benutzen, bei der der Benutzer nur begrenzte Verwendungsmöglichkeiten hat. Diese Versionen werden auch Crippleware oder Liteware genannt. Sharewareprogramme liegen fast nie im Quellcode vor, sodass hier auch nicht von Open Source Software ausgegangen werden darf.

2.4.7. Kommerzielle und Proprietäre Software

„Making free software affordable.“

Der ehemalige Slogan von *“Cygnus Solutions“*

Kommerzielle und proprietäre Software ist nicht unbedingt das Gleiche. Zwar ist fast jede kommerzielle Software auch proprietär, aber es gibt Ausnahmen. So gibt es bspw kommerzielle freie Software und nicht-kommerzielle proprietäre Software.

Kommerzielle Software ist Software, die von einer Firma mit dem Ziel entwickelt wird, aus der Benutzung dieser Software Profit zu erwirtschaften (bspw SAP). Es gibt oder gab, wie bereits angeführt auch Beispiele, bei denen Firmen mit freier Software Geld verdienen oder verdient haben, die sie kommerziell benutzen bzw. benutzt haben. Pionier auf diesem Gebiet war die Firma Cygnus Solutions, die 1989 gegründet und Ende 1999 von Red Hat für über 670 Millionen Dollar aufgekauft wurde⁹³. Cygnus Solutions war ein führendes Unternehmen im Bereich frei verfügbarer Software-Lösungen und der Entwicklung von Internet-Software und bot Entwicklungsplattformen auf Linux-Basis an. Entwicklungstools von Cygnus unterstützten mehr als 125 unterschiedliche Umgebungen für Embedded-Mikroprozessoren und wurden weltweit von namhaften Firmen eingesetzt. Zur Produktpalette von Cygnus Solutions gehörten Entwicklungswerkzeuge und -services für Desktops unterschiedlicher Hersteller, Client/Server-, Echtzeit-(RTOS) und Embedded-Systeme sowie Handheld-Geräte.

Das Beispiel zeigt, dass Open Source wirtschaftlich den Durchbruch geschafft hat. Viele Firmen haben die Idee von Cygnus Solutions aufgegriffen und verdienen mit dem Support und der Migrierung von Open Source Software ihr Geld. Auf die wirtschaftliche Komponente, die neben dem Dienstleistungssektor auch die Distributionen und „*Embedded Systems*“ umfasst, soll im Rahmen dieser Arbeit aber nicht weiter eingegangen werden, da dies zu weit führen würde⁹⁴.

Proprietäre Software kann als das komplette Gegenteil von Open Source Software gesehen werden. Der Begriff proprietär ist allerdings nicht eindeutig

⁹³ <http://www.redhat.com/about/corporate/timeline.html>.

⁹⁴ Siehe aber bspw *Stallman*, *Voices from the Open Source Revolution* (1999); <http://www.oreilly.com/catalog/opensources/book/toc.html>, <http://www.heise.de/newsticker/meldung/38206> ; http://www.ecomm-online.de/fileadmin/template-ecomm/uploads/uin/061103_ecomm_linux.pdf.

besetzt, eher veraltet und ieS auch inkorrekt⁹⁵. Im juristischen Sinne darf der Begriff „proprietär“ wohl gleichbedeutend mit „urheberrechtlich geschützt“ gesehen werden, wobei diese Ausführung insofern irreführend ist, als das Open Source Software natürlich auch durch das Urheberrecht geschützt wird. Abweichend davon benutzt die FSF den Begriff für Objekte, die nicht „frei“ sind. „Proprietäre Software“ ist hier jegliche Software, die keine „freie Software“ ist⁹⁶. Mit dem juristischen Begriff hat diese Definition jedoch wenig zu tun: So ist für die FSF bspw ein urheberrechtlich geschütztes Programm, das unter der GPL lizenziert ist, hiernach „frei“, ein urheberrechtlich nicht geschütztes Programm ohne Quellcode hingegen „proprietär“. Da aber der Begriff Closed Source als Gegenteil zu Open Source nicht alternativ benutzt werden kann – es gibt durchaus Software wie bspw die „Microsoft Foundation Classes“, die zwar im Quellcode vorliegen, die aber keinesfalls Open Source sind – wird der Begriff proprietär auch in der Praxis herangezogen um Software zu bezeichnen, die nicht Open Source ist. Proprietäre Software darf nicht modifiziert, vervielfältigt⁹⁷ und weiterverbreitet werden. Der Benutzer wird nie Eigentümer der Software, sondern erhält gegen Entgelt das Recht auf die Benutzung, eine Lizenz.

2.5. Ergebnis

Die Geschichte der Softwareindustrie sollte dazu dienen, ein besseres Verständnis für die Entwicklung von Open Source als eigenständige Softwarevariante zu bekommen und die folgenden Überlegungen zu den Problemstellungen, die Open Source Lizenzen aufwerfen, nachvollziehen zu können.

⁹⁵ Jaeger/Metzger, Open Source Software 3.

⁹⁶ <http://www.fsf.org/licensing/essays/categories.html#TOCProprietarySoftware>.

⁹⁷ Ausgenommen Sicherungskopien zum eigenen Gebrauch.

Die aufgezeigten dogmatischen Unterschiede in den Definitionen von Freier Software und Open Source Software machen deutlich, dass die Thematik ernsthaft diskutiert und weiterentwickelt wird. Sowohl *Richard Stallmann* und die FSF als auch die OSI mögen in Detailfragen unterschiedlicher Ansicht sein, verfolgen jedoch das gleiche Ziel, Software ohne Patentrecht durch die Nutzung der dafür geschaffenen Lizenzen entwickeln, verändern und verbreiten zu dürfen. Die Beurteilung der unterschiedlichen Argumentationen muss dem Einzelnen überlassen bleiben.

Da eine rechtliche Definition von Open Source Software nicht existiert, ist es sinnvoll die OSD als Hilfsmittel zur Einordnung heranzuziehen. Die dort aufgeführten Kriterien, die eine Software zu erfüllen hat, um als Open Source zu gelten, haben sich bislang bei der Einordnung bewährt und werden von den Verfassern von Open Source Lizenzen als richtungweisend akzeptiert.

Des Weiteren wurde deutlich, dass Open Source Software eine wirklich eigenständige Softwarevariante ist. Public Domain Software kann abgewandelt und Teil von nicht freien Programmen werden. Microsoft Shared Source fehlt es nicht nur an Anerkennung, sondern auch an der wirklichen Freiheit der Lizenz. Freeware und Shareware sind zwar preiswerte, benutzerfreundliche und teilweise auch kostenlose Software, da man sie aber nicht weiterentwickeln kann und darf, sind sie keinesfalls Freie Software. Um also frei zu bleiben, muss Software urheberrechtlich geschützt und lizenziert werden, was teilweise durch den angesprochenen Copyleft-Effekt erreicht wird. Im nächsten Kapitel folgen zunächst die urheberrechtlichen Vorgaben des österreichischen Gesetzgebers, bevor auf die einzelnen Problemstellungen der Open Source Lizenzen selbst eingegangen wird.

3. URHEBERRECHTLICHE ASPEKTE

„Und doch wären wir arm dran ohne Urheber. Sie stellen sich, uns und die Welt in Frage. Sie sind uns Stachel im Fleisch, irritierend belebend.“

*Günther Grass*⁹⁸

Als der römische Epigrammatiker *Martial*, der im ersten Jahrhundert n Chr lebte, feststellen musste, dass ein gewisser *Fidentinus* seine Werke als dessen eigene ausgab, verglich er ihn mit einem Plagiator, einem Menschenräuber, und somit den Raub an seiner geistigen Leistung mit dem des Sklavenraubes. Sanktionen hatte *Fidentinus* allerdings keine zu befürchten⁹⁹, da das Urheberrecht, wie wir es heute kennen, erstmals mit der Erfindung des Buchdrucks Eingang in die Gesetzbücher fand¹⁰⁰.

In Österreich findet man das Urheberrecht erst im Patent zum österreichischen Urheberrecht vom 19.10.1846, das bis 1895 in Geltung war¹⁰¹. Das jetzt noch geltende Urheberrechtsgesetz in Österreich stammt aus dem Jahr 1936 und wurde bis heute, bedingt durch technische Neuerungen und den Beitritt zur EU, häufig novelliert. Das Urheberrecht kann als das momentan umfassendste Schutzmerkmal für Software angesehen werden, auch wenn durch die Möglichkeit der Patentierung von Software versucht wurde¹⁰², einen noch stärkeren Schutz aufzubauen¹⁰³. Sieht man sich die wirtschaftliche

⁹⁸ Aus dem Aufsatz „Wir Urheber“, FAZ vom 18.01.2005, 36.

⁹⁹ Das römische Urheberrecht kannte den Schutz der eigenen geistigen Leistung nur als „res extra commercium“.

¹⁰⁰ *Rehbinder*, Urheberrecht: ein Studienbuch¹³ (2004) 7; *Potz*, Urheberrecht aus kirchenrechtlicher Sicht, in *Dittrich* (Hrsg), Woher kommt das Urheberrecht und wohin geht es? (1988) 43; *Junger*, Überblick über das Urheberrecht, <http://remus.jura.uni-sb.de/urheberrecht/gw01.html>.

¹⁰¹ *Dillenz*, Praxiskommentar zum österreichischen Urheberrecht und Verwertungsgesellschaftenrecht (1999) 10.

¹⁰² Bspw FAZ vom 21.01.2005 und 08.02.2005.

¹⁰³ Zur patentrechtlichen Problematik von Open Source-Software siehe Kap 9.

Entwicklung des Internets und des Softwaremarktes genauer an, wird ersichtlich, dass mit deren wachsender Bedeutung auch deren Interesse an von hoher Rechtssicherheit geprägten Schutzinstrumenten wächst.¹⁰⁴

Welche Auswirkungen das österreichische UrhG auf die urheberrechtliche Konstruktion des Open Source Modells hat, soll das folgende Kapitel klären. Zunächst wird eine kurze Einordnung des Schutzes von Computerprogrammen unter das UrhG sowie die rechtliche Beurteilung des Urhebers als Rechteinhaber vorgenommen, bevor abschließend auf praktische Problemfelder im Bereich des Schutzzumfangs eingegangen wird.

3.1. Allgemeines zum Schutz nach dem österreichischen Urheberrechtsgesetz

In Österreich gilt das Enumerationsprinzip dh, dass zum Schutz des „geistigen Eigentums“ nur jene Immaterialgüterrechte zur Verfügung stehen, die vom Gesetzgeber taxativ festgelegt wurden. Zu den wichtigsten Immaterialgüterrechten, neben dem Urheberrecht (geregelt im UrhG), zählen das Markenschutzrecht (geregelt im MSchG), das Patentrecht (geregelt im PatG), das Musterschutzrecht (geregelt im MuSchG) und das Gebrauchsmusterrecht (geregelt im GMG). Diese werden auch häufig unter dem Begriff „gewerblicher Rechtsschutz“ zusammengefasst, auf den in diesem Kapitel nicht weiter eingegangen wird.

Alle Immaterialgüterrechte – also auch der gewerbliche Rechtsschutz – werden nach dem „*lex loci protectionis*“, auch Territorialitätsprinzip genannt,

¹⁰⁴ Siehe auch *Scheuch/Holz Müller*, Die wirtschaftliche Bedeutung des Urheberrechts in Österreich – Die Wertschöpfung der Copyright Industries (1989) 59.

behandelt¹⁰⁵. Gem § 34 Abs 1 IPRG sind das „*Entstehen, der Inhalt und das Erlöschen von Immaterialgüterrechten [...] nach dem Recht des Staates zu beurteilen, in dem eine Benützung- oder Verletzungshandlung gesetzt wird*“. Begeht bspw ein Italiener in Italien eine Urheberrechtsverletzung eines Werks, das von einem Deutschen und einer Österreicherin geschrieben wurde und begehrt jetzt die Österreicherin den Schutz dieses Werks in Italien, kommt in diesem Fall das italienische Urheberrecht zur Anwendung. Auch die Verordnung Nr. 864/2007 des Europäischen Parlaments und des Rates (Rom II)¹⁰⁶ sieht eine solche Kollisionsnorm für außervertragliche Schuldverhältnisse, die aus der Verletzung von Rechten an geistigem Eigentum entstehen, in Artikel 8 vor. Während Rom II für das Urheberrecht festlegt, dass das nationale Recht jenes Landes zur Anwendung kommen soll, in dem die Verletzungshandlung begangen wurde, soll bei einer Verletzung des gewerblichen Rechtsschutzes jenes nationale Recht zur Anwendung kommen, in dem das Recht begründet wurde. Dies hätte dann im Falle einer möglichen Patentierung von Software weitreichende Konsequenzen.

Im österreichischen UrhG sind sowohl das *Urheberrecht* iS – Schutzgegenstand ist hier das Werk iSd § 1 UrhG – als auch die *verwandten Schutzrechte* iSd II. Hauptstücks des UrhG geregelt, die auch als Leistungsschutzrechte oder Nachbarrechte bezeichnet werden¹⁰⁷. Durch sie werden Gegenstände geschützt, die keinen Werkcharakter haben, für die aber auch ein Schutzbedürfnis bestehen soll, wie bspw die Rechte von ausübenden Künstlern oder Datenbankherstellern. Das *Urheberrecht im engeren Sinne* und die *verwandten Schutzrechte* werden zusammen auch als *Urheberrecht im weiteren Sinne* bezeichnet.

¹⁰⁵ Posch, Bürgerliches Recht VII – Internationales Privatrecht³ (2002) Rz 13/8.

¹⁰⁶ Verordnung (EG) Nr. 864/2007 des Europäischen Parlaments und des Rates vom 11. Juli 2007 über das auf außervertragliche Schuldverhältnisse anzuwendende Recht („Rom II“), ABI 2007 L 199, 40.

¹⁰⁷ Reinprecht in Kucsko, Geistiges Eigentum (2003) 1078.

3.2. Der Schutz von Computerprogrammen im österreichischen UrhG

Der Schutz von Computerprogrammen wurde erst 1993 im österreichischen UrhG expressis verbis in § 2 UrhG iVm § 40a UrhG eingeführt¹⁰⁸, da die ComputerRL¹⁰⁹ der EG im Zuge des Beitritts Österreichs zum EWR eine Anpassung der damals geltenden Rechtslage an das EG-Recht forderte. Es löste damit das UWG als bis dahin primäres Schutzgesetz für Software ab¹¹⁰. Anzumerken ist, dass der urheberrechtliche und der wettbewerbsrechtliche Schutz dabei subsidiär nebeneinander bestehen und einander nicht ausschließen.¹¹¹

Bereits vor dieser Verankerung wurde darüber diskutiert, ob Computerprogramme Schutz durch das UrhG bekommen sollten¹¹². Die Rsp in Österreich musste sich erstmals 1985 mit diesem Problem in der Zahnarzt-Programm-Entscheidung¹¹³ befassen, bei der zwar dem streitgegenständlichen Computerprogramm Urheberrechtsschutz gewährt, aber keine klare Lösung zur Urheberproblematik ausgesprochen wurde. Auch in der späteren Commodore-Diskettenbetriebssystem-Entscheidung¹¹⁴ aus dem Jahr 1986 musste sich der OGH nicht zum Thema Urheberrechtsschutz für Computerprogramme äußern, da das Verfahren mit einem Vergleich endete¹¹⁵. Interessant ist, dass beide Entscheidungen die urheberrechtliche Schutzfähigkeit von Zeitaufwand und

¹⁰⁸ BGBl 1993/93.

¹⁰⁹ Richtlinie 91/250/EWG des Rates vom 14. Mai 1991 über den Rechtsschutz von Computerprogrammen, ABi 1991 L 122, 42, in der Folge als ComputerRL bezeichnet.

¹¹⁰ Der Gesetzgeber hatte schon 1974 erwogen, den Softwareschutz in das UrhG aufzunehmen, dies aber letztendlich nicht durchgeführt; siehe dazu *Dittrich*, Urheberrechtsschutz für Computersoftware? RdW 1983, 39.

¹¹¹ *Wachte/Winter*, Computerrecht für die Praxis (1996) 109.

¹¹² Bspw. *Röttlinger*, Der Urheberrechtsschutz von Computersoftware in Österreich, ÖJZ 1990, 33 ff.

¹¹³ OGH 19.05.1987, 4 Ob 323/86, EDV&Recht 1/86, 26.

¹¹⁴ OLG Wien 08.08.1986, 3 R 101/86, EDV&Recht 4/86, 34.

¹¹⁵ OLG Wien 08.08.1986, 3 R 101/86, EDV&Recht 4/86, 35.

Kosten der Programmierung abhängig machen, also wirtschaftlichen Kriterien¹¹⁶, welche man eher in den wettbewerbsrechtlichen Bestimmungen des UWG findet und die dem UrhG an sich fremd sind.¹¹⁷

3.2.1. Schutzgegenstand

Der Schutzgegenstand – das Computerprogramm – wird nachfolgend unter das UrhG eingeordnet, wobei die Einordnung der Werkhöhe, der Begriff des Computerprogramms an sich und die Arten der Urheberschaft mit den damit verbundenen rechtlichen Problemen aus Sicht der Open Source Projekte behandelt werden. Abschließend werden die zivilrechtlichen Schutzmöglichkeiten skizziert und die Relevanz für Open Source Software untersucht.

3.2.1.1. Einordnung und Werkhöhe

Gem § 2 Z 1 UrhG zählen Computerprogramme zu den Sprachwerken¹¹⁸ und damit zu den Werken der Literatur. Ihre genaue Regelung findet sich seit der UrhG-Novelle 1993 in § 40a UrhG, der auf „*das Ergebnis der eigenen geistigen Schöpfung ihres Urhebers*“ abzielt und der sich damit wesentlich von § 1 UrhG unterscheidet. Während § 1 UrhG von einem Werk „*Eigentümlichkeit*“

¹¹⁶ Renner, Rechtsschutz von Computerprogrammen (1998) 10.

¹¹⁷ „Commodore“-Entscheidung OLG Wien, 08.08.1986, 3 R 101/86 zur Softwarepiraterie MR 6/86, 7. Wirtschaftliche Kriterien findet man im UrhG in den §§ 76c ff UrhG. Diese können aber nicht als Grundcharakteristika des UrhG angesehen werden.

¹¹⁸ *Fallenböck* in *Kucsko*, Geistiges Eigentum 1115; kritisch dazu *Rehbinder*, Urheberrecht: ein Studienbuch¹⁰ (1998) Rz 127 oder *Blocher/Walter* in *Walter* (Hrsg), Europäisches Urheberrecht (2001) 126.

als Unterscheidungsmerkmal fordert¹¹⁹, verlangt § 40a UrhG vom Urheber, dass sein Programm eine „*eigene geistige*“ Leistung darstellt.

Der Werkbegriff selbst ist von den einzelnen Werkgattungen unabhängig¹²⁰. Die genaue Definition eines Werkes iSd UrhG würde an dieser Stelle zu weit führen und ist darüber hinaus in der Fachliteratur nachzuschlagen. Daher wird im Folgenden nur auf die für diese Arbeit wichtige Einordnung der Computerprogramme eingegangen.

Ein Computerprogramm musste bis zur "*Bundesheer-Formblatt*"-Entscheidung¹²¹ eine gewisse Werkhöhe¹²² haben, damit es urheberrechtlich geschützt wird. Der OGH hatte dies bereits 1984 in der „*Mart Stam-Stuhl I*“-Entscheidung¹²³ ausgesprochen und festgestellt, dass eine statistische Einmaligkeit nicht ausreichen würde, um urheberrechtlichen Schutz zu genießen. Von dieser Judikatur ist der OGH jedoch wieder abgekommen und hat bspw in der „*Ranking*“-Entscheidung¹²⁴ festgestellt, dass ein Computerprogramm dann urheberrechtlichen Schutz genießt, wenn es sich um eine individuell geprägte Problemlösung handelt, die durch die Kombination vieler Programmschritte erreicht wird, nicht aber ein durch seine Anwendung erzieltetes Arbeitsergebnis.

¹¹⁹ Siehe dazu auch: OGH 02.03.1982, 4 Ob 427, 428/81 – Blumenstück – ÖBI 1982, 164. Bei der eigentümlichen geistigen Leistung kommt es hauptsächlich auf die Individualität an, während es bei der eigenen geistigen Leistung nur darauf ankommt, dass das Werk vom Urheber selbst entwickelt wurde.

¹²⁰ *Fallenböck* in *Kucsko*, Geistiges Eigentum 1105.

¹²¹ OGH 07.04.1992 4 Ob 36/92 – „*Bundesheer-Formblatt*“ – MR 1992, 199 (*Walter*) = ÖBI 1992, 81 = SZ 65/51 = WBI 1992, 340 = EvBI 1993/36 = ÖJZ NRsp 1992/183, 184 = GRUR Int 1993/565.

¹²² *Renner*, Rechtsschutz von Computerprogrammen 12.

¹²³ OGH 10.07.1984, 4 Ob 337/84 – „*Mart Stam-Stuhl I*“ – MR 1/92, 21 (*Walter*) = ÖBI 1985, 24 = GRUR Int 1985, 684.

¹²⁴ OGH 09.11.1999, 4 Ob 282/99w, MR 1999, 346; siehe auch OGH 23.05.2000, 4 Ob 30/00s – Handwerkerpaket – ÖBI 2001, 141; OGH 19.12.2000, 4 Ob 256/00a – steuerprofi.at – wbl 2001, 237. In jüngster Zeit hat der OGH 12.07.2005, 4 Ob 45/05d – TerraCAD – MR 2005, 379 ausgesprochen, dass ein Computerprogramm eine gewisse Komplexität ausweisen muss.

3.2.1.2. Begriff

Der Begriff „Computerprogramm“ umfasst gem § 40a Abs 2 UrhG alle Ausdrucksformen¹²⁵ – einschließlich des Maschinencodes – sowie das Material zur Entwicklung und entspricht damit Art 1 Abs 1 letzter Satz iVm Abs 2 erster Satz der ComputerRL sowie Art 10 Abs 1 des TRIPS-Abk. Nicht dem Computerprogramm zugehörig sind nach *Blocher* das Pflichtenheft oder auch eine reine Systemanalyse¹²⁶.

Die ComputerRL sieht in Art 1 Abs 2 erster Satz weiter vor, dass „Ideen und Grundsätze, die irgendeinem Element eines Computerprogramms zugrunde liegen, einschließlich der den Schnittstellen zugrunde liegenden Ideen und Grundsätze, [...] nicht im Sinne dieser Richtlinie urheberrechtlich geschützt“ sind. Die europäische Gemeinschaft geht damit einen Schritt in die richtige Richtung, indem sie die Schaffung von Standards fördert¹²⁷, da die Funktion von Computerprogrammen darin besteht, mit den anderen Komponenten eines Computersystems und den Benutzern in Verbindung zu treten¹²⁸. Die Schnittstellen von Software und Hardware und deren Benutzern gewährleisten die beabsichtigte Funktionalität. Diese funktionale Verbindung und Interaktion ist als "Interoperabilität" bekannt. Diese Interoperabilität wird in der ComputerRL als die Fähigkeit zum Austausch von Informationen und zur wechselseitigen Verwendung der ausgetauschten Informationen definiert¹²⁹ und kann auch als Kompatibilität bezeichnet werden.

¹²⁵ Der Schutz gilt somit auch für Embedded Systeme.

¹²⁶ *Blocher*, Gewerblicher Rechtsschutz und Urheberrecht, in *Jahnel/Schramm/Staudegger* (Hrsg), Informatikrecht² (2002) 143.

¹²⁷ Erwägungsgrund 9 der ComputerRL; siehe auch *Fallenböck* in *Kucsko*, Geistiges Eigentum 1114.

¹²⁸ Erwägungsgrund 10 der ComputerRL.

¹²⁹ Erwägungsgrund 12 der ComputerRL.

Wie erwähnt, gilt der Rechtsschutz nur für die Ausdrucksform eines Computerprogramms und nicht für die Ideen und Grundsätze, die irgendeinem Element des Programms einschließlich seiner Schnittstellen zugrunde liegen. Dies wurde nur im dt UrhG in § 69a Abs 2, nicht aber im österreichischen UrhG übernommen.

Strittig ist der urheberrechtliche Schutz des Algorithmus¹³⁰. Diesem wurde schon am 29.10.1970 von der Beschwerdeabteilung des Österreichischen Patentamtes¹³¹ der Schutz durch das Patentrecht verweigert, „da wissenschaftliche Lehr- und Grundsätze nicht unter den Begriff der Erfindung“ fallen könnten. Urheberrechtlich treffen hier verschiedene Meinungen aufeinander¹³². ME ist *Blocher*¹³³ zu folgen, der auf die Individualität des Lösungswegs abzielt und von dem Schutz des „Implementationsalgorithmus“ spricht. An dieser Stelle muss beachtet werden, dass eine nicht durch das Urheberrecht geschützte Idee oder ein Grundsatz durchaus Schutz durch andere Rechtsvorschriften – wie bspw durch das UWG – erfahren kann¹³⁴.

Zusammenfassend bleibt festzuhalten, dass auch die „kleine Münze“¹³⁵ insbesondere in Deutschland – aber auch in Österreich¹³⁶ – Schutz durch das Urheberrecht genießt. Im Streitfall ist der schöpferische Teil des Originals mit

¹³⁰ Algorithmus ist die Bezeichnung einer Folge von Anweisungen, mit deren Hilfe aus vorgegebenden Anfangswerten in endlich vielen Schritten ein Ergebnis berechnet werden kann.

¹³¹ ÖPA 29.10.1970, Algorithmus, GRUR Int 1971, 337 f.

¹³² Siehe bspw *Horvath/Ortner* in *Ulrich/Körner* (Hrsg), Der internationale Softwarevertrag (1995) Rz 1063.

¹³³ *Blocher* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 143.

¹³⁴ *Rehbinder*, Urheberrecht¹³, Rz 128.

¹³⁵ Die „kleine Münze“ ist ein im Urheberrecht gebräuchlicher Begriff, der Werke bezeichnet, die nur wenig Eigentümlichkeit oder geistige Schöpfung aufweisen, also am unteren Rand des Urheberrechtsschutzes angesiedelt sind, wie bspw Landkarten.

¹³⁶ Siehe auch OGH 01.09.1970, 4 Ob 336/70 – ZahnärztekammerG I – ÖBI 1970, 146; OGH 07.03.1978, 4 Ob 317/78 – Stichwörterverzeichnis – ÖBI 1978, 107; OGH 12.09.1989, 4 Ob 76/89, ÖBI 1990, 136 – Happy Skiing – MR 1989, 210 (*Walter*); OGH 06.11.1990, 4 Ob 145/90 „Oberndorfer Gschichtn“ = ÖBI 1991, 1188 = MR 1990, 230 (*Walter*)

dem Verletzungsgegenstand zu vergleichen. Dies ist ein wesentlicher Faktor für Open Source Software, da oft freie und bekannte Programmbausteine aneinandergereiht werden. Das Ergebnis genießt dann keinen eigenen Schutz¹³⁷.

3.2.2. Der Urheber als Rechteinhaber und die verschiedenen Arten der Urheberschaft an Open Source Software

Die Urheberschaft hat bestimmenden Einfluss auf die Lizenzierung der Software, da nur derjenige Rechte übertragen bzw einräumen kann, der sie besitzt. Das Urheberrecht stellt daher die „*nötige Vorbedingung für die Wirksamkeit*“¹³⁸ der Open Source Lizenzen dar. Da die Philosophie, die hinter dem Open Source Gedanken steckt, einen offenen Austausch an Informationen vorsieht, kann Software zwar von einzelnen Personen entwickelt werden, jedoch führt die Verbreitung der Software im Internet zu neuen urheberrechtlichen Konstellationen¹³⁹.

Bei der Entwicklung von Open Source Software ist oft schwierig festzustellen, wer als Inhaber des Urheberrechts anzusehen ist, da die Zusammenarbeit vieler Menschen an einem Open Source Projekt sehr komplexe Umfänge annehmen kann¹⁴⁰. Zwar gibt es auch im Open Source Bereich Projekte, die klar strukturiert sind, weil sie entweder von Einzelnen bzw einer überschaubaren Gemeinschaft weiterentwickelt werden oder eine Firma, welche die Urheberrechte an der Open Source Software besitzt, diese – meist durch „Dual Licensing“¹⁴¹ – Open Source zur Verfügung stellt. Viele der erfolgreichen Open Source Programme – wie bspw „Linux“ – werden weltweit

¹³⁷ *Marly*, Softwareüberlassungsverträge³ (2000) Rz 123.

¹³⁸ *Spindler*, Rechtsfragen der Open Source Software (2004) 22.

¹³⁹ Vgl Kap 3.2.2.2.

¹⁴⁰ Siehe Kap 1.

¹⁴¹ Vgl Kap 5.10.

von einer großen Anzahl an Programmierern ständig weiterentwickelt und angepasst. Aufgrund dessen sind verschiedene urheberrechtliche Konstellationen denkbar¹⁴².

3.2.2.1. Der oder die Alleinurheber

Die ComputerRL gibt in Art 2 Abs 1 die Vorgabe, dass der oder die Urheber eines Computerprogramms eine natürliche Person bzw eine Gruppe natürlicher Personen oder – soweit es das nationale Recht eines MS zulässt – eine juristische Person, welche nach den jeweiligen Rechtsvorschriften als Rechteinhaber gilt, sein kann. In Österreich, wie bspw auch in Deutschland, können nur physische Personen als Urheber auftreten, ein originärer Erwerb an Urheberrechten durch eine juristische Person ist nicht möglich¹⁴³.

In Österreich gilt gem § 10 Abs 1 UrhG das Schöpferprinzip¹⁴⁴, nach dem „*Urheber eines Werkes ist, wer es geschaffen hat*“. Er erhält dieses Recht also „ipso iure“. Auch die Personen, auf welche das Urheberrecht nach dem Tod des Schöpfers übergegangen ist, werden iSd § 10 Abs 2 UrhG als Urheber bezeichnet.

Gem § 94 UrhG ist jedes Werk geschützt, dessen Urheber oder Miturheber österreichischer Staatsbürger ist. Darüber hinaus schützt das UrhG gem § 95 alle im Inland erschienenen Werke unabhängig von der Staatsbürgerschaft ihres Urhebers. Ausländische Werke genießen nach dem Reziprozitätsprinzip gem § 96 UrhG dann urheberrechtlichen Schutz, wenn dies entweder durch einen Staatsvertrag festgelegt wurde oder Werke von

¹⁴² Jaeger/Metzger, Open Source Software 27.

¹⁴³ Dittrich, Österreichisches und Internationales Urheberrecht⁴ (2004) 83 E1.

¹⁴⁴ Siehe auch Rehbindner, Urheberrecht¹³ Rz 164.

österreichischen Urhebern in dem Staat, dem ein ausländischer Urheber angehört, gleichrangig geschützt sind.

Für Computerprogramme gilt, dass es nicht nur darauf ankommt, wer es geschaffen hat, auch die Selbständigkeit ist ein wichtiges Merkmal, um den Alleinurheber iSd § 10 UrhG vom Miturheber iSd § 11 UrhG abgrenzen zu können. Programme werden in der Praxis häufig zu einem Produkt zusammengefasst und als „ein Programm“ angeboten, wie dies bspw bei „Windows“, „Adobe Photoshop“ oder „OpenOffice“ geschieht. Diese Tatsache darf allerdings nicht über eine mögliche Trennbarkeit bzw unabhängige Verwertbarkeit der einzelnen Softwareteile hinwegtäuschen. In solchen Fällen muss aus urheberrechtlicher Sicht stets darauf geachtet werden, ob das Programmpaket ein einheitliches Werk darstellt oder ob es zu einer Werkverbindung gekommen ist¹⁴⁵.

Ein einheitliches Werk liegt dann vor, wenn es nicht einmal theoretisch möglich ist, einzelne Programme oder Programmbestandteile getrennt von den übrigen Teilen des Programms zu verwerten¹⁴⁶. Dies ergibt sich aus dem Umkehrschluss zu § 11 UrhG, nach dem das Urheberrecht allen Miturhebern gemeinschaftlich zusteht, wenn diese gemeinsam ein Werk geschaffen haben, bei dem die Ergebnisse ihres Schaffens eine untrennbare Einheit bilden. Für Computerprogramme ist dies im Einzelfall zu entscheiden. Während bspw die Implementierung des Internetprotokolls („TCP/IP-Stack“¹⁴⁷) sowie die Dateisystemtreiber von Windows mit diesem so verbunden sind, dass es theoretisch unmöglich ist, sie getrennt vom Windows-Kernel zu verwerten, ist dies bei den diversen Office-Paketen durchaus möglich. Ein Nutzer erwirbt zwar

¹⁴⁵ *Omsels*, Open Source und das deutsche Vertrags- und Urheberrecht, in *Schertz/Omsels*, Festschrift *Hertin* 2000, 141 (160).

¹⁴⁶ *Omsels*, Open Source und das deutsche Vertrags- und Urheberrecht, in *Schertz/Omsels*, Festschrift für *Paul W. Hertin* zum 60. Geburtstag am 15. November 2000 (2000) 141 (160); *Jaeger/Metzger*, Open Source Software 27; *Spindler*, Rechtsfragen der Open Source Software 27.

¹⁴⁷ Der TCP/IP-Stack ist auf die Internet-Protokolle zugeschnitten, die den Datenaustausch über die Grenzen lokaler Netzwerke hinaus ermöglichen.

meist das Gesamtpaket, dennoch können Textverarbeitungsprogramme (bspw „Word“) unabhängig von Tabellenkalkulationsprogrammen (bspw „Excel“) oder Präsentationsprogrammen (bspw „Power Point“) verwertet werden. Die rechtliche Bewertung der Eigenständigkeit wird in den Fällen proprietärer Software in der Regel keine Probleme bereiten. Zum einen stehen alle Nutzungs- bzw Verwertungsrechte, meist aufgrund des § 40b UrhG oder sonstiger Vereinbarungen, einem Einzigen oder einer Firma zu, zum anderen ist die Bewertung der Eigenständigkeit auf das Programm in Objektformat beschränkt. Bei Open Source Software muss die Bewertung der Eigenständigkeit jedoch um die Bewertung der Eigenständigkeit des Quelltextes erweitert werden, da die Eigenständigkeit des ausführbaren Programms nicht zwingend gleichbedeutend mit der Eigenständigkeit des Quelltextes ist.

Viele Forensysteme im Internet, welche auf Open Source Software basieren, benutzen bspw die gleichen Befehle, um so genannte „Emoticons“ („Smileys“) automatisch nach Eingabe des Textes einzufügen¹⁴⁸. Auch wenn jedes Forensystem getrennt von den anderen verwertet werden kann, weist der Quelltext womöglich Teile auf, welche nicht nur von unterschiedlichen Urhebern stammen, sondern welche auch unabhängig von einander verwertet werden können. Dieser „UBB-Parser“ kann nicht nur in Forenprogrammen genutzt werden, auch Chat- oder Blogprogramme können den Quelltext übernehmen und einsetzen. Die Bewertung der Eigenständigkeit des Quelltextes muss daher nach anderen Kriterien erfolgen.

Der Quelltext ist als einheitliches Werk zu betrachten, wenn kein Abschnitt des Quelltextes eigenständig verwertet werden kann. Unter einer eigenständigen Verwertung ist die funktionelle Selbständigkeit zu sehen. Kann der relevante Part unabhängig vom restlichen Teil des Quelltextes verwertet

¹⁴⁸ Der Nutzer des Forensystems gibt bspw „;-)“ ein, nach dem Absenden des Textes an den Server wird Lesern des Beitrages das Emoticon „☺“ angezeigt.

werden – wie bspw im Falle der „UBB-Parsers“¹⁴⁹ – liegt hinsichtlich des Quelltextes eine Werkverbindung vor, welche eine Alleinurheberschaft iSd § 10 UrhG für jeden Urheber des jeweils eigenständig verwertbaren Teils begründet. Während für den Quelltext in einem solchen Fall eine Alleinurheberschaft für die jeweiligen Teile angenommen werden muss, kommt bei der Vorlage des Programms in Objektform eine Teilurheberschaft oder eine Miturheberschaft iSd § 11 UrhG in Betracht, da das Ergebnis (das Objektformat) eine untrennbare Einheit bildet. Im oben angesprochenen Fall kommt es durch die Möglichkeit der klar trennbaren Verwertungsmöglichkeiten zu einer Teilurheberschaft¹⁵⁰.

3.2.2.2. Miturheber

Für Open Source Software sind die Vorschriften über die Miturheberschaft iSd § 11 UrhG besonders interessant, arbeiten doch tagtäglich viele Personen – bedingt durch das Internet – aus allen Ländern der Welt an der Entwicklung eines Computerprogramms zusammen¹⁵¹.

Schon Art 2 Abs 2 der ComputerRL gibt vor, sollte ein Computerprogramm von einer Gruppe natürlicher Personen gemeinsam geschaffen worden sein, stehen dieser Gruppe auch gemeinsam die ausschließlichen Rechte daran zu. Diesem Aspekt trägt das österreichische UrhG in § 11 Abs 1 Rechnung, der besagt, dass allen Miturhebern das Urheberrecht gemeinschaftlich zusteht, wenn das Ergebnis von ihrem – wissentlich und willentlich¹⁵² – gemeinsam geschaffenen Werk eine

¹⁴⁹ Ein Parser wird dazu verwendet, einen Text in eine neue Struktur zu übersetzen.

¹⁵⁰ Zur Teilurheberschaft siehe Kap 3.2.2.4; zur Bewertung der „*derivative works*“ als eigenständiges Werk siehe Kap 5.4.1.2.2.

¹⁵¹ Vgl Kap 2.3.2. über die Basarmethode.

¹⁵² Ciresa, Urheberrecht aktuell: Wegweiser durch das österreichische Urheber- und Leistungsschutzrecht für die Praxis (1997) 34.

untrennbare Einheit bildet. Verbundene Werke, also die Verknüpfung von mehr oder weniger selbständigen Werken gleicher Art, fallen nicht unter die Miturheberschaft¹⁵³. Hier fehlt es an der eigenen geistigen Leistung. Der Verbindung von Werken verschiedener Art ist gem § 11 Abs 3 UrhG schon ex lege die Begründung der Miturheberschaft untersagt.

Für Open Source Software kommt es in Abgrenzung zur vorher behandelten Alleinurheberschaft in jenen Fällen zu einer Miturheberschaft iSd § 11 UrhG, wenn ein Programm nach einem einheitlichen Konzept¹⁵⁴ gemeinsam entwickelt wurde und kein von einer bestimmten Person entwickelter Programmbestandteil alleine funktionstüchtig bzw verwertbar ist¹⁵⁵. Andere Kriterien – wie bspw eine zeitgleiche Zusammenarbeit oder der Umfang der Arbeit – sind für die Beurteilung der Miturheberschaft unerheblich. Eine Abgrenzung wird in der Praxis umso schwieriger, je größer die Anzahl und die räumliche Distanz der Miturheber sind. Die damit verbundenen Problemstellungen hinsichtlich der Aktivlegitimation werden in Kap 3.2.2.6. behandelt.

Des Weiteren ist von der Miturheberschaft die Bearbeitung iSd § 5 UrhG zu unterscheiden. Eine Bearbeitung iSd § 5 UrhG findet statt, wenn eine Weiterentwicklung getrennt vom ursprünglichen Konzept des Programms vorgenommen wird, bspw durch die Integration eines neuen Moduls zu einer Software¹⁵⁶. Erfüllt die Bearbeitung die Mindestanforderung an eine geistige Schöpfung, kommt dem Bearbeiter ein eigenes Urheberrecht zu¹⁵⁷. Gem § 14 Abs 2 UrhG darf der Bearbeiter allerdings die Bearbeitung „auf die ihm vorbehaltenen Arten nur verwerten, soweit ihm der Urheber des bearbeiteten

¹⁵³ Blocher in Jahnel/Schramm/Staudegger, Informatikrecht², 144.

¹⁵⁴ Wiebe/Prändl, ÖJZ 2004, 630.

¹⁵⁵ Küng, MR 2004, 24; Spindler, Rechtsfragen der Open Source Software 22; Schneider in Kucsko, Geistiges Eigentum 1135.

¹⁵⁶ Wiebe/Prändl, ÖJZ 2004, 630.

¹⁵⁷ So auch Horvath/Ortner in Ulrich/Körner, Der internationale Softwarevertrag Rz 1082. Siehe auch Kap 5.2.

Werkes das ausschließliche Recht oder die Bewilligung dazu erteilt“ hat. Diese Rechtseinräumung ist bei jeder weiteren Bearbeitung zur Verwertung notwendig und muss daher von allen Rechtsinhabern – ursprünglicher Urheber und Bearbeitern – eingeholt werden. Im Falle von Open Source Software geschieht dies, wenn der ursprüngliche Urheber seine Software unter eine Open Source Lizenz stellt, welche einem Bearbeiter nur dann Verwertungsrechte einräumt, sollte dieser seinerseits die Bearbeitung unter diese Open Source Lizenz stellen (Copyleft-Effekt).

3.2.2.3. Gehilfe

Vom Miturheber ist auch der Gehilfe zu unterscheiden. Dieser hat zur Schaffung des Werkes keinen eigenen geistigen Beitrag geliefert¹⁵⁸ und wird daher auch nicht zum Miturheber iSd § 11 UrhG oder Bearbeiter iSd § 5 UrhG. Diesem Umstand muss Beachtung geschenkt werden, da Entwicklungsprozesse im Softwarebereich oft auf kurze Anregungen angewiesen sind, die mitunter bspw nur darin bestehen, Eingabemasken auf dem Bildschirm sinnvoll anzuordnen. Diese Leistungen reichen demnach nicht aus, um iSd § 10 UrhG als Schöpfer und damit als (Mit-) Urheber anerkannt zu werden. Im Open Source Bereich findet man viele solcher „Gehilfen“ vor allem in Internetforen oder Newsgroups, die – sei es als Anwender, als Softwareentwickler oder als interessierter Laie – den Softwareentwicklern, und damit den eigentlichen Urhebern, Anregungen zur Weiterentwicklung ihrer Software geben. Miturheber werden sie dadurch nicht, auch wenn sie vielleicht wesentlichen Einfluss auf die Funktionalität einer Software haben.¹⁵⁹

¹⁵⁸ *Graninger in Kucsko*, Geistiges Eigentum 1133.

¹⁵⁹ Es sei denn, sie leisten einen eigenständigen geistigen Beitrag, bspw durch das Schreiben eines Bugfixes, mit welchem sie wissentlich und willentlich einen vom restlichen Werk wirtschaftlich untrennbaren Beitrag leisten. Dann würden sie vom Gehilfen zum Miturheber iSd § 11 UrhG.

3.2.2.4. Teilurheberschaft

Die Verbindung von zwei selbständig verwert- und trennbaren Werken, bspw eines Musikstückes mit einem Computerprogramm, begründet eine Teilurheberschaft¹⁶⁰, die eine Gesellschaft bürgerlichen Rechts (GesbR) iSd § 1175 ABGB erschafft. Die eingebrachten Werke bilden dabei das Gesellschaftsvermögen¹⁶¹. Dies ändert nichts an der Urheberschaft für die einzelnen Bestandteile, die verbunden werden. Die jeweiligen Rechtsinhaber – Allein- oder Miturheber – bleiben Urheber ihres eigenständigen Teils¹⁶². Hierbei ist entscheidend, dass ein Teil immer dann als eigenständiges Werk gesehen werden kann, wenn eine unabhängige Verwertung der einzelnen Werke durch ihren jeweiligen Teilurheber möglich ist¹⁶³. Hierzu ein Exkurs über die gesellschaftsrechtlichen Aspekte:

Exkurs: Gesellschaftsrechtliche Aspekte im Open Source Bereich

An dieser Stelle wird kurz auf die Entstehung einer GesbR im Open Source Bereich eingegangen werden; ebenso interessant und für die Praxis relevant hat sich die Genossenschaft gezeigt, durch die sich eine große Anzahl von Open Source Dienstleistern im KMU-Bereich zur OSAlliance zusammengeschlossen haben. Diese wird daher nach der GesbR ebenfalls einer Betrachtung unterzogen.

¹⁶⁰ *Graninger* in *Kucsko*, Geistiges Eigentum 1135.

¹⁶¹ *Ciresa*, Urheberrecht aktuell 35.

¹⁶² *Spindler*, Rechtsfragen der Open Source Software 24.

¹⁶³ *Dillenz*, Praxiskommentar 43.

Die GesbR im Open Source Bereich

Die Begründung einer GesbR kann nur durch einen – auch formlosen – Vertrag geschehen¹⁶⁴. Die Entstehung einer GesbR durch die Verbindung verschiedener eigenständiger Teile kann im Open Source Bereich daher nur dann entstehen, wenn das Projekt wissentlich und willentlich koordiniert realisiert wird¹⁶⁵. In den anderen Fällen kann keine konkludente Willensübereinstimmung iSd § 863 ABGB angenommen werden. Im Gegensatz zu einer schlichten Rechtsgemeinschaft besteht hinsichtlich des Entstehens einer GesbR das Erfordernis des gemeinsamen Wirkens, im Gegensatz zu einem gemeinsamen Haben oder Verwalten¹⁶⁶. Der wirtschaftliche Erfolg ist also ein direktes Bedürfnis der GesbR, weshalb nur solche Zusammenschlüsse von Open Source Projekten in die Anwendungsbereiche der §§ 1175 ff ABGB fallen, die auf die Erzielung eines gemeinschaftlichen Nutzens gerichtet sind. Die von der Rsp geforderte Existenz einer Gemeinschaftsorganisation ist im Open Source Bereich gegeben, da die Eigenart der Softwareentwicklung das Projektmanagement geradezu fordert. Das Projektmanagement legt Aufgaben und Verantwortung fest, weshalb dieses Kriterium bei auf Erwerbszwecke ausgerichteten Open Source Projekten – und nur solche werden ja von den §§ 1175 ff ABGB abgedeckt – immer gegeben sein wird.

Da die §§ 1175 ff ABGB des Weiteren überwiegend dispositiven Charakter haben und eine GesbR ohne großen formalen Aufwand gegründet werden kann, sind einige Anwendungsfälle für den Open Source Bereich denkbar, die in Folge kurz skizziert und auf ihre Relevanz überprüft werden:

¹⁶⁴ OGH 06.12.1961, 5 Ob 399/61 = SZ 34/184.

¹⁶⁵ *Wiebe/Prändl*, ÖJZ 2004, 630.

¹⁶⁶ OGH 26.01.1989, 8 Ob 620/88 = RdW 1989, 189 = ARD 4086/19/89 = JBI 1989, 383.

3.2.2.4.1. Gruppenarbeitsverträge im Open Source Bereich

Im Falle von Gruppenarbeitsverträgen sind verschiedene rechtliche Gestaltungsformen denkbar. Die Rsp hat bereits zwei Varianten herausgearbeitet¹⁶⁷, bei denen Gruppenarbeitsverträge zustande kommen können. Danach entscheidet man „*je nachdem, ob die Gruppe überhaupt nur gemeinsam ein Vertragsverhältnis zum Arbeitgeber eingeht oder ob vom Vorliegen von Arbeitsverträgen jedes einzelnen Arbeitnehmers auszugehen ist.*“ Da dies im Einzelfall zu beurteilen sein wird, kann für den Open Source Bereich festgestellt werden, dass es sich um einen Gruppenarbeitsvertrag handelt, wenn hinsichtlich eigenständiger Programmteile ein gemeinsamer Arbeitsvertrag vorliegt (ein Anbieter von ERP-Systemen bietet eine Lösung zusammen mit einem Voice over IP-Spezialisten an) oder zwar getrennte Arbeitsverträge vorliegen (der Anbieter des ERP-Systems erbringt seine Leistung eigenständig neben der des Voice over IP-Spezialisten), diese aber dem Willen der Vertragsparteien nach verbunden sein sollen¹⁶⁸.

Gruppenarbeitsverhältnisse findet man im Open Source Bereich häufig. Open Source Entwickler gehören überwiegend dem Bereich der KMUs an, weswegen sich diese bei größeren Projekten zusammenschließen, um auf Dauer oder auch einmalig Projekte abzuwickeln. Steht bei einem Gruppenarbeitsvertrag der Zusammenschluss eigenständiger Werke im Vordergrund, welche die Bestimmungen über die Teilurheberschaft betreffen, decken Gelegenheitsgesellschaften einen Großteil jener Fälle ab, die durch Weiterentwicklungen durch Miturheber geschehen. Die Gelegenheitsgesellschaften werden deshalb nachfolgend erörtert.

¹⁶⁷ OGH 29.08.2002, 8 Oba 130/02d = ARD 5377/1/2003.

¹⁶⁸ OGH 29.08.2002, 8 Oba 130/02d = ARD 5377/1/2003.

3.2.2.4.2. Gelegenheitsgesellschaften im Open Source Bereich

Gelegenheitsgesellschaften machen für Open Source Projekte dann Sinn, wenn sie nur einem bestimmten Zweck dienen und nicht dauerhaft geschlossen werden sollen. Denkbar wäre etwa der Zusammenschluss zu einer Bietergemeinschaft, um an einer Ausschreibung teilzunehmen, die Durchführung einer Veranstaltung im Open Source Bereich, das Auftreiben von Venture Kapital oder der Zusammenschluss, um ein Projekt gemeinsam durchführen zu können. Letzteres weist große Ähnlichkeit zu den Gruppenarbeitsverträgen auf. Im Gegensatz zu diesen zeichnet die Gelegenheitsgesellschaft jedoch die gewollte befristete und damit vorübergehende Zusammenarbeit aus. Die Gelegenheitsgesellschaft wird sich regelmäßig aus dem Zusammenschluss von Entwicklern ergeben, die eigenständige Programme einbringen. Ebenso ist eine gemeinsame Entwicklung vorstellbar, weshalb die Gelegenheitsgesellschaft auch in den Fällen entstehen kann, in denen Miturheberschaft begründet wird. Die Gelegenheitsgesellschaften sind für Open Source Softwareentwickler auch dann eine sinnvolle Alternative, wenn diese einen Geschäftspartner erst einmal besser kennen lernen wollen. Die Gelegenheitsgesellschaft kann dabei jederzeit in ein Gruppenarbeitsverhältnis umgewandelt werden, wenn die Partnerschaft doch längerfristig – im Sinne wiederkehrender Erbringung von Leistungen – sein soll.

3.2.2.4.3. Kooperationsverträge im Open Source Bereich

Kooperationsverträge machen im Open Source Bereich dann Sinn, wenn eine strategische Partnerschaft eingegangen werden soll. Unternehmen, die rechtlich und selbständig arbeiten, haben durch den Kooperationsvertrag die Möglichkeit, gemeinsam zu versuchen ihre Wettbewerbsfähigkeit sowie Produktivität und Rentabilität zu erhöhen. Kooperationsverträge können sowohl

in den Fällen der Teil- als auch der Miturheberschaft zur Anwendung kommen. Ausschlaggebend sind die Inhalte und Tatsachen, die Sinn und Zweck der Partnerschaft festlegen und begründen.

3.2.2.4.4. Die Genossenschaft im Open Source Bereich

Das Wesen der Genossenschaft kommt den Entwicklern von Open Source Software sehr entgegen, da die Grundwerte der Genossenschaft nicht nur auf Selbsthilfe und –verantwortung basieren, sondern vor allem auf Demokratie, Gleichheit, Gerechtigkeit und Solidarität; Werte, die der Open Source Philosophie entsprechen¹⁶⁹. Es ist daher nicht verwunderlich, dass sich in Österreich eine Anzahl von ca 30 Open Source Entwicklern zur „osAlliance¹⁷⁰“ in Form einer Genossenschaft zusammengeschlossen hat, um mit Hilfe dieses Netzwerks gemeinsam am Markt tätig zu werden. Auf die Frage, ob die Genossenschaft sich als sinnvolle Unterstützung für Existenzgründungen im Open Source Bereich anbietet, wird an dieser Stelle nur kurz eingegangen, da die positiven Faktoren im Dienstleistungsbereich deutlich erkennbar sind¹⁷¹. Wie mehrmals erwähnt, handelt es sich bei den meisten Open Source Entwicklern um Freiberufler oder KMUs, die über ein Spezialwissen verfügen, wie bspw Datenbankspezialisten, ERP-Spezialisten, usw. Diesen fehlt es häufig an den betriebswirtschaftlich notwendigen Unternehmensstrukturen, wie Bilanzierung, Finanzierung oder Verwaltung sowie an Vertriebs- oder Marketingressourcen. Hierfür bietet die Genossenschaft einen geeigneten Rahmen, welcher Synergieeffekte nutzt und dadurch die Wettbewerbsfähigkeit der einzelnen Open Source Entwickler erhöht.

¹⁶⁹ In der Realität wird die Entscheidung eine Genossenschaft zu gründen, aber in den seltensten Fällen von diesen hehren Ziele geleitet sein, sondern eher trivialen Motivationen haben.

¹⁷⁰ <http://osalliance.com>.

¹⁷¹ Vgl auch: *Harbrecht*, Die Genossenschaft als Rechtsform für junge Unternehmen (2001) 21f.

Urheberrechtlich gesehen ist in den Fällen, bei denen Open Source Entwickler als Genossenschaft auftreten, ein Blick auf die Zusammenstellung der einzelnen Leistungen nötig. Entwickeln mehrere Personen ein Programm zusammen und treten nach außen als Genossenschaft auf, kann es in den Fällen zu Lizenzkollisionen kommen, bei denen Open Source Programme, die unter unterschiedlichen Open Source Lizenzen stehen, zusammengeführt werden und nur noch gemeinsam – dh als ein eigenständiges Programm – genutzt werden können. In diesen Fällen sollten die Urheber bereits im Vorhinein dafür Sorge tragen, ob das weiterentwickelte Programm unter einer einzigen Lizenz weiterlizenzieren kann, ob „Dual Licensing“¹⁷² möglich ist oder ob eine Zusammenführung gar ausgeschlossen ist. Im letzteren Fall ist es notwendig, die Programme eigenständig zu belassen und das jeweils andere Programm über Schnittstellen einzubinden.

In den Situationen, in denen die Eigenständigkeit des Programms gewahrt bleibt, kommt es zur Teilurheberschaft, bei der Zusammenführung oder gemeinsamen Neuentwicklung zur Miturheberschaft iSd § 11 UrhG mit den jeweiligen angesprochenen rechtlichen Konsequenzen. Es ist ebenfalls möglich der Genossenschaft als nach außen hin auftretenden Gesellschaft das Werknutzungsrecht¹⁷³ einzuräumen und ihr dadurch die Verantwortung zu überlassen, auf welche Weise die weiterentwickelten Werke lizenziert zur Verfügung gestellt werden sollen. Diese tritt dann im Außenverhältnis gleich einem Alleinurheber auf, da die maßgeblichen Rechte in der Hand der Genossenschaft liegen.

Dies zeigt, dass die urheberrechtliche Konstellation dem Arbeitnehmerurheberrecht – wie im Folgenden aufgeführt – gleicht, da auch hier der Arbeitgeber als Rechteinhaber nach außen auftreten darf.

¹⁷² Dazu mehr in Kap 5.10.

¹⁷³ Vgl Kap 3.3.1.

3.2.2.5. Arbeitnehmerurheberrecht

Hier gibt ebenfalls das Gemeinschaftsrecht in Art 2 Abs 3 der ComputerRL vor, dass ein Computerprogramm, sollte es von einem AN in Wahrnehmung seiner Aufgaben oder nach den Anweisungen seines Arbeitgebers geschaffen worden sein, ausschließlich der AG zur Ausübung aller wirtschaftlichen Rechte an dem so geschaffenen Werk berechtigt ist, wenn keine andere vertragliche Vereinbarung getroffen wurde. In Österreich wurde diese Bestimmung in § 40b UrhG übernommen und nur in der Formulierung dem § 7 PatG angeglichen¹⁷⁴. Die Urheberschaft bei der Entwicklung im Arbeits- und Dienstverhältnis erfährt dadurch eine Sonderregelung. Dies wird vom Gesetzgeber damit gerechtfertigt, hierdurch das spezielle Problem der schwierigen Zuordnung der Urheberschaft bei der Entwicklung von Software zu lösen. Im Grünbuch der Kommission über „Urheberrecht und die technologische Herausforderung“¹⁷⁵ steht dazu, dass *“im Interesse aller Mitgliedstaaten gewährleistet sein (sollte), dass in Ihren Gesetzen zumindest klar festgelegt wird, wer Rechtsinhaber ist, falls keine gegenteilige Abmachung besteht.”*¹⁷⁶ Dieser pragmatischen Ansicht hat sich die Lehre angeschlossen und dies damit begründet, dass es sich bei Software um Werke handeln würde, die eine „Schnittstelle zwischen Urheberrecht und gewerblichen Rechtsschutz“ bilden und denen daher die für das Urheberrecht fundamentale individuelle Prägung fehle.¹⁷⁷ Dieser Annahme ist unbestritten zuzustimmen, da die Urheberschaft hier für Außenstehende leicht zu erkennen ist und dadurch ein hohes Maß an Rechtssicherheit gegeben ist. Nicht unbeachtlich erscheint in diesem

¹⁷⁴ § 40b UrhG findet keine Anwendung auf vor dem 1.3.1993 geschaffene Computerprogramme, siehe OGH 28.10.1997, 4 Ob 304/97b – „einzigartiges“ EDV-Programm – ÖBI 1999, 57 = MR 1998, 72 (Walter) = wbl 1998, 181.

¹⁷⁵ Grünbuch der Kommission, Urheberrecht und die technologische Herausforderung, KOM(88) 172, abgedruckt in UFITA, Band 110, 1989, 241.

¹⁷⁶ KOM(88) 172, abgedruckt in UFITA, Band 110, 1989, 262.

¹⁷⁷ Wartinger, Urheberrecht und Arbeitsverhältnis (2006) 130 mwA.

Zusammenhang die Durchsetzung der mit der Urheberschaft verbundenen Rechte¹⁷⁸, die hier vom AG wahrgenommen werden können.

Wichtig anzumerken ist, dass die Bestimmung des § 40b UrhG nur auf Dienstnehmer abstellt, Auftragswerke bleiben davon unberührt¹⁷⁹. Diese „widerlegliche Vermutung“¹⁸⁰ erlaubt es den Vertragspartnern bei der Entwicklung von Open Source Software zwischen verschiedenen Vertragskonstellationen auszuwählen. Dies ist für den Open Source Bereich von großer Bedeutung, arbeiten doch viele Programmierer in diesem Bereich freiberuflich. Nachfolgend werden kurz die relevantesten Arbeitsverhältnisse und ihre rechtlichen Auswirkungen skizziert.

3.2.2.5.1. Neuentwicklung durch einen einzelnen Dienstnehmer

Dieser Fall stellt rechtlich die geringsten Schwierigkeiten dar¹⁸¹. § 40b UrhG behält das Schöpferprinzip bei¹⁸². Dem AG steht dabei ein unbeschränktes Werknutzungsrecht an dem von seinem AN geschaffenen Werk zu. Im Gegensatz zu Deutschland ist der AG auch gem § 40b 2. Satz 1. Teil UrhG zur Ausübung des Rechts auf Urheberbezeichnung iSd § 20 UrhG und des Werkschutzes iSd § 21 Abs 1 UrhG berechtigt. Da die Urheberschaft durch § 19 UrhG geschützt wird, räumt § 40b 2. Satz 2. Teil UrhG dem Urheber ein, diese für sich in Anspruch zu nehmen¹⁸³.

¹⁷⁸ Vgl Kap 3.2.3.

¹⁷⁹ Blocher in *Jahnel/Schramm/Staudegger*, Informatikrecht², 144; Loewenheim in *Schricker*, Urheberrecht² § 69b Rz 2. Siehe auch Kap 3.2.2.5.4.

¹⁸⁰ *Dillenz*, Praxiskommentar 120.

¹⁸¹ *Dinnes*, Softwareentwicklung im Vertragsverhältnis (2002) 90.

¹⁸² *Blocher/Walter* in *Walter*, Europäisches Urheberrecht 149.

¹⁸³ *Blocher/Walter* in *Walter*, Europäisches Urheberrecht 150.

Die Bestimmungen des § 40b UrhG sind als investitionsschützende Maßnahmen zu sehen, da der Arbeitgeber direkt in den Genuss kommt, die Software nach seinen eigenen Vorstellungen verwerten zu können.¹⁸⁴ Dies bedeutet im Ergebnis, dass ihm alle damit verbundenen Rechte zukommen und er selbst darüber entscheiden kann, ob er das neuentwickelte Werk – sollte er es gegenüber Dritten anbieten wollen – unter eine Open Source oder eine proprietäre Lizenz stellt. Der AN hat in diesem Fall kein Mitspracherecht, da seine Rechte gem § 40b UrhG direkt auf den AG übergegangen sind.

3.2.2.5.2. Neuentwicklung durch mehrere Dienstnehmer

§ 40b UrhG enthält keine speziellen Vorschriften, wie mit Neuentwicklungen durch mehrere Dienstnehmer umzugehen ist. Dieser Fall kommt in der Praxis jedoch sehr häufig vor, da Software oft in Teamarbeit erstellt wird. Entwickeln mehr als eine Person Open Source Software gemeinschaftlich als Dienstnehmer desselben Dienstgebers, sind sie hinsichtlich der entwickelten Software als Miturheber iSd § 11 UrhG anzusehen, da keiner von ihnen unmittelbarer alleine Rechtsträger des Urheberrechts an der Software wird, womit auch die verwertungsrechtlichen Befugnisse bei ihnen gesamthändisch gebunden sind¹⁸⁵. Wendet man wiederum § 40b UrhG auf den Fall an, so muss man feststellen, dass die ausschließlichen Nutzungsrechte durch die Zuweisungsanordnung des § 40b UrhG auf den AG zum Zeitpunkt des Entstehens des Miturheberrechts ex lege übergehen und nur die Urheberpersönlichkeitsrechte iSd § 19 UrhG der Miturhebergemeinschaft erhalten bleiben.

¹⁸⁴ *Dinnes*, Softwareentwicklung 97.

¹⁸⁵ *Dinnes*, Softwareentwicklung 106.

Dieses Ergebnis ist schlüssig, da alles andere als eine analoge Behandlung zur Neuentwicklung durch einen Einzelnen rechtlich schwer nachzuvollziehen und zu rechtfertigen wäre.

3.2.2.5.3. Weiterentwicklung durch einen einzelnen oder mehrere Arbeitnehmer

Open Source Software etabliert sich immer mehr im unternehmerischen Einsatz. Daher ist es für viele AG wirtschaftlich interessant, das Bearbeitungsrecht, welches ihnen in der jeweiligen Open Source Lizenz eingeräumt wird, zu nutzen, indem sie einen einzelnen oder mehrere AN zur Weiterentwicklung der Open Source Software einsetzen.

Während dem AG bei einer Neuentwicklung das alleinige Urheberrecht gem § 40b UrhG zukommt und er über das entstandene Werk frei verfügen kann, ist er bei einer Weiterentwicklung durch einen AN an die Open Source Lizenz gebunden, unter welcher das weiterentwickelte Werk lizenziert ist. Dies ergibt sich aus der Lizenz, welche der AG mit dem Open Source Lizenzgeber abschließt¹⁸⁶ und an dessen Bestimmungen er selbstverständlich gebunden ist. Die Annahme eines ausschließlichen Lizenzverhältnisses zwischen Open Source Lizenzgeber und AN muss ausgeschlossen werden, da dem AG nicht nur die Rechte an der Weiterentwicklung der Software zukommen, sondern auch die damit verbundenen Pflichten. Er kann deshalb nicht gegen die Open Source Lizenzbedingungen verstoßen unter dem Hinweis, dass sich die urheberrechtlichen Pflichten ihm gegenüber erschöpft hätten¹⁸⁷.

¹⁸⁶ Zum Vertragsabschluss siehe Kap 4.

¹⁸⁷ Zum Erschöpfungsgrundsatz siehe Kap 3.3.2.1.

Wichtig anzumerken ist in diesem Fall, dass dem AN keinerlei Verfügungsrechte über die weitere Verwertung der Open Source Software zukommen, da er de facto nicht Vertragspartner geworden ist. Entschließt sich ein AG dazu, gegen die Lizenzbedingungen zu verstoßen und die Weiterentwicklung proprietär weiterzuvertreiben, darf der AN nicht die weiterentwickelte Software seinerseits unter eine Open Source Lizenz stellen, da er dann das Urheberrecht seines AG verletzen würde. Dieses mögliche „Dilemma“ wird in der Praxis schwer zu lösen sein, da es in der Regel zu einem Interessenkonflikt beim AN kommen wird.

3.2.2.5.4. Neu- oder Weiterentwicklung auf Grund eines freien Dienstvertrags oder durch arbeitnehmerähnliche Personen

Die Übertragung der Nutzungs- und Verwertungsrechten an Neu- und Weiterentwicklungen von Open Source Software durch freie Dienstnehmer iSd § 1151 ABGB oder durch arbeitnehmerähnliche Personen iSd § 51 Abs 3 Z 1 ASGG sind ohne Rücksicht auf § 40b UrhG zu klären. Hier kommt dispositives Recht zur Anwendung, wonach die Werknutzungsrechte vertraglich dem Auftraggeber oder –nehmer eingeräumt werden können.¹⁸⁸ Im Falle einer Neuentwicklung wird die Entscheidung über die Rechteeinräumung den beiden Vertragspartnern alleine überlassen. Kommt es zu einer Weiterentwicklung müssen evtl auch die Rechte von Dritten berücksichtigt werden¹⁸⁹. Hierbei ist dann vorrangig auf die Rechte Rücksicht zu nehmen, welche einem vom Rechteinhaber des Ursprungswerks – bspw mittels Lizenz – eingeräumt wurden. Die Dispositionsfreiheit zwischen Auftraggeber und –nehmer bewegt sich dann ausschließlich in diesem Rahmen und es dürfen keine weiterführenden Rechte gegenseitig eingeräumt werden. Eine Überführung von

¹⁸⁸ Blocher in *Jahnel/Schramm/Staudegger*, Informatikrecht², 145.

¹⁸⁹ Dies ist dann der Fall, wenn die Urheberrechte am Ursprungswerk, das weiterentwickelt werden soll, weder dem Auftragnehmer, noch dem Auftraggeber zukommen.

GPL-lizenzierter Software in eine BSD-lizenzierte bspw ist ebenso wenig gestattet wie die Einräumung der Nutzung von Markenrechten, wenn diese durch die Ursprungslizenz untersagt wurde.

3.2.2.5.5. Abgrenzung zwischen in „Erfüllung der Arbeitspflicht“ entwickelte Open Source Software, bei „Gelegenheit des Arbeitsverhältnis“ entwickelte Open Source Software und „außerhalb des Arbeitsverhältnis“ entwickelte Open Source Software.

§40b UrhG spricht des Weiteren von Werken, die der AN „in Erfüllung seiner dienstlichen Obliegenheiten geschaffen“ hat. Hierbei kann es zu Grenzfällen kommen, welche in Folge erörtert und bewertet werden.

a) In Erfüllung der Arbeitspflicht entwickelte Open Source Software

In Erfüllung der Arbeitspflicht entwickelte Software liegt dann vor, wenn die Software unter Eingliederung in die Arbeitsorganisation des AG und aufgrund seines Auftrags und mit dessen Finanzierung entwickelt wurde. Bei der Bewertung muss darauf Rücksicht genommen werden, dass nur die Open Source Software als in Erfüllung der Arbeitspflicht angesehen werden kann, für die der AN entlohnt wurde. Der OGH spricht in diesem Zusammenhang von Pflichtwerken¹⁹⁰.

¹⁹⁰ OGH 16.06.1992, 4 Ob 65/92 – Übungsprogramm – EDVuR 1992/2, 133 = JBI 1993, 116 = ÖBI 1992, 281 = SZ 65/89.

b) Bei „Gelegenheit des Arbeitsverhältnis“ entwickelte Open Source Software

Bei „Gelegenheit des Arbeitsverhältnis“ entwickelte Software liegt dann vor, wenn der AN Open Source Software entwickelt, welche zwar außerhalb seiner arbeitsrechtlichen Pflichten liegt, aber trotzdem eine so enge Verbindung zu seinem Dienstverhältnis hat, dass die Software nicht als außerhalb des Arbeitsverhältnisses anzusehen ist. Ausgangspunkt der Bewertung muss dabei immer die Gegenüberstellung des Berufsbildes mit der vertraglich vereinbarten Aufgabe sein. Diese Abgrenzung ist besonders schwierig, da eindeutige Kriterien zur Bewertung fehlen. Das dt Bundesarbeitsgericht hat in seiner „Abdampfverwertungsentscheidung“¹⁹¹ dazu ausgeführt, dass sich eine solche Leistung eines AN dadurch qualifiziert, dass sie *„insbesondere eine Leistung schöpferischer Art (ist), die über die übliche Arbeitsleistung hinausgeht und eine echte Sonderleistung darstellt.“* Während der wirtschaftliche Nutzen der besonderen Leistung für den AG nur für die Frage der zusätzlichen Vergütung eine Rolle spielt, kann dieser für die urheberrechtliche Bewertung außer Acht gelassen werden. Beachtenswert dagegen ist die Vergütung der Sonderleistung durch den AG selbst. Erkennt dieser die Entwicklung als Sonderleistung an, kann er sie im Rahmen des Arbeitsverhältnisses dem AN vergüten. Somit ergeben sich die Wirkungen des § 40b UrhG. Vergütet der AG die Sonderleistung des AN diesem nicht, darf er sie in keinem Fall eigenmächtig verwerten, da er nicht durch § 40b UrhG die Verwertungsrechte daran erhalten hat. Streitbar erscheint in diesem Fall die Möglichkeit des AN, die Sonderleistung nach eigenem Interesse zu verwerten, da die Sonderleistung nicht gänzlich außerhalb des Dienstverhältnisses entwickelt wurde. Eine Lösung in diesem Fall könnte in Hinblick auf das Spannungsverhältnis des Rechts am Arbeitsergebnis darin gesehen werden, dass dem AG ein „Verwertungsprivileg“ zukommt. Nutzt der AG diese Option nicht, kann der AN die Sonderleistung in eigenem Interesse verwerten. Diskutierbar erscheint

¹⁹¹ Urteil des dt Bundesarbeitsgerichts vom 30. April 1965, 3 AZR 291/63 – Abdampferwertung – GRUR 1966, 88 (Vollmer).

dabei die Möglichkeit, den AG in angemessener Weise am Verwertungserlös zu beteiligen.

c) Außerhalb des Arbeitsverhältnis entwickelte Open Source Software

Hier muss man zwei Fälle unterscheiden, solche die in zeitlicher und solche die in sachlicher Hinsicht außerhalb des Arbeitsverhältnisses liegen.

In zeitlicher Hinsicht außerhalb des Dienstverhältnisses geschaffene Software liegt dann vor, wenn diese vor dem Beginn oder nach der Beendigung des Dienstverhältnisses geschaffen¹⁹² oder außerhalb der Dienstzeit entwickelt wurde. Die entwickelte Software ist dann nicht von der Regelung des § 40b UrhG betroffen.

In sachlicher Hinsicht außerhalb des Dienstverhältnisses geschaffene Software liegt dann vor, wenn die entwickelte Software nach objektiven Kriterien außerhalb des Zwecks des Dienstvertrags liegt (ein ausschließlich zur Softwarewartung angestellter Mitarbeiter entwickelt ein Buchhaltungsprogramm oder ein Programm zur grafischen Darstellung von 3D-Modellen), die Mittel für die Entwicklung nicht überwiegend beim AG liegen (die entwickelte Software des Mitarbeiters liegt zwar außerhalb des Zwecks seines Dienstvertrags, sie konnte aber nur durch die Nutzung des Hochleistungsrechners des AG geschaffen werden) und der AN auf eigene Initiative, das heißt ohne Auftrag des AG tätig geworden ist¹⁹³. Diese strengen Kriterien sind bei der Bewertung unbedingt heranzuziehen, da es im Streitfall schwer fallen wird, eine Abgrenzung vornehmen zu können.

¹⁹² Thiele, Übertragung von Urheberrechten auf den Arbeitgeber, RdW 2002, 537 (538).

¹⁹³ Blatz, Das Recht des Urhebers an literarischen Arbeitsergebnissen (1967) 102.

Schwierig wird die Beurteilung auch dann, wenn die Entwicklung von Open Source Software während der Dauer des Arbeitsverhältnisses begonnen, aber erst nach Beendigung des Dienstverhältnisses vollendet wurde. Jener Teil der Software, welcher während des Dienstverhältnisses entwickelt wurde, ist unbestritten dem Dienstgeber zuzurechnen¹⁹⁴. Der ausgeschiedene AN darf deshalb nicht den angefangenen Teil des Werks nehmen und selbst weiterentwickeln. Eine Ausnahme erfährt diese Bestimmung nur dann, wenn die Open Source Software durch den AG veröffentlicht wird und der ausgeschiedene AN nun selbst Lizenznehmer der Software werden kann - dies jeweils nach Maßgabe der genutzten Open Source Lizenz.

3.2.3. Der zivilrechtliche Schutz und die Auswirkung der verschiedenen Arten der Urheberschaft auf die Aktivlegitimation

Der zivilrechtliche Schutz für Urheber ist in den §§ 81 – 90d UrhG geregelt. Insbesondere der Unterlassungsanspruch iSd § 81 UrhG, der Beseitigungsanspruch iSd des § 82 UrhG, die Ansprüche auf angemessenes Entgelt iSd § 86 sowie Schadenersatz und Herausgabe des Gewinnes iSd § 87 UrhG sind im Zusammenhang mit Open Source Lizenzen zu erwähnen. Bislang wurde die Durchsetzung einer Open Source Lizenz weltweit nur einmal gerichtlich behandelt¹⁹⁵. Dennoch sind die in den §§ 81 – 90d UrhG geregelten Ansprüche wichtige Schutzinstrumente, mit deren Hilfe Verstöße gegen eine urheberrechtliche Bestimmung geahndet und sanktioniert werden können.

Auch die Feststellung der rechtmäßigen Aktivlegitimation ist im Falle einer gerichtlichen Auseinandersetzung von grundlegender Bedeutung, da eine

¹⁹⁴ Thiele, RdW 2002, 538.

¹⁹⁵ Dabei wurde die Wirksamkeit der Ziffer 4 GPL durch das LG München (Urteil vom 19.05.2004, Az 21 O 6123/04) bestätigt. Siehe Kap 5.4.1.2.2.

mangelnde Aktivlegitimation über Antrag zur Abweisung der Klage führen würde. Während im Falle eines Alleinurhebers die Sachlegitimation, ob der behauptete Anspruch dem Kläger zusteht, einfach festzustellen sein wird, kann dies im Fall der Miturheber schwierig werden. Dies gilt im Besonderen für Open Source Software, da deren Urheber – wie erwähnt – überall auf der Welt sitzen können. Die zivilrechtlichen Schutzmöglichkeiten und die Möglichkeit, urheberrechtliche Ansprüche durchzusetzen, werden daher nachfolgend erörtert.

3.2.3.1. Unterlassungsanspruch

Den Unterlassungsanspruch des Urhebers regelt § 81 UrhG, welcher die Forderung des Art 8 Abs 3 der Info-RL umsetzt, wonach einem Dritten bspw die Verwertung eines Programms, an dem der Kläger die Werknutzungsrechte¹⁹⁶ besitzt, untersagt werden kann¹⁹⁷. Ein Verschulden des Beklagten ist für den Unterlassungsanspruch nicht nötig¹⁹⁸, es reicht Begehungs- bzw Wiederholungsfahr. Den Wegfall der Wiederholungsfahr hat nach ständiger Rsp der Beklagte zu behaupten und zu beweisen¹⁹⁹. Der OGH gewährt gegen den Nachweis der Begehungsgefahr auch eine vorbeugende Unterlassungsklage²⁰⁰, die für den Softwarebereich von großer Bedeutung ist. Die allgemeine Internetnutzung bringt es mit sich, dass es dem Kläger in der Regel nicht hilft, ein dort bereits unrechtmäßig eingespeistes Programm mittels einer auf die Wiederholungsfahr gerichteten Unterlassungsklage zu schützen. Für Open Source Software ist ein Unterlassungsanspruch in jenen Fällen denkbar, in denen die Software entgegen den Lizenzbestimmungen

¹⁹⁶ Vgl Kap 3.3.1.

¹⁹⁷ OGH 17.12.1996, 4 Ob 2363/96w – Head-Kaufvertrag – ÖBI 1997, 256 = MR 1997, 93 (Walter) = SZ 69/283 = wbl 1997, 175.

¹⁹⁸ OGH 13.09.1999, 4 Ob 151/99f – Roll up – ÖBI 2000, 133 (Kucsko) = MR 1999, 343 (Walter).

¹⁹⁹ OGH 28.05.2002, 4 Ob 108/02i – Figurstudio – MR 2002, 236 (Walter).

²⁰⁰ OGH 12.08.1998, 4 Ob 193/98f – Wiener Gruppe – MR 1998, 287 (Walter) = GRUR Int 1999, 360.

verwertet wird, bspw durch das Verlangen einer Lizenzgebühr bei der Verbreitung oder die Geheimhaltung des Quellcodes bei einer Weiterentwicklung. Für Teilnehmer an Open Source Projekten ohne Firmenhintergrund, die ihre Projekte dezentral entwickeln und aufgrund dessen nicht so schnell auf einen solchen Verstoß reagieren können, ist wichtig anzumerken, dass auch ein Zuwarten mit der Antragstellung den Unterlassungsanspruch nicht verwirkt²⁰¹.

Gem § 81 Abs 1a UrhG muss sich der Unterlassungsanspruch jedoch nicht nur gegen denjenigen richten, welcher die urheberrechtliche Verletzung begangen hat, sondern er kann ebenso gegen den „Vermittler“ gerichtet sein, mit dessen Hilfe die Verletzung begangen wird. Für Open Source Software ist an dieser Stelle der Host-Provider als Vermittler zu erwähnen, da dieser als Erbringer von Diensten der Informationsgesellschaft – hier Bereitstellung des Webspaces – gesehen wird²⁰². Bevor der Urheber einen Unterlassungsanspruch mittels Klage durchsetzen kann, muss er gem § 81 Abs 1a Satz 2 UrhG dem Host-Provider den rechtsverletzenden Inhalt zur Kenntnis bringen, ihn „abmahnen“, solange die Tatbestände der §§ 13 - 17 ECG für die Haftungsfreistellung erfüllt sind. Der Host-Provider soll dabei die Möglichkeit erhalten, den rechtsverletzenden Inhalt, der ihm in der Regel nicht bekannt ist, zu entfernen, ohne sich einer Klage ausgesetzt zu sehen. Entfernt er diesen binnen angemessener Frist, bzw unterbindet er die Möglichkeit, diesen abzurufen, besteht der Unterlassungsanspruch des Urhebers nicht länger. Bisher ungeklärt ist der Anspruch, der sich aus § 81 Abs 1a Satz 2 UrhG gegenüber einem (Hyper-) Linksetzer²⁰³ ergibt, da auch gegen diesen mit einer Unterlassungsklage vorgegangen werden kann²⁰⁴. Es ist allerdings fraglich, ob dieser als Vermittler iSd § 81 Abs 1a UrhG angesehen werden kann. ME ist der Linksetzer dann als Vermittler iSd § 81 Abs 1a UrhG zu sehen, wenn er den

²⁰¹ OLG Wien 18.4.1986, 4 R 48/86 – UIP-Repertoire – MR 1986, 18 (*Walter*) in der ein Zuwarten von 5 Monaten hinsichtlich der Verwirkung zu bewerten war.

²⁰² *Walter* in *Walter* (Hrsg), Europäisches Urheberrecht Rz 139.

²⁰³ *Fallenböck/Galla/Stockinger*, Urheberrecht in der digitalen Wirtschaft (2005) 150.

²⁰⁴ OGH, 19.12.2000, 4Ob274/00y – jobmonitor.com – ÖBl 2001, 164.

Link – steht für „Verknüpfung“, „Verbindung“ oder „Verweis“ – bewusst auf den rechtsverletzenden Inhalt gesetzt hat. Die Kenntnis des Linksetzers über die urheberrechtliche Verletzung darf dabei nicht relevant sein, da sich der Begriff des „Vermittlers“ allein durch den Sinn und Zweck eines Links logisch erklärt. Links dienen dem Querverweis auf bereits vorhandene Inhalte, aufgrund deren es einem Benutzer möglich ist, auf diese „vermittelt“ zu werden. Die „Vermittlung“ ist der vorrangige Sinn und Zweck des Links²⁰⁵, weshalb es kaum nachvollziehbar ist, einen Linksetzer nicht als Vermittler iSd § 81 Abs 1a UrhG anzusehen. Im Ergebnis muss daher der Urheber einen Linksetzer vor Klageeinbringung des Unterlassungsanspruchs²⁰⁶ abmahnen.

Die Aktivlegitimation der Unterlassungsklage betreffend ist festzuhalten, dass bei einer Verletzung des Urheberrechtes gem § 11 Abs 2 Satz 1 UrhG jeder Miturheber alleine berechtigt ist, diese gerichtlich verfolgen zu lassen. Bei anderer Gesetzeslage wäre es schwierig, eine aktiv legitimierte Streitgenossenschaft für ein Gerichtsverfahren zu Stande zu bringen, je größer die Anzahl an Miturhebern ist. Ein wirksamer Rechtsschutz für Open Source Software wäre ohne diese Bestimmung nicht möglich, bedenkt man die weltweite Streuung der verschiedenen Urheber. Des Weiteren besteht nach § 11 Abs 2 UrhG die Möglichkeit – ungeachtet der Bestimmungen der Open Source Lizenzen – einen Miturheber, der sein Einverständnis ohne guten Grund verweigert, auf die Erteilung zu klagen. Umgekehrt kann ein Dritter, der seine Miturheberschaft zu Unrecht behauptet, mittels negativer Feststellungsklage bekämpft werden²⁰⁷.

²⁰⁵ „*The intention in the design of the web was that normal links should simply be references, with no implied meaning.*“ Eine Aussage des „Erfinders“ des Internet *Tim Berners-Lee* <http://www.w3.org/DesignIssues/LinkLaw.html>.

²⁰⁶ Interessant in diesem Zusammenhang ist, dass die hier aufgeführten Unterlassungsansprüche im eigentlichen Sinn Beseitigungsansprüche sind. § 82 UrhG weist daher auch auf die analoge Anwendung des § 81 Abs 1a UrhG hin.

²⁰⁷ OGH 14.03.2000, 4 Ob 41/00h – Zahnarztprogramm – MR 2000, 312 (*Walter*).

3.2.3.2. Beseitigungsanspruch

Dem Unterlassungsbegehren in der Praxis nachfolgend ist in der Regel der Beseitigungsanspruch, welcher in § 82 UrhG geregelt ist, wonach derjenige, dessen Ausschließungsrecht verletzt wurde, die Beseitigung dieses Zustandes verlangen kann. Neben der Vernichtung der *„zuwider hergestellten oder verbreiteten sowie der zur widerrechtlichen Verbreitung bestimmten Vervielfältigungsstücke“* ist bei Verbreitung über das Internet dafür Sorge zu tragen, dass auch die Caches der allgemein gebräuchlichen Suchmaschinen auf Abrufbarkeit dieser Vervielfältigungsstücke hin überprüft werden. Hier reicht eine Löschung der Software auf dem Server nicht aus, da die rechtswidrige Verbreitung hierdurch allein nicht beseitigt sein muss. Gängige Suchmaschinen speichern Daten in eigenen Caches ab, welche in Folge weiterhin von Dritten abgerufen werden können. Deshalb kann der Kläger im Rahmen des Beseitigungsanspruchs die Löschung dieser Caches begehren, um deren wirksame Beseitigung zu erreichen²⁰⁸.

Wie beim Unterlassungsanspruch gilt für den Beseitigungsanspruch gem § 11 Abs 2 Satz 1 UrhG, dass jeder Miturheber alleine berechtigt ist, diesen zu begehren. Da der Beseitigungsanspruch nach erfolgter Verletzung des Urheberrechts in der Praxis mit dem Unterlassungsanspruch verbunden sein wird, wäre eine unterschiedliche Behandlung in Bezug auf die Aktivlegitimation kaum zu rechtfertigen.

²⁰⁸ Vgl Ausführungen zu § 81 Abs 1a UrhG in Kap 3.2.3.1.

3.2.3.3. Anspruch auf angemessenes Entgelt und der Anspruch auf Schadenersatz und die Herausgabe des Gewinns

Das UrhG stattet Urheber in den §§ 86 und 87 UrhG mit Leistungsklagen aus. Gem § 86 UrhG haben diese einen verschuldensunabhängigen Anspruch²⁰⁹ auf angemessenes Entgelt (weitergehende Ersatzansprüche nach dem ABGB sind dabei ausgeschlossen²¹⁰), wobei für Software die Höhe der Lizenzgebühr als angemessenes Entgelt gesehen wird²¹¹. Diese Lösung ist für Open Source Software unbefriedigend, da Open Source Lizenzen Lizenzgebühren ausschließen. Hierbei sollte auf die Bereitstellungskosten Rücksicht genommen werden, da diese dem Entgelt entsprechen, das üblicherweise für eine gleichartige Leistung bei im Vorhinein eingeholter Einwilligung des Urhebers hätte bezahlt werden müssen. Das Fehlen einer Gewinnerzielungsabsicht ist bei der Bemessung des angemessenen Entgelts nach Meinung des OGH irrelevant²¹², weshalb jedes Open Source Projekt das Recht hat, dies einzufordern.

Des Weiteren stehen den Rechteinhabern gem § 87 UrhG die verschuldensabhängigen Ansprüche auf Schadenersatz und Herausgabe des Gewinns zu. Liegt ein Verschulden – leichte Fahrlässigkeit ist ausreichend²¹³ – auf Seiten des Beklagten vor, ist dem Urheber gem § 87 Abs 3 UrhG Schadenersatz in mindestens doppelter Höhe des angemessenen Entgelts – hier müssen ebenfalls die Bereitstellungskosten herangezogen werden – zu leisten. Dieser Betrag entspricht einem pauschalierten Schadenersatz, bei dem

²⁰⁹ OGH 29.01.2002, 4 Ob 279/01k – Aufzugsanlagen – MR 2002, 156 (*Walter*)

²¹⁰ *Dittrich*, Österreichisches und internationales Urheberrecht⁴ § 86 E1 mwA.

²¹¹ OGH 13.11.2001, 4 Ob 249/01y – Wirtschaftskurier – MR 2002, 101.

²¹² OGH 13.11.2001, 4 Ob 249/01y – Wirtschaftskurier – MR 2002, 101.

²¹³ OGH 26.05.1998, 4 Ob 63/98p – Rauchfänge – MR 1998, 197 (*Walter*) = JBL 1998, 793 (*Mahr*).

weder nachgewiesen werden muss, dass ein Schaden eingetreten ist noch wie hoch dieser tatsächlich ist²¹⁴.

Der entgangene Gewinn ist gem § 87 Abs 1 UrhG ohne Rücksicht auf den Grad des Verschuldens zu ersetzen und dem Rechteinhaber herauszugeben. In der Praxis wird es in der Regel schwierig sein, die Höhe des entgangenen Gewinns zu beweisen. Festzuhalten ist, dass der OGH ausgesprochen hat, die §§ 86 und 87 UrhG alternativ heranzuziehen²¹⁵. Dies führt zu dem Ergebnis, dass dem Urheber bei Verschulden des Beklagten die Zahlung des doppelten, ansonsten die Zahlung des einfachen ihm gebührenden Entgelts zusteht.

Bei Schadenersatzansprüchen kommt dem Host-Provider ebenso wie bei den Unterlassungs- und Beseitigungsansprüchen eine wichtige Rolle zu. Urheberrechtsverletzungen in Zusammenhang mit Open Source Software bestehen nicht zuletzt durch die Abrufbarkeit rechtsverletzender Inhalte im Internet, welche durch den Host-Provider bereitgestellt werden. Fraglich in diesem Zusammenhang ist daher, ob Schadenersatzansprüche gem § 87 UrhG gegen den Host-Provider gerichtet werden können²¹⁶. Das könnte der Fall sein, wenn der Schaden diesem kausal zumindest mittelbar²¹⁷ zugerechnet werden könnte. Dies ist bspw der Fall, wenn der Host-Provider unrechtmäßig (Open Source) Software zum Herunterladen bereitstellt. Dies muss darüber hinaus eine rechtswidrige Handlung darstellen. Der Host-Provider handelt schon dann rechtswidrig, wenn gegen den Inhalt, den er bereitstellt, ein

²¹⁴ OGH 26.005.1998, 4 Ob 63/98p – Rauchfänge – MR 1998, 197 (*Walter*) = JBL 1998, 793 (*Mahr*).

²¹⁵ OGH 26.05.1998, 4 Ob 63/98p – Rauchfänge – MR 1998, 197 (*Walter*) = JBL 1998, 793 (*Mahr*); OGH 24.11.1998, 4 Ob 292/98i – Mittelschulatlant – ecolex 1999, 167 (409, *Tahedi*) = MR 1999, 171 (*Walter*).

²¹⁶ *Fallenböck/Galla/Stockinger*, Urheberrecht in der digitalen Wirtschaft 153.

²¹⁷ Eine unmittelbare Zurechnung wäre in den Fällen anzunehmen, in denen der Host-Provider bspw selbst unrechtmäßig Software auf seinen Servern einsetzt, um seine Dienste anbieten zu können. Im vorliegenden Fall ist der Host-Provider aber nur mittelbarer Täter, da die Inhalte von einem Dritten auf den Webspace des Host-Providers gespeichert wurden.

Unterlassungsanspruch eines Urhebers besteht. Letztendlich müsste der Host-Provider ein Verschulden am Schaden haben. Hier reicht – wie ausgeführt – leichte Fahrlässigkeit aus, wobei sich diese an der Kenntnis bzw. das „Kennenmüssen“ iSd Haftungsprivilegs gem § 16 ECG des Host-Providers über die rechtswidrige Tätigkeit bemisst²¹⁸.

Während die Durchsetzung der Unterlassungs- und Beseitigungsansprüche kaum rechtliche Schwierigkeiten aufweisen, kann es bei Leistungsklagen iSd problematisch werden²¹⁹. Sollten die Miturheber eine Gesamthandgemeinschaft iSd § 890 iVm § 848 ABGB bilden²²⁰, ist im Streitfall der Schuldner nicht verpflichtet, die geschuldete Leistung an einen Einzelnen zu erbringen. Drängt der Schuldner auf die Übereinkunft aller Mitgläubiger – Name und Anteil der einzelnen Miturheber müssen dabei nachgewiesen werden, was bei größeren Open Source Projekten oftmals unmöglich festzustellen sein wird²²¹ – bleibt die Klage erfolglos. Bei Fällen, in denen die Urheber im Quelltext oder an sonstiger Stelle dokumentiert sind, kann die Urhebervermutung des § 12 Abs 1 UrhG helfen²²², wonach es zu einer Umkehr der Beweislast kommt. Allerdings wird der Schuldner ohne Sicherstellung nur frei, wenn die von ihm begehrte Leistung allen Gläubigern oder demjenigen, der dieser Gesellschaft vorsteht²²³, zugekommen ist²²⁴. Hier wird ein möglicher Lösungsansatz für Open Source Projekte ersichtlich, bei denen eine bestimmte Person oder ein Gremium dem Projekt vorsteht. Ein Problem in diesem

²¹⁸ *Fallenböck/Galla/Stockinger*, Urheberrecht in der digitalen Wirtschaft 154.

²¹⁹ *Küng*, Urheberrechtliche Aspekte freier Software, MR 2004, 21 (25).

²²⁰ In die Richtungweisend: OGH 14.02.1958, 3 Ob 577/57, ÖBI 1958, 71 (73); weiterführend: *Dillenz*, Urheberrechtsschutz heute, ÖBI 1990, 1 (4); nur als „stark annähernd“ bezeichnend: *Gamerith*, Sind Rechtsgemeinschaften an Immaterialgüterrechten Gesamthandgemeinschaften? ÖBI 1996, 63 (68 f); in Bezug auf die Verwertungsrechte: OGH 10.07.2001, 4 Ob 155/01 – C-Villas – MR 2001, 311 = ÖBI 2003, 252 (254) = EvBI 2002/7 = RdW 2001/750 = ecolex 2001/352 (*Schanda*) = ZUM-RD 2002, 135 = GRURInt 2002, 452.

²²¹ Vgl die Suchaktion vom mozilla.org unter: <http://www.mozilla.org/MPL/missing.html>.

²²² *Wiebe/Prändl*, ÖJZ 2004, 630.

²²³ OGH 04.10.1979, 7 Ob 727/79, JBI 1980, 318.

²²⁴ *Gamerith* in *Rummel* (Hrsg), Kommentar zum Allgemeinen bürgerlichen Gesetzbuch I³ (2000) § 890 Rz 3.

Zusammenhang ist dabei die verfahrensrechtliche Beurteilung der Aktivlegitimation, da die Klagebefugnis als öffentlichrechtlicher und unverzichtbarer Anspruch nicht ohne den zugrunde liegenden materiellrechtlichen Anspruch abgetreten werden kann²²⁵.

Die Aktivlegitimation könnte durch den Zusammenschluss zu einer Streitgenossenschaft iSd § 11 ZPO gegeben sein. Diese entsteht nicht ausschließlich durch die in § 11 Z 1 und 2 ZPO aufgezählten Fälle, sondern immer dann, wenn andere Gesetze die Möglichkeit der Klage von oder gegen eine Personenmehrheit ausdrücklich zulassen²²⁶. Das Entstehen der Streitgenossenschaft liegt im Ermessen der klagenden Partei und kann nicht vom Prozessgegner oder durch das Gericht erzwungen werden. Darüber hinaus müssen die übrigen Streitgenossen nicht als Hauptparteien am Prozess teilnehmen, was Open Source Projekten mit einer großen Anzahl an Miturhebern entgegenkommt. Die Nichtteilnahme begründet dabei nicht das Fehlen einer Prozessvoraussetzung, kann allerdings bei der Beurteilung der Aktivlegitimation zur Abweisung der Klage führen.²²⁷

Der Zusammenschluss zu einer Streitgenossenschaft durch die Miturheber von Open Source Software führt zu einer eigentlichen bzw materiellen Streitgenossenschaft iSd § 11 Z 1 ZPO. Die Streitgenossen können in Anbetracht des Streitgegenstands in Rechtsgemeinschaft stehen (§ 11 Z 1 Fall 1 ZPO), bspw wenn sich Mit- oder Teilurheber zu einer GesbR zusammengeschlossen haben. Sie können auch aus demselben tatsächlichen Grund oder solidarisch berechtigt sein (§ 11 Z 1 Fall 2 ZPO), bspw bei der Verwertung einer Open Source Software, welche nicht den Lizenzbedingungen entspricht. Die Streitgenossenschaft iSd § 11 ZPO bietet für Open Source

²²⁵ Zu dieser Ansicht kommen sowohl die hL, sie bspw *Grillberger*, Zur Einziehung fremder Forderungen in eigenem Namen, ÖJZ 1978, 141; *Fasching*, Lehrbuch des österreichischen Zivilprozeßrechts² (1990) Rz 344, als auch die Rsp, siehe bspw OGH 07.02.1969 = SZ 42/105 = EvBl 1970, 147 = JBI, 198; OGH 15.10.2002, 4 Ob 209/02t = EvBl 2003, 26.

²²⁶ *Fasching*, Kommentar zu den Zivilprozessgesetzen Bd 2 Teilbd 1² (2002), § 11 Rz 1.

²²⁷ *Fasching*, Kommentar Bd 2 Teilbd 1² § 11 Rz 4.

Projekte mit zahlreichen Miturhebern viele Vorteile. Eine Klage muss nicht daran scheitern, dass Miturheber in anderen Ländern wohnhaft sind und nicht als Hauptpartei am Prozess teilnehmen müssen. Des Weiteren besteht die Möglichkeit, sich nachträglich am Verfahren zu beteiligen. Die Leistungsdurchsetzung für eine Open Source Gemeinschaft kann demnach durch den Zusammenschluss zu einer Streitgenossenschaft erleichtert werden, wenn dies auch keine optimale Lösung darstellt. Hierbei zeigt sich, dass es dem Gesetzgeber bisher schwer gefallen ist, auf internationale Sachverhalte einzugehen, um für mehr Rechtssicherheit zu sorgen.

3.3. Das Urhebervertragsrecht und die Open Source Lizenzen

Das Urhebervertragsrecht stellt die rechtlichen Rahmenbedingungen für Software-Lizenzverträge dar, die sowohl im Geschäftsleben als auch im Privatbereich immer mehr an Bedeutung gewinnen. Ein Hauptmerkmal des Urheberrechts bzw des Immaterialgüterrechts liegt darin, dass der Urheber bzw der Rechteinhaber die Möglichkeit hat, Dritte vom bestimmungsgemäßen Gebrauch auszuschließen – siehe bspw § 354 ABGB – oder zumindest einen Anspruch auf angemessene Bezahlung – siehe bspw §§ 16a Abs 1 oder 16b UrhG - hat²²⁸. Diese Ansprüche können gegen jedermann ohne das Erfordernis durchgesetzt werden, dass eine vertragliche Bindung zwischen dem Rechteinhaber und demjenigen, der die Rechtsverletzung begeht, bestehen muss. Der Urheber genießt daher ein „*ius excludendi*“ – ein Ausschließungsrecht – welches ihm die Möglichkeit vorbehält, Dritten Rechte an seinem Werk einzuräumen²²⁹. Dieses Instrument der Rechtseinräumung bietet der „Lizenzvertrag“, durch welchen ein Urheber die in den §§ 14 – 18a UrhG ihm vorbehaltenen Verwertungsarten auch Dritten einräumen kann. Der

²²⁸ Kucsko in Kucsko, Geistiges Eigentum, 94.

²²⁹ Kucsko in Kucsko, Geistiges Eigentum, 95.

Abschluss eines Lizenzvertrags für die Rechtseinräumung hat sich in Zusammenhang mit Software bewährt²³⁰. In diesem können nicht nur Rechte aus einem bestimmten Teil des Immaterialgüterrechts übertragen werden, sondern es besteht die Möglichkeit, alle gewünschten Rechtseinräumungen vorzunehmen. In Open Source Lizenzen finden sich daher nicht nur Bestimmungen zum urheberrechtlichen Umgang, sondern oft auch zum Marken- oder Patentrecht.

Während bei proprietären Lizenzverträgen der Schutz des geistigen Eigentums vorrangig dazu dient, kommerzielle Interessen zu schützen, ist der Schutzgedanke bei Open Source Lizenzen darauf ausgerichtet, die freie Verfügbarkeit des Quellcodes zu garantieren. Dies schließt eine kommerzielle Verwertung der Software durch einen Lizenzgeber nicht aus. Allerdings bleibt es ihm verwehrt, aufgrund der Maxime der Open Source Lizenzen, keine Lizenzgebühren verlangen zu dürfen, Entgelt aus der Lizenzierung selbst zu erhalten. Die Geschäftsfelder im Open Source Bereich unterscheiden sich daher von denen „klassischer“ Softwareverwertung und werden in Folge der wachsenden Popularität von Open Source Software ständig erweitert. Die typischen Einsatzfelder von Open Source Anbietern sind dabei der Verkauf von Softwaredistributionen, Dienstleistungen, wie der technische Support und die Schulung der Benutzer, die Softwareentwicklung und die Implementation von Open Source Software als Embedded Systems.

Die Rechtseinräumung kann durch den Alleinurheber oder die Miturheber geschehen, wobei § 11 Abs 2 UrhG besagt, dass jede Änderung oder Verwertung des Werks das Einverständnis aller Miturheber benötigt. Dieses Erfordernis findet in den Open Source Lizenzen besondere Berücksichtigung²³¹. Diese verlangen von einem potentiellen Miturheber, dass er das Recht zur Veränderung und Verwertung erteilt, sollte er durch seine Mitarbeit zum

²³⁰ Siehe auch: *Staudegger in Jahn/Schramm/Staudegger*, Informatikrecht², 85.

²³¹ Vgl Kap 5.

Miturheber werden. Die GPL spricht in ihrer Präambel bspw „*you must give the recipients all the rights that you have*“ und meint damit, dass einem anderen die gleichen Rechte, die einem selbst eingeräumt wurden, nach einer Änderung weitergegeben werden müssen. Sollte er sich damit nicht einverstanden erklären, wird ihm das Recht zur Veränderung oder Verwertung ex nunc entzogen. Ein weiterer wesentlicher Aspekt in Bezug auf die Open Source Lizenzen ist dabei die in § 883 ABGB statuierte Formfreiheit bei Abschluss eines Vertrags iSd § 861 ABGB, die dem Lizenzgeber erlaubt, die Rechtseinräumung individuell zu regeln. Diese Rechte werden in Folge näher dargestellt.

3.3.1. Die Open Source Lizenzen – Werknutzungsbewilligung oder Werknutzungsrecht?

Die Unterscheidung zwischen der in § 24 UrhG geregelten Einräumung einer Werknutzungsbewilligung oder einem Werknutzungsrecht ist für Open Source Lizenzen von grundlegender Bedeutung. Die Werknutzungsbewilligung iSd § 24 Abs 1 Satz 1 UrhG ist dadurch charakterisiert, dass der Lizenzgeber (bzw der Urheber) dem Lizenznehmer das Recht einräumt, das Programm auf einzelne oder alle nach den §§ 14 bis 18a UrhG dem Urheber vorbehaltenen Verwertungsarten zu nutzen. Eine Werknutzungsbewilligung gewährt daher kein absolutes, sondern nur ein relatives Recht²³². Ein Werknutzungsrecht iSd § 24 Abs 1 Satz 2 UrhG dagegen überträgt dem Lizenznehmer eine ausschließliche Nutzungsberechtigung und damit die alleinige positive Befugnis, die durch den Lizenzvertrag eingeräumten Rechte auszuüben²³³. Der Lizenznehmer verliert durch die Übertragung des Werknutzungsrechts die

²³² OGH 09.04.2002, 4 Ob 77/02f – Geleitwort – MR 2002, 387.

²³³ *Stumpf/Groß*, Der Lizenzvertrag⁸ (2005) Rz 36; *Reinprecht* in *Kucsko*, Geistiges Eigentum 1244.

Möglichkeit, das Programm selbst zu verwerten oder Dritten dieses Recht erneut einzuräumen²³⁴.

Der Urheber räumt dem Nutzer daher durch die Open Source Lizenzen eine nicht ausschließliche Werknutzungsbewilligung iSd § 24 Abs 1 Satz 1 UrhG ein, da die Rechtseinräumung einer unbegrenzten Anzahl an Lizenznehmern zukommen soll und der Lizenzgeber selbst weiterhin in der Lage bleiben will, sein Programm weiterzuentwickeln und zu verwerten²³⁵.

3.3.2. Schutzrechtsumfang und -wirkungen

Der Schutzrechtsumfang durch die Werknutzungsbewilligung iSd § 24 Abs 1 Satz 1 UrhG und dessen Auswirkungen sind im Open Source Bereich in den jeweiligen Lizenzen geregelt. Vorrangig ist die Behandlung des Erschöpfungsgrundsatzes, welchem im Bereich des Schutzrechtsumfangs eine zentrale Rolle zukommt und dessen Auswirkungen auf Open Source Lizenzen eine wesentliche Antwort auf die Frage ermöglichen, ob die in den Open Source Lizenzen eingeräumten Verbreitungsrechte eine Erschöpfung erfahren. Nachfolgend wird die Problematik der Zugänglichmachung im Internet, die des Vermietens und abschließend die des Application Service Providing (ASP) behandelt, wobei bei Letzterem besonderer Bezug auf die Einräumung von Nutzungsrechten für unbekanntere Nutzungsarten genommen wird.

²³⁴ OGH 04.07.2000, 4 Ob 171/00a – Glückwunschkartenmotive – MR 2000, 318 (*Walter*).

²³⁵ *Küng*, MR 2004, 23; *Wiebe/Prändl*, ÖJZ 631.

3.3.2.1. Die Problematik des Erschöpfungsgrundsatzes

Der Zweck der Erschöpfung im Rahmen des Verbreitungsrechts ist es, einerseits dem Urheber die Belohnung für seine Arbeit zuzusichern²³⁶, indem er angemessen entlohnt wird, andererseits das Interesse der Allgemeinheit an klaren und übersichtlichen Rechtsverhältnissen zu berücksichtigen²³⁷. Aus diesem Grund behält § 16 Abs 1 UrhG dem Urheber das ausschließliche Recht zu, Werkstücke zu verbreiten. Ohne dessen Einwilligung dürfen seine Werkstücke daher nicht feilgehalten werden oder auf eine Art, die das Werk der Öffentlichkeit zugänglich macht, in Verkehr gebracht werden. Dieser Grundsatz wird durch die Bestimmung des § 16 Abs 3 UrhG dahingehend eingeschränkt, dass Werkstücke, die mit Einwilligung des Urhebers durch Übertragung des Eigentums in einem Mitgliedsstaat der EG oder einem Vertragsstaat des EWR in Verkehr gebracht worden sind – vorbehaltlich den Bestimmungen über das Vermieten und Verleihen gem § 16a UrhG – nicht mehr dem Verbreitungsrecht unterliegen. Damit wurde durch die UrhG-Novelle 2003²³⁸ die Regelung der Erschöpfung des Verbreitungsrechts iSd § 16 Abs 3 UrhG an Art 4 Abs 2 InfoRL angepasst und die bisherige Einschränkung gestrichen, nach der eine Verbreitung innerhalb des genehmigten territorialen Gebiets nicht zur Erschöpfung des Verbreitungsrechts auch außerhalb von diesem Gebiet führte²³⁹.

²³⁶ ISd „Belohnungstheorie“; vgl *Reimer*, Der Erschöpfungsgrundsatz im Urheberrecht und gewerblichen Rechtsschutz unter Berücksichtigung der Rechtsprechung des Europäischen Gerichtshofes, GRURInt 1972, 221 (225).

²³⁷ *Plöckinger*, Der Erschöpfungsgrundsatz im Marken- und Urheberrecht (2002) 101.

²³⁸ EB UrhG-Nov 2003 zu Z 3, 40 BlgNR XXII. GP.

²³⁹ *Reinprecht* in *Kucsko*, Geistiges Eigentum 1178; *Walter*, Urheberrechtsgesetz – UrhGNov 2003 (2003) 24.

Für Open Source Lizenzen hat der Erschöpfungsgrundsatz zentralen Charakter. Die Verpflichtung zur Einhaltung der Lizenzbedingungen bei Weitergabe der Software ist ein wesentliches Schutzinstrument²⁴⁰, da nur so die „Freiheit“ der Software gewährleistet werden kann. Die Weitergabe der Software kann im Open Source Bereich auf verschiedene Arten vorgenommen werden. Bei Vermietung oder bei Veränderung der Software durch den Anwender findet keine Erschöpfung statt, da es in letzterem Fall zu einer neuerlichen Erstverbreitung kommt²⁴¹. Das im nächsten Punkt angesprochene Herunterladen der Software wirft allerdings in Hinblick auf die Erschöpfung Fragen auf.

3.3.2.1.1. Download

Eine Erschöpfung beim Download von Software ist in der Lehre umstritten²⁴². In den Erwägungsgründen der Info-RL²⁴³ wird in Erwägungsgrund 29 eine Erschöpfung bei Online-Diensten abgelehnt, *„anders als bei CD-ROM oder CD-I, wo das geistige Eigentum in einem materiellen Träger, d.h. in einem Gegenstand verkörpert ist, ist jede Bereitstellung eines Online-Dienstes im Grunde eine Handlung, die zustimmungsbedürftig ist, wenn das Urheberrecht oder ein verwandtes Schutzrecht dies vorsieht.“*

²⁴⁰ *Wiebe/Prändl*, Open Source Software, Rechtliche Rahmenbedingungen nach österreichischem Recht, ÖJZ 2004, 628 (632).

²⁴¹ *Wiebe/Prändl*, ÖJZ 2004, 632. An dieser Stelle wird darauf hingewiesen, dass die Frage, ob die Erschöpfungswirkung das gesamte Verbreitungsrecht oder nur den eingeräumten Teil erfasst, in Österreich noch weitgehend ungeklärt ist. Für die Konstellation des „Ersterwerbs“ der GPL siehe Kap 5.4.1.2.

²⁴² Gegen eine Erschöpfung: *Spindler/Wiebe*, Open Source Vertrieb, CR 2003, 873 (876) mwA; *Spindler*, Rechtsfragen der Open Source Software (2004) 50. Dafür: *Berger*, Urheberrechtliche Erschöpfungslehre und digitale Informationstechnologie, GRUR 2002, 198 (203); *Handig*, Urheberrechtliche Erschöpfung von Downloads im World Wide Web, RdW 2003, 2 (3); *Knies*, Erschöpfung online? GRUR Int 2002, 314; *Küng*, Urheberrechtliche Aspekte Freier Software, MR 2004, 21 (29).

²⁴³ Richtlinie 2001/29/EG des Europäischen Parlaments und des Rates vom 22. Mai 2001 zur Harmonisierung bestimmter Aspekte des Urheberrechts und der verwandten Schutzrechte in der Informationsgesellschaft, ABI 2001 L 167, 10, in der Folge als InfoRL bezeichnet.

Kritiker²⁴⁴ bemängeln, dass es hier zur Ungleichbehandlung kommt, da sich zwar der Transferweg – Download einerseits und körperliche Übergabe des Werkstücks andererseits – unterscheiden mag, der Zweck aber der Gleiche ist: Beide Male wird die Software auf den Computer des Benutzers geladen werden müssen, um genutzt werden zu können. Vielfach wird dabei darauf verwiesen, dass der Berechtigte auch beim Download die Möglichkeit besitzt, dafür eine angemessene Vergütung zu verlangen²⁴⁵, um im Sinne der vorher erwähnten „Belohnungstheorie“ befriedigt zu werden. Ebenso spräche die Verkehrssicherungstheorie²⁴⁶ für eine Erweiterung, da eine Erschöpfungswirkung durch das Interesse des Käufers und der Allgemeinheit an der freien Zirkulation der Waren gerechtfertigt ist. Daher ist der Meinung zuzustimmen, dass eine Erschöpfung bei der Online-Übertragung eintritt. Wichtig ist anzumerken, dass der Erwerber von Open Source Software mittels Download auf den Erwerb der urheberrechtlichen Nutzungsrechte angewiesen ist²⁴⁷, da sich der Erschöpfungsgrundsatz ausschließlich auf die Verbreitung bezieht, nicht jedoch auf andere Verwertungsrechte wie bspw. das Vervielfältigungsrecht²⁴⁸. Diese Nutzungsrechte räumt dem Anwender die jeweilige Open Source Lizenz, verbunden mit den darin enthaltenen Verpflichtungen, ein. Da der Vertragsabschluss hinsichtlich der Nutzungsrechte und -pflichten allerdings dem Download zeitlich nachfolgt²⁴⁹, kommt man zu folgendem Ergebnis²⁵⁰:

²⁴⁴ *Knies*, GRURInt 2002, 316; *Küng*, MR 2004, 29; *Handig*, RdW 2003, 3; *Berger*, Urheberrechtliche Erschöpfungslehre und digitale Informationstechnologie, GRUR 2002, 198 (200).

²⁴⁵ *Knies*, GRUR Int 2002, 316; *Handig*, RdW 2003, 4; *Berger*, GRUR 2002, 200.

²⁴⁶ *Joos*, Die Erschöpfungslehre im Urheberrecht (1991) 53.

²⁴⁷ *Knies*, GRUR Int 2002, 316.

²⁴⁸ *Blocher* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 146.

²⁴⁹ Siehe dazu Kap 4.2.3.

²⁵⁰ Siehe auch *Küng*, MR 2004, 29 f.

Hinsichtlich der mittels Download erhaltenen Erstkopie kommt es zur Erschöpfung, die keinem Verbreiterungsrecht des Urhebers mehr unterliegt und deshalb ohne Rücksichtnahme auf die Bestimmungen der jeweiligen Open Source Lizenz weitergegeben werden kann. Jede weitere Kopie der Erstkopie muss dagegen die Bestimmungen der Open Source Lizenz beachten, da hieran das Verbreiterungsrecht mangels „Zustimmung des Berechtigten“ noch nicht erschöpft ist²⁵¹. Die Frage nach der Geltung dieser Bestimmungen in Hinblick auf den Erschöpfungsgrundsatz wird zusammen mit der Eigentumsübertragung ohne Download nachfolgend behandelt.

3.3.2.1.2. Eigentumsübertragung ohne Download

Es bleiben jene Fälle übrig, bei denen eine bereits in Verkehr gebrachte Open Source Software durch Weitergabe im Objekt- oder Quellformat verbreitet wird, wobei § 16 Abs 3 UrhG auf jede Form der Veräußerung abstellt, also auch solche Eigentumsübertragungen wie Tausch oder Schenkung umfasst²⁵². Erschöpfung tritt demnach in jenen Fällen ein, in denen ein Urheber Kopien seiner Open Source Software einem Distributor auf Dauer zur Weiterverbreitung, bspw durch Verkauf, überlässt und diese dann an einen Anwender²⁵³ weitergibt. Dies bedeutet, dass der Zweiterwerber der Open Source Software urheberrechtlich keiner urheberrechtlich-dinglichen Beschränkung des Verbreiterungsrechts mehr unterliegt²⁵⁴. Als letzter Ausweg bleibt, die in einer Open Source Lizenz enthaltenen Klauseln als rein schuldrechtliche Beschränkungen anzusehen, die dann der Inhaltskontrolle

²⁵¹ Blocher, Urheberrecht im Internet, in *Studiengesellschaft für Wirtschaft und Recht* (Hrsg), Internet und Recht. Rechtsfragen von E-Commerce und E-Government (2002) 223 (231).

²⁵² Walter, UrhGNov 2003, 26. So auch Art 4 Abs 2 der InfoRL.

²⁵³ Wiebe/Prändl, ÖJZ 2004, 632.

²⁵⁴ Spindler/Wiebe, CR 2003, 877. AA Koch, Urheber und kartellrechtliche Aspekte der Nutzung von Open-Source-Software (II) CR, 2000, 333 (335) f.

gem § 864a und insbesondere 879 Abs 3 ABGB unterliegen²⁵⁵. Hierbei muss eine ausführliche Interessensabwägung stattfinden, die den Erschöpfungsgrundsatz als Maßstab der Inhaltskontrolle hat.

Berücksichtigt man die eigenständige Art der Open Source Software, muss man zu dem Ergebnis kommen, dass die in den Open Source Lizenzen zu findenden „Beschränkungsklauseln“ keine gröbliche Benachteiligung darstellen und daher schuldrechtliche Wirkung entfalten, da die Rechte, die man in den Open Source Lizenzen eingeräumt bekommt, durch die für die Weitergabe auferlegten Pflichten zwar eingeschränkt werden, im Endeffekt aber allen Beteiligten zum Vorteil gereichen. Der Copyleft-Effekt wird daher nicht vom Erschöpfungsgrundsatz beseitigt, sondern muss im Rahmen der Interessenabwägung gegen diesen durchdringen²⁵⁶.

3.3.2.2. Zugänglichmachung im Internet

Die Problematik der Einräumung des Rechts zur öffentlichen Zugänglichmachung hat für Open Source Software große praktische Bedeutung, da diese meistens im Internet zum Download bereitgestellt wird und nur populärere Produkte über den Handel vertrieben werden. § 2 der OSD fordert daher von einer Open Source Software: „*The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost—preferably, downloading via the Internet without charge.*“ In Österreich regelt das Zuverfügungstellungsrecht der durch die UrhG-Nov 2003 geschaffene § 18a UrhG, der auch klärt, dass die Nutzung von

²⁵⁵ *Wiebe/Prändl*, ÖJZ 2004, 632, die unter dem Verweis auf OGH 5 Ob 504/96, JBI 1998, 577, davon ausgehen, dass die auferlegte Beschränkung eine Nebenbestimmung iSd § 879 Abs 3 ABGB darstellt; siehe auch *Spindler*, Rechtsfragen 49. Ausführlich zur Geltungskontrolle siehe Kap 4 2.

²⁵⁶ Zum gleichen Ergebnis kommt bspw auch *Spindler*, Rechtsfragen 50.

Werken und Leistungen in digitalen Netzen als öffentliche Wiedergabe und nicht als Vervielfältigung und Verbreitung besonderer Art gewertet werden muss²⁵⁷. Ausgenommen von dieser Regelung ist insbesondere die Übermittlung von Werken und Leistungen per E-Mail und durch „push-Dienste“. Diese finden in den Bestimmungen des § 18a UrhG keine Deckung, da hier jeweils die Initiative vom Versender ausgeht. In diesen Fällen ist daher auf die Bestimmungen des Vervielfältigungs- und Verbreitungsrechts zu verweisen.

In Deutschland ist die Problematik der Zugänglichmachung der Software im Internet kontrovers diskutiert worden. *Koch* verweist bspw darauf, dass die GPL die Zugänglichmachung im Internet gar nicht erfasst und deshalb ergänzungsbedürftig ist.²⁵⁸ *Jaeger/Metger*²⁵⁹ sind der Ansicht, dass sich dieses Verwertungsrecht aus dem Wortlaut des amerikanischen Lizenztextes ergibt, der von „distribute“²⁶⁰ spricht. *Spindler*²⁶¹ sieht dies wiederum nur bei einer extensiven Auslegung des Begriffs „distribute“ als gegeben an und weist darauf hin, dass wegen des Territorialitätsprinzips nicht nur von der US-amerikanischen Begrifflichkeit des Lizenztextes ausgegangen werden dürfe, zudem seien die nicht bekannten Nutzungsarten zum Zeitpunkt der Lizenzerteilung nicht vom Umfang der eingeräumten Nutzungsrechte erfasst²⁶². Die Rechtmäßigkeit der Zugänglichmachung im Internet hänge daher davon ab, ob die Verbreitung mittels Datennetze bereits zur Zeit der Formulierung der GPL als neue Nutzungsart bekannt war. *Omsels*²⁶³ bejaht diese Frage und weist darauf hin, dass die Zusammenarbeit der Programmierer, die die GPL nutzen, von Anfang an und somit auch zur Zeit der Formulierung der GPL über

²⁵⁷ *Walter*, UrhGNov 2003, 35.

²⁵⁸ *Koch*, CR 2000, 338.

²⁵⁹ *Jaeger/Metger*, Open Source Software 33.

²⁶⁰ Siehe auch *Jaeger/Schulz*, Gutachten zu ausgewählten rechtlichen Aspekten der Open Source Software (2005) 21.

²⁶¹ *Spindler*, Rechtsfragen der Open Source Software 44.

²⁶² Zur Problematik der Einräumung unbekannter Nutzungsarten im Zusammenhang mit § 40a UrhG siehe Kap 3.2.3.4.

²⁶³ *Omsels*, Open Source und das deutsche Vertrags- und Urheberrecht, in *Schertz/Omsels*, Festschrift *Hertin* 2000, 141 (158).

Datennetze stattgefunden hat. Darüber hinaus enthalte der Begriff „Distribution“ keine Restriktion auf die Verbreitung von Open Source Software nur mittels Datenträger.

Die angeführte Diskussion mag im Hinblick auf die GPL zwar sinnvoll erscheinen, führt auf Open Source Lizenzen allgemein bezogen jedoch nicht zum Ziel, wenn man bedenkt, dass Open Source Software nicht ausschließlich unter der GPL lizenziert wird. Andere Lizenzen wie die „Academic Free License v2.1“ sprechen in ihrem Lizenztext bspw von „*to distribute copies of the Original Work and Derivative Work to the public; [...] to perform the Original Work publicly and to display the Original Work publicly*“. Die „X11“ oder auch „MIT License“ wiederum räumt dem Lizenznehmer ua das Recht ein „*to publish [...] the Software*“. Hier zeigt sich, dass eine Zugänglichmachung im Internet kaum rechtliche Probleme aufwirft. Zwar wird die Verbreitung mittels Datennetzen nicht explizit erwähnt, aber gerade die weit gefassten Bestimmungen erlauben die Annahme, dass die jeweiligen Lizenzgeber dem Lizenznehmer keine Restriktionen auferlegen, sondern möglichst uneingeschränkte Möglichkeiten zur Verwertung und Verbreitung einräumen wollen. Des Weiteren ist es wichtig anzumerken, dass die Lizenzgeber und -nehmer der GPL es als „usus“ und „opinio iuris“ erachten, GPL-lizenzierte Software im Internet bereitzustellen und herunterzuladen.

3.3.2.3. Die Problematik der Vermietung

Der klassische Bestandvertrag iSd § 1090 ABGB iVm §§ 16a und 40c UrhG in Zusammenhang mit Open Source Software wirft Fragen auf, die bislang nur durch die Lehre beantwortet wurden²⁶⁴.

²⁶⁴ Staudegger, Rechtsfragen beim Erwerb von IT-Systemen in *Jahnel/Schramm/Staudegger, Informatikrecht*², 87.

*Koch*²⁶⁵ sieht ein Vermietrecht im Rahmen der GPL als nicht gegeben an, da dieses ein eigenständiges ausschließliches Verwertungsrecht darstelle, welches nicht durch die Einräumung eines Verbreitungsrechts gedeckt wäre und die Erschöpfung des Verbreitungsrechts nicht zur Erschöpfung des Vermietrechts führe. *Jaeger/Metzger*²⁶⁶ legen den in § 0 Abs 2 GPL benutzten Begriff „Distribution“ weit aus und kommen zu dem Ergebnis, dass ein Vermietrecht als Teil des Verbreitungsrechts mit umfasst sei, da es Zweck der Lizenz sei, Freie Software so effektiv wie möglich zu verbreiten. Wichtig zur Unterscheidung seien dabei die Lizenzgebühren für das Vermietrecht einerseits und das Mietentgelt andererseits. Während die Vermietung gegen Entgelt gestattet sei, sei es unzulässig für die Einräumung des Vermietrechts eine Lizenzgebühr zu verlangen. *Spindler*²⁶⁷ zieht beides in Betracht und kommt zu dem Ergebnis, dass ein Vermietrecht jedenfalls mit den Grundsätzen des deutschen Urheberrechts zur Auslegung von Nutzungsrechtseinräumungen nicht vereinbar sei.

Nach österreichischer Gesetzeslage ist eine Vermietung von Software grundsätzlich nicht untersagt²⁶⁸. Deshalb muss auf den jeweiligen Lizenzinhalt Rücksicht genommen werden, bei dem man in den Open Source Lizenzen zu teilweise unterschiedlichen Interpretationen kommt. In den meisten Open Source Lizenzen wird das Recht „to distribute“ eingeräumt ohne dieses näher zu definieren. Die MIT License²⁶⁹ bspw benutzt den Ausdruck „*to deal in the software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software*“. Dies deutet darauf hin, dass das Recht zur Vermietung gestattet wird.

²⁶⁵ *Koch*, CR 2000, 336.

²⁶⁶ *Jaeger/Metzger*, Open Source Software 33 f; siehe auch *Jaeger/Schulz*, Gutachten zu ausgewählten rechtlichen Aspekten der Open Source Software 22.

²⁶⁷ *Spindler*, Rechtsfragen der Open Source Software 45.

²⁶⁸ § 40c UrhG besagt, dass Werknutzungsrechte an Computerprogrammen, wenn mit dem Urheber nichts anderes vereinbart worden ist, ohne dessen Einwilligung auf einen anderen übertragen werden können.

²⁶⁹ Siehe Kap 5.2.5.

Wenn man die Vermietung als Dienstleistung betrachtet, muss man zu dem Ergebnis kommen, dass eine Vermietung gestattet ist, da hierbei der Supportgedanke im Vordergrund steht. In allen Fällen, in denen Geld für eine Vermietung eine versteckte Lizenzgebühr bedeutet, bspw durch Freigabe des Sourcecodes nur über einen Mietvertrag, muss das Recht zur Vermietung verneint werden.

3.3.2.4. Die Problematik des Application Service Providing und Software as a Service

Das Besondere am „Application Service Providing“ (ASP)²⁷⁰ sowie bei „Software as a Service“ (SaaS)²⁷¹ im Gegensatz zur Vermietung ist, dass der Nutzer keine Kopie der Software besitzt, sondern Software nutzt, die auf den Servern des Providers liegt. Eine Übergabe von körperlichen Werkstücken wie beim Mietvertrag liegt hier also nicht vor²⁷². Diese Besonderheit muss im Ergebnis jedoch nicht dazu führen, dass eine bestandrechtliche Qualifizierung ausgeschlossen wird²⁷³. ASP mit Hilfe von Open Source Software kann als Pacht betrachtet werden, da der Nutzer des ASP in diesem Fall die Software weiter wirtschaftlich verwerten darf und ihm dafür auch die Erträge zustehen²⁷⁴.

²⁷⁰ Hierbei werden Softwareanwendungen, sogenannte „Applications“, über Datennetze, meistens das Internet, dem Kunden zur Verfügung gestellt, ohne dass bei diesem eine Installation der Software lokal, dh auf dessen Rechnersystem, zu erfolgen hat. Die Software kann so zentral beim „Provider“ gewartet, angepasst und weiterentwickelt werden, und dem Kunden so zur Verfügung zu stehen.

²⁷¹ SaaS ist eine Weiterentwicklung des ASP und unterscheidet sich teilweise von diesem. ASP ist im Gegensatz zu SaaS nicht mandantenfähig, dies hat praktische Auswirkungen. Während bei ASP jedem Kunden eine eigene Plattform zur Verfügung steht (sogenannte 1:1-Beziehung), betreibt der Anbieter bei SaaS eine Plattform für alle Kunden (sogenannte 1:n-Beziehung).

²⁷² Jaeger/Metzger, Open Source Software 34.

²⁷³ *Fallenböck/Trappitsch*, Application Service Providing (ASP) – rechtlich betrachtet, MR 2002 3 (5).

²⁷⁴ Ergebnis e contrario aus *Fallenböck/Trappitsch*, MR 2002, 4.

Das Fruchtziehungs- und Fruchtgenussrecht iSd § 1091 ABGB²⁷⁵ ist damit gewährleistet, da die Open Source Lizenzen hier keine Restriktionen vorsehen.

Die Bereitstellung der Open Source Software an sich und die damit entsprechenden Handlungen können dem Recht der öffentlichen Zugänglichmachung, die Eingangs erörtert wurde, zugeordnet werden²⁷⁶.

Die Problematik²⁷⁷, dass unbekannte Nutzungsarten – wie bspw das ASP von älterer Open Source Software – nicht eingeräumt werden konnten, ist bislang nur in Deutschland ausreichend geklärt. In Deutschland verbietet § 31 Abs 4 dt UrhG die Einräumung von Nutzungsrechten für noch nicht bekannte Nutzungsarten. Diese Bestimmung ist dem österreichischen UrhG unbekannt. Die Problematik der Zweckübertragungstheorie wurde hier zu Lande von Lehre und Judikatur bislang mit teilweise unterschiedlichen Ergebnissen beantwortet²⁷⁸.

Der überwiegende Teil der Lehre spricht sich iSd „Trägheit des UrhG“ aus, indem er dafür plädiert, dass nach österreichischem UrhG nur Nutzungsrechte für bei Vertragsabschluss objektiv vorhersehbare Nutzungsarten eingeräumt werden können²⁷⁹ und verlangt darüber hinaus eine Klarstellung durch den Gesetzgeber²⁸⁰. Von anderer Seite hingegen wird die

²⁷⁵ *Binder* in *Schwimann* (Hrsg), Praxiskommentar ABGB VI² (2006) § 1091 Rz 1 ff.

²⁷⁶ *Jaeger/Schulz*, NOW Gutachten 22.

²⁷⁷ *Jaeger/Metzger*, Open Source Software 36; *Jaeger/Schulz*, NOW Gutachten 22.

²⁷⁸ Ausführlich dazu: *Küng*, MR 2004, 27 f.

²⁷⁹ *Ciresa*, Urheberrecht aktuell, 151 f.; *Haller*, Music on demand, Internet, Abrufdienste und Urheberrecht (2001), 116 f; *Walter* ua in Verbreitungsrecht und Öffentlichkeit, MR 1998, 344 (341) und *Walter*, Entwurf eines Gesetzes zur Verbesserung der vertraglichen Stellung von Urhebern und ausübenden Künstlern in Österreich, GRURInt 2001 602 (604, 607 f); *Reindl*, Die Nebenrechtseinräumung im Musikverlagsvertrag, ÖSGRUM 12/1993; *Thiele*, Verwendung von Mitarbeiterfotos auf Firmenwebsites, wbl 2002 (397) 399 f.

²⁸⁰ *Walter*, Die vier Säulen des Urheberrechtes, ZfRV 1999, 88 (94); *Küng*, MR 2004, 28.

Einräumung von Nutzungsrechten an unbekanntem Nutzungsarten für gültig²⁸¹ erachtet.

Der OGH hat 1993 ausgesprochen²⁸², dass urheberrechtliche Verträge hinsichtlich der Nutzungsrechte so auszulegen seien, dass „*das Ausmaß der Befugnisse, die der Werknutzungsberechtigte durch den Werknutzungsvertrag erhält [...] nicht weiter reicht, als es für den praktischen Zweck der ins Auge gefassten Werknutzung erforderlich erscheint*“. In jüngster Zeit hat sich der OGH dem überwiegenden Teil der Lehre angeschlossen²⁸³ und sich dabei sowohl auf die Lehre in Deutschland als auch in Österreich berufen. Befremdlich scheint hier die Tatsache, dass der OGH die Lehre in Deutschland zitiert, aber außer Acht zu lassen scheint, dass sich in § 31 Abs 4 dt UrhG eine entsprechende Regelung findet, welche man im österreichischen UrhG bislang vergeblich sucht. Des Weiteren vermeidet es der OGH, Stellung zur Zweckübertragungstheorie zu nehmen und zieht die Überlegungen von *Dittrich* bspw erst gar nicht in seine Erwägungen ein. Daher bleibt eine klare Antwort auf die Frage, ob eine Einräumung für noch nicht bekannte Nutzungsrechte möglich ist, vorerst offen.

3.3.2.5. Der Einfluss des GebG auf Open Source Lizenzverträge

Der VwGH hat am 7. September 2006²⁸⁴ eine Entscheidung gefällt, die weitreichende Konsequenzen für die Lizenzgeber von Software in Österreich hat. Demnach wurde der Lizenzgeber eines Computerprogramms dazu verurteilt, der Finanzbehörde eine Rechtsgeschäftsgebühr in Höhe von EUR 218.474,78 nach dem GebG zu bezahlen, nachdem er mit dem

²⁸¹ *Dittrich*, Gedanken zur so genannten Zweckübertragungstheorie, RfR 1979, 41 (44) und *Dittrich*, Noch einmal: Gedanken zur so genannten Zweckübertragungstheorie, RfR 1984, 1.

²⁸² OGH 08.06.1993, 4 Ob 53/93, MR 1993, 187 (*Walter*). Diese Meinung wurde vom OGH am 21.03.2000 in 4 Ob 77/00b MR 2001, 171 (*Walter*) bestätigt.

²⁸³ OGH 19.10.2004, 4 Ob 182/04z, *ecolex* 2005/370, 776.

²⁸⁴ VwGH 07.09.2006, 2006/16/0054 = MR 2006, 408.

Lizenznehmer eine Lizenz über die nicht-ausschließliche Nutzung der Software abgeschlossen hatte.

Die Entscheidung begründet sich auf dem Gebührengesetz von 1957 (GebG), welches für verschiedene Arten von Rechtsgeschäften, wie bspw Bestands- oder Darlehensverträge, einen bestimmten Prozentsatz als Rechtsgeschäftsgebühr vom Gesamtbetrag des Geschäfts vorsieht. Da das Gesamtvolumen des strittigen Lizenzvertrags relativ hoch war und die Behörde sowie der VwGH das vorliegende Geschäft nicht als Werknutzungsvertrag einstufen, ergab sich die eindrucksvolle Summe von EUR 218.474,78.

Knackpunkt der Entscheidung ist die Bewertung des Lizenzvertrags. Der VwGH verneint das Vorliegen eines Werknutzungsvertrags mit der Begründung, in der Beschwerdevorbringung sei kein Anhaltspunkt dafür zu entnehmen, *„dass das dem Kunden des Vertrages eingeräumte Nutzungsrecht am Computerprogramm über das hinausgeht, was diesem Kunden nicht ohnehin schon nach § 40d UrhG ex lege eingeräumt“* worden sei. Weiter heißt es: *„Da der vorliegende Vertrag dem Kunden keine Nutzung des Computerprogramms in einem solchen Umfang einräumt, die einem Verwertungsrecht nach den §§ 14ff UrhG gleichzuhalten wäre, liegt schon deshalb kein Werknutzungsvertrag im Sinn des § 33 TP 5 Abs. 4 Z. 2 GebG vor.“*

Dieses Ergebnis des VwGH bedeutete noch zum Jahresbeginn 2007 für die meisten Softwarelizenzverträge eine ernsthafte Gefahr. Hielt sich der Lizenzgeber in seiner Rechteinräumung nur an das UrhG, musste er damit rechnen, Gebühren nach dem GebG an die Finanzbehörde abführen zu müssen. Dies war für Lizenzpartner in der IT-Branche eine absolut unbefriedigende Lösung, da es zu Hindernissen bei Vertragsabschluss führte. Der Lizenzgeber musste sich Gedanken machen, wie er den Lizenznehmer besser als das Gesetz stellen könne, um keine weiteren Kosten zu verursachen. Eine mögliche Lösung konnte hier nur eine Nachbesserung durch den Gesetzgeber bringen, welche schließlich durch die Änderungen im BudgetbegleitG 2007 beschlossen wurde und für alle Sachverhalte gilt, für die

die Gebührenpflicht gem § 37 Abs 19 GebG nach dem 31.5.2007 eingetreten ist. Gem neu beschlossenen § 33 Abs 4 Z 2 sind „*Werknutzungs-, Patent-, Marken- und Musterlizenzverträge sowie Verträge über die Nutzung von Software*“ von der Gebührenpflicht befreit.

Bis dahin konnte den Betroffenen nur geraten werden, darauf zu achten, bei Abschluss des Lizenzvertrags weiterführende Rechte einzuräumen. Dies konnte bspw durch das Einräumen eines Bearbeitungsrechts über die persönliche Nutzung hinaus oder das Recht zur Vervielfältigung außerhalb der persönlichen Nutzung geschehen. Ebenso sollte auf die genaue Aufschlüsselung der Kosten Wert gelegt werden, um so die Bemessungsgrundlage einer möglichen Vergebüßung so niedrig wie möglich zu halten.

Im Ergebnis muss festgehalten werden, dass der Wirbel um die Gebührenpflicht die Dienstleister von Open Source Software nicht betroffen hat, da diese in den meisten Fällen dem Lizenznehmer mehr Rechte einräumen, als er ex lege erhält. Open Source Lizenzpartner mussten in dieser Phase also wenig Bedenken haben, dass sie ihre Lizenzen nach dem GebG vergebühren mussten und konnten darauf vertrauen, dass Open Source Lizenzverträge von der Anwendung des GebG ausgenommen waren.

Dieses in der rechtlichen Betrachtung ungleiche Ergebnis demonstriert ein weiteres Mal, dass Open Source Software und ihre Lizenzen eine Sonderstellung im Bereich des Softwarevertragsrechts innehaben.

3.4. Ergebnis

Das Urheberrecht bietet dem Entwickler von Software den bislang größten Schutz im Rahmen der Immaterialgüterrechte und stellt die „*nötige Vorbedingung für die Wirksamkeit*“²⁸⁵ der Open Source Lizenzen. Da das Schöpferprinzip iSd § 10 UrhG gesetzlich ausschließt, dass ein Werk in die „public domain“²⁸⁶ gelangt, kommt dem Urheber als Rechteinhaber im Open Source Bereich eine zentrale Rolle zu. Die in vielen Open Source Lizenzen angefügten Bestimmungen, die das Werk vor der „public domain“ schützen wollen, sind damit für Österreich unerheblich.

Bei Open Source Software ist die Feststellung der Urheberschaft nicht einfach, teilweise sogar unmöglich, da viele Personen an einem Projekt über das Internet mitarbeiten.²⁸⁷ Trotzdem soll Rechteinhabern die Möglichkeit gegeben werden, ihre Urheberrechte effektiv durchsetzen zu können. Im Fall der Teil- und Miturheberschaft gibt es fließende Übergänge, die jeweils im Einzelfall zu beurteilen sind. Die Beurteilung, ab wann eine eigenständige Weiterentwicklung eines Programms vorliegt, wird dadurch erschwert, dass klare Abgrenzungskriterien fehlen. Wichtig zu beachten bei Open Source Software ist die Möglichkeit, die Urheberschaft am Quelltext und am Objektformat getrennt zu beurteilen, da diese nicht zwingend übereinstimmen muss. Das Kriterium der Eigenständigkeit führt bspw in vielen Fällen dazu, dass Teilurheber am Quelltext als Miturheber am Objektformat angesehen werden müssen.

Mit- oder Teilurheberschaft liegt bei den überwiegenden Teilen von Open Source Software vor, weshalb die gesetzlichen Bestimmungen hierzu von besonderer Bedeutung sind. Diese führen jedoch hinsichtlich des Entstehens

²⁸⁵ Spindler, Rechtsfragen 22.

²⁸⁶ Vgl Kap 2.4.2.

²⁸⁷ Vgl auch Küng, MR 2004, 23.

einer GesbR iSd § 1175 ABGB in Bezug auf Open Source Projekte zu teils ungewünschten Ergebnissen. Dieser Bereich bedarf entweder einer größeren Aufklärung in der Softwaregemeinschaft, da sich diese meist nur unzureichend über die rechtlichen Auswirkungen informiert oder einer effektiveren rechtlichen Beurteilung durch den Gesetzgeber. Denkbar wäre bspw die Entstehung einer GesbR im Softwarebereich mit übernationalem Bezug vom tatsächlichen Willen der Beteiligten abhängig zu machen.

Beim Arbeitnehmerurheberrechts ist es wichtig, auf die Art des Dienstverhältnisses einzugehen, da die Bestimmungen des § 40b UrhG nur auf Dienstnehmer anzuwenden sind. Bei allen anderen Dienstverhältnissen kann dispositives Recht zur Anwendung kommen. In der Praxis wird darauf zu achten sein, wem das Recht zukommt, die Werknutzungsrechte zu regeln und damit die Entscheidung zu treffen, ob eine neu entwickelte Software unter eine Open Source Lizenz gestellt wird oder Closed Source werden kann.

Der zivilrechtliche Schutz der §§ 81-90d UrhG ist für Urheber von Open Source Software nur begrenzt ausreichend. Wie bereits erwähnt kommt es vor allem beim Vorliegen von Mit- oder Teilurheberschaften zu tatsächlichen Problemen, die sich teilweise aus dem Zivilprozessrecht begründen. Während es bei Unterlassungs- und Beseitigungsbegehren keine Probleme bei der Durchsetzung geben sollte, da diese Ansprüche gegebenenfalls auch gegenüber dem Host-Provider oder einem Linksetzer bestehen, kann es im Fall der Leistungsklagen ieS schwierig werden, wenn die Anzahl an Mit- oder Teilurhebern an einer Open Source Software einmal eine große Menge oder einen bestimmten Grad an Internationalität erreicht hat. Hier erscheint die Durchsetzung eines Anspruchs durch eine Streitgenossenschaft iSd § 11 ZPO am sinnvollsten, da sich Miturheber auch noch während des Gerichtsverfahrens anschließen können. Problematisch erscheint dabei die zu erwartende physische Abwesenheit von Miturhebern, die nicht in Österreich ihren gewöhnlichen Aufenthalt haben. Hier sollte die Möglichkeit bestehen, einen Streitgenossen, bzw den Kläger als nationalen Stellvertreter des Projekts

anzuerkennen und so dem Schuldner die Möglichkeit zu eröffnen, durch Leistung an diesen frei zu werden. Der Stellvertreter der Gesamthandgemeinschaft hat in Folge für die rechtmäßige Verteilung des Geleisteten zu sorgen, widrigenfalls er sich selbst zivilrechtlichen Ansprüchen der anderen Gesamthandigentümer ausgesetzt sehen muss.

Die Open Source Lizenzen leiten ihre Existenz aus dem Urhebervertragsrecht ab, welches die Rahmenbedingungen für die Lizenzen aufstellt. Der Intention der Open Source Philosophie folgend erhalten Lizenzgeber die Möglichkeit, die ihnen vorbehaltenen Schutzrechte Dritten einzuräumen²⁸⁸ und dadurch die freie Verfügbarkeit der Software bzw des Quelltextes sicherzustellen. Durch die Formfreiheit iSd § 883 ABGB bestehen individuelle Gestaltungsmöglichkeiten für den Inhalt der Lizenz, was auch durch die große Anzahl an Open Source Lizenzen belegt wird. Wichtig zu erwähnen ist dabei, dass die Open Source Lizenzgeber den Lizenznehmern nur eine Werknutzungsbewilligung iSd § 24 Abs 1 Satz 1 UrhG einräumen, um sich nicht selbst von der Verwertung des Programms auszuschließen.

Interessant sind der Schutzrechtsumfang und dessen Wirkungen. Die Problematik des Erschöpfungsgrundsatzes tritt vor allem beim Download von Open Source Software ein, da es hier zu einer extremen Ungleichbehandlung kommt. Erschöpfung tritt hier nur bei der heruntergeladenen Erstkopie ein, das Verbreitungsrecht bei jeder weiteren Kopie der Erstkopie wird mangels „Zustimmung des Berechtigten“ nicht erschöpft. Bei der Eigentumsübertragung ohne Download unterliegt zwar der Zweiterwerber keiner urheberrechtlich-dinglichen Beschränkung, die schuldrechtlichen Bestimmungen setzen sich aber im Rahmen der Interessenabwägung mit Rücksichtnahme auf die eigenständige Art der Open Source Software gegen diese durch.

²⁸⁸ Zur Stellung der Vertragspartner siehe Kap 4.

Die Zugänglichmachung im Internet iSd § 18a UrhG betreffend, muss festgestellt werden, dass ein Blick in die Open Source Lizenzen und die OSD weiterhilft, wobei man feststellen muss, dass die Lizenzen die Zugänglichmachung von Open Source Software mittels Download nicht explizit erwähnen, der Telos der Lizenzen jedoch eine restriktionslose Verbreitung der Open Source Software gewährleisten will. § 2 OSD fordert explizit die Bereitstellung von Open Source Software im Internet, weshalb diese Bestimmung generelle Geltung für alle Open Source Lizenzen hat. Würde der Download von Open Source Software nicht in Einklang mit § 18a UrhG stehen, würde dies außerdem einen schwerwiegenden, wenn nicht vernichtenden Eingriff in die Open Source Philosophie darstellen. Für die Vermietung von Open Source Software bleibt zu erwähnen, dass das Recht zur Vermietung dann zu verneinen ist, wenn die Miete als versteckte Lizenzgebühr eingesetzt wird oder die Freigabe des Quelltextes davon abhängt.

Auch wenn eine abschließende Antwort, ob ASP mit Open Source Software gestattet ist, offen gelassen werden muss, ist mE den Überlegungen *Dittrichs* zu folgen, nach denen die Einräumung von Nutzungsrechten an unbekanntem Nutzungsarten unter Berücksichtigung der Open Source Philosophie gültig ist. Dem Lizenzgeber steht es frei, die Einräumung von Nutzungsrechten an unbekanntem Nutzungsarten von vorneherein auszuschließen. Hierbei muss in Betracht gezogen werden, dass gerade im technologischen Bereich von Lizenzgebern das Wissen vorausgesetzt werden kann, dass für Software stets neue Nutzungsarten entwickelt werden und daher schon im Lizenzvertrag die Einräumung von Nutzungsrechten für derzeit unbekanntem Nutzungsarten geregelt werden sollte.

Die in diesem Kapitel angesprochenen allgemeinen Probleme werden in Kapitel 5 anhand der Bestimmungen einzelner Lizenztexte einer näheren Prüfung unterzogen.

4. DIE ANNAHME DES LIZENZVERTRAGS

Die Lizenzgeber von Open Source Lizenzen wenden sich an einen unbestimmten internationalen Kreis natürlicher und juristischer Personen, ihre unter einer bestimmten Open Source Lizenz stehende Software zu benutzen, zu vervielfältigen, zu modifizieren und zu verbreiten. Damit einem Nutzer jedoch diese Rechte eingeräumt werden, muss er die Lizenzbedingungen der jeweiligen Open Source Lizenz akzeptieren, indem er einen Vertrag mit dem Lizenzgeber abschließt. Der Abschluss eines solchen Lizenzvertrages wirkt in rechtlicher Hinsicht etliche Fragen auf, die in Folge dargestellt werden.

4.1. Die Anwendbarkeit von österreichischem Recht auf Open Source Lizenzverträge

Das EVÜ, welches in Österreich seit dem 1.12.1998 in Kraft ist, regelt in Art 3, dass ein Vertrag dem von den Parteien gewählten Recht unterliegt. Diese Vereinbarung iSd Art 3 Abs 1 S 2 EVÜ muss ausdrücklich oder schlüssig vorliegen²⁸⁹. Eine ausdrückliche Rechtswahl liegt bspw bei einer „Rechtswahlklausel“ in einem Vertrag vor, wenn die Parteien eine unmittelbar auf die Bestimmungen des anzuwendenden Rechts gerichtete Willenserklärung abgegeben haben, während es bei einer schlüssigen Rechtswahl eines auf die Rechtswahl gerichteten realen Parteiwillens bedarf.²⁹⁰

²⁸⁹ Heiss in Czernich/Heiss (Hrsg), EVÜ – Das Europäische Schuldvertragsübereinkommen: Übereinkommen über das auf vertragliche Schuldverhältnisse anzuwendende Recht: Kommentar (1999) Art 3 Rz 5 ff.

²⁹⁰ von Hoffmann in Soergel, Kommentar zum Bürgerlichen Gesetzbuch X Art 27 EGBGB Rz 41.

Eine schlüssige Rechtswahl kann bei Open Source Lizenzen in den überwiegenden Fällen nicht angenommen werden²⁹¹. Es scheidet sowohl eine hypothetische Rechtswahl als auch das österreichische Spezifikum der Geltungsannahme aus²⁹², da für diese eine Vermutung der bestimmten und als maßgebend angenommenen Rechtsordnung nicht ausreicht²⁹³. Die Schlüssigkeit der Rechtswahl fällt daher schon deshalb weg, da die Vertragspartner bei Open Source Software in den wenigstens Fällen explizite Kenntnis von einander haben.

Für diese Open Source Lizenzen gilt daher die objektive Anknüpfung des Art 4 EVÜ, wonach ein Vertrag jenem Recht unterliegt, mit dem er die engste Verbindung im Zeitpunkt des Vertragsabschlusses²⁹⁴ aufweist. Die Faktoren für die engste Verbindung iSd Art 4 EVÜ sind dabei:

- der gewöhnliche Aufenthalt des Lizenznehmers,
- die Staatsangehörigkeit des Lizenznehmers,
- die Währung mit der das Rechtsgeschäft bezahlt wird (schwacher bis gar keine Indizwirkung bei Open Source Software) und
- der Abschlussort des Lizenzvertrages (mE gar keine Indizwirkung, da er im Open Source Bereich durch die Interaktionen der Vertragspartner über das Internet zufällig ist).

²⁹¹ Für eine genaue Übersicht siehe Anhang A.

²⁹² § 35 Abs 1 IPRG aF; *Schwimann* in *Rummel*, Kommentar zum Allgemeinen bürgerlichen Gesetzbuch II² § 35 IPRG Rz 6.

²⁹³ *Schwimann* in *Rummel*, Kommentar II² § 35 IPRG Rz 6a.

²⁹⁴ *Czernich* in *Czernich/Heiss* (Hrsg), EVÜ Art 4 Rz 8; *Posch*, Internationales Privatrecht³ Rz 15/11.

Für jene Open Source Lizenzen, die eine Rechtswahl in ihrem Lizenztext verankert haben, wie bspw die „Common Public License“, die „Eclipse Public License“, die „IBM Public License“ oder die „Lucent Public License“, die New York als Gerichtstand festlegen und auf das amerikanische Urheberrecht verweisen, muss auf Art 5 EVÜ Rücksicht genommen werden. Ist der Lizenznehmer ein Verbraucher iSd Art 5 Abs 1 EVÜ, gilt für ihn die in einer Open Source Lizenz ausgewählte Rechtswahl nur bedingt. Der Lizenznehmer darf durch die Rechtswahl nicht den Schutz verlieren, der ihm durch die zwingenden gesetzlichen österreichischen Bestimmungen – bzw durch das Recht jenes Staates, in dem er seinen gewöhnlichen Aufenthalt hat – gewährt wird. Hier ist im Einzelfall ein Günstigkeitsvergleich im Hinblick auf das Ergebnis durchzuführen.²⁹⁵

4.2. Die Annahme des Lizenzvertrags

Die Wirksamkeit des Lizenzinhaltes ist von einer gültigen übereinstimmenden Willenserklärung iSd § 861 ABGB abhängig. Dieser Teil der Arbeit beschreibt vorrangig verschiedene Abschlussvarianten von Open Source Lizenzen, die damit verbundenen Grundsätze und die Einordnung der AGB bei einem Vertragsabschluss im Internet. Dabei wird auf die Problematik hingewiesen, dass die von der OSI approbierten Lizenzen in Englisch verfasst sind und Übersetzungen von den Lizenzgebern meist abgelehnt werden. Die mit einem Vertragsabschluss verbundenen Haftungs- und Gewährleistungsfragen, werden eigenständig in Kapitel 6 getrennt behandelt.

²⁹⁵ Heiss in Czernich/Heiss (Hrsg), EVÜ Art 5 Rz 59 f.

4.2.1. Abschlussvarianten

Vorrangig werden Open Source Lizenzen über das Internet abgeschlossen, aber auch im Einzelhandel kann man immer mehr Open Source Software, wie bspw Linux-Distributionen, erwerben. Darüber hinaus kann man Open Source Software bspw noch als OEM oder Embedded System erhalten. Aus diesem Grund werden diese drei Varianten besprochen. Andere Fallkonstruktionen werden hier außer Acht gelassen. Davor wird ein kurzer Überblick über die Akteure gegeben, die beim Umgang mit Open Source Software auftreten.

4.2.2. Überblick über die Notwendigkeit zum Abschluss eines Open Source Lizenzvertrags

Nachfolgende Grafik gibt einen Überblick über mögliche Lizenzpartner und verdeutlicht, dass nicht jeder der Open Source Software einsetzt einen Lizenzvertrag zwingend abschließen muss:



Abb 6

Die drei wichtigsten Varianten zeigen, dass ein reiner Nutzer von Open Source Software einen Lizenzvertrag nicht zwingend abschließen muss. Eine Zustimmung zum Lizenzvertrag ist ihm deshalb aber nicht verwehrt. Sollte ein Nutzer das Interesse haben oder die Notwendigkeit empfinden, die Open Source Software weiterzuentwickeln, muss er zum Lizenznehmer werden, damit er die Rechte aus dem Lizenzvertrag erhält, desgleichen jemand, der die Open Source Software ausschließlich weiterentwickelt, ohne sie selbst zu nutzen. Dies wird im weiteren Verlauf der Arbeit immer wieder verdeutlicht.

4.2.3. Download und Kaufvertrag

Beim unkörperlichen Vertrieb von Open Source Software über Datennetze, wie bspw das Internet, wird der Vertrag zwischen Lizenzgeber und -nehmer meist asynchron²⁹⁶ abgeschlossen, da in den wenigsten Fällen beide Vertragspartner gleichzeitig zum Vertragsabschluss anwesend sind. Aus diesem Grund müssen solche Verträge auch als Verträge unter Abwesenden angesehen werden²⁹⁷.

Im Onlinebereich ist der Vertragsabschluss per E-Mail durch das Ausfüllen eines Webformulars oder durch Zustimmung zum Lizenztext bei der Installation oder erstmaligen Benutzung der Software besonders zu erwähnen. Letztere Variante kann zwar nicht alleine dem Onlinebereich zugerechnet werden – hier muss auf alle möglichen Arten von Vertriebs- und Verbreitungswegen verwiesen werden – dennoch ist sie in der Praxis wohl am häufigsten im Onlinebereich zu finden, da Open Source Software klassischer Weise über das Internet heruntergeladen wird. Diese Variante hat für Open Source Software die größte Bedeutung, da es für die Software selbst ausreicht, den verpflichtend beigefügten Lizenztext anzunehmen. Ein Vertragsabschluss per E-Mail oder Webformular kann daher nur für ein schuldrechtliches Verpflichtungsgeschäft – bspw Kauf der Softwaredistribution über einen Webshop – relevant sein, nicht aber für die Einräumung der Nutzungsrechte, welche vom „Grundgeschäft“ getrennt behandelt gehören²⁹⁸.

Daraus folgt, dass eine Annahme der Open Source Lizenz nur im Falle einer Weiterverbreitung, einer Vervielfältigung außerhalb des privaten Bereichs oder bei einer Bearbeitung notwendig ist. Eine reine Benutzung von Open

²⁹⁶ *Tangl*, Leitfaden für die Einbeziehung elektronischer AGB, *ecolex* 2001, 896.

²⁹⁷ *Mottl*, Vertragsrechtliche Rahmenbedingungen für den Electronic-Commerce in *Jahnel/Schramm/Staudegger* (Hrsg), *Informatikrecht*², 34 f.

²⁹⁸ *Wiebel/Prändl*, *ÖJZ* 2004, 631 halten diese Trennung für „vertretbar“.

Source Software ist im Rahmen des § 40d UrhG ohne Vertragsabschluss möglich²⁹⁹. Dabei stellt sich die Frage, wann von einer Einräumung der Nutzungsrechte und einem Vertragsabschluss in Betracht auf die zeitliche Reihenfolge ausgegangen werden kann. Der Ansicht, dass die Nutzungsrechte bereits bei „Ingangsetzung“ der Software eingeräumt werden³⁰⁰, steht jene entgegen, dass die Übertragung der Nutzungsrechte dem Erwerb der Software zeitlich nachfolgt und daher auch nicht als einheitlicher Vertragsgegenstand anzusehen ist³⁰¹. Letztere Ansicht stützt sich auf § 5 der GPL, welcher besagt, dass erst durch eine Veränderung oder Verbreitung der Software ein Einverständnis mit der Lizenz notwendig ist und hier als Beispiel stellvertretend für alle Open Source Lizenzen stehen soll.

Die Wirksamkeit des § 5 GPL muss im Falle eines Rechtsgeschäftes – bspw Kaufvertrag – zwischen Unternehmer und Verbraucher iSd KSchG verneint werden³⁰², da § 6 Abs 1 Z 2 KSchG die dort verankerte Fiktion der Vertragsannahme nicht zulässt. *Wiebe* und *Prändl* schlagen zur Lösung vor, dass der Verbraucher in Folge einen neuen Vertrag abschließt, in welchem ihm durch die GPL über die gesetzlichen Mindestrechte des § 40d UrhG hinaus Rechte eingeräumt werden.

Die Tendenz einer nachfolgenden Rechtseinräumung zeigt sich mE auch in der Präambel der GPL, die mit den Worten „*The precise terms and conditions for copying, distribution and modification follow*“ endet. § 0 der GPL spricht des Weiteren davon „*Activities other than copying, distribution and modification are not covered by this license; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).*“ Die Übertragung der Nutzungsrechte ist daher

²⁹⁹ So auch in Anwendung des dt UrhG *Spindler*, Rechtsfragen der Open Source Software 72.

³⁰⁰ *Spindler*, Rechtsfragen der Open Source Software 72.

³⁰¹ *Jaeger/Metzger*, Open Source Software 138.

³⁰² *Wiebe/Prändl*, ÖJZ 2004, 631.

unabhängig von der Nutzung zu sehen, die wie erwähnt ex lege bewilligt wird³⁰³.

Andere Open Source Lizenzen beantworten die angesprochene Problematik in ihren jeweiligen Lizenztexten teilweise weniger deutlich als die GPL. Allerdings darf davon ausgegangen werden, dass auch dort die Einräumung eines einfachen und unbeschränkten Nutzungsrechts zu einem späteren Zeitpunkt erfolgt und nicht schon bei erstmaliger Nutzung bzw. Installation.

4.2.4. OEM und Embedded-Systems

Mittlerweile haben sich Open Source Lösungen für den Einsatz in elektronischen Geräten für eine breite Anwendermasse etabliert. So wird Linux inzwischen häufig von Computerherstellern auf deren Hardware vorinstalliert und somit ein einsatzfähiges Betriebssystem gleich mitgeliefert. Bei anderen Anwendungen, wie bspw Mobiltelefonen und PDAs, werden so genannte Embedded Systems genutzt, die vielfach aus dem Open Source Bereich stammen.

Da § 6 Abs 1 Z 2 KSchG bei Verbrauchergeschäften, wie vorher erwähnt, die in § 5 GPL verankerte Fiktion der Vertragsannahme nicht zulässt, kann eine Wirksamkeit des Vertrags nicht bereits automatisch beim Kauf erfolgen. Sollte der Nutzer weitere - über die gesetzlichen Mindestrechte des § 40d UrhG hinausgehende - Rechte benötigen, muss er einen neuen Vertrag mit den Lizenzgebern abschließen.

³⁰³ So auch *Jaeger/Schulz*, NOW Gutachten 20.

4.3. AGB

Während die bereits erwähnten Ausführungen den wirksamen Vertragsabschluss behandelt haben, beziehen sich die folgenden Überlegungen auf die inhaltlichen, in den Open Source Lizenzen enthaltenen Bedingungen. Diese Bedingungen der einzelnen Open Source Lizenzen sind allgemeine Geschäftsbedingungen³⁰⁴ (AGB), die der Geltungskontrolle der §§ 864a und 879 Abs 3 ABGB unterliegen sowie im Falle eines Verbrauchergeschäftes nicht gegen die Bestimmungen des § 6 Abs 1, 2 und 3 KSchG verstoßen dürfen.

Die Geltung der jeweiligen AGB-Bestimmungen in den jeweiligen Open Source Lizenzen ist wiederum davon abhängig, ob der Lizenznehmer vor Abgabe seiner Vertragserklärung erkennen kann, dass der Lizenzgeber diese zum Lizenzbestandteil machen will³⁰⁵. Dabei kommt es nicht darauf an, ob der Lizenzgeber die AGB auf seiner Webseite zum Herunterladen und Speichern bereitgestellt hat. Dies ist nur von verwaltungsstrafrechtlicher Relevanz iSd § 11 ECG iVm § 26 ECG³⁰⁶. Trotzdem wird meist auf jeder Webseite, auf der ein Open Source Programm heruntergeladen werden kann, die dazugehörige Lizenz zum Abrufen bereitgestellt.

³⁰⁴ *Wiebe/Prändl*, ÖJZ 2004, 634. Für die Einrodung als AGB im dt Recht auch *Sujecki*, Open Source Software im deutschen Vertrags- und Urheberrecht, MR 2005, 40 (41 f); *Koch*, CR 2000, 339; *Omsels*, Festschrift *Hertin* 2000, 147; *Jaeger/Metzger*, GRUR Int 1999, 846, *Deike*, CR 2003, 13.

³⁰⁵ *Zankl*, Online-AGB: Erste Entscheidung zum E-Commerce-Gesetz, *ecolex* 2003, 669 (670); *Staudegger* in *Jahnel/Schramm/Staudegger*, Rechtsfragen beim Erwerb von IT-Systemen, 54.

³⁰⁶ *Zankl*, *ecolex* 2003, 670. AA *Mottl* in *Jahnel/Schramm/Staudegger* (Hrsg), Informatikrecht² 42.

Verstöße der Open Source Lizenzen gegen die Geltungskontrolle werden nachfolgend, vor allem im Kapitel über die Haftungs- und Gewährleistungsausschlüsse, untersucht und daher an dieser Stelle nicht weiter behandelt.

4.4. Lizenzsprache

Alle von der OSI approbierten Lizenzen liegen ausschließlich in englischer Sprache vor. Die meisten darüber hinaus existierenden Lizenztexte sind ebenfalls in englischer Sprache abgefasst und es gibt nur wenige Lizenzen mit von ihren Urhebern anerkannten Übersetzungen. Diese sind häufig nicht an einer Übersetzung interessiert, da das für sie mit erheblichem Mehraufwand verbunden ist. Die Lizenztexte müssten auf eine korrekte Übersetzung hin überprüft werden, was dem Lizenzurheber selbst oft nicht möglich ist und er in Folge Dritte damit beauftragen müsste, deren Ergebnis dann aber trotzdem noch von seinen Intentionen abweichen könnte. Der Lizenznehmer muss daher mit dem englischen Lizenztext kontrahieren.

Der OGH hat die Problematik der Lizenzsprache in jüngster Zeit dahingehend präzisiert³⁰⁷, dass die Abfassung von AGB "*in einer anderen als der Vertragssprache (oder der Sprache des Adressaten) eine Einbeziehung in das Vertragsverhältnis*" jedenfalls nicht hindere. Die Lösung der Frage, "*ob AGB als Bestandteil des Angebotes des erklärenden Vertragspartners anzusehen*" sei, hänge vielmehr davon ab, "*ob ihre gewollte Einbeziehung für den Adressaten erkennbar und ihm auch zumutbar*" sei. Insofern kann auch die Sprache, in der auf AGB verwiesen wird und in der diese abgefasst sind, Bedeutung erlangen. Ob dem Adressaten des Vertragsangebots das Verständnis übermittelter AGB, die in einer anderen als seiner Muttersprache

³⁰⁷ OGH 17.12.2003, 7 Ob 275/03x, JBI 2004, 449 = ZfRV 2004, 20 = IHR 2004, 148 = RdW 2004, 252.

oder der Vertragssprache oder einer ihm sonst geläufigen Sprache abgefasst sind - etwa durch Herstellung einer Übersetzung - zugemutet werden kann, ist deshalb nach Länge, Intensität und Bedeutung von Geschäftsbeziehungen sowie nach der Verbreitung der verwendeten Sprache im betreffenden Kulturkreis zu beurteilen. Es genügt hierbei die Übersendung von AGB in einer Weltsprache oder in einer Sprache, die der Vertragspartner beherrscht.

Für Open Source Lizenzen muss vorrangig auf die Verbreitung der Sprache im jeweiligen Kulturkreis verwiesen werden. Vor allem beim Herunterladen der Open Source Software von einer Webseite werden meist unbestritten Englischkenntnisse beim Nutzer erwartet³⁰⁸. Beim Bezug der Open Source Software über einen deutschsprachigen Distributor bspw stellt die Beifügung des englischsprachigen Lizenztextes entsprechend der vorher erwähnten OGH-Entscheidung kein Problem dar, da Englisch als Weltsprache anerkannt ist³⁰⁹.

Letztendlich bleibt noch der Hinweis auf die Konsequenz einer Verweigerung der Annahme einer Lizenz aufgrund des Lizenztextes. In diesem Fall ist selbstverständlich eine Modifikation oder Weiterverbreitung der Software nicht gestattet. Gibt der Nutzer die Software weiter oder modifiziert er sie, muss davon ausgegangen werden, dass er sich auf den englischen Lizenztext eingelassen hat³¹⁰.

³⁰⁸ So auch *Jaeger/Metzger*, Open Source Software 149 mwA.

³⁰⁹ Für D siehe auch: *Spindler*, Rechtsfragen der Open Source Software 38; *Jaeger/Koglin/Kreutzer/Metger/Schulz*, Die GPL kommentiert und erklärt (2005) 100 f.

³¹⁰ *Jaeger/Metzger*, Open Source Software 149; *Spindler*, Rechtsfragen der Open Source Software 38.

4.5. Erwerb der Lizenzrechte von einem unberechtigten Dritten

Der Erwerb der Lizenzrechte durch den Lizenznehmer kann nur dann gültig erfolgen, wenn der Lizenzgeber Inhaber aller Rechte ist, die er dem Lizenznehmer einräumen will oder er die Befugnis dazu vom Urheber hat (bspw bei Entwicklungen durch einen Auftragsgeber). Sollte dies nicht der Fall sein, so kann es nicht zum Erwerb durch den Lizenznehmer kommen, da nach österreichischem Recht ein gutgläubiger Erwerb von Urheberrechten ausgeschlossen ist³¹¹.

Diese Tatsache wird im Open Source Bereich dadurch umgangen, dass die in einer Lizenz eingeräumten Rechte nicht jeweils von Lizenznehmer zu Lizenznehmer übertragen, sondern direkt vom Rechtsinhaber erworben werden³¹². Dieses Konstrukt ist besonders deutlich in Ziffer 6 GPL beschrieben und wird deshalb auch in diesem Zusammenhang erst näher besprochen³¹³. Die Problematik des Erwerbs der Lizenzrechte von einem unberechtigten Dritten ist im Open Source Bereich sehr brisant, da es oft zur Weitergabe über viele Teilnehmer kommt und Fehler in dieser „Lizenzkette“ weit reichende Folgen haben, die schließlich zu Rechtsunsicherheiten bei den Lizenznehmern führen.

³¹¹ *Ertl*, Gutgläubiger Erwerb von Softwarepiraten – Zugleich ein Beitrag zum Gutgläubenserwerb an Forderungen, MR 1997, 314; *Walter*, Gutgläubiger Erwerb urheberrechtlicher Nutzungsrechte nach österreichischem Recht? in *Loewenheim* (Hrsg), Urheberrecht im Informationszeitalter: Festschrift für *Wilhelm Nordemann* zum 70. Geburtstag am 8. Januar 2004 (2004) 625; OGH 29.04.2003, 4 Ob 57/03s – "Die Puppenfee" – MR 2003, 239 = RdW 2003/617, 700 (LS) = *ecolex* 2004, 45 (*Schumacher*); OGH 24.05.2005, 4 Ob 63/05a, MR 2005, 252 (*Walter*).

³¹² *Jaeger/Schulz*, NOW Gutachten 10.

³¹³ Siehe Kap 5.4.1.2.

4.6. Ergebnis

Der Abschluss eines Open Source Lizenzvertrags weist bereits auf sehr spezifische Problemstellungen hin. Schon die Art des Vertragsabschlusses birgt Diskussionsstoff, da die Übertragung von Nutzungsrechten und –bewilligungen über das Internet erst eine kurze Tradition hat und eine fundierte rechtlich definierte Einordnung noch gefunden werden muss. Die zeitliche Differenz zwischen Abschluss des schuldrechtlichen Verpflichtungsgeschäfts und der Einräumung der Nutzungsrechte ist ein Merkmal, das ebenfalls die außergewöhnliche Stellung von Open Source Software zeigt. Die zeitliche Differenz lässt sich deshalb rechtfertigen, da sie kein zwingendes Indiz für die Nutzung von Open Source Software ist. Auch wenn Open Source Lizenzen in ihren Lizenztexten das Recht „to use“ einräumen, muss dies mehr als Hinweis, denn als rechtliche Notwendigkeit gesehen werden, da § 40d UrhG bereits ein gesetzlich gewährtes Nutzungsrecht einräumt. Der Lizenznehmer, der im Normalfall die Software nur nutzt, braucht deshalb keinen Lizenzvertrag abzuschließen, er ist reiner Erwerber der Software. Dies wird erst dann notwendig, wenn er Nutzungsrechte benötigt, welche über die gesetzlich gewährten hinausgehen und daher nur vom Lizenzgeber gegeben werden können. Aus diesem Grund wird in dieser Arbeit auch bei der Verwendung der Begriffe für einen Vertragspartner unterschieden. Ein Lizenzgeber kann auch gleichzeitig Anbieter der Software sein, bei Verbreitung durch einen Distributor ist er es normalerweise jedoch nicht. Deshalb wird zwischen Lizenzgeber und Anbieter unterschieden.

Auch die Rechtmäßigkeit des Englischen als Lizenzsprache ist nachvollziehbar. Da ein Abschluss des Lizenzvertrags nur nach dem bisher Erörterten notwendig ist, stellt sich die Problematik der Lizenzsprache ausschließlich für Weiterentwickler. Sie müssen alleine aufgrund der Tatsache, dass alle Programmiersprachen (und deren Dokumentationen) auf dem Englischen basieren und ebenso alle Programmbibliotheken in englischer Sprache vorliegen, die englische Sprache beherrschen. Daher erschiene es

unverständlich, die Lizenzsprache Englisch als nicht ausreichend für den Abschluss eines Lizenzvertrags zu bewerten.

5. DIE OPEN SOURCE LIZENZEN – RECHTE UND PFLICHTEN

Da hinter dem Freiheitsgedanken von Open Source - wie eingangs erwähnt - viele verschiedene Philosophien und Theorien stehen, gibt es nicht nur eine Open Source Lizenz, sondern eine größere Anzahl davon, die mit unterschiedlichen Ansätzen versuchen, die Freiheit von Open Source Software zu schützen. Die OSI hat daher bislang 64 Open Source Lizenzen zertifiziert³¹⁴. Die Bestrebungen der OSI gehen allerdings in Richtung Reduzierung der Lizenzen, da deren stetig wachsende Anzahl einige Probleme für Open Source mit sich bringt³¹⁵. Auf den Linuxtagen 2005 in Boston wurde daher gefordert, die Lizenzvielfalt auf drei Open Source Lizenzen zu beschränken oder eine Einteilung der Lizenzen in Klassen (Gold, Silber, Bronze) vorzunehmen. Darüber hinaus hat die OSI ihr Lizenzierungsverfahren erweitert, indem sie die Kriterien zur Zertifizierung von Open Source Lizenzen verschärft³¹⁶. Neue Lizenzen sollen von jetzt an nicht mehr ausschließlich aufgrund bestehender Kriterien der OSD überprüft werden. In Zukunft müssen sie folgende weitere Kriterien erfüllen:

1. Die Lizenz darf keine Lösung für ein bereits durch bestehende lizenzierte Software gelöstes Problem darstellen.
2. Die Lizenz muss klar, einfach und verständlich geschrieben sein und soll auch für andere Projekte wieder verwendbar sein.

Die OSI versucht mit diesem Schritt, das Problem der Inkompatibilität verschiedener Open Source Lizenzen zu lösen, um die uneingeschränkte Wiederverwendbarkeit des Programmcodes weiterhin effektiv fördern zu

³¹⁴ <http://www.opensource.org/licenses/alphabetical> (Stand Oktober 2007).

³¹⁵ Siehe auch *Raymond*, Software Release Practice HOWTO, <http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/licensepractice.html>.

³¹⁶ <http://opensource.org/docs/policy/licenseproliferation.php>.

können. Dies tritt bspw auf, wenn es bei der Weiterentwicklung, Neuerstellung und Neukombination vom Sourcecode, der unter den zahlreichen unterschiedlichen Lizenzen geschrieben wurde, zu Vermischungen der jeweiligen Lizenzen kommt. Einen Ausweg daraus kann es nur über eine kleine Anzahl von Lizenzmodellen, zwischen denen Interaktionen handhabbar sind, geben³¹⁷. Diese Möglichkeit zur Interaktion wird nach Meinung der OSI durch die mittlerweile zu zahlreichen Open Source Lizenzmodelle nicht mehr in ausreichendem Maß gewährleistet sein. Neben der Entscheidung zu einer erweiterten Zertifizierung teilt die OSI die Lizenzen in drei Gruppen:

1. normale ("ordinary"),
2. bevorzugte ("preferred") und
3. abgelehnte ("deprecated") Lizenzen.

Wichtig zu bemerken ist, dass die von der OSI angebotene Zertifizierung letztendlich nur eine Informationshilfe für diejenigen ist, die ihr Programm freigeben bzw für jene, die es nutzen und weiterentwickeln wollen. Auch darf diese Zertifizierung keinesfalls als rechtliche Definition dienen, ob Software als Open Source angesehen werden kann. Allerdings stellt sie ein starkes Indiz für deren Klassifizierung dar.

Wie bereits erwähnt versteht man unter einer „Softwarelizenz“ die Übertragung bestimmter (Werknutzungs-)Rechte und Pflichten, welche mit dem Besitz und der Nutzung der Software verbunden sind. Um einen Open Source Lizenzvertrag abschließen zu können, muss der Urheber einer Software diese unter eine der Open Source Lizenzen stellen und dies auch bekannt geben - sei es durch einen Hinweis auf der Webseite, auf der die Software bezogen werden kann oder durch einen Hinweis im Quelltext selbst. In diesem Kapitel werden Inhalte dieser Lizenzen näher betrachtet, außerdem Rechte und Pflichten, die

³¹⁷ Zu dieser Problematik siehe auch die jeweilige Kompatibilität der Lizenzen und das „*Dual Licensing*“ siehe Kap 5.10.

sie einem Nutzer auferlegen erläutert und mit den gesetzlichen Bestimmungen in Österreich verglichen.

5.1. Die Lizenzarten

„Don't write your own license if you can possibly avoid it.“³¹⁸

- Eric S. Raymond

Es gibt neben den zertifizierten Lizenzen der OSI eine große Anzahl an Lizenzen, die den Gedanken freier Software aufgreifen und den Anforderungen der Free Software Definition genügen. Deswegen entsprechen die hier zu besprechenden Lizenzen sowohl der Free Software Definition als auch der Open Source Definition. Das IFROSS³¹⁹ in Deutschland teilt sie nach ihren lizenzrechtlichen Charakteristika wie folgt ein:

1. Lizenzen ohne Copyleft-Effekt,
 - a. BSDartige Lizenzen
 - b. Sonstige Lizenzen ohne Copyleft-Effekt
2. Lizenzen mit strengem Copyleft-Effekt,
 - a. GPLartige Lizenzen
 - b. Sonstige Lizenzen mit strengem Copyleft-Effekt
3. Lizenzen mit beschränktem Copyleft-Effekt,
 - a. MPLartige Lizenzen
 - b. Sonstige Lizenzen mit beschränktem Copyleft-Effekt
4. Lizenzen mit Wahlmöglichkeiten,
5. Lizenzen mit Sonderrechten und
6. Sonstige Lizenzen zur freien Nutzung von Immaterialgütern.

³¹⁸ <http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/licensepractice.html>.

³¹⁹ <http://www.ifross.de>.

Jede Software mit einer unter eine solche Kategorie fallenden Lizenz kann als Freie oder Open Source Software bezeichnet werden.

5.2. Lizenzen ohne Copyleft-Effekt – BSDartige Lizenzen

Die Lizenzen ohne Copyleft-Effekt, iW auch Non-Copyleft-Lizenzen genannt, gehen zu einem großen Anteil aus der „BSD License“ hervor und werden deshalb auch als BSDartige Lizenzen bezeichnet. Folgende Lizenzen gehören zu den BSDartigen Lizenzen³²⁰:

1. **Apache License (v1.0; v1.1 und v2.0)**^{*321}
2. Attribution Assurance License*
3. Berkley Database License
4. 4.4 BSD Copyright License (Originale BSD-Lizenz)
5. **FreeBSD Copyright (Modifikation der BSD-Lizenz)**
6. Christian Software Public License
7. Cougaar Open Source License
8. Cryptix General License
9. Eiffel Forum License (v1.0 und v2.0)*
10. Entessa Public License v1.0*
11. Free Fuzzy Logic Library Open Source License
12. Intel Open Source License for CDSA/CSSM Implementation*
13. Lua Copyright notice
14. MIT License*
15. Mozart License
16. **OpenLDAP Public License (v2.3, v2.5 und v2.7)**
17. Open Media Group Open Source License
18. Pangeia Informatica Copyright v1.2

³²⁰ Aus Gründen der Übersichtlichkeit werden Lizenzen, die in verschiedenen Versionen vorliegen nur einmal genannt und ihre verschiedenen Versionsnummern in Klammern erwähnt. Trotzdem wird natürlich auch eine ältere Lizenzversion durch eine neuere nicht außer Kraft gesetzt, sondern steht gleichwertig zu der neueren Version. Teilweise fällt eine ältere oder neuere Version auch unter eine andere Kategorie. In der Praxis ändert sich eventuell nur etwas an der Popularität der Lizenz, da eine neue Version Rechte entweder erweitern oder einschränken kann.

³²¹ Die mit einem * versehenen Lizenzen zeigen Lizenzen an, die von OSI approbiert wurden und somit zu den 58 bislang zertifizierten Lizenzen gehören.

19. Phorum License v2.0
20. PHP License v3.0*
21. PLT License
22. Python Copyright
23. skybuilders Open Source License
24. SpectSoft General Open Source License
25. Speechworks Public License – Software v1.1
26. 4Suite License v1.1
27. Tea SoftwareLicense
28. Udanax TM Open-Source License
29. Vovida Software License
- 30. W3C Software Notice and License**
31. Wide Open License
32. X.Net License*
- 33. X Window System License**
34. The license of ZLib
- 35. Zope Public License v2.0***

Es würde zu weit führen, auf jede einzelne Lizenz einzugehen, auch wenn ein Hauptmerkmal der BSDartigen Lizenzen der kurz gehaltene Lizenztext ist. Deshalb wird an dieser Stelle nur ein Einblick in die wichtigsten BSDartigen Lizenzen gegeben – dazu zählen neben den verschiedenen Versionen der BSD-Lizenz die „Apache Software License“, die „OpenLDAP Public License“, die „W3C Software Notice and License“, die „X Window License“ und die „Zope Public License“ – und ihre Unterschiede zu den im Open Source Bereich häufiger anzutreffenden Copyleft-Lizenzen aufgezeigt.

5.2.1. Die BSD-Lizenz

Die BSD Lizenz muss als wichtigste Alternative zur GPL angesehen werden, da sie im Bereich der Non-Copyleft-Lizenzen die wichtigste Rolle spielt. Diese Tatsache führt immer wieder zu generellen Diskussionen über die

Vorzüge der jeweiligen Lizenzen³²², welche jedoch ohne Ergebnis bleiben muss, da die Wahl eines Lizenzgebers für eine bestimmte Lizenz immer im Einzelfall hinterfragt werden muss.

5.2.1.1. Hintergrund

Die BSD-Lizenz ist nicht nur eine der ersten Open Source Lizenzen³²³, ebenso dient sie als Ursprungslizenz für viele Derivate, die oftmals fast wörtlich die BSD-Lizenz wiedergeben. Auch die BSD-Lizenz selbst wurde überarbeitet und enthielt bis 1999 in Punkt 3. die so genannte „advertising clause“ – „Werbeklausel“, welchen die FSF, die OSI und andere als Angriffspunkt auf die Lizenz nutzten. Diese Klausel stellte in der Praxis ein großes Problem dar. Da viele Programmierer ihre Miturheberschaft in der Lizenz angegeben haben wollten, wurde die an sich kurze BSD-Lizenz durch die Werbeklausel ein umfangreiches Papierwerk an Urhebernennungen³²⁴. Ein anderes noch wichtigeres Problem bestand darin, dass die ursprüngliche BSD-Lizenz durch die Werbeklausel nicht mit der GPL kompatibel war und damit eine strikte Trennung von BSD- und GNU-Software schuf. Die in der Werbeklausel genannten Pflichten sind der GPL unbekannt und finden darin nirgends Deckung. Durch das Einwirken der FSF wurde Punkt 3 am 22. Juli 1999 aus der Lizenz entfernt, sodass nun zwei Varianten verbreitet werden.

³²² Siehe bspw: http://www.softpanorama.org/Copyright/License_classification/social_dynamics_of_BSD_and_GPL.shtml.

³²³ Vgl Kap 2.1.2.

³²⁴ Als Bsp dafür mag die NetBSD-Lizenz dienen, die eine solche Autorenliste enthält.

5.2.1.2. Lizenzmerkmale

Beide Varianten beginnen mit einem Urhebervermerk und räumen jeder Person ein einfaches unbeschränktes Nutzungsrecht ein, die Software sowohl zu vervielfältigen, zu verbreiten und zu verändern als auch die veränderten Versionen zu verbreiten,³²⁵ solange

1. der Quelltext der veränderten Version im Falle der Weitergabe ebenfalls den Urhebervermerk, die Liste der Lizenzbestimmungen und den Haftungs- und Gewährleistungsausschluss enthält (Punkt 1.),
2. bei Weitergabe in Binärform der Urhebervermerk, die Liste der Lizenzbestimmungen und der Haftungs- und Gewährleistungsausschluss in der Dokumentation oder anderem mitgelieferten Material enthalten sind (Punkt 2.) und
3. die Namen der ursprünglichen Autoren nur mit deren ausdrücklicher schriftlichen Erlaubnis erwähnt werden dürfen (Punkt 3. bzw 4.).

Die eingangs erwähnte „Werbeklausel“ findet sich nur in der ursprünglichen Variante in Punkt 3. wieder.

Der Non-Copyleft-Effekt erreicht die BSD-Lizenz jedoch nicht durch ihren Lizenztext, sondern durch das, was nicht in ihr enthalten ist. Während die GPL in § 2 b) bestimmt, dass alle Weiterentwicklungen von ihren Urhebern unter den gleichen Bedingungen wieder freigegeben werden müssen, überlässt die BSD-Lizenz diese „Freiheit der Wahl“ dem Benutzer. Dieser kann Weiterentwicklungen in proprietäre Software überführen und braucht den Code der Weiterentwicklung nicht offen zu legen.

³²⁵ Jaeger/Metzger, Open Source Software 54.

5.2.1.3. Kompatibilität

Während Software von BSD-Derivaten häufig unter einer der BSD-Lizenz-Varianten lizenziert wird, ist Software, die mit Linux verbunden ist, meistens unter der GPL lizenziert. Quelltext, der unter der neuen Variante der BSD-Lizenz steht, kann durch die Non-Copyleft-Eigenschaft ohne Probleme in eine andere Lizenzform überführt werden, bspw auch unter der GPL. Dazu ist das Einverständnis aller Urheber nicht notwendig. Dieses wird umgekehrt aber von allen Urhebern benötigt, wenn Software, welche unter der GPL lizenziert ist, in eine BSD-Lizenz überführt werden soll, da diese im Gegensatz zur GPL nicht fordert, dass Weiterentwicklungen wieder „frei“ sein müssen.

5.2.2. Die Apache Software License

Das Apache-Projekt stellt sich als „meritokratisch“ dar: Je mehr jemand an der Entwicklung des Projekts mitgewirkt hat, desto mehr Rechte zur Weiterentwicklung besitzt er³²⁶. Es existiert eine aus den ursprünglichen Projektgründern bestehende Kerngruppe von Teilnehmern, welche von Zeit zu Zeit erweitert wird. Diese Kerngruppe besitzt den vollen Zugang zum „CVS Repository“ (die zentrale Versionsablage), zu den „CVS-Snapshots“ dagegen hat jeder Zugriff.

³²⁶ Vgl http://apache.org/about_apache.html.

5.2.2.1. Hintergrund

Die Apache Software License gibt es mittlerweile in den Versionen 1.0, 1.1 und 2.0, welche alle von der „Apache Software Foundation“ stammen³²⁷. Hauptanteil an der Popularität der Lizenz hat der Apache-Webserver, welcher überragende Marktanteile auf dem Webservermarkt besitzt und seit 1995 entwickelt wird.

5.2.2.2. Lizenzmerkmale der Version 1.0 und 1.1

Die Ursprungslizenz 1.0 kann in ihrer Betrachtung vernachlässigt werden, da sie nur für ältere Versionen der Webserver-Software zur Anwendung kommt, die mittlerweile größtenteils durch neuere Versionen ersetzt wurden, welche schon unter die Lizenzversion 1.1 oder 2.0 fallen.

Die Version 1.1 – sie stammt aus dem Jahr 2000 – hat (wie auch Version 1.0) große Übereinstimmungen mit der originalen BSD-Lizenz, nicht nur was den Text betrifft, sondern auch was Rechte und Pflichten der Nutzer angeht. Wie bei der originalen BSD-Lizenz findet man auch hier eine Werbeklausel, die den Nutzer bei Weitergabe verpflichtet, den Hinweis aufzunehmen, dass das weitergegebene Produkt Software enthält, welche von der Apache Software Foundation entwickelt wurde³²⁸. Um den Namen Apache zu schützen, bestimmt Punkt 5. des Lizenzvertrags, dass der Name Apache für Weiterentwicklungen nur mit schriftlicher Genehmigung der Apache Software Foundation verwendet werden darf. Dieser markenrechtliche Aspekt ist – wie auch *Jaeger/Metzger*³²⁹

³²⁷ Siehe auch <http://www.apache.org>. Software die unter einer der Apache-Lizenzen steht, ist bspw. ua *Apache Ant* (ein Java-Tool), *Apache HTTP server* und verschiedene seiner Module, *Apache Xindice* (Datenbankserver) und *SpamAssassin* (Spam mail-Filter).

³²⁸ Der englische Lizenztext lautet: „*This product includes software developed by the Apache Software Foundation (<http://www.apache.org>)*“.

³²⁹ *Jaeger/Metzger*, Open Source Software 56.

anmerken – nicht unproblematisch, da die Marke „*Apache*“ nicht entsprechend eingetragen wurde³³⁰.

5.2.2.3. Lizenzmerkmale der Version 2.0

Die Version 2.0 wurde im Januar 2004 mit der Intention veröffentlicht, die Lizenz für solche Produkte leichter verwenden zu können, die nicht von der Apache Software Foundation stammen. Ein wichtiges Merkmal war der Versuch, die Lizenz GPL-kompatibel zu machen. Der Lizenztext ist gegenüber den früheren Versionen stark erweitert.

Die Lizenz beginnt mit einer Liste an Begriffsdefinitionen. Danach folgt der „Grant of Copyright License“, gefolgt vom „Grant of Patent License“. Beide Punkte räumen jedem ein weltweites, nicht exklusives, kostenloses, lizenzfreies, unaufhebbares Recht ein, das Werk an sich als auch Weiterentwicklungen, in Quell- oder in Objektcode vorliegend, öffentlich zu benutzen, aufzuführen, unter eine Sublizenz zu stellen oder weiterzugeben. Die „*Patent License*“ will mit ihren zusätzlichen Bestimmungen dafür sorgen, Werke vor der Patentierung zu schützen. Sollte es zu einer Patentierung kommen, müssen die Werknutzungsrechte sowie die Verwertungsrechte jeweils weitergegeben werden.

Punkt 4. der Bestimmung hat große Ähnlichkeit mit den vorherigen Versionen der Lizenz. Dieser räumt jeder Person ein einfaches, unbeschränktes Nutzungs- und Verwertungsrecht ein, die Software sowohl zu vervielfältigen, zu verbreiten und zu verändern als auch die veränderten Versionen zu verbreiten, solange

³³⁰ Zu dieser Problematik im österreichischen Recht und der Frage der Verkehrsgeltung siehe Kap 7.

1. jedem des veränderten oder unveränderten weitergegebenen Werks eine Kopie des Lizenztextes beiliegt,
2. jede veränderte Datei einen Hinweis auf die Veränderung enthält,
3. im Quellcode jeder weiterentwickelten Version, welche weitergegeben wird, eindeutig darauf hingewiesen wird, welcher Quellcode mit welchen urheber-, patentmarkenrechtlichen oder sonstigen Merkmalen verwendet wurde. Ausgenommen davon sind jene Teile, die in der weiterentwickelten Form nicht übernommen wurden.
4. das Werk eine „Notice“ – Textdatei enthält, muss diese mit dem Quelltext oder in einer generierten Anzeige der Weiterentwicklung weitergegeben werden. Diese Datei dient ausschließlich zu Informationszwecken und hat keinen Einfluss auf die Lizenzbestimmungen. Auch eine vom Nutzer gemachte Veränderung darf keinerlei Einfluss auf die Lizenzbestimmungen haben.

Weiterentwickelte Werke dürfen unter zusätzliche oder veränderte Lizenzbedingungen gestellt werden, wodurch die Lizenz ihren Non-Copyleft-Charakter behält.

Die markenrechtliche Problematik, welche sich noch in den Vorgängerversionen fand, wird in Version 2.0 insofern aufgeweicht, als nun Markennamen verwendet werden dürfen, um den Ursprung des Werks zu definieren. Dafür muss der Inhalt der „*Notice*“-Textdatei wiedergegeben werden.

Die Lizenz endet mit einem Haftungs- und Gewährleistungsausschluss, welcher durch einen gesonderten Servicevertrag³³¹ geregelt werden kann. Die

³³¹ Diese Klausel ist in der Praxis relativ wichtig, da sich Firmen ohne ausreichenden Support selten auf einen Softwarevertrag einlassen. Diese Möglichkeit wird hier nun zur freien Wahl gestellt.

Lizenz stellt allerdings klar, dass ein solcher Vertrag keinen anderen Miturheber in die Pflicht nehmen darf.

5.2.2.4. Kompatibilität

Die Kompatibilität der Lizenzversionen ist teilweise umstritten. Während die Version 1.0 aus den gleichen Gründen wie die BSD-Lizenz mit der GPL nicht kompatibel ist, wird Version 1.1 von der FSF als inkompatibel eingestuft³³², da die Werbeklausel nicht mit letzter Konsequenz entfernt wurde. Die FSF spricht von „*a few requirements*“ – einigen Voraussetzungen – welche die Version 1.1 beinhaltet, die sie inkompatibel macht.

Die Version 2.0 wurde von der ASF mit der Intention herausgegeben, sie GPL-kompatibel zu machen. Hierzu gibt es zwei Sichtweisen:

5.2.2.4.1. Sichtweise der FSF

Die FSF knüpft die Inkompatibilität an die Patentklausel. Die Bestimmungen in der Lizenz, nach welcher eingeräumte Patentrechte wieder entzogen werden können, sieht die FSF als zu weitläufig an, da diese in der GPL kein Erfordernis sind. Die FSF stellt zwar fest, dass sie diese Bestimmungen nicht für ganz falsch erachtet, sie die Lizenz aber trotzdem inkompatibel zur GPL macht.

³³² <http://www.fsf.org/license-list.html#GPLIncompatibleLicenses>.

5.2.2.4.2. Sichtweise der ASF

Die kurz gehaltene Sichtweise der FSF hat bei der ASF für Verärgerung gesorgt. Diese ist daraufhin mit der FSF mit folgendem Ergebnis in Dialog getreten³³³:

- Die Einräumung eines ausdrücklichen Patentrechts würde ein unbegriffenes Patentrecht vernichten.
- Durch die Zurücknahme des ausdrücklichen Patentrechts würde jemand, der gegen eine Patentverletzung rechtlich vorgeht, seine Patentrechte verlieren, die er durch die GPL erlangt hätte.
- Der Verlust der Patentrechte geht einher mit dem Verlust der Nutzungsrechte
- Punkt 7 der GPL erlaubt gegen eine Patentverletzung in einem Werk, welches unter der GPL steht, rechtlich vorzugehen, ohne dass das Recht auf Weitergabe unter der GPL verloren geht. Die Weitergabe darf solange erfolgen bis ein Dritter das Recht zur Weitergabe, bspw durch Gerichtsurteil oder einstweilige Verfügung, einschränkt.
- Punkt 6 der GPL bestimmt, "*You may not impose any further restrictions on the recipients' exercise of the rights granted herein*", welches nur auf Urheber- nicht aber auf Patentrechte Anwendung findet.

³³³ <http://www.apache.org/GPL-compatibility.htm>. Die nachfolgenden Punkte wurden von der FSF noch nicht als richtige oder falsche Wiedergabe des Dialoges bestätigt.

5.2.2.4.3. Kommentar zu den Sichtweisen

Die ASF war vor dem Gespräch davon ausgegangen, dass Nutzungsrechte verloren gehen würden, sollte jemand versuchen, gegen eine Patentverletzung rechtlich vorzugehen. Diese Sichtweise wurde von der FSF dahingehend korrigiert, dass sie eine Klage für keine Beeinträchtigung des Rechts auf Weitergabe sähe. Die Beeinträchtigung könne nur von einem Gericht festgestellt und durch ein Unterlassungsurteil das Recht auf Weitergabe entzogen werden. Dieser Sichtweise ist zuzustimmen, da im Geschäftsleben ein Mitbewerber alleine durch das Behaupten einer Patentrechtsverletzung den Vertrieb des Gegners behindern kann.

Die ASF beklagt des Weiteren, dass die Inkompatibilität nur dann vorliegen könne, wenn in dem Werk ein Patent genutzt wurde. Sei dies nicht der Fall, käme die von der FSF bemängelte Klausel nicht zur Geltung. Deswegen betreffe die Inkompatibilität nur eine gerichtlich durchgesetzte Patentrechtsverletzung und nicht die Lizenz an sich.

Hintergrund dieser Diskussion mag einmal mehr der unterschiedliche Standpunkt sein, den Copyleft- und Non-Copyleft-Anhänger einnehmen. Eine Lösung des Kompatibilitätskonflikts kann demzufolge nur durch eine andere Interpretation der Lizenzauslegung durch die Konfliktpartner oder eine neue Version eines oder beider Lizenztexte gefunden werden.

5.2.3. Die OpenLDAP Public License

„I caution you to pay less attention to RMS and more attention to reality. The two are not always in sync with each other...“

- Aus einer **Diskussion** über die Kompatibilität der OpenLDAP Public License mit der GPL.

OpenLDAP ist eine Open Source Implementation des **Lightweight Directory Access Protocol**, ein Netzwerkprotokoll, das die Abfrage und die Veränderung von Informationen eines Verzeichnisdienstes (eine im Netzwerk verteilte hierarchische Datenbank) erlaubt³³⁴ und auf dem Netscape Directory Server basiert.

5.2.3.1. Hintergrund

Die OpenLDAP Public License liegt in verschiedenen Versionen vor. Die aktuellste ist Version 2.8.1. Auf die vorherigen Lizenzversionen wird an dieser Stelle nicht eingegangen.

5.2.3.2. Lizenzmerkmale

Die kurzen Bestimmungen der OpenLDAP License Version 2.8.1 ähneln im Großteil den Lizenzmerkmalen der BSD-Lizenz. Es wird das einfache und unbeschränkte Recht eingeräumt, die Software und die dazugehörigen Dokumentationen in veränderter oder unveränderter Form weiterzugeben, solange bei der Weitergabe im Quellcode die Urheberhinweise – bei Weitergabe in Binärform zusätzlich auch der Lizenztext – mitgeliefert werden. Die Lizenz endet mit den Garantie- und Haftungsausschlüssen.

³³⁴ <http://www.mitlinx.de/ldap>.

5.2.3.3. Kompatibilität

Bis zur Version 2.7 war die OpenLDAP Public License mit der GPL nach Ansicht der FSF nicht kompatibel³³⁵, da die früheren Versionen eine Werbeklausel enthielten. In Version 2.5 sollte in Ziffer 5. bspw auf die Verdienste der Urheber in „gebührender Weise“ hingewiesen werden. Viele Programmierer sahen darin jedoch keinen Grund für eine Inkompatibilität³³⁶. Letztendlich wurden diese Klauseln in den neueren Versionen entfernt und die OpenLDAP Public License ist seitdem allgemein anerkannt GPL kompatibel.

5.2.4. W3C Software Notice und License

Das World Wide Web Consortium (W3C) ist eine Vereinigung, die freie Softwarestandards – „recommendations³³⁷“ genannt – für das World Wide Web produziert³³⁸. Die „recommendations“ sollen dazu dienen, die Interoperabilität der verschiedenen im World Wide Web benutzten Protokolle zu sichern³³⁹.

³³⁵ <http://www.gnu.org/philosophy/license-list.html>.

³³⁶ <http://mail.gnome.org/archives/evolution-list/2001-April/msg00452.html>.

³³⁷ Ein W3C-Standard durchläuft die Bezeichnungen „*Working Draft*“, „*Last Call*“, „*Candidate Recommendation*“ und „*Proposed Recommendation*“, bevor es zu einer „*Recommendation*“ wird.

³³⁸ Das W3C leitet *Tim Berners-Lee*, der Erfinder des URL HTTP und von HTML, jenen grundlegenden Techniken, die das World Wide Web begründet haben. Eine gute Übersicht ist unter <http://www.w3.org/2005/01/timelines/timeline-2500x998.png> zu erhalten.

³³⁹ <http://www.w3c.org/consortium>.

5.2.4.1. Hintergrund

Um diese Interoperabilität zu gewährleisten, untersteht jede im Rahmen der W3C entwickelte Software³⁴⁰ der W3C Software Notice and License, die aus dem Jahr 1998 stammt.

5.2.4.2. Lizenzmerkmale

Die W3C Software Notice and License ist ebenfalls der BSD-Lizenz sehr ähnlich. Die Lizenz beginnt jedoch mit einer „Lizenzannahmebestimmung“, welche man in anderen BSDartigen-Lizenzen nicht findet. Bereits der Erhalt („obtaining“) eines unter der W3C Software Notice and License stehenden Werks soll, ebenso wie die Nutzung („using“) oder die Vervielfältigung („copying“), nach dem Lizenztext zum Vertragsabschluss mit dem Nutzer führen³⁴¹. Nach dieser Bestimmung folgt die für BSDartige-Lizenzen typische Einräumung eines einfachen, unbeschränkten Nutzungsrechts, die Software in veränderter oder unveränderter Form zu verbreiten und zu vervielfältigen, solange den Vervielfältigungen der Lizenztext beigelegt wird und die Verpflichtung zur Beibehaltung des Urhebervermerks (falls ein solcher nicht vorhanden ist, muss er in allen vervielfältigten, weitergegebenen Kopien enthalten sein). Der Urhebervermerk darf auch nicht zu Werbezwecken ohne schriftliche Genehmigung der betroffenen Urheber verwendet werden. Jede Änderung an dem Ursprungswerk muss deutlich und mit Datumsangabe gekennzeichnet sein. Die W3C Software Notice and License endet mit einem Haftungs- und Gewährleistungsausschluss.

³⁴⁰ Bspw HTML, XHTML, CSS, RDF, SVG.

³⁴¹ Zur Frage der Wirksamkeit dieser Bestimmung siehe Kap 4.2.

5.2.4.3. Kompatibilität

Die W3C Software Notice and License ist mit der GPL kompatibel³⁴². Die W3C hat ausdrücklich auf die Kompatibilität der beiden Lizenzen geachtet und den Lizenztext danach ausgerichtet. Es ist daher möglich, Software als Ganzes unter der GPL zu vertreiben, auch wenn ein Teil davon unter der W3C Software Notice and License lizenziert ist³⁴³.

5.2.5. „X11“ oder auch „MIT License“

Das „X Window System“ (auch „X11“ oder „X“ genannt) ist ein Standard-Toolkit und Protokoll, um graphische Benutzeroberflächen (GUI) auf Unix, Unix-ähnlichen Betriebssystemen und OpenVMS zu entwerfen³⁴⁴. X schafft die Basis für solche „Fenster“, die auf dem Bildschirm gezeichnet und dann verschoben werden können und mit der Computermaus und dem Keyboard interagieren. Eines der Hauptmerkmale des „X Window System“ ist die Netzwerkcharakteristik. Die Verwendung der Begriffe "client" und "server" wird hier umgekehrt zu der allgemeinen Bezeichnung genutzt, indem der "server" die lokale Bildschirmoberfläche bezeichnet und nicht eine „remote-machine“.

5.2.5.1. Hintergrund

Das „X Window System“ wurde 1984 am MIT entwickelt. Die aktuelle Protokollversion X11 wurde im September 1987 herausgegeben. Das Projekt

³⁴² <http://www.fsf.org/compatible-license-list.htm>.

³⁴³ *Jaeger/Metzger*, Open Source Software 58.

³⁴⁴ <http://www.x.org/wiki>.

wird von der X.Org Foundation³⁴⁵ geleitet, welche die verschiedenen Versionen unter die X11 oder auch MIT License gestellt hat. Der Name der Lizenz variiert im allgemeinen Gebrauch. Es taucht häufig der Name „MIT License“ auf, der aber aufgrund der Tatsache, dass vom MIT einige Softwarelizenzen entwickelt wurden, nach Meinung der FSF zu Missverständnissen führen kann³⁴⁶. Die FSF bevorzugt daher den Ausdruck „X11 License“, während die OSI im Gegenzug die Lizenz als „MIT License“ zertifiziert hat.

5.2.5.2. Lizenzmerkmale

Der Lizenztext der „X11“ oder „MIT License“ ist ihrem Inhalt nach wieder fast identisch der BSD-Lizenz, aber kürzer als diese. Die Einräumung des einfachen und unbeschränkten Nutzungsrechts regelt bereits die Bestimmung des ersten Satzes, in dem auch ausdrücklich die Lizenzgebührenfreiheit („*free of charge*“) erwähnt wird. Das Recht zur Nutzung, Vervielfältigung, Veränderung, Zusammenführung, Veröffentlichung, Weitergabe, Sublicenzierung und Verkauf der Software ist an die Pflicht zur Mitgabe des Urhebervermerks und der Lizenzbestimmungen bei jedem zur Weitergabe vervielfältigten Werk geknüpft³⁴⁷.

Eine andere Version des Lizenztextes enthält noch den Zusatz: *“Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.”*, der die Lizenz in ihren Auswirkungen dann fast identisch zu der BSD-Lizenz macht.

³⁴⁵ <http://www.x.org>.

³⁴⁶ <http://www.gnu.org/licenses/license-list.html#X11License>.

³⁴⁷ Die FSF nutzt bspw. eine Lizenz identischen Inhalts zur Lizenzierung ihrer „*ncurses library*“.

5.2.5.3. Kompatibilität

Die „X11“ oder „MIT License“ ist mit der GPL kompatibel³⁴⁸. Einige bevorzugen beim Umgang mit der X11 oder „MIT License“ die Wahlmöglichkeit des „dual-licensing“; so kann man bspw bei älteren Versionen der „cURL library“ auswählen, ob man die „Mozilla Public License“³⁴⁹ oder die „X11“ oder „MIT License“ der Lizenzannahme zu Grunde legen will.

Wohin Inkompatibilität mit der GPL führen kann, zeigt der Weg von XFree86³⁵⁰, einem ehemals äußerst populären X Server Systems. Als der Präsident des Projekts (*David Dawes*) entschied, XFree86-Software nicht mehr unter die „X11“ oder „MIT License“ zu stellen, sondern eine Lizenz zu benutzen, welche GPL-inkompatibel war, hatte das zur Folge, dass die X.org Foundation keinen Quelltext mehr in der neusten X11-Version übernahm, der unter der neuen GPL-inkompatiblen Lizenz veröffentlicht wurde³⁵¹. Nach und nach verließen die meisten der früheren XFree86-Entwickler das Projekt³⁵². Damit war der Versuch gescheitert, XFree86 von der GPL loszulösen und das Projekt wurde bedeutungslos.

³⁴⁸ <http://www.gnu.org/licenses/license-list.html#X11License>.

³⁴⁹ Vgl Kap 5.6.1.

³⁵⁰ Siehe *Wheeler*, Make Your Open Source Software GPL-Compatible or Else. <http://www.dwheeler.com/essays/gpl-compatible.html>.

³⁵¹ Das Statement lautete: "*The X.Org repository is based on XFree86 4.4 RC2. Just before its 4.4 release, XFree86 adopted a new license possibly incompatible with the GPL. For this reason, we have recreated its tree as closely as possible without importing files affected by the new license.*"

³⁵² <http://lwn.net/Articles/94763/>.

5.2.6. Zope Public License

Zope³⁵³ ist ein „web application server“, der in der Programmiersprache Python geschrieben wurde³⁵⁴. Dieser kann fast vollständig mittels einer webbasierten Benutzeroberfläche administriert werden und wird dazu benutzt, Objekte – wie bspw Dokumente, Bilder und „page templates“ – zu erschaffen und zu verwalten. Spezielle Objekte – wie bspw „wikis“, „weblogs“ oder Fotogalerien - können dabei als so genannte „add-ons“ eingebunden werden.³⁵⁵

5.2.6.1. Hintergrund

Die Software Zope wurde von der Firma Digital Creations (jetzt Zope Corporation) entwickelt und 1998 bei der Zusammenführung der Softwareprodukte „Bobo“ und „Principia“ unter dem Namen „Zope“ unter der „Zope Public License“ freigegeben. Diese wurde bis zum heutigen Tag mehrmals verändert. Die erste offizielle Version war 1.0, die bisherigen Nachfolgeversionen 1.1, 2.0 und 2.1³⁵⁶.

5.2.6.2. Lizenzmerkmale der Versionen 1.0 und 1.1

Die erste offizielle Zope Public License Version 1.0 hat große Ähnlichkeit mit der originalen BSD-Lizenz, trotzdem weist sie einige Modifikationen und Zusätze auf. Version 1.0 räumt zu Beginn des Lizenztextes ein einfaches und unbeschränktes Nutzungsrecht ein, gefolgt von der für BSDartige Lizenzen

³⁵³ Zope steht für „Z Object Publishing Environment“.

³⁵⁴ <http://www.zope.de/ueber-zope/was-ist-zope>.

³⁵⁵ Eine Eigenschaft, die „Zope“ wirtschaftlich interessant macht.

³⁵⁶ Die unter <http://www.zope.org/Resources/License> aufgeführten Versionen: 0.92, 0.93, 0.94, 0.95, 0.96 und 0.97 wurden nie öffentlich gemacht und haben daher fast keinerlei praktische Bedeutung. Stand 01.06.2005.

typischen Verpflichtung des Lizenznehmers – bei Weitergabe der Software als Quelltext oder in Binärform – den Lizenztext, den Urhebervermerk und den Haftungsausschluss mitzuliefern. Punkt 3. der Version 1.0 fordert alle Nutzer auf, den "Powered by Zope"-Button ebenfalls mitzuliefern. Dies ist eine Kann- und keine Muss-Bestimmung, da eine Verletzung von Punkt 3. keinerlei Konsequenzen nach sich zieht. Punkt 3. wurde in der Version 1.1 ersatzlos gestrichen. Die Punkte 4. und 6. sind der originalen BSD-Lizenz sehr ähnlich und bestimmen, dass bei Weitergabe der veränderten Software, in der Software und in den Werbematerialien der Hinweis „*This product includes software developed by Digital Creations for use in the Z Object Publishing Environment (http://www.zope.org/).*“ beigefügt wird. Nur bei unveränderter Weitergabe entfällt diese Pflicht. Punkt 5. verbietet jedoch für die Weitergabe einer veränderten Version die Nutzung der Namen „Zope“, „Digital Creations“ oder eines ähnlichen Namen ohne vorherige schriftliche Genehmigung. Punkt 7. ist eine besondere Bestimmung der Zope Public License und in der Version 1.0 und 1.1 enthalten. Die Bestimmung lautet: „*Modifications are encouraged but must be packaged separately as patches to official Zope releases. Distributions that do not clearly separate the patches from the original work must be clearly labeled as unofficial distributions. Modifications which do not carry the name Zope may be packaged in any form, as long as they conform to all of the clauses above.*“ Diese Bestimmung ist kontrovers diskutiert worden³⁵⁷, da sie sehr unterschiedlich ausgelegt werden kann. Nach Punkt 7. müssen Weiterentwicklungen als eigene „Patches“ der Originalsoftware beigefügt werden. Sollten die Weiterentwicklungen nicht als „Patch“ beigefügt, sondern direkt in die Software integriert worden sein, muss die Distribution als „inoffiziell“ bezeichnet werden. Diese Bestimmung führt zu einem Problem mit der OSD. Diese bestimmt in ihrem Punkt 4: „*The license may restrict Source code from being distributed in modified form only if the license allows the distribution of "patch files" with the Source code for the purpose of modifying the program at build time.*“ Die Zope Public License fordert in den Versionen 1.0 und 1.1, dass die Veränderungen "*packaged separately*", also getrennt mitgeliefert werden

³⁵⁷ Siehe bspw <http://lists.debian.org/debian-legal/2001/09/msg00018.html>.

müssen (keine Kann-Bestimmung), die OSD jedoch verlangt von Open Source Software, dass „Patches“ in einer Lizenz nur solange eingebunden werden können, solange diese „Patches“ nur "with the Source code" weitergegeben werden dürfen. Deshalb verlangt die Zope Public License hier eine Trennung vom Quelltext, die gegen Punkt 4. der OSD verstößt. Der letzte Satz in Punkt 7. ist nach *Jaeger/Metzger*³⁵⁸ ohne praktische Bedeutung, da eine Weitergabe der Software in veränderter Form gem Punkt 4. und 6. der Zope Public License ohnehin einen Hinweis auf Zope enthalten muss und eine Weitergabe der Software ohne diesen Hinweis daher nicht möglich ist. Um diesen Problematiken entgegenzuwirken, entwickelte Zope die Nachfolger der 1.x – Versionen, die Version 2.0 und 2.1.

5.2.6.3. Lizenzmerkmale der Versionen 2.0 und 2.1

Die Zope Corporation hat aus den Erfahrungen der 1.x-Versionen gelernt und drückt dies zu Beginn der Zope Public License 2.0 aus, indem dort einerseits auf die Zertifizierung der Lizenz durch die OSI hingewiesen andererseits die Kompatibilität zur GPL hervorgehoben wird. Erst danach kommt die Einräumung des einfachen und unbeschränkten Nutzungsrechts. Die Punkte 1. und 2. gleichen dabei durchgehend der BSD-Lizenz. Die Version 2.0 bestimmt in Punkt 3., dass der Name „Zope Corporation“ nicht ohne vorherige schriftliche Genehmigung bei der Bewerbung von weiterentwickelten Versionen verwendet werden darf. In Punkt 4 regelt die Version 2.0 markenrechtliche Aspekte. Die Nutzung von „Servicemarks (sm)“ oder „Trademarks (tm)“³⁵⁹ der Zope Cooperation sind in einem besonderen Vertrag geregelt³⁶⁰. Der letzte

³⁵⁸ *Jaeger/Metzger*, Open Source Software 60; ebenso: *Brandon*, FYI: Zope Public License 1.1 vague, contradictory, and not DFSG-free, <http://lists.debian.org/debian-legal/2001/09/msg00014.html>.

³⁵⁹ Diese Bestimmungen gelten vorrangig für den amerikanischen Raum. Zum Schutz von Markenrechten siehe Kap 7.

³⁶⁰ Siehe <http://www.zope.com/Marks>. Die Einräumung zur Nutzung der Markenrechte soll kostenfrei sein, die Lizenzstrategie befindet sich aber zum jetzigen Zeitpunkt noch in der Testphase.

Punkt 5. betrifft die Kennzeichnung von Veränderung mit Datum und Urhebervermerk. Zum Abschluss folgt der Haftungsausschluss.

Die Zope Public License wurde in Version 2.1 abgeändert, um die Lizenz „wieder benutzbar“³⁶¹ zu machen. Dazu wird der Lizenz ein separater Urhebervermerk beigefügt, auf den gleich zu Beginn der Version 2.1 hingewiesen wird. Die Lizenz selbst unterscheidet sich von der Version 2.0 in den Punkten 1. und 2.; anstelle der Forderung, den Urhebervermerk am Anfang der Lizenz anzubringen, wird gefordert, diesen in dem separaten Urhebervermerk mitzuliefern. Die Punkte 3. und 4. beziehen sich nicht mehr wie in Version 2.0 auf die Zope Cooperation, sondern auf die jeweiligen Urheber.

5.2.6.4. Kompatibilität

Die Versionen 1.x sind mit der GPL – oft aus den gleichen Gründen wie die originale BSD-Lizenz – inkompatibel³⁶². Ab Version 2.0 besteht Kompatibilität mit der GPL, da die angesprochenen problematischen Klauseln aus dem Lizenztext entfernt wurden.

³⁶¹ <http://www.zope.org/Resources/License/>.

³⁶² Siehe auch: <http://www.dzug.org/maillinglisten/zope/archive/2003/2003-03/1047467083020>.

5.3. Lizenzen ohne Copyleft-Effekt – Sonstige Lizenzen

Über die BSDartigen Lizenzen hinaus gibt es weitere Lizenzen ohne Copyleft Effekt. Zu diesen „Sonstigen Lizenzen“ gehören:

1. Academic Free License (v1.1 und v1.2)*
2. CNRI Open Source License Agreement (bis Python 1.6)
3. Free Software License
4. Galen Open Source License
5. Globus Toolkit Public License
6. ICU License
7. ISC License
8. MirOS License
- 9. Open Group Test Suite License**
10. PSF License Agreement (Python)
11. Ruby License
12. Sendmail License
13. SFL License Agreement
14. Standard ML of New Jersey Copyright Notice
15. Suneido Free Software License
16. Tcl/Tk License Terms
17. xinetd License
18. Zope Public License v2.1*

In diesem Bereich wurde nur die „Open Group Test Suite License“ von der OSI zertifiziert, weshalb auch nur auf diese eingegangen wird.

5.3.1. Die “Open Group Test Suite License”

Die “Open Group Test Suite License” dient – wie der Name bereits aussagt – einem Programm, mit dessen Hilfe die Funktionalität von LDAP überprüft werden kann.

5.3.1.1. Hintergrund

Die Open Group Test Suite License (OgTSL) wurde von der Open Group entwickelt, einem IT-Dienstleistungsunternehmen, welches Beratungen unter anderem in den Bereichen Strategie, Management und dem Testen von Softwareentwicklungen anbietet. Hauptprodukt ist die „Basic LDAP Interoperability Test suite“ (BLITS), welche der Funktionsüberprüfung von LDAP-Client und Server-Paaren dient.

5.3.1.2. Lizenzmerkmale

Die Lizenz ist unwesentlich länger als die BSD-Lizenz. Sie beginnt mit einer Präambel, welche die Intentionen der Lizenzautoren erklärt und ein Manifest enthält, warum das Testen im Softwarebereich so wichtig für Nutzer, Entwickler und Dienstleistende ist. Sie endet mit der Bitte, „bugs“ an den Lizenzgeber zu melden, damit diese in der Originalsoftware ausgebessert werden können.

Ziffer 1 OgTSL räumt dem Lizenznehmer das Recht ein, den Quellcode der „Standard Version of this Package“ zu vervielfältigen. Dies stets unter der Voraussetzung, dass die Urhebervermerke und sonstige Bemerkungen mitverbreitet werden.

Ziffer 2 OgTSL gestattet „bug“ und „portability fixes“ sowie andere Weiterentwicklungen an der Software, die der Lizenznehmer vom Rechteinhaber oder aus der Public Domain erhalten hat. Dies ändert nichts an der Qualifizierung der Software als „Standard Software“. Die Klausel ist nach dem UrhG unwesentlich, da dieses dem Nutzer der Software gem § 40d Abs 2 UrhG das Recht einräumt, das Programm zu bearbeiten soweit dies für die bestimmungsgemäße Benutzung durch den zur Benutzung Berechtigten

notwendig ist. Dazu gehört ebenfalls die Anpassung an die Bedürfnisse des Nutzers.

In Ziffer 3 OgTSL erhält der Lizenznehmer das Recht, Kopien der Software weiterzuentwickeln, solange

- die Weiterentwicklungen am Programm bedeutend gekennzeichnet werden, beinhaltend die Erklärung, in welcher Weise und zu welchem Datum dies geschehen ist.
- nicht zum „*Standard Package*“ gehörende Executables und Testcases umbenannt werden, damit sie nicht mit denen des „*Standard Packages*“ in Konflikt geraten.
- die weiterentwickelten Executables auf einer eigenen mitzuverbreitenden Seite dokumentiert werden.

Das Recht zur Verbreitung in Objektform befindet sich in Ziffer 4 OgTSL, nachdem dies gestattet wird, solange

- bei jedem nicht zum „*Standard Package*“ gehörenden Executable das (korrespondierende) Executable des „*Standard Packages*“ mitverbreitet wird.
- alle nicht zum „*Standard Package*“ zugehörigen Executables umbenannt werden.
- die Unterschiede der weiterentwickelten Executables zu jenen des „*Standard Packages*“ dokumentiert werden und darüber hinaus,
- ein Hinweis darauf gegeben wird, wo die Executables des „*Standard Packes*“ bezogen werden können.

Ziffer 5 OgTSL räumt dem Lizenznehmer das Recht ein, für Kopien die zum Zweck der Weitergabe gemacht werden oder für Dienstleistungen, die er

einem Dritten hinsichtlich des Programms anbietet, Geld zu verlangen. Das „*Package*“ selbst muss unentgeltlich weitergegeben werden. Darüber hinaus ist es einem Lizenznehmer gestattet, das „*Package*“ in Verbindung mit einem unter einer anderen Lizenz stehenden Programm gemeinsam zu vertreiben, solange der Lizenznehmer sich nicht als Urheber des Gesamtwerkes präsentiert.

Gem Ziffer 6 OgTSL dürfen Skripte oder Programmbibliotheken, welche aus dem Einsatz des Programms resultieren, eigenständig und unter vom Urheber zu bestimmenden Bedingungen weiterverbreitet werden. Subroutinen fallen gem Ziffer 7 OgTSL außerhalb des Lizenzbereichs.

Ziffer 8 OgTSL untersagt es dem Lizenznehmer, den Namen des Urhebers ohne dessen vorherige schriftliche Einwilligung zu Werbezwecken zu benutzen. Dies ist aus Gründen des Namensschutzes iSd § 43 ABGB als Hinweis aufzufassen, da eine unberechtigte Verwendung zu den in § 42 ABGB genannten Konsequenzen führen könnte.

Die OgTSL endet mit einem kurzen Gewährleistungs- und Haftungsausschluss

5.3.1.3. Kompatibilität

Die OgTSL ist mit der GPL kompatibel, da die OgTSL es jedem Lizenznehmer selbst überlässt, ein unter der OgTSL lizenziertes Programm mit einem unter einer anderen Lizenz stehenden Programm zu verbinden. Hinsichtlich anderer Open Source Lizenzen ohne oder mit eingeschränktem Copyleft-Effekt besteht daher ebenfalls Kompatibilität, da diese Lizenztypen die Möglichkeit einräumen, Weiterentwicklungen oder Verbindungen von eigenständigen Programmen unter eine selbst zu wählende Lizenz zu stellen.

5.4. Lizenzen mit strengem Copyleft-Effekt – GPL-artige Lizenzen

Die Lizenzen mit strengem Copyleft-Effekt sind die am meist genutzten Lizenzen im Open Source Bereich, vor allem da die GPL und die GNU Lesser General Public License (LGPL) in diese Kategorie fallen. Hauptmerkmal dieser Lizenzen ist die Bedingung, Weiterentwicklungen ebenfalls unter die genutzte Lizenz zu stellen³⁶³. Folgende Lizenzen gehören zu den GPL-artigen Lizenzen:

1. Affero General Public License
2. Alternate Route Open Source License (v1.1)
3. CrossPoint Quelltextlizenz
4. eCos license (v2.0)
5. FreeCard License
6. GNU Emacs General Public License
7. **GNU General Public License (GPL) (v1.0 und v2.0)**
8. Open RTLinux Patent License (v2.0)

Da in diesem Bereich der GPL die größte Aufmerksamkeit zukommt, soll diese stellvertretend für die anderen GPL-artigen Lizenzen besprochen werden.

5.4.1. Die GNU General Public License 2.0

“We are about to make a sweeping, revolutionary change in the General Public License. The terms for copying will be essentially unchanged, but the architecture of the legal framework used to embody them will make a quantum leap.”

- Aus dem **GNU Bulletin Board 1989**

Die meistverbreitete und deshalb am meisten beachtete Lizenz ist unbestreitbar die GPL. Diese liegt seit 1991 in Version 2.0 vor und ist 2007

³⁶³ Vgl Kap 2.4.3.

durch Version 3.0 aktualisiert worden³⁶⁴. Aufgrund der Tatsache, dass die meisten Arbeiten, die sich mit Open Source Lizenzen beschäftigen, oftmals nur die GPL 2 zum Inhalt haben und zu dieser Version viele Aufsätze und Bücher vorliegen, wird hier eine ausführliche Betrachtung unterbleiben und an gegebener Stelle auf weiterführende Literatur verwiesen.

5.4.1.1. Hintergrund

Die „Mutter aller Open Source Lizenzen“, war die erste Lizenz, die den Coyleft-Effekt verwendet hat³⁶⁵ und die *Richard Stallman* zusammen mit dem Juristen *Eben Moglen* im Januar 1989 der Öffentlichkeit präsentierte³⁶⁶. Sie basiert auf einer Vereinheitlichung gleichartiger Lizenzen, die für die früheren Versionen von „GNU Emacs“, dem „GNU Debugger“ und der „GNU Compiler Collection“ verwendet wurden. Das Ziel von *Stallman* und *Moglen* war es, eine Lizenz zu entwickeln, die universell einsetzbar war, im Gegensatz zu den früheren Lizenzen für freie Software, die man für die jeweilige Software angepasst hatte. Seit ihrer Einführung ist die GPL 2 die am weitesten verbreitete Softwarelizenz im Open Source Bereich, daher sind die meisten Programme im GNU-Projekt unter der GPL 2 und der LGPL lizenziert³⁶⁷. Die bekanntesten Beispiele dafür sind der Linux Kernel (von *Stallman* deshalb auch als GNU/Linux bezeichnet³⁶⁸) und die grafischen Benutzeroberflächen KDE und GNOME.

Die GPL 2 sticht nicht nur als „Ursprungslizenz“ aus den anderen Open Source Lizenzen heraus, sie zeichnet sich auch durch Länge und sprachliche Exaktheit aus, die andere Open Source Lizenzen teilweise vermissen lassen.

³⁶⁴ <http://gplv3.fsf.org/>.

³⁶⁵ *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt XI.

³⁶⁶ http://www.free-soft.org/gpl_history/ (20.09.2005).

³⁶⁷ Siehe auch *Sieckmann*, Bravehack 62 mwA.

³⁶⁸ Siehe „*Revolution OS*“ (Film).

Zuletzt hat die GPL 2 Aufmerksamkeit durch das Urteil des Landgerichts München I³⁶⁹ erlangt, in dem weltweit erstmalig über die Wirksamkeit der GPL 2 entschieden wurde. Ob dieses Urteil für die GPL 2 positiv zu bewerten ist, bleibt der persönlichen Betrachtung überlassen. Die neben der GPL 2 existierenden Open Source Lizenzen mussten noch nie von einem Gericht interpretiert werden. Das lässt darauf schließen, dass diese ausreichend formuliert sind und einen Gang vor Gericht unnötig machen.

5.4.1.2. Lizenzmerkmale

Gleich zu Beginn der Lizenz wird – durch den dort stehenden Urhebervermerk – die FSF als Inhaberin der Rechte am Lizenztext ausgewiesen. Dies bedeutet, dass die FSF iSd Ziffer³⁷⁰ 9 GPL 2 und den in Ziffer 8 GPL 2 genannten Ausnahmen alleine berechtigt ist, Änderungen am Lizenztext vorzunehmen³⁷¹. Dies wird auch im Vorspann der Lizenz deutlich, welcher jedem die Vervielfältigung und Weitergabe des Lizenztextes, nicht aber eine Änderung erlaubt.

Die nachfolgende Präambel beschreibt die Intentionen und Ziele der Lizenz sowie einige ihrer Hauptcharaktereigenschaften. Die Präambel gehört nicht zu den eigentlichen Lizenzbedingungen dazu³⁷², dennoch kann sie zur Auslegung des Lizenzvertrags iSd §§ 914 und 915 ABGB herangezogen werden. Die in Deutschland herrschende Zweckübertragungslehre³⁷³ nimmt die Präambel gerne heran, um bestimmte Auslegungen zu stützen. Da die

³⁶⁹ Urteil vom 19.05.2004, Az 21 O 6123/04.

³⁷⁰ Vielfach taucht in der einschlägigen Fachliteratur „§“ oder „Nr.“ auf, um Lizenzabschnitte der GPL zu bezeichnen. In dieser Arbeit wird der Ausdruck „Ziffer“ verwendet, der sich auch in der neueren Fachliteratur findet.

³⁷¹ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 26.

³⁷² Die Präambel endet mit: „*The precise terms and conditions for copying, distribution and modification follow.*“

³⁷³ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 28.

Zweckübertragungstheorie jedoch in Österreich nicht angewendet wird³⁷⁴ und allgemeine Formulierungen in Verträgen als wirksam angesehen werden, kommt der Präambel hier keine so große Wirkung zu wie in Deutschland³⁷⁵.

Ziffer 0 GPL 2 regelt, dass die GPL 2 nur für jene Programme Anwendung findet, bei denen der Rechtsinhaber³⁷⁶ auf die Lizenzbedingungen hinweist. Dadurch wird gewährleistet, dass jeder, der eine Kopie des Programms erhält, Kenntnis von dem Vertragsangebot der GPL 2 hat³⁷⁷. Ziffer 0 GPL 2 definiert des Weiteren zentrale und wiederkehrende Begriffe des Lizenztextes, um weitere Auslegungshilfe zu geben. Der Begriff „Programm“ bezeichnet demnach jedes (Computer-)programm ebenso wie jegliche Form eines Werks. Das bedeutet, dass die GPL 2 nicht ausschließlich für die Lizenzierung von Software verwendet werden kann, was aber nicht zu empfehlen ist, da die Lizenz überwiegend auf Software eingeht³⁷⁸. „A work based on the program“ ist nach Ziffer 0 GPL 2 einerseits das Programm selbst andererseits bezeichnet es jegliche urheberrechtlich relevante Weiterentwicklung³⁷⁹. Eine Weiterentwicklung ist demnach jedes Werk, welches ein unter der GPL 2 stehendes Programm ganz oder teilweise verändert oder unverändert beinhaltet. Zu einer solchen Veränderung zählt auch das bloße Übersetzen des Werks in eine andere Sprache.

Der zweite Absatz der Ziffer 0 GPL 2 erklärt recht umgangssprachlich durch die negative Ausschließung von Nutzungsarten, welche „Anwendungsbereiche“ durch die Lizenz abgedeckt werden. Wie bereits

³⁷⁴ Reinprecht in Kucsko, Geistiges Eigentum 1243.

³⁷⁵ Weiters siehe Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 29 ff.

³⁷⁶ Rechtsinhaber können der oder die Urheber des Programms sein, der oder die Inhaber ausschließlicher Nutzungsrechte an dem Programm oder der Arbeitgeber eines Programmierers (vgl Kap 3.2.2.).

³⁷⁷ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 34; Omsels in FS Hertin, Open Source und das deutsche Vertrags- und Urheberrecht 149.

³⁷⁸ Siehe auch Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 37.

³⁷⁹ Eine relevante urheberrechtliche Weiterentwicklung betreffend siehe Kap 5.4.1.2.2.

erwähnt³⁸⁰ hat diese Bestimmung eine zentrale Bedeutung für die Nutzungsarten der Open Source Software. Die reine Nutzung der Open Source Software bedarf keines Vertragsabschlusses unter der GPL 2. „*The act of running the program is not restricted*“ verweist damit auf die nationalen Rechte, die dem Nutzer vom Gesetzgeber zugestanden werden und die er in Österreich durch § 40 d UrhG erhält. Erst wenn der Nutzer über diese gesetzlichen Mindestrechte hinausgehende Rechte benötigt, bspw bei Weiterentwicklung und Vertrieb, muss er einen GPL 2-Lizenzvertrag abschließen.

Die folgenden Ziffern 1 und 2 GPL 2 verleihen dem Nutzer die typischen für die Einordnung von Open Source Software notwendigen Rechte und knüpfen dies gleichzeitig an Bedingungen. Ziffer 1 behandelt dabei die Weiterverbreitung und Vervielfältigung des ursprünglichen Werks und fordert den Nutzer auf, den Urhebervermerk sowie den Haftungsausschluss in unveränderter Form beizugeben, sollte er von diesen Rechten Gebrauch machen. Der Lizenznehmer hat dabei die Wahl, ob er durch einen Hinweis auf den Lizenztext auf dem Datenträger selbst dieser Verpflichtung nachkommt oder ob er einen digitalen Hinweis anbringt (wenn dieser nicht bereits existiert)³⁸¹. Der zweite Absatz in Ziffer 1 erlaubt es, eine Gebühr in selbst zu wählender Höhe für den Kopiervorgang zu erheben sowie gegen (jedoch auch ohne) Entgelt eine Garantie für das Programm abzugeben³⁸². Dabei ist wichtig darauf zu achten, dass damit keine Lizenzgebühren gemeint sind, was im Einzelfall teilweise schwierig zu überprüfen sein dürfte, da Gebühren für sonstige Leistungen – wie bspw Installation der Software oder Dokumentations- und Hilfsmaterial – durchaus erlaubt sind³⁸³. Dies wird nochmals in Ziffer 2 b) GPL 2 deutlich ausgesprochen.

³⁸⁰ Vgl Kap 3.2.3.

³⁸¹ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 50.

³⁸² Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 40; Wiebe/Prändl, ÖJZ 2004, 631.

³⁸³ Küng, MR 2004, 25.

Ziffer 2 GPL 2 räumt einem Lizenznehmer Nutzungsrechte für weiterentwickelte Werke ein und knüpft auch dies wieder an bestimmte Bedingungen. Sie wird oft genug als das „Herz“ der Lizenz bezeichnet³⁸⁴, da Ziffer 2 b) GPL 2 den „copyleft-Effekt“³⁸⁵ beinhaltet. Zu Beginn der Ziffer 2 GPL 2 wird dem Lizenznehmer erlaubt, seine Programmkopie weiterzuentwickeln sowie die modifizierte Version zu vervielfältigen und zu verbreiten. Diese Erlaubnis ist wichtig, da das modifizierte Werk sonst nur den gesetzlichen Beschränkungen des § 40d UrhG unterliegen würde und eine Vervielfältigung und Verbreitung nur in diesem durch das Gesetz eingeschränkten Rahmen möglich wäre. Die Bedingungen in Ziffer 2 GPL 2 verweisen einerseits auf jene der Ziffer 1 GPL 2 und fordern darüber hinaus, dass iSd Ziffer 2 a) GPL 2 jede weiterentwickelte Programmdatei einen Hinweis darauf enthält, wer die Änderung vorgenommen hat und wann dies geschehen ist³⁸⁶.

Ziffer 2 b) GPL 2 beinhaltet – wie erwähnt – das Copyleft. Dadurch wird jeder Lizenznehmer verpflichtet, bei einer Weiterentwicklung eines unter der GPL 2-lizenzierten Programms das daraus neu entstandene Werk lizenzgebührenfrei wiederum unter der GPL 2 zu lizenzieren. Diese knappe Formulierung ist nicht nur eine der zentralen Vorschriften, sondern gehört auch zu den am meisten umstrittenen Punkten bei der Auslegung der GPL 2³⁸⁷. So ist einerseits der Punkt, wann es zu einer Veröffentlichung der Software kommt, von Interesse andererseits ist es wichtig abgrenzen zu können, ab wann der Beitrag einer GPL 2-lizenzierten Software zu einer neu entwickelten Software so groß ist, dass das Ergebnis, sprich die gesamte Software, unter die GPL 2 gestellt werden muss.

³⁸⁴ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 59.

³⁸⁵ Siehe Kap 2.4.3.

³⁸⁶ Wie ein solcher Hinweis in der Praxis anzubringen ist, siehe Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 61 ff.

³⁸⁷ Jaeger/Metzger, Open Source Software 39; Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 63 (hier auch ausführlich zum „viralen Effekt“); Spindler, Rechtsfragen der Open Source Software 52. Auf die Probleme, die dabei in Zusammenhang mit der Kompatibilität, bspw bei der Verknüpfung von unter der GPL-lizenzierte Software mit proprietärer Software, auftauchen, wird später in einem eigenen Punkt „Kompatibilität“ eingegangen.

5.4.1.2.1. Veröffentlichung der Software

Für Unternehmen wie für Behörden ist die Frage nach der Pflicht zur Offenlegung besonders wichtig, da sie beim Einsatz von Open Source Software in ihrem Bereich oftmals Teile der Software für ihre Zwecke weiterentwickeln. In der Regel sind sie jedoch nicht daran interessiert, diese mitunter mit viel Know-how und Aufwand erstellten Weiterentwicklungen, die nur Behörden oder Firmen intern zum Einsatz kommen, iSd Ziffer 2 GPL 2 als Ganzes der GPL 2 zu unterstellen. Dadurch würde der eigenständige, individuelle Bearbeitungsteil veröffentlicht und bspw einem Mitbewerber lizenzgebührenfrei zur Verfügung gestellt werden³⁸⁸.

Da die GPL 2 selbst keine nähere Auslegung zur Thematik der Veröffentlichung enthält, sondern es dem jeweiligen nationalen Recht überlässt zu definieren, wann es zu einer Offenlegung der Weiterentwicklung kommt, muss man nach österreichischem Recht beurteilen, ab wann von einer „Öffentlichkeit“ im Zuge der Vervielfältigung und Verbreitung auszugehen ist.³⁸⁹

Wie bereits ausgeführt³⁹⁰ ist unter einer Veröffentlichung iSd Ziffer 2 GPL 2 jede öffentliche Zugänglichmachung und jede öffentliche Zurverfügungstellung gemeint, unabhängig davon, ob die Software körperlich (mittels Datenträger) oder nicht körperlich (via Download) zur Verfügung gestellt wird³⁹¹. Die hiesige Rechtsprechung hat inzwischen einige Definitionen

³⁸⁸ *Spindler*, Rechtsfragen der Open Source Software 52; *Andreewitch*, Risiken bei firmen- und behördeninterner Bearbeitung von Open Source Software, MR 2005, 36; *Jaeger/Metzger*, Open Source Software 39.

³⁸⁹ Wie *Andreewitch* richtig bemerkt, ist auch der Begriff der Öffentlichkeit nicht durch die Info-RL geregelt, sondern wurde der rechtlichen Auslegung eines jeden MS überlassen; siehe auch *Spindler*, Europäisches Urheberrecht in der Informationsgesellschaft, GRUR 2002, 105 (108); *Reinbothe*, Die EG-Richtlinie zum Urheberrecht in der Informationsgesellschaft, GRUR Int 2001, 733 (736).

³⁹⁰ Vgl die analoge Auslegung beim Erschöpfungsgrundsatz Kap 3.2.3.1 sowie die rechtlichen Grundlagen beim Zuverfügungstellungsrecht Kap 3.2.3.2.

³⁹¹ *Andreewitch*, MR 2005, 36.

herausgearbeitet, was unter "öffentlich" bzw. "nicht-öffentlich" zu verstehen ist und sich dabei an der deutschen Rechtsprechung orientiert³⁹². Danach sind einerseits qualitative Kriterien – wie die der Art der persönlichen Beziehung der (potenziellen) Teilnehmer (hier: Lizenznehmer) zueinander und dem Zweck der Sache (hier: Zurverfügungstellung der Software) – zu beachten andererseits muss das quantitative Kriterium, nämlich die Anzahl dieser tatsächlichen oder potenziellen Teilnehmer, berücksichtigt werden³⁹³.

Bei Anwendung dieser Definition auf die angesprochene Problemstellung muss man zu dem Ergebnis kommen, dass die Zugehörigkeit zu einer Firma oder zu einer bestimmten Behörde nicht ausreicht, um jenes von der Rechtsprechung geforderte Kriterium der persönlichen Verbundenheit zu erfüllen³⁹⁴. Auch weitgehend gleichgerichtete sachbezogene Interessen eines Personenkreises – bspw bei der Nutzung einer innerhalb Firmen oder Behörden weiterentwickelten Open Source Software, die unter der GPL 2 steht und nur im internen Kreis verwendet wird – erscheinen nicht zweckdienlich, um die Nicht-Öffentlichkeit zu begründen³⁹⁵. Wird GPL 2 lizenzierte Software bspw Abteilungs intern angepasst, indem eine Schnittstelle definiert wird, um mit der „hauseigenen“ Software interagieren zu können, wird trotz möglicher Auswirkungen dieser Änderung für die gesamte Firma oder Behörde keine persönliche Verbundenheit mit dieser Weiterentwicklung gegeben sein. Andererseits werden umfangreiche Weiterentwicklungen an der Software, an deren Entstehung ein Großteil der IT-Abteilung mitgearbeitet hat, die geforderte persönliche Verbundenheit begründen. Eine Veröffentlichung sollte in diesem Fall auch im Interesse der Firma- oder Behörde liegen, da ein fortlaufender Support der Software durch die Open Source Gemeinschaft nur dann gegeben

³⁹² *Dittrich*, Österreichisches und internationales Urheberrecht⁴ § 18 E 11 mwA.

³⁹³ *Andreewitch*, MR 2005, 36.

³⁹⁴ *Jaeger/Metzger*, Open Source Software 40; *Andreewitch*, MR 2005, 39, jeweils unter Berücksichtigung der dt Rechtsprechung, insbesondere des BGH Urteils vom 24.6.1955, I ZR 178/53, BGHZ 17, 376 (378 f) (Betriebsfeiern); ebenso: *Spindler*, Rechtsfragen der Open Source Software 53, der dabei noch einen Versuch unternimmt, den Begriff der Öffentlichkeit negativ abzugrenzen.

³⁹⁵ *Andreewitch*, MR 2005, 39.

ist, wenn diese auch umfassende Kenntnis von der Weiterentwicklung erlangen kann. Ein solches Ergebnis ist mE nachvollziehbar, auch wenn der Telos der Norm durch die unpräzise Formulierung breiten Spielraum für Interpretationen gibt³⁹⁶.

5.4.1.2.2. Der Begriff „*derived work*“

Ein anderes Problem, das in der Praxis häufig auftaucht, betrifft die Eigenständigkeit eines Werks. Wie bereits erwähnt³⁹⁷ sind die Übergänge zwischen einer Gehilfenstellung einer Teil- und einer Miturheberschaft fließend und nur im Einzelfall zu beurteilen. Dennoch sind ein paar wesentliche Grundsätze hervorzuheben.

Im zweiten Hauptsatz von Ziffer 2 GPL 2 heißt es: *„If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works“*. Demnach ist es für die GPL 2 nicht ausreichend, dass einzelne Softwarebestandteile inhaltlich selbständig sind, denn auch deren Weitergabe muss als eigenständiges Werk erfolgen³⁹⁸. Diese Intention wird durch den dritten Hauptsatz der Ziffer 2 GPL 2 unterstrichen: *„Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.“*

³⁹⁶Die Konsequenzen bei einer Verletzung der Offenlegungspflichten iSd Ziffer 2 GPL betreffend werden im Rahmen der Ziffer 4 GPL behandelt.

³⁹⁷Siehe Kap 3.2.2.2.

³⁹⁸Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 67; mit einer weiterführenden Kritik an der Widersprüchlichkeit der GPL im Hinblick auf die unterschiedlichen Definitionen zum eigenständigen Werksbegriff.

Kriterien, durch die – je nach Art der Verbreitung – eine Abgrenzung erfolgen kann, haben einerseits inhaltliche andererseits formale Voraussetzungen³⁹⁹. Diese sollten bei der Entscheidung, ob ein Softwaremodul als eigenständig gelten kann, herangezogen werden, da die gesetzlichen Vorgaben zu einer Bearbeitung iSd § 5 UrhG iVm § 14 Abs 2 UrhG und § 40d UrhG keine rechtliche Relevanz in der Tatsache sehen, ob einzelne Bestandteile gemeinsam oder getrennt verbreitet werden. Eigenständigkeit wird bspw immer dann anzunehmen sein, wenn zwar die Algorithmen einer Neuimplementierung auf der Grundlage der unter der GPL 2 lizenzierten Software entwickelt wurden, der Quelltext dabei jedoch nicht übernommen bzw kopiert wurde.

Ziffer 3 GPL 2 behandelt den Umgang mit „Binaries“⁴⁰⁰ und räumt dem Lizenznehmer die gleichen Rechte im Umgang mit dem Programm als Binary ein wie die Ziffern 1 und 2 GPL 2 dies für die Programmversionen tun, die im Original oder als Weiterentwicklung vorliegen. Ziffer 3 gleicht Ziffer 1 Abs 1 GPL 2, welche die parallelen Rechte zum Vervielfältigen und Verbreiten des Quellcodes gewährt⁴⁰¹.

Da der Freiheitsgedanke, der hinter Open Source steht, einschließt, dass der Quellcode für den Lizenznehmer ebenfalls nutz- und einsehbar ist, werden in den weiterführenden Pflichten, die Ziffer 3 GPL 2 vorschreibt, Kriterien aufgestellt, die in technischer Hinsicht von Belang sind. Bei der Weitergabe des

³⁹⁹ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 67; zu einer „wertenden Gesamtbetrachtung“ siehe Jaeger/Metzger, Open Source Software 42, sowie Spindler, Rechtsfragen der Open Source Software 56.

⁴⁰⁰ Eine sogenannte Binärdatei (engl. binary file) enthält im Unterschied zu einer reinen Textdatei auch nicht-alphabetische Zeichen. Binärdateien werden im Allgemeinen meist dazu verwendet, Daten abzuspeichern. Grundsätzlich zählen all jene Dateien zu Binärys, die keinen lesbaren Text enthalten, wobei für die häufigsten der heute verwendeten 8-Bit-Zeichensätze gilt: nicht lesbare Sonderzeichen sind Zeichen mit ASCII-Werten von 0 bis 31 und lesbare Zeichen sind Zeichen mit Werten von 32 bis 126. Die Lesbarkeit von Zeichen mit den darüber hinausgehenden Werten ab 127 ist abhängig vom verwendeten Zeichensatz.

⁴⁰¹ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 79.

Programms als Binary kann der Lizenznehmer aus folgenden Möglichkeiten auswählen:

- a) Angebot der Mitgabe des Quellcodes auf einem handelsüblichen Datenträger⁴⁰²,
- b) Mitgabe eines schriftlichen Angebots, den Quellcode innerhalb eines Zeitraums von mindestens drei Jahren zum Selbstkostenpreis auszuliefern (wiederum mittels handelsüblichen Datenträger),
- c) unter Hinweis auf ein in Punkt b) genanntes Angebot eines Dritten ist die Verbreitung in nicht-kommerzieller Form gestattet,
- d) Verfügbarkeit des Quellcodes durch Nutzung eines Datennetzes (vorrangig Internet) von derselben Stelle aus, von der auch die Binary bezogen werden kann.

Während die ersten beiden Wahlmöglichkeiten jedem Lizenznehmer stets offen stehen, kann er die dritte nur nutzen, wenn er sie nicht-kommerziell verwendet und er selbst iSd lit b) ein schriftliches Angebot erhalten hat. Bei der letzten Variante kann der Vertrieb der Binary nur online erfolgen.

Der dritte Absatz der Ziffer 3 GPL 2 definiert schließlich, was der anzubietende Quellcode enthalten muss. Dabei ist zu berücksichtigen, dass sich die in Ziffer 3 GPL 2 gemachten Ausführungen nicht auf die Verbreitung des Quellcodes nach Ziffer 1 GPL 2 beziehen, da dort bestimmt wird, dass der Quellcode nur in der Form weiterzugeben ist, die man selbst erhalten hat⁴⁰³.

⁴⁰² Als handelsübliche Datenträger müssen wohl 3,5 Zoll Disketten, CD, Flash EEPROM (USB-Stick) und DVD angesehen werden, da diese üblicherweise für den Austausch von Software verwendet werden. Dies immer in Hinblick auf den Umfang des Quellcodes.

⁴⁰³ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 85.

Nachfolgende Abbildung gibt einen Überblick, in welchen Ziffern der GPL 2 die unterschiedlichen Nutzungsrechte behandelt werden:



Werk	Quelltext	Binärdatei
Original	Ziffer 1 GPL	Ziffer 3 + 1 GPL
Weiterentwicklung	Ziffer 2 GPL	Ziffer 3 + 2 GPL

Abb 6⁴⁰⁴

Ziffer 4 GPL 2 enthält Regeln darüber, welche Folgen ein Verstoß gegen die Lizenzbestimmungen der GPL 2 hat. Die Bestimmung der Ziffer 4 GPL 2 entspricht dabei im österreichischen Recht im Wesentlichen § 14 Abs 2 UrhG. Sie stellt klar, dass ein Lizenznehmer ein unter der GPL 2 lizenziertes Programm nicht vervielfältigen, verändern, weiter lizenzieren oder verbreiten darf, sofern es nicht durch die GPL 2 ausdrücklich gestattet ist. Des Weiteren ist Ziffer 4 GPL 2 in Einklang mit § 40c UrhG zu sehen, der über Computerprogramme aussagt, dass Werknutzungsrechte an diesen ohne Einwilligung nur auf einen anderen übertragen werden dürfen, wenn mit dem Urheber nichts anderes vereinbart worden ist. Das ist im Falle einer Unterlizenzierung von einem unter der GPL 2 lizenzierten Programm von wesentlicher Bedeutung, da dies nach Ziffer 4 nur dann gestattet ist, wenn die teilweise bereits angeführten Bedingungen beachtet werden.

Die Folgen eines Verstoßes werden ebenfalls in Ziffer 4 GPL 2 behandelt. Die rechtliche Qualifizierung hierbei ist interessant, da die GPL 2 selbst keine rechtliche Konstruktion vorgibt⁴⁰⁵. Nach österreichischem Recht wird darin eine auflösende Bedingung mit dinglicher Wirkung ex nunc mit Rücksichtnahme auf §§ 696 und 704 ABGB zu sehen sein⁴⁰⁶. Dem bisherigen Lizenznehmer werden die Rechte, die er vorher aufgrund des Vertragsabschlusses mit den Lizenzgebern durch die GPL 2 erhalten hatte,

⁴⁰⁴ Siehe auch: *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 41.

⁴⁰⁵ Für das dt Recht siehe: *Spindler/Wiebe*, CR 2003, 874; *Jaeger/Metzger*, Open Source Software 38; *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 87.

⁴⁰⁶ *Wiebe/Prändl*, ÖJZ 2004, 631.

entzogen⁴⁰⁷. Wird Ziffer 4 GPL 2 von Unternehmen oder Behörden⁴⁰⁸ – bspw aus Erwägungen der Vertraulichkeit – verletzt, verliert das Unternehmen oder die Behörde nach § 4 GPL 2 das Recht zur Nutzung der unter der GPL 2 lizenzierten Software⁴⁰⁹.

Im Gegensatz dazu entstehen iSd Ziffer 4 Satz 3 einem Dritten keine unmittelbaren Beschränkungen, da dieser gem Ziffer 6 GPL 2 die Nutzungsrechte stets unmittelbar vom Lizenzgeber bzw Rechteinhaber erwirbt.

In Hinblick auf § 21 Abs 3 UrhG ist anzuführen, dass eine Entstellung oder Verstümmelung des Werks zwar keine Verletzung iSd Ziffer 4 GPL 2 darstellen kann, der Gesetzgeber hier aber dem Urheber weiterreichende Rechte zum Einschreiten einräumt. Dies muss von Fall zu Fall entschieden werden, da die subjektive Ansicht über die Qualität einer Fortentwicklung eines Programms von Programmierer zu Programmierer anders sein kann⁴¹⁰.

Die Folgen aus einer Verletzung iSd Ziffer 4 GPL 2 finden sich vor allem im Unterlassungsanspruch des § 81 UrhG, den Schadenersatzansprüchen des § 87 UrhG und den strafrechtlichen Bestimmungen des § 91 UrhG. Allerdings ist eine Verletzung gegen die Bestimmungen der GPL 2 kein Ausschließungsgrund auf Lebenszeit für den „Lizenzverletzer“. Dieser kann sein Vergehen „heilen“, sobald er den ex nunc nichtig gewordenen Vertrag neu annimmt und die Lizenzbestimmungen einhält, indem er bspw ein Programm, welches er vorher nicht unter der GPL 2 lizenziert hatte, nun unter dieser anbietet⁴¹¹.

⁴⁰⁷ Die Wirksamkeit der Ziffer 4 GPL wurde weltweit erstmalig durch das LG München (Urteil vom 19.05.2004, Az 21 O 6123/04) bestätigt.

⁴⁰⁸ Vgl Kap 5.4.1.2.1.

⁴⁰⁹ *Andréewitch*, MR 2005, 38.

⁴¹⁰ Siehe auch *Küng*, MR 2004, 27.

⁴¹¹ *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 94.

Ziffer 5 GPL 2 regelt die Annahme des an jedermann gerichteten Lizenzvertrags durch den Lizenznehmer iSd § 861 ABGB⁴¹². Das Angebot des Lizenzgebers muss nicht mehr ausdrücklich abgewartet oder angefordert werden. Es liegt in dem Moment vor, in dem sich der Lizenzgeber entschließt, sein Programm unter die GPL 2 zu stellen und zu verbreiten und damit ein Angebot zur Vertragsannahme an jedermann richtet.

Gem Ziffer 6 GPL 2 kommt es bei der Weitergabe des Lizenzvertrags zum unmittelbaren Erwerb der Nutzungsrechte vom Urheber⁴¹³. Von der GPL 2 wird demnach kein Kettenlizenzverhältnis gewünscht, stattdessen kommt bei jeder neuerlichen Verbreitung ein neuer Lizenzvertrag zwischen dem neuen Lizenznehmer und dem Urheber oder den Miturhebern zustande. Der Lizenznehmer darf gem Ziffer 6 GPL 2 mit Nutzern, die GPL 2-lizenzierte Software über ihn beziehen, keine Vertragsbedingungen vereinbaren, welche die durch die Lizenz zugestandenen Rechte einschränken würden. Somit wird sichergestellt, dass der Lizenznehmer keine Nutzungsrechte an Dritte einräumen kann, nicht einmal aus dem einfachen Nutzungsrecht heraus, welches er selbst erworben hat⁴¹⁴. Der Lizenznehmer hat hier also nur eine Botenstellung⁴¹⁵.

Unter Berücksichtigung der Philosophie von Open Source wird diese Vorschrift verständlich. Eine unproblematische Weitergabe der Software kann nur durch die direkte Vertragsstellung zwischen Urheber und Lizenznehmer gewährleistet werden, da eine Kettenlizenzierung schwerwiegende Probleme mit sich bringen könnte. Mit ursächlich hierfür ist, dass Kettenlizenzierungen davon abhängig sind, ob alle Kettenglieder wirksam ihre Rechte (hier die

⁴¹² Zum Vertragsabschluss im Open Source Bereich im Allgemeinen vgl Kap 4.1.2; zu Ziffer 5 GPL im Speziellen siehe Kap 4.1.2.1.1.

⁴¹³ Siehe auch: *Wiebe/Prändl*, ÖJZ 2004, 631; *Omsels* in FS *Hertin* 169; *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 103.

⁴¹⁴ *Omsels* in FS *Hertin* 169.

⁴¹⁵ *Wiebe/Prändl*, ÖJZ 2004, 631.

Nutzungsrechte) weitergeben können. Fehlt es einem Kettenglied bspw an der Geschäftsfähigkeit, sind alle folgenden Unterlizenzierungen nichtig⁴¹⁶. Ebenso problematisch wäre für den Urheber die Möglichkeit, die in der GPL 2 erwähnten schuldrechtlichen Verpflichtungen bei einer Verletzung durchzusetzen, da er sich bei einer Kettenlizenzierung stets an seinen direkten Vertragspartner wenden und diesen dazu anhalten könnte, die Durchsetzung zu veranlassen⁴¹⁷. Diese Problematik wird im letzten Satz der Ziffer 6 GPL 2 klar angesprochen, die bestimmt, dass ein Lizenznehmer nicht für die Durchsetzung der GPL 2 gegenüber Dritten verantwortlich ist.

Ziffer 7 ist eine „Kollisionsklausel“, da sie Fälle behandelt, bei denen einem Lizenznehmer durch einen Gerichtsbeschluss, Vergleich (gerichtlich oder außergerichtlich) oder durch einen sonstigen Akt Bedingungen auferlegt werden, die jenen der GPL 2 widersprechen. Sollte dies der Fall sein, verliert der Lizenznehmer das Recht, das Programm zu verbreiten. Ziffer 7 GPL 2 schützt damit die Freiheit eines Programms, wobei es keine Rücksicht auf die Gründe nimmt, die einen Lizenznehmer dazu verleiten könnten, ob gewollt oder ungewollt, die Rechte, die in der GPL 2 beschrieben werden, einzuschränken. Des Weiteren nimmt die GPL 2 dadurch Rücksicht auf den jeweils herrschenden nationalen Rechtsschutz, der einem Lizenznehmer - bspw die Verbreitung eines bestimmten Programmteils - nur unter Abgabe von Lizenzgebühren erlauben könnte.

Die salvatorische Klausel der Ziffer 7 zweiter Absatz GPL 2 ist bisher nur von einem dt Gericht behandelt und für unbedeutend erachtet worden⁴¹⁸. In Österreich wäre ein Einschnitt in Ziffer 7 GPL 2 ebenfalls kaum zu rechtfertigen, da die Intention dieses Lizenzbestandteils in Einklang mit der österreichischen Gesetzgebung, Judikatur und Lehre steht und eine anderweitige Auslegung

⁴¹⁶ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 105.

⁴¹⁷ Zur Problematik der „Durcharbeitung“ einer solchen Kette siehe: *Omsels* in FS *Hertin* 169.

⁴¹⁸ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 113; LG München I, Urteil vom 19.05.2004, Az 21 O 6123/04.

kaum Rückhalt finden dürfte. Auch die darin getroffene Entscheidung, Open Source Software in dem Sinne zu schützen, dass unter den in Ziffer 7 GPL 2 angesprochenen Umständen eventuell die Verbreitung eines Programms untersagt werden muss.

Ziffer 8 GPL 2 legt Ausnahmen fest, unter denen eine Einschränkung der Rechteeinräumung iSd GPL 2 möglich ist. Demnach ist dies für den geografischen Geltungsbereich möglich, „*if the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces*“ Die Klausel ist demnach konditional zu sehen, da sie schon mit „*if*“ beginnt. Nur „wenn“ die in Ziffer 8 GPL 2 genannten Hindernisse in Bezug auf Verbreitung und jegliche Art der Verwendung⁴¹⁹ auftreten, ist es dem ursprünglichen Rechteinhaber, der das Programm unter die GPL 2 gestellt hat, erlaubt, eine räumliche Beschränkung aufzunehmen. Einem „reinen“ Lizenznehmer, der das Programm weiterentwickelt, ist dies hingegen nicht gestattet. Ziffer 8 letzter Satz GPL 2 räumt dem ursprünglichen Rechteinhaber die Genehmigung ein, die von ihm in zulässigerweise Weise eingefügte räumliche Beschränkung der Nutzungsrechte zum Bestandteil der GPL 2 werden zu lassen. Diese Einräumung ist einmal in Hinblick auf § 14 Abs 2 UrhG wichtig zum ändern wird auch im Header der GPL 2 eine Veränderung des Lizenztextes untersagt.

Die Frage, ab wann ein Programm hinsichtlich seiner Verbreitung und Verwendung durch Patente oder durch urheberrechtlich geschützte Schnittstellen eingeschränkt ist, lässt sich nur schwer beantworten. Hinsichtlich bestehender Handelsembargos ist auf die Free Software Definition zu verweisen, die besagt, dass der ursprüngliche Rechteinhaber die Verbreitung eines Programms nicht aufgrund gesetzlicher Verwertungshindernisse durch

⁴¹⁹ Ausführlich dazu Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 115.

Änderung der GPL 2 einschränken darf⁴²⁰. Generell wird davon auszugehen sein, dass dem ursprünglichen Rechteinhaber die Möglichkeit gegeben werden soll, ihm im Vorhinein bekannte Verwertungshindernisse umgehen zu können⁴²¹, ohne ihn dabei von den generellen Pflichten der GPL 2 zu entbinden. Aus diesem Grund wird das Recht zur Änderung nur dem ursprünglichen Rechteinhaber vorbehalten, nicht aber einem späteren Lizenznehmer.

Ziffer 9 GPL 2 regelt zum einen die Lizenzhoheit der FSF und zum anderen den Umgang und die Anwendung verschiedener Lizenzversionen. Wie erwähnt, wurde seit Anfang 2006 die Version 3 der GPL diskutiert, die 2007 dann offiziell durch die FSF herausgegeben wurde⁴²².

Ziffer 10 GPL 2 spricht die Lizenzkompatibilität der GPL 2 mit anderen Lizenzen an. Die damit verbundenen Fälle des „dual licensing“ werden in einem folgenden Teil der Arbeit besprochen⁴²³, da Doppellizenzierungen alle Open Source Lizenzen betreffen. Ziffer 10 GPL 2 hat die Intention Lizenznehmer, die Teile eines Programms in einem anderen Open Source Programm verwenden wollen, darauf hinzuweisen, dass dies nur möglich ist, wenn dafür die Erlaubnis des Rechteinhabers – also des Urhebers oder eines Dritten, dem die Rechte übertragen wurden – vorliegt.

Aus gutem Grund weist die FSF in Satz 2 der Ziffer 10 GPL 2 auf sich selbst als Rechteinhaber hin: In den USA kann das Copyright durch „Copyright Assignments“⁴²⁴ auf die FSF übertragen werden, was in Österreich der Wirkung

⁴²⁰ <http://www.gnu.org/philosophy/free-sw.html>; *Deike*, Open Source Software: IPR-Fragen und Einordnung ins deutsche Rechtssystem, CR 1/2003, 9 (11); *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 116.

⁴²¹ Vgl. *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 114, siehe dort ebenfalls zu der Auswirkung einer ungerechtfertigten Änderung.

⁴²² <http://gplv3.fsf.org/draft>; <http://www.fsf.org/news/gplv3launch>; <http://www.heise.de/newsticker/meldung/65576>.

⁴²³ Siehe Kap 5.10.

⁴²⁴ <http://www.fsf.org/licensing/licenses/why-assign.html>.

nach einer Übertragung des vollumfänglichen exklusiven Verwertungsbefugnisses entspricht⁴²⁵. Die FSF nimmt diese Rechte dann treuhänderisch für die Rechteinhaber wahr. Für den kontinentaleuropäischen Raum sind solche „Copyright Assignments“ unbrauchbar, da die vom französischen „Droit-d’Auteur-System“ beeinflussten Urheberrechtsordnungen die Urheberpersönlichkeitsrechte hervorheben und daher das Urheberrecht nicht übertragbar ist⁴²⁶. Die FSF Europe bietet daher unter Beachtung dieser Unterschiede die treuhänderische Lizenzvereinbarung (FLA) als europäischen Ansatz an, der vom Prinzip her dem „Copyright Assignments“ sehr ähnlich ist und die FSF Europe zum Treuhänder macht⁴²⁷.

Das FLA geht dabei sowohl auf Rechtsordnungen ein, in denen eine uneingeschränkte Übertragung möglich ist (§ 1 FLA) als auch auf solche, in denen dies keine rechtliche Deckung findet (§ 2 FLA). Die Aktivlegitimation betreffend hat sich das FLA dafür entschieden, dem Treuhänder, also der FSF Europe, die umfassenden Nutzungsrechte an der Software zu übertragen, damit kein Zweifel daran besteht, dass die FSF Europe eigene Rechte wahrnimmt. Der Rechteinhaber soll dadurch aber keinesfalls seiner Verwertungsrechte beraubt werden, es soll nur klargestellt werden, dass der FSF Europe eine ausreichende Bevollmächtigung zur Prozessführung zukommt. § 2 FLA listet einerseits taxativ die Nutzungsrechte auf, die übertragen werden, andererseits werden durch § 5 FLA eine unbegrenzte Anzahl an einfachen Nutzungsrechten zurückübertragen, wonach auch hier die Möglichkeit zum „dual licensing“ offen bleibt⁴²⁸.

Ziffer 11 und 12 GPL 2 regeln die Haftung und Gewährleistung. Da diese Klauseln in vielen Open Source Lizenzen vorkommen und eine wichtige Rolle

⁴²⁵ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 133.

⁴²⁶ Vgl Kap 2.4.2.

⁴²⁷ <http://www.germany.fsf europe.org/projects/fla/fla.de.html>. Metzger, Das FLA - Die treuhänderische Lizenzvereinbarung der FSF Europe, Linux-Magazin 5/2003, 75 (76).

⁴²⁸ Metzger, Linux-Magazin 5/2003, 77.

beim Umgang mit Open Source Software spielen, werden sie in einem eigenen Kapitel an anderer Stelle gemeinsam besprochen⁴²⁹.

Abgeschlossen wird die GPL 2 mit einem Anhang, der praktische Hinweise für den Rechteinhaber enthält, wie er sein Programm am besten unter der GPL 2 lizenzieren kann und was er dabei zu beachten hat.

5.4.1.3. Kompatibilität

Die Kompatibilität der GPL 2 mit anderen Open Source Lizenzen führt immer wieder zu Streitigkeiten innerhalb der Open Source Gemeinschaft. Die geführten Diskussionen unterstreichen allerdings die herausragende Rolle der GPL 2, da die meisten Open Source Lizenzen die Kompatibilität mit der GPL 2 suchen und teilweise sogar, wie bspw die originale BSD-Lizenz oder die Zope Public License, ihre Lizenztexte in veränderter Form herausgeben, um eine Kompatibilität zu erreichen. Neben der Kompatibilität mit anderen Open Source Lizenzen ist es wichtig, die Kompatibilität der GPL 2 zu proprietärer Software zu betrachten.

5.4.1.3.1. Kompatibilität mit anderen Open Source Lizenzen

Wie bereits bei den BSDartigen Lizenzen erwähnt gibt es bei der Eingliederung von Programmen, die unter einer BSDartigen Lizenz stehen, nur wenige Hindernisse. Da die BSDartigen Lizenzen dem Lizenznehmer große Freiheiten einräumen, Weiterentwicklungen unter verschiedene Lizenztypen zu stellen, die nicht ausschließlich aus dem Open Source Bereich kommen

⁴²⁹ Vgl Kap 6.

müssen, muss hinsichtlich der GPL 2 darauf geachtet werden, dass die in einer BSDartigen Lizenz einem Lizenznehmer auferlegten Pflichten nicht mit Ziffer 6 GPL 2 kollidieren. Dieser besagt „*You may not impose any further restrictions on the recipients` exercise of the rights granted herein*“ und verbietet damit jedem Lizenznehmer bei der Weitergabe von GPL 2 lizenzierte Software zusätzliche Bedingungen aufzustellen. Dadurch sind Lizenzen, die eine „Werbeklausel“ enthalten⁴³⁰, wie bspw die originale BSD-Lizenz und die Apache Software License v1.0, mit der GPL 2 inkompatibel.

Hinsichtlich der Kompatibilität mit anderen „Copyleft“-Lizenzen muss auf eine ausdrückliche Regelung im Lizenztext geachtet werden⁴³¹, da der Schutz des „Copyleft“ eine andere Lizenzwahl grundsätzlich ausschließt. Nur wenn eine Lizenz eine „devote Klausel“ zu Gunsten der GPL 2 enthält (die Lizenz tritt in diesem Fall zu Gunsten der GPL 2 zurück), liegt Kompatibilität vor und ein Programm oder Programmbestandteil kann vollständig unter die GPL 2 (oder gegebenenfalls auch eine andere Open Source Lizenz) gestellt werden.

5.4.1.3.2. Kompatibilität mit proprietären Lizenzen

Die Verbindung von unter der GPL 2 lizenzierte mit proprietärer Software ist für die Wirtschaft von großem Interessen, da es bei einem gemeinsamen Vertrieb dazu kommen könnte, dass das Gesamtwerk unter die GPL 2 gestellt werden muss. Hier wird auf Ziffer 2b GPL 2 verwiesen⁴³², welche die Eigenständigkeit des Werks behandelt. Hinsichtlich der Kompatibilität kommt es darauf an, ob beide Werke getrennt vertrieben werden können⁴³³. Die Kompatibilität ist daher im Einzelfall zu beurteilen, da neben den technischen

⁴³⁰ Vgl Kap 5.2.1.

⁴³¹ Jaeger/Schulz, NOW Gutachten 37.

⁴³² Vgl Kap 5.4.1.2.

⁴³³ Wiebe/Prändl, ÖJZ 2004, 633.

Kriterien auch die Verkehrsanschauung für die Qualifizierung herangezogen werden muss⁴³⁴.

Letztlich bleibt es der persönlichen Ansicht überlassen, bis wohin der Copyleft-Effekt der GPL 2 reicht, da bislang keine Rechtsprechung zu dieser Problematik vorliegt, eine Diskussion über mögliche Fallkonstruktionen aber an dieser Stelle zu weit führen würde⁴³⁵.

5.4.2. Die GNU General Public License 3.0

„The Linux kernel is under the GPL version 2. Not anything else. Some individual files are licenceable under v3, but not the kernel in general.“⁴³⁶

- **Linus Torvalds** über die Chancen Linux unter der GPL 3 zu lizenzieren.

Der aktuellsten Version der GPL ist ein langer Entwicklungsprozess vorausgegangen. In einer demokratischen – fast zweijährigen – weltweiten Entscheidungsfindung wurden die GPL 3 in vier Vorentwürfen diskutiert und am 29.07.2007 in der finalen Version herausgegeben.

5.4.2.1. Hintergrund

Die GPL 2 wurde seit Anfang des Jahres 2000 einer immer kritischeren Diskussion ausgesetzt, da viele Lizenznutzer der Meinung waren, die GPL 2 sei nicht mehr zeitgemäß und daher nicht mehr geeignet, auf aktuelle Entwicklungen in der IT-Branche einzugehen. Diese Sorge wurde von der FSF – den Urhebern der GPL 2 – geteilt, sodass Anfang 2006 schließlich öffentlich

⁴³⁴ Jaeger/Metzger, Open Source Software 42.

⁴³⁵ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 66.

⁴³⁶ <http://www.linux.com/articles/51613>.

über den Lizenz-Update diskutiert wurde. Diese Öffnung der FSF nach außen stellte sich im Endeffekt als positiv heraus, was ein Vergleich der neuen Version gegenüber der alten zeigt. Die nachfolgende Darstellung beschränkt sich nur auf die Erörterung der wesentlichen Neurungen und zeigt kurz die dahinter stehenden Überlegungen auf.

5.4.2.2. Lizenzmerkmale

Die GPL 3 beginnt mit einer Präambel in welcher der erste Satz – „*The GNU General Public License is a free, copyleft license for software and other kinds of work*“ – bereits auf die über Software hinausgehenden Verwendungsmöglichkeiten hinweist. Dieser Hinweis befand sich schon in Punkt 0 der GPL 2 – „*This License applies to any program or other work*“ – durch die Platzierung an den Anfang der GPL 3 wird aber die Intention der FSF deutlich, die GPL nicht nur für Software zu nutzen, sondern zukünftig auch verstärkt für die übrigen in § 1 UrhG geschützten Werke, wie bspw Musik oder Literatur bereitzustellen. Es macht daher Sinn für die GPL 3 in den Fällen von Werken zu sprechen, in denen nicht ausdrücklich auf Computerprogramme Bezug genommen wird oder dies aus dem Text erkenntlich wird.

Die Präambel der GPL 3 stimmt größtenteils mit jener der GPL 2 überein, der sprachlich überarbeitete Text enthält darüber hinaus einen anlassbezogenen Abschnitt, der sich mit Embedded Systems auseinandersetzt und den Fall „TiVo“ behandelt. Der TiVo (Television Input / Video Output) ist ein digitaler Videorekorder, der unter Linux läuft und ermöglicht, Fernsehsignale aufzuzeichnen und bei Bedarf wiederzugeben⁴³⁷. Die Entwickler des TiVo hatten den Quelltext des Programms iSd GPL offen gelegt, das Programm führte aber nur jenen Code aus, der von den Entwicklern digital signiert wurde. Fremde Weiterentwicklungen waren dadurch möglich, konnten jedoch nicht

⁴³⁷ Der Service des TiVo steht bislang nur in den USA und Großbritannien zur Verfügung.

mehr mit der TiVo-Hardware benutzt werden. Dieses Ergebnis verstörte die FSF – insbesondere *Richard Stallman* und *Eben Moglen* – weshalb die GPL 3 nun in der Präambel gegen die Möglichkeit einer „technischen Unterwanderung“ der Lizenz vorgeht. Demnach verstoßen jene Lizenzgeber gegen die GPL 3, deren Geräte so entworfen sind, dass es einem Endbenutzer nicht möglich ist, weiterentwickelte Versionen der darauf laufenden Software zu installieren oder nutzen zu können, während dies dem Hersteller durchaus möglich ist. Die Urheber der GPL 3 haben hier neben dem „TiVo-Fall“ auch an den immer größeren Einsatz von PDAs, Smartphones oder ähnlichem gedacht, die meist mit einem eigenen Betriebssystem ausgeliefert werden und bei denen die Möglichkeit der Weiterentwicklung oder Anpassung der Software nur geringfügig oder gar nicht gegeben ist. Der Lizenzgeber muss hier sicherstellen, dass es dem Lizenznehmer technisch möglich ist, die Rechte, die er durch die GPL 3 erhält, auch tatsächlich umsetzen zu können.

Punkt 0 GPL 3 enthält Begriffsdefinitionen der Lizenz. Das Wort „propagate“ soll stellvertretend für die jeweiligen Begriffe der Nutzungsrechte der nationalen Rechtsordnungen stehen und damit die Auslegungsschwierigkeiten beseitigen, die bei der GPL 2 aufgetreten sind⁴³⁸. Neben dieser begrifflichen Klarstellung verwendet die GPL 3 das Wort „convey“ neu. Der Begriff „convey“ ersetzt damit den aus der GPL 2 bekannten Begriff „distribute“ und bezieht sich auf Vertriebshandlungen der GPL 3⁴³⁹. *Jaeger*⁴⁴⁰ weist darauf hin, dass der Begriff sich nicht auf alle Vertriebshandlungen bezieht, sondern auf jene beschränkt, bei denen der Empfänger eine Kopie der Software erhält. In den Fällen von SaaS entfiel damit die Pflicht, den Quellcode offen zu legen. Dieser Einschätzung ist zuzustimmen, da Punkt 0 GPL in Zusammenhang mit dem Begriff „convey“ deklariert, dass „*mere interaction with a user through a computer network, **with no transfer of a***

⁴³⁸ Vgl Kap 5.4.1. sowie: *Wiebe/Heidinger*, GPL 3.0 und EUPL – Aktuelle Entwicklungen im Bereich der Open Source Lizenzen, MR 2006, 258.

⁴³⁹ *Jaeger*, Version 3 der GPL fertiggestellt, c` t 2007 Heft 15, 38.

⁴⁴⁰ *Jaeger*, c` t 2007 Heft 15, 38.

copy, is not conveying". Die reine Interaktion über ein Intra- oder über das Internet befreien einen Lizenzgeber von der Pflicht, den Quellcode zugänglich zu machen.

Punkt 1 GPL 3 behandelt den Sourcecode und entspricht damit wesentlich Punkt 1 GPL 2, auch wenn die Definition des Sourcecodes in der GPL 3 wesentlich genauer vorgenommen wurde, um Interpretationsschwierigkeiten entgegenzuwirken. Punkt 2 GPL 3 betrifft die grundlegenden Rechte und entspricht damit ebenfalls Punkt 2 GPL 2.

Punkt 3 GPL 3 behandelt den Schutz der Rechte des Lizenznehmers vor „Umgehungsverbotgesetzen“ und war in der Diskussionsphase heftig umstritten, da die ersten Lizenzvorschläge dezidiert auf den US-amerikanischen „Digital Millenium Copyright Act“ (DMCA) eingingen, was vielen Kritikern zu eng gefasst erschien⁴⁴¹. Die Behandlung des „Digital Rights Managements“ (DRM) wurde dennoch nicht ersatzlos gestrichen, sondern lediglich internationaler gestaltet. Gem Punkt 3 GPL 3 darf kein unter der GPL 3 lizenziertes Werk den Schutz gegen die Umgehung technischer Schutzmaßnahmen iSd „WIPO Copyright Treaty“⁴⁴² (WCT) oder vergleichbarer Normen für sich beanspruchen. Die GPL 3 betont dies besonders, indem sie in Punkt 3 Satz 2 GPL festlegt, dass ein Lizenzgeber auf das Recht, die Umgehung technischer Schutzmaßnahmen zu verbieten, ausdrücklich verzichtet. Der Grund für diesen Einschub sieht *Jaeger* darin, dass die Möglichkeit des „Reengineering“ erhalten bleibt und durch diese Klausel geschützt werden soll⁴⁴³. Tatsächlich benutzt die FSF den Terminus „DRM“ als „Digital Restrictions Management“ und hat diesem Thema eine eigene Kampagne gewidmet⁴⁴⁴, in der zum Ausdruck

⁴⁴¹ Siehe *Jaeger*, c` t 2007 Heft 15, 38, der die Kritikfähigkeit der FSF diesbezüglich lobt und: *Wiebe/Heidinger*, MR 2006, 258, welche eine neutralere Formulierung forderten.

⁴⁴² <http://www.wipo.int/treaties/en/ip/wct>.

⁴⁴³ *Jaeger*, c` t 2007 Heft 15, 39.

⁴⁴⁴ <http://www.fsf.org/campaigns/drm.html>.

kommt, dass die FSF die Gesetzgebung zu DRM-Maßnahmen als große Gefahr für die Freiheit von Werken ansieht.

Punkt 4, 5 und 6 GPL 3 entsprechen Punkt 3 GPL 2. Die Aufteilung in drei unterschiedliche Punkte erscheint sinnvoll, da nach der neuen Systematik einerseits auf die lizenzrechtliche Behandlung von unveränderten Kopien eingegangen (Punkt 4 GPL 3) und andererseits die Übertragung weiterentwickelter Quelltext- (Punkt 5 GPL 3) und Nicht-Quelltextversionen (Punkt 6 GPL3) geregelt wird. Inhaltlich führen die Punkte 4 und 5 GPL 3 schon das in der GPL 2 Gesagte aus, weshalb sich hier nur kleine – und für die Praxis unbedeutende – Änderungen ergeben. Hinsichtlich der Ausführungen zu Punkt 6 GPL 3 fällt auf, dass hier ein „Update“ auf die aktuellen technischen Gegebenheiten stattgefunden hat. Programme, die in Nicht-Quelltext-Form weitergegeben werden, müssen den maschinenlesbaren korrespondierenden Quelltext unter bestimmten Bedingungen bereitstellen. Zu diesen Bedingungen zählen bspw die Übertragung des Quelltextes gemeinsam mit dem Objektcode, wobei sich dieser *„fixed on a durable physical medium customarily used for software interchange“* befinden muss. Die Übertragung auf einem USB-Stick oder einer SD-Speicherkarte ist demnach ebenso möglich wie die Übergabe einer CD-ROM oder DVD-ROM, da diese unveränderliche, haltbare, physikalische Speichermedien darstellen, die üblicherweise für den Austausch von Software verwendet werden. Gegen Punkt 6a GPL 3 verstoßen würde ein Speicherträger, der sich nach einer bestimmten Zeit selbständig löscht oder codiert oder, der so selten ist, dass der Lizenznehmer erst ein bestimmtes Lesegerät oder eine spezielle Software erwerben muss, um den Quelltext einsehen und verwenden zu können. Darüber hinaus kann der Quelltext mit einem schriftlichen Angebot bereitgestellt werden, dass mindestens drei Jahre bzw solange gültig ist, solange Ersatzteile und Support angeboten werden und dessen Bereitstellung keine höheren Kosten verursachen darf, als dies durch den physikalischen Vorgang der Übertragung des Quelltextes begründbar wäre. Ein ebenfalls als Bezugsmöglichkeit angegebene Download muss jedoch kostenfrei erfolgen. *Wiebe/Heidinger* weisen hier auf die Möglichkeit eines

Verstoßes durch das Bereitstellen einer nur niedrigen Bandbreite bei dem Bezug des Quelltextes hin. Diesen Bedenken kann mE dann gefolgt werden, wenn der Quelltext eine bestimmte Größe erreicht hat, welche die Downloadzeit wesentlich verlängert, da eine Downloadzeit von über einer Stunde für einen 10 MB großen Quelltext bspw inakzeptabel erscheint. Ebenso muss auf das zu Punkt 6a GPL 3 Ausgeführte verwiesen und bezüglich des Download ergänzt werden, dass die Bereitstellung des Quelltextes auf einem kostenlosen, aber registrierungspflichtigen Server, mE als Verstoß gegen Punkt 6b GPL 3 gewertet werden muss. Letztendlich geht Punkt 6e GPL 3 auf die Möglichkeit der Bereitstellung des Quelltextes über ein Peer-to-Peer-Netzwerk – File-Sharing-Netzwerke nutzen diese Möglichkeit besonders – ein. Die Möglichkeit der Bereitstellung ist gem Punkt 6e GPL gegeben, wenn die Teilnehmer des Netzwerks darüber informiert werden, wo der Objektcode und der korrespondierende Quelltext des Programms zu finden sind.

Ein weiterer interessanter Aspekt befindet sich im zweiten Teil von Punkt 6 GPL 3. Die Lizenz geht hier explizit auf die Rechte ein, die einem Lizenznehmer eines „User Products“ eingeräumt werden müssen. Dieser Verweis auf die besonderen Rechte des Endverbrauchers muss unter dem gesetzlichen Hintergrund des US-amerikanischen Rechts betrachtet werden. Dort regelt § 2301 des 15. Titel des „United States Code“⁴⁴⁵ – auch bekannt als „Magnuson-Moss Warranty Act“ – die Gewährleistungsrechte bei Verbrauchergeschäften, die EUR 25,- übersteigen. Ursprünglich war nur ein Verweis auf diese Norm vorgesehen, im Zuge der „Internationalisierung“ der Lizenz wurde jedoch dem Verbraucherprodukt ein eigener Abschnitt gewidmet. Der wichtigste Teil des Abschnitts behandelt den Umgang mit „Installation Information“, welche sich auf die Möglichkeit beziehen, weiterentwickelte Versionen eines Programms auf einem Endbenutzerprodukt zu installieren und auszuführen⁴⁴⁶. Dabei muss zwischen Bugs bzw Interaktionsfehlern und gewollten Ausführungsblockaden unterschieden werden. Dem Lizenzgeber ist

⁴⁴⁵ http://www.law.cornell.edu/uscode/html/uscode15/usc_sec_15_00002301----000-.html.

⁴⁴⁶ Vgl die schon in der Präambel der GPL 3 ausgeführten Hinweise dazu.

es untersagt, Ausführungsblockaden in dem Programm einzubauen, die einen gewünschten Nutzen des Endverbrauchers durch das Installieren einer weiterentwickelten Version verhindern. Daher kann er selbstverständlich nicht für die Gewährleistung der einwandfreien Interaktion der weiterentwickelten Software herangezogen werden. Reagiert ein Programm nicht wunschgemäß, muss überprüft werden, ob der Grund dafür bspw in einer veralteten oder inkompatiblen Schnittstelle zu sehen ist oder aufgrund einer vom Lizenzgeber eingebauten „Sperrung“. Sollte der Grund in einer Sperrung liegen, muss der Lizenzgeber die Informationen zur Beseitigung oder Aufhebung derselben liefern.

Punkt 7 GPL 3 regelt die Zulässigkeit zusätzlicher Bedingungen und inwiefern diese mit der GPL 3 konform sind. Wichtig hierbei ist die Erlaubnis, gem Punkt 7a GPL 3 abweichende Gewährleistungs- oder Haftungsbegrenzungen für das Werk zu nutzen. Die Prüfung der Haftungs- und Gewährleistungsklauseln, wie in Kapitel 6 ausgeführt, kann dadurch wesentlich vereinfacht werden, indem die Klauseln dem nationalen Recht angepasst werden. Ebenso können die Bestimmungen einer Markenlizenz, die von der GPL getrennt vereinbart wird⁴⁴⁷, durch eine zusätzliche Bedingung geregelt werden. Ob diese Möglichkeiten ausreichen, um die von der FSF gewünschten Kompatibilitäten zu anderen Lizenzen herzustellen, wird in Kap 5.4.2.3. überprüft.

Punkt 8 GPL 3 behandelt die Kündigung der Lizenz. Diese enthält eine „automatic termination“-Klausel, nach der die Lizenzrechte bei einer ungerechtfertigten Verbreitung oder Weiterentwicklung automatisch ex nunc entzogen werden. Die Kündigung ist schwebend konzipiert und kann vorübergehend oder dauerhaft geheilt werden. Eine vorübergehende Heilung findet statt, indem der Lizenznehmer alle GPL 3-widrigen Handlungen einstellt und ein Lizenzgeber dem Lizenznehmer nicht ausdrücklich und endgültig

⁴⁴⁷ Siehe Kap 7.

kündigt. Eine dauerhafte Heilung findet statt, wenn ein Lizenzgeber es verabsäumt, den Lizenznehmer innerhalb von 60 Tagen ab Beendigung der lizenzverletzenden Handlung auf diese sinnvoll hinzuweisen. Ein sinnvoller Hinweis wird einerseits die genaue Beschreibung der Lizenzverletzung sein und andererseits in die Sphäre des Lizenznehmers gelangen. Findet eine Kündigung statt, muss der Lizenznehmer eine neue Lizenz abschließen und nachweisen, dass er alle lizenzverletzenden Handlungen eingestellt hat. Die Durchführbarkeit dieser Handhabung in der Praxis erscheint fraglich, da die Kenntnis über Lizenzverstöße für die meisten Lizenzgeber im Verborgenen bleibt. Punkt 8 GPL 3 behandelt iVm Punkt 10 GPL 3 das Problem der Lizenzkette. Durch die „automatic termination“-Klausel wird eine Erschöpfung des Verbreitungsrechts verhindert⁴⁴⁸. Eine Lizenzkündigung befreit daher weitere Lizenznehmer, die in einem Lizenzkettenverhältnis zum Lizenzverletzer stehen, nicht von der Einhaltung der GPL 3. Andererseits müssen diese keine neuen Lizenzverträge abschließen, um einen rechtmäßigen Vertrieb weiterführen zu können.

Punkt 9 GPL 3 entspricht Punkt 5 GPL 2, Punkt 10 GPL 3 führt noch einmal das in Punkt 8 GPL 3 zu den Kettenlizenzverhältnissen Angesprochene näher aus und entspricht damit Punkt 6 GPL 2.

Punkt 11 GPL 3 behandelt ein Reizthema der Open Source Gemeinde, die Patente. Für große Aufregung während des Entscheidungsprozesses der GPL 3 sorgte die Neuigkeit, dass Microsoft und Novell⁴⁴⁹ ein Abkommen⁴⁵⁰ abgeschlossen haben, in welchem vereinbart wurde, keine Patentansprüche gegeneinander oder gegen die Kunden des anderen geltend zu machen. Die FSF wurde daraufhin mit der Sorge konfrontiert, dass die Lizenznehmer von GPL 3-lizenzierter Software in zwei Lager gespalten werden können: In

⁴⁴⁸ http://www.www.ifross.de/ifross_html/home2_2007.html#ARTIKEL26.

⁴⁴⁹ Novell vertreibt die Linuxdistribution „openSUSE“, siehe: <http://www.opensuse.org>.

⁴⁵⁰ <http://www.heise.de/open/artikel/80523/1>.

diejenigen mit Patentschutz und diejenigen ohne⁴⁵¹. Die patentrechtlichen Intentionen der FSF verfolgten jedoch das Ziel, alle Lizenznehmer gleichzustellen und keine Zweiklassengesellschaft einzuführen. Die finale Patentklausel der GPL 3 enthält daher die Bestimmung, dass der Lizenzgeber dem Lizenznehmer eine Verzichtsgarantie abgibt, seine Patentrechte gegen diesen durchzusetzen. Der Lizenzgeber erlaubt dem Lizenznehmer die Weiterentwicklung, Weiterverbreitung und versichert keine „diskriminierenden“ Bestimmungen in die Lizenz mit aufzunehmen. Das Diskriminierungsverbot ist die Antwort der FSF auf das Microsoft-Novell-Abkommen und enthält zu Gunsten von Novell eine „Großvaterklausel“, indem man die Klausel so konstruiert hat, dass sie nur für jene Abkommen gilt, die nach dem 28. März 2007 – das Abkommen zwischen Novell und Microsoft stammt vom 2. November 2006 – geschlossen wurden. Dieser Kompromiss erscheint halbherzig, wenn man bedenkt, dass gegenseitige Freistellungen von Patentansprüchen gängige Praxis sind und es für Nachzügler keine Übergangsfrist gab bzw gibt. Die FSF hat festgestellt, dass die Formulierung des Punkt 11 GPL 3 schwierig war⁴⁵². Es bleibt abzuwarten, ob diese Art der Formulierung eine rechtliche Absicherung gebracht hat. Die Patentproblematik, die in Kapitel 10 extra erörtert wird, hat ebenfalls ihren Einzug in die GPL 3 gefunden, die artikulierte Lösung scheint dem Sinn nach gerechtfertigt, rechtlich wird sie zumindest in den USA einem präjudiziellen Vergleich standhalten müssen.

Punkt 12 GPL 3 entspricht den Intentionen der Punkte 7 und 8 GPL 2, der Punkt 13 GPL 3 wird in der nachfolgenden Kompatibilitätsprüfung erklärt, Punkt 14 GPL 3 entspricht Punkt 9 GPL 2. Die abschließenden Gewährleistungs- und Haftungsbestimmungen sind insofern zu erwähnen, als diese explizit auf nationales Recht verweisen. Die Klauseln sind daher zwar

⁴⁵¹ <http://www.fsf.org/blogs/licensing/2007-03-28-gplv3-grandfather>.

⁴⁵² FSF, GPLv3 Third Discussion Draft Rationale 25, <http://gplv3.fsf.org/gpl3-dd3-rationale.pdf>.

nicht von einer Überprüfung durch nationales Recht⁴⁵³ befreit, nehmen aber erstmals Bezug auf dieses.

5.4.2.3. Kompatibilität

Die Überprüfung der Kompatibilität der GPL 3 ist nicht nur hinsichtlich der anderen Open Source Lizenzen interessant, auch die Kompatibilität mit der GPL 2 sorgte während des Diskussionsprozesses für Kritik.

5.4.2.3.1. Kompatibilität mit anderen Open Source Lizenzen

Hinsichtlich der BSDartigen Lizenzen gibt es bei der GPL 3 die geringsten Kompatibilitätsschwierigkeiten. Die Überführung von BSDartig lizenzierter Software in GPL 3-lizenzierte Software ist demnach ohne Probleme möglich, die Überführung von GPL 3 lizenzierter Software in BSDartige weiterhin nicht. Die GPL 3 enthält in den Punkten 7 und 13 einige Hinweise, die bei der Kompatibilitätsprüfung beachtet werden müssen. Das Ziel, die GPL kompatibel zur Apache License⁴⁵⁴ zu machen, musste letztendlich nicht – wie befürchtet – aufgegeben werden⁴⁵⁵, da die FSF in der letzten Phase der Diskussion herausfand, dass die Haftungs- und Gewährleistungsbestimmungen des § 9 APL mit der GPL 3 nicht vereinbar sein könnten. Bezüglich der GPL 2 ist festzustellen, dass viele Lizenzgeber – der prominenteste unter ihnen ist wohl Linus Torvalds – die Möglichkeit ergriffen, den Zusatz „*or any later version*“ von ihren Lizenztexten zu streichen, um so einen automatischen Übergang ihres Werks von GPL 2 zu GPL 3 zu verhindern. Die Möglichkeit ein

⁴⁵³ Vgl Kap 6.

⁴⁵⁴ Vgl Kap 5.2.2.

⁴⁵⁵ Siehe: FSF, GPLv3 Third Discussion Draft Rationale 33, <http://gplv3.fsf.org/gpl3-dd3-rationale.pdf>.

Programm, welches nur unter der GPL 2 lizenziert wird, mit Werken, die unter der GPL 3 lizenziert sind, zu verbinden ist damit unmöglich⁴⁵⁶. In Bezug auf die Affero GPL⁴⁵⁷ hat die FSF in der GPL 3 mit Punkt 13 eine eigene Kompatibilitätsklausel eingeführt, nach der es einem Lizenznehmer gestattet ist, ein GPL 3-lizenziertes Werk mit einem Affero GPL -lizenziertem Werk zu verbinden und das Ergebnis unter die Affero GPL zu stellen. *Jaeger* weist hierbei darauf hin, dass diese Regelung für jede Verbindung von Programmelementen gültig ist und daher ein Anreiz für einige Lizenznehmer sein könnte, ihre Weiterentwicklungen unter die Affero GPL zu stellen, um eine Umgehung der Lizenzpflichten ausschließen zu können⁴⁵⁸. Dies stellt mE eine im Ergebnis unbefriedigende Lösung dar, da die GPL 3 hier ausgehöhlt wird und Kunden verunsichert werden, die einen Lizenzwechsel ihres Anbieters nur schwer nachvollziehen können. Die GPL 3 hat diesbezüglich noch Verbesserungsbedarf, ohne dass es gleich einer GPL 4 bedürfe. Die Kompatibilitätsproblematik muss aber noch genauer analysiert und in einem engen Dialog mit anderen Open Source Projekten überprüft und gegebenenfalls angepasst werden.

Hinsichtlich der Kompatibilität mit anderen „Copyleft“-Lizenzen muss wie bei der GPL 2 auf eine ausdrückliche Regelung im Lizenztext geachtet werden, da der Schutz des „Copyleft“ eine andere Lizenzwahl grundsätzlich ausschließt. Nur wenn eine Lizenz eine „devote Klausel“ zu Gunsten der GPL 3 enthält, liegt diese Kompatibilität vor.

⁴⁵⁶ Unter: <http://gpl3.palamida.com:8080/index.jsp> findet man einen Überblick, welche Projekte von der GPL 2 zur GPL 3 gewechselt haben.

⁴⁵⁷ <http://www.affero.org/oagpl.html>.

⁴⁵⁸ *Jaeger*, c't 2007 Heft 15, 38.

5.4.2.3.2. Kompatibilität mit proprietären Lizenzen

Was die Verbindung von unter der GPL 3 lizenzierten Werken mit proprietärer Software betrifft, ist auf das zu GPL 2 in Kapitel 5.4.1.3.2. Ausgeführte zu verweisen.

5.5. Lizenzen mit strengem Copyleft-Effekt – Sonstige Lizenzen

Folgende Lizenzen weisen ebenfalls einen strengen Copyleft-effekt auf:

1. Arphic Public License
2. **Common Public License**,
3. Democratic Software Public License (v1.1)
4. **Eclipse Public License (v1.0)**
5. **European Union Public License**⁴⁵⁹
6. **IBM Public License**
7. **Jabber Open Source License**
8. **Nethack General Public License**
9. Open Group Public License
10. **Open Software License (OSL) (v1.0, v1.1 und v2.0)**
11. RedHat eCos Public License (v. 1.1)
12. Salutation Public License
13. Software AG License Terms (Quip License) (v. 1.3)
14. Vim License

Die übrigen Open Source Lizenzen mit strengem Copyleft-Effekt werden anhand der „European Union Public License, der „Common Public License“ und der „Nethack General Public License“ erörtert.

⁴⁵⁹ Die EUPL wurde bislang noch nicht durch die OSI approbiert, wird aber auf Grund ihrer Relevanz für den europäischen Raum besprochen.

5.5.1. Die European Union Public License

Die European Union Public License (EUPL) ist eine der jüngsten Entwicklungen auf dem Lizenzsektor im Open Source Bereich. Im Rahmen dieses Überblicks ist es nötig zu überprüfen, ob die Lizenz den Anforderungen der OSD genügt, da der Name bereits eine Einschränkung aufweist.

5.5.1.1. Hintergrund

Die EUPL wurde im Rahmen des EU-Programms der Kommission IDABC (steht für „*Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens*“) im Zeitraum 2005 – 2007 geschrieben und von der Kommission am 9. Januar 2007 gebilligt⁴⁶⁰. Das IDABC hat die Aufgabe, die Interoperabilität europaweiter elektronischer Behördendienste für die öffentliche Verwaltung, Unternehmen und Bürger zu fördern. Da es im Rahmen dieser Aufgaben zur Entwicklung verschiedener Computerprogramme kam, deren Urheber die EG ist, wurde überlegt diese Programme unter eine Open Source Lizenz zu stellen. Dies nahm die EG zum Anlass, eine eigene Lizenz als Gegenstück zu den anderen Open Source Lizenzen zu schaffen, die auf amerikanischem Recht beruhen und nur unzureichend auf europäische Rahmenbedingungen Rücksicht nehmen. Die EUPL ist ein Versuch der EG, eine gemeinschaftliche Lösung zu schaffen. Mangels einer vollendeten Harmonisierung in den Bereichen des gewerblichen Rechtsschutzes und des Vertragsrechts muss die EUPL jedoch stets auf ihre Auswirkungen im Rahmen der einzelnen nationalen Rechtsordnungen überprüft werden.

⁴⁶⁰ <http://ec.europa.eu/idabc/en/document/6523>.

5.5.1.2. Lizenzmerkmale

Ein wesentliches Lizenzmerkmal ist die Lizenzsprache. Die EG hat die EUPL in drei Sprachen herausgegeben, neben der englischen gibt es eine deutsche und eine französische Version. In dieser Arbeit wird auf den Text der deutschen Version eingegangen, der sich inhaltlich nicht von den anderen Sprachversionen unterscheidet

Die EUPL beginnt mit einer Präambel, welche nach eigener Aussage kein Bestandteil der EUPL ist, sondern als Einleitung und Hintergrundinformation dient. Der Präambel folgen zwei Bedingungen, welche zum einen die Nutzung der EUPL-lizenzierten Programme betreffen und zum anderen die Gültigkeit der Lizenz hinsichtlich des Programms allgemein behandeln. Diese Bedingungen sind wie folgt:

1. Die Nutzung der Software darf nur in der Form genutzt werden, die durch die EUPL vorgegeben wird
2. Die EUPL ist nur dann für Programme maßgeblich bzw anwendbar, wenn der Lizenzgeber unmittelbar hinter dem Urheberrechtshinweis den Satz „*Lizenziert unter der EUPL V 1.0*“ als Vermerk anbringt oder in einer sonstigen Form darauf hinweist, dass er sein Programm unter der EUPL lizenzieren möchte.

Diese Bedingungen sind mE nicht klar definiert und werfen einige Fragen auf. Die erste Bedingung setzt ein (Ein-)Verständnis des Lizenztextes voraus, um ein unter der EUPL lizenziertes Werk überhaupt benutzen zu dürfen. Diese Bestimmung ist hinsichtlich des gesetzlichen Nutzungsrechts des § 40d UrhG nicht unproblematisch und in der Praxis als hinderlich einzustufen. Es erscheint unrealistisch, dass ein Großteil der reinen Nutzer EUPL-lizenzierter Software eine genaue Kenntnis des Lizenztextes hat, geschweige denn ein genaues Verständnis desselben. Dieses wird jedoch von der EUPL vorausgesetzt, um die EUPL-lizenzierten Programme nutzen zu dürfen. Die zweite Bedingung ist

mE ebenfalls missverständlich formuliert. Demnach ist die EUPL nur dann auf ein Programm anwendbar, wenn der Hinweis auf die Lizenz hinter dem Urhebervermerk angebracht ist. Der Sinn dieser Regelung mag sinnvoll erscheinen, da sie aber als Ausschlussregel konzipiert ist, geht sie über das Ziel hinaus, den Anwender bzw. Weiterentwickler über die Lizenzierung der Software in Kenntnis zu setzen. Sinnvoller erschiene hier die Platzierung des Lizenzhinweises als Vorschlag – wie dies in weiterer Folge auch beim Textvorschlag „Lizenziert unter der EUPL V 1.0“ gehandhabt wird – zu regeln und Ausnahmen zuzulassen, welche den Zweck der Regelung erfüllen. Anzudenken wäre eine Regelung, welche den Hinweis an prominenter Stelle erlaubt (wie bei anderen Open Source Lizenzen auch).

Den Bedingungen am Anfang folgen in Punkt 1 EUPL die Begriffsdefinitionen, wie man sie aus anderen Open Source Lizenzen kennt. Punkt 2 EUPL beinhaltet den Umfang der Lizenzrechte. Diese gleichen der GPL und räumen einem Lizenznehmer eine weltweite, kostenlose, nicht-ausschließliche Werknutzungsbewilligung sowie das Recht zur Unterlizenzierung ein. Diese Werknutzungsbewilligung gilt auch für unbekannte Nutzungsarten, soweit dies nicht durch nationale Gesetze anders geregelt wird.⁴⁶¹ Punkt 2 EUPL umfasst ebenfalls den Umgang mit bestehenden Patentrechten. Sollte das Programm von einem Patent betroffen sein, räumt der Lizenzgeber einem Lizenznehmer hinsichtlich des Patents ebenfalls ein nicht-ausschließliches, kostenloses Nutzungsrecht ein.

Punkt 3 EUPL betrifft die Zurverfügungstellung des Quellcodes. Dem Lizenzgeber wird hier ein Wahlrecht eingeräumt. Verbreitet er das Programm nicht im Quellcode, muss er wie bei dem Hinweis auf die Lizenz selbst nach dem Urhebervermerk einen Ort angeben, von wo der Quellcode bezogen werden kann. Praktisch erscheint hier der Hinweis, dass ein Entwickler den

⁴⁶¹ Zur Problematik der unbekanntenen Nutzungsarten siehe Kap 3.3.24.

Quellcode nur solange bereitstellen muss, solange er das Programm selbst verbreitet.

Punkt 4 EUPL ist eher als Hinweis und nicht als Regelung zu sehen, da hier nur der „Zweck“ der Lizenz beschrieben wird. Der Hinweis darauf, dass die EUPL keine „*Ausnahmen oder Schranken der ausschließlichen Rechte des Urhebers am Originalwerk*“ bezweckt, ist mE unzureichend formuliert. Abgesehen davon, dass die Erklärung betreffend den Zweck der Lizenz ihrer Systematik nach an den Anfang von Punkt 2 EUPL gehört, ist der Hinweis auf die Erschöpfung dieser Rechte überflüssig, da die Bestimmungen zum Erschöpfungsgrundsatz durch den Gesetzgeber, die Rsp und die Lehre herausgearbeitet wurden und daher von der EUPL sowieso nicht berührt werden⁴⁶².

Punkt 5 EUPL behandelt die Pflichten des Lizenznehmers und ist als Bedingung für die Rechtseinräumung durch die EUPL konzipiert. Dem Lizenznehmer ist es danach untersagt, Urheber-, Marken- und Patentrechtshinweise, den Lizenztext oder den Haftungsausschluss zu verändern oder zu entfernen. Darüber hinaus muss bei jeder Verbreitungshandlung der Lizenztext mitgeliefert werden. Sollte eine Weiterentwicklung vorgenommen worden sein, ist diese zu dokumentieren. Diesen Bedingungen folgt die „copyleft“- und die Kompatibilitätsklausel der EUPL, auf welche im Zuge der Kompatibilitätsprüfung im Anschluss eingegangen wird sowie die Pflicht, den Quellcode bei der Verbreitung mitzuliefern oder eine Bezugsquelle anzugeben. Die abschließende Bedingung, die Kennzeichen- oder Markenbezeichnung bzw das Namensrecht des Lizenzgebers nur dann benutzen zu dürfen, wenn dies für eine „angemessene oder übliche Beschreibung der Herkunft des Werkes“ sinnvoll erscheint, sorgt dafür, dass der Lizenznehmer selbst erkennen muss, wann er bspw die Markenbezeichnung des Lizenzgebers nutzen darf. Diese Lösung der EUPL

⁴⁶² Zum Erschöpfungsgrundsatz vergleiche Kap 3.3.2.1.

sollte mE ebenfalls klarer formuliert werden, da diese Bedingung für einen juristisch unkundigen Lizenznehmer schwer verständlich sein wird. Hierbei wäre eine „angemessene oder übliche Beschreibung der Herkunft des Werkes“ zumindest in einem Glossar zu kommentieren oder besser noch im Lizenztext selbst näher zu definieren.

Punkt 6 EUPL weist darauf hin, dass ein Lizenzgeber – als Urheber des Originalwerks oder als Weiterentwickler – mit der Verbreitung versichert, dass er die für die Lizenzierung nötigen Rechte besitzt und diese auch im Falle einer Kreuzlizenzierung einräumt. Dieser Punkt erscheint unspektakulär, sorgt aber für mehr Rechtssicherheit unter den Lizenzvertragspartnern, da er die rechtmäßige Weitergabe der Lizenzrechte klar regelt, indem die Partner darauf hingewiesen werden, sicherzustellen, die benötigten Rechte dafür inne zu haben.

Die in den Punkten 7 und 8 EUPL beinhalteten Gewährleistungs- und Haftungsausschlüsse bzw. Haftungsbeschränkungen unterscheiden sich wesentlich von den üblichen Haftungs- und Gewährleistungsausschlüssen der Open Source Lizenzen, da sie die bestehenden Gesetze nicht vollkommen außer acht lassen. Der erste Satz von Punkt 7 EUPL beinhaltet mE wiederum eine unglücklich gewählte Formulierung. „*Die Arbeit an diesem Werk wird laufend fortgeführt*“ impliziert eine dauernde Weiterentwicklung des Werkes des Anwenders bzw. Lizenznehmers und sorgt damit für ein Sicherheitsgefühl bei diesem, da er davon ausgehen kann, dass er ein „lebendiges“ Programm benutzt. Dies mag eventuell auf die Werke zutreffen, die unter der EUPL stehen und von der Kommission der EG entwickelt wurden, da diese stetig weiterentwickelt werden. Dennoch wird man erstens nicht davon ausgehen können, dass das EUPL-lizenzierte Programm des Anwenders immer weiterentwickelt wird und zweitens ist es möglich, dass dies auf solche Programme nicht zutrifft, die zwar unter der EUPL lizenziert wurden, die aber nicht von der EG betreut werden. Der zweite Satz des Punkts 7 EUPL ist ebenso unverständlich, da er zwar den Lizenznehmer auf mögliche Bugs im

Programm hinweist, dies aber keinen rechtlichen Nutzen hat⁴⁶³. Punkt 8 EUPL begrenzt die Haftung auf Vorsatz und geht sogar auf die Produkthaftung ein.⁴⁶⁴

Punkt 9 EUPL behandelt Zusatzvereinbarungen, welche für die entgeltliche Erbringung von Zusatz- oder Supportleistungen, die Gewährleistung, die Haftungsfreistellung oder andere Haftungsverpflichtungen vereinbart werden dürfen, solange sie in Einklang mit der Lizenz stehen und auf eigenen Namen passieren. Diese Klausel ähnelt damit Formulierungen der anderen Open Source Lizenzen, wobei sie die Möglichkeiten der Zusatzvereinbarungen exemplarisch und taxativ aufzählt. In der Praxis kann auch dieser Punkt mE zu Schwierigkeiten führen, da die Prüfung über die Konformität mit der EUPL im Auslegungsweg geschieht.

Punkt 10 EUPL betrifft die Annahme des Lizenztextes. Hierbei wird darauf hingewiesen, dass die EUPL durch das Anklicken des Symbols „*Lizenz annehmen*“ unter dem Fenster des Lizenztextes als Lizenzvereinbarung anerkannt wird. Zwar wird eine „*vergleichbare Weise in einer nach anwendbaren Recht zulässigen Form*“ ebenso durch Punkt 10 EUPL anerkannt, diese Formulierung weist jedoch wiederum Einschränkungen auf, da andere Lizenzgeber als die EG, die ihr Programm unter der EUPL lizenziert haben, einen anderen Weg gewählt haben könnten, eine Lizenzvereinbarung gültig zustande kommen zu lassen.

Punkt 11 EUPL betrifft die nötigen Informationspflichten, welche sich in Österreich in den §§ 5 sowie 9-11 ECG, 5c KSchG und 14 UGB finden⁴⁶⁵.

⁴⁶³ Siehe dazu auch: *Wiebe/Heidinger*, MR 05/06, 262.

⁴⁶⁴ Zur näheren Betrachtung dieser Problematik siehe Kap 6.

⁴⁶⁵ Die Offenlegungspflichten gem §§ 24, 25 MedienG können in diesem Fall außer acht gelassen werden. Diese Pflichten können zwar für Webseiten in Betracht kommen, von denen aus das Programm bezogen werden kann, dies ist aber unabhängig vom Programm selbst.

Dieser Hinweis erscheint sinnvoll, da er Lizenzgeber und Weiterentwickler daran erinnert, die Herkunftsinformationen zu pflegen bzw aktuell zu halten.

Punkt 12 EUPL enthält die auflösende Bedingung ähnlich der GPL, wonach die eingeräumten Rechte automatisch ex nunc revidiert werden, sobald ein Lizenznehmer gegen die Bestimmungen der EUPL verstößt. Der zweite Absatz von Punkt 12 EUPL, welcher Unterlizenznehmer betrifft, die ihre Lizenz von einem Lizenzgeber erhalten haben, der gegen die EUPL verstoßen hat, kann als Indiz dafür gesehen werden, dass die EUPL nicht durch eine Lizenzkette weitergegeben wird, sondern wie bei der GPL ein Lizenznehmer stets mit dem ursprünglichen Lizenzgeber kontrahiert. Absurd wird das Ganze, wenn der Ursprungslizenzgeber sich dazu entschließt, gegen die Lizenzbedingungen zu verstoßen, indem er das Werk bspw in Folge nur noch proprietär vertreibt. Die EUPL wäre nach Punkt 11 für Lizenznehmer damit immer noch gültig. Die Konsequenz wäre, dass das Programm weiterhin unter der EUPL vertrieben werden könnte, eine Änderung des Lizenztextes aber nicht mehr möglich wäre⁴⁶⁶.

Punkt 13 EUPL betrifft die Wirksamkeit der EUPL in den Fällen, in denen einzelne Teile der Lizenz nicht mit einem jeweiligen nationalen Recht zu vereinbaren sind. Diese Klausel entspricht ebenfalls der Lösung der GPL, die als Konsequenz eine teleologische Reduktion vorsieht.

Die Punkte 14 und 15 EUPL betreffen den Gerichtsstand und das anzuwendende Recht. In den Fällen, in denen die EG als Lizenzgeber betroffen ist, wird der Gerichtshof der EG gem Art 238 EG als Gerichtsstand vereinbart, in allen anderen Fällen der Ort, an dem der Lizenzgeber seinen Wohnsitz oder den wirtschaftlichen Mittelpunkt seiner Tätigkeit hat. Dies entspricht den §§ 66

⁴⁶⁶ Dies wäre dann der Fall, wenn die EG selbst gegen die EUPL verstoßen würde. Sie wäre zwar weiterhin gem Punkt 13 EUPL in der Lage, durch die Europäische Kommission die EUPL zu modifizieren, könnte aber nicht mehr als Lizenzgeber auftreten.

und 75 JN. Solange die EG als Lizenzgeber auftritt, ist das belgische Recht auf die EUPL anwendbar, in den anderen Fällen richtet sich das anzuwendende Recht nach dem Recht jenes Mitgliedstaates der EU, in dem der Lizenzgeber seinen Sitz hat.

Die EUPL muss als Schritt in die richtige Richtung gewertet werden. Die Intention, eine europäische Lizenz zu schaffen hat seine Berechtigung, da sich die europäischen Immaterialgüterrechte, aber auch das Haftungs- und Gewährleistungsrecht teilweise eklatant vom amerikanischen Recht unterscheiden. Die EUPL hat jedoch aus verschiedenen Gründen ihr Ziel verfehlt, eine sinnvolle Alternative zu den anderen Open Source Lizenzen darzustellen. Die Gründe hierfür werden nachfolgend aufgezeigt:

1. Ein wesentlicher Kritikpunkt ist, dass mit der EUPL eine weitere Open Source Lizenz geschaffen wurde und eine Fülle von Open Source Lizenzen nicht dazu beiträgt, eine größere Rechtssicherheit unter den Vertragspartnern herzustellen. Die Zielsetzung eine „gesamteuropäische“ Open Source Lizenz zu erstellen, erscheint mir momentan noch zum Scheitern verurteilt, da eine gemeinschaftsrechtliche Harmonisierung in den jeweiligen Rechtsgebieten zwar in Gang gebracht, aber noch nicht abschließend hergestellt wurde. Die Problematik der nicht einwandfreien Umsetzbarkeit der „*amerikanischen*“ Open Source Lizenzen in Europa haftet damit auch der EUPL an, deren Geltung ebenfalls jeweils im nationalen Recht überprüft werden muss.
2. Die EUPL wurde geschrieben, um die von der EG entwickelten Computerprogramme als Open Source Software verbreiten zu können. Auch wenn diese Einstellung der EG loblich ist, wird die Lizenz doch meist für die Eigenentwicklungen der EG benutzt werden und eher selten von Dritten für ihre Ursprungsprogramme.

3. Weiterentwickler müssen sich daher mit einer weiteren neuen Open Source Lizenz auseinandersetzen, die noch keinen großen Bekanntheitsgrad hat und infolge noch kein Vertrauen genießt.
4. Open Source Lizenzen sollten für den globalen Einsatz nutzbar sein. Dies ergibt sich schon aus der Philosophie des Open Source Gedankens. Die EUPL schafft für Europa nur ein schwaches Gegenstück zu den amerikanischen Lizenzen, da amerikanische Weiterentwickler hier gleiche rechtliche Auslegungsprobleme haben werden wie ihre europäischen Kollegen mit den „amerikanischen“ Lizenzen.
5. Ein letzter Kritikpunkt betrifft den Lizenztext der EUPL. Dieser ist teilweise sehr umständlich geschrieben und erschwert so eine einfache Auslegung der Lizenz. Dieses wiederum vermindert die Rechtssicherheit der Vertragspartner, da diese bei der Auslegung zu unterschiedlichen Ergebnissen kommen können. Deutlich wird dies bspw bei der Auslegung der Kompatibilität der Lizenz, welche zu einer Diskussion geführt hat, die in Folge dargestellt und bewertet wird.

5.5.1.3. Kompatibilität

Die Problematik der Kompatibilität ist von den Verfassern der EUPL erkannt worden. Die EUPL hat deshalb in Punkt 5 eine eigene Kompatibilitätsklausel, welche es einem Weiterentwickler erlaubt, seine Weiterentwicklung auch unter jenen kompatiblen Lizenzen zu verbreiten, die im Anhang taxativ aufgelistet sind. Die EUPL tritt dabei gem Punkt 5 zugunsten einer kompatiblen Lizenz im Konfliktfall zurück. Die aufgelisteten kompatiblen Open Source Lizenzen sind:

1. GPL (Version 2)
2. OSL (Version 2.1 und 3.0)
3. CPL (Version 1.0)
4. EPL (Version 1.0)
5. Cecill (Version 2.0)

Im Zuge der Kompatibilitätsüberprüfung gilt es drei Dinge zu klären. Zum einen muss überprüft werden, ob die aufgelisteten Lizenzen tatsächlich kompatibel zur EUPL sind, zum anderen werden auch die übrigen Open Source Lizenzen einer Kompatibilitätsprüfung unterzogen, bevor abschließend die Kompatibilität gegenüber proprietären Lizenzen erörtert wird.

5.5.1.3.1. Die Kompatibilität hinsichtlich der aufgeführten Open Source Lizenzen

Es muss unterstrichen werden, dass Punkt 5 EUPL den Sinn hat, jene Open Source Lizenzen taxativ aufzuzählen, hinter denen die EUPL im Falle einer Weiterentwicklung zurücktritt („Downstream“ – Kompatibilität⁴⁶⁷). Dies bedeutet, dass die EUPL im Gegensatz zur GPL eine devote Klausel – ähnlich Ziffer 3 LGPL – enthält und daher flexibler als die GPL auftritt. Punkt 5 EUPL darf daher nicht so verstanden werden, dass die aufgezählten Lizenzen jeweils ihrerseits mit der EUPL kompatibel sind, da keine der aufgezählten Lizenzen eine devote Klausel gegenüber der EUPL enthält. Eine inhaltliche Überprüfung kann an dieser Stelle entfallen, da die EUPL keinerlei Bedingungen an das „Zurücktreten“ stellt, sondern lapidar feststellt: *„Sollten die Verpflichtungen des Lizenznehmers aus der kompatiblen Lizenz mit denjenigen aus der vorliegenden Lizenz (EUPL) in Konflikt stehen, werden die Verpflichtung aus der kompatiblen Lizenz Vorrang haben.“* Diese Lösung ist in praktischer Hinsicht löblich, da sie durch die taxative Aufzählung nur jene Lizenzen betrifft,

⁴⁶⁷ Bastin/Laurent, Report on Study of the compatibility mechanism of the EUPL, <http://ec.europa.eu/idabc/servlets/Doc?id=27472>.

die unstrittig den Open Source Gedanken in sich tragen. Strittiger vom Standpunkt der EUPL aus gesehen ist die „Kann“-Bestimmungen in Punkt 5 EUPL. Dem Weiterentwickler wird hier ein Wahlrecht eingeräumt, sein Werk unter einer der taxativ aufgezählten Lizenzen weiterzuverbreiten. Diese Bestimmung schwächt die EUPL nachhaltig, da Weiterentwickler sich für die Lizenzierung allein unter einer anderen aufgezählten Lizenz entscheiden könnten, ohne dass daraufhin die Notwendigkeit bestünde, Weiterentwicklungen wiederum unter die EUPL zu stellen. Diese Klausel sollte dahingehend erweitert werden, dass die „Kann“-Option nur dann zum Tragen kommt, wenn nachvollziehbare Gründe für einen Lizenzwechsel angeführt werden können. Aktuell birgt diese Bestimmung in Punkt 5 EUPL die Gefahr zur „Ausverkaufsklausel“ zu werden.

5.5.1.3.2. Die Kompatibilität hinsichtlich der nicht aufgeführten Lizenzen

Im Falle der nicht im Annex zur EUPL aufgeführten Open Source Lizenzen ist auf das zur GPL Ausgeführte zu verweisen⁴⁶⁸. Durch das strenge Copyleft, das die EUPL beinhaltet, ist ihre Kompatibilität mit BSDartigen Lizenzen insofern gegeben, dass diese in die EUPL eingegliedert werden können. Gegenüber anderen „Copyleft“-Lizenzen besteht nur im Sinne der oben erörterten Kompatibilitätsklausel Kompatibilität.

5.5.1.3.3. Die Kompatibilität mit proprietären Lizenzen

Die EUPL hat durch ihr starkes Copyleft keine Kompatibilität mit proprietären Lizenzen. Eine solche Intention findet sich an keiner Stelle im Lizenztext der EUPL, im Gegenteil, die Wahl der Kommission die Lizenz mit

⁴⁶⁸ Vgl Kap 5.4.

einem starken Copyleft auszustatten, kann als Indiz dafür dienen, dass eine solche Kompatibilität in keiner Weise gewünscht oder vorgesehen ist. Hierzu bleibt zu erwähnen, dass eine Kompatibilität im Einzelfall dann gegeben sein kann, wenn ein unter der EUPL lizenziertes Programm und ein proprietäres Programm eine so große Eigenständigkeit jeweils aufweisen, dass beide Programme auch getrennt von einander vertrieben werden könnten. Dies ergibt sich analog aus den Kompatibilitätsuntersuchungen zur GPL⁴⁶⁹.

5.5.2. Die Common Public License

Die Common Public License (CPL) ist die weiterentwickelte Lizenzversion der IBM Public License und regelt die Bestimmungen des Lizenztextes allgemeiner als die IBM Public License, die speziell auf IBM als Rechteinhaber eingeht bzw einging. IBM spielt im Open Source Bereich eine große Rolle und hat im Jahr 2005 fünfhundert Patente für die Nutzung im Open Source Bereich freigegeben⁴⁷⁰.

5.5.2.1. Hintergrund

IBM stellte 2001 „Eclipse“, ein Framework, das meist als Entwicklungsumgebung (IDE) genutzt wird, unter die CPL Version 0.5⁴⁷¹. Durch die aktuelle Version 1.0 der CPL, welche 2002 folgte, wurden lediglich Redaktionsfehler bereinigt. Die CPL hat sogar Microsoft überzeugen können, die das Entwicklungstool „Windows Installer XML“ (WiX), die „Windows Template Library“ (WTL) und die „FlexWiki“-Engine“ unter der CPL lizenziert

⁴⁶⁹ Vgl Kap 5.4.1.3.

⁴⁷⁰ http://news.com.com/IBM+offers+500+patents+for+open-source+use/2100-7344_3-5524680.html?tag=nl.

⁴⁷¹ <http://www-128.ibm.com/developerworks/library/os-cplfaq.html>.

haben. IBM dagegen hat „Eclipse“ inzwischen unter eine eigene Lizenz – die „Eclipse Public License“ – gestellt, da Mitglieder der „*Eclipse Foundation*“ mit Ziffer 7 CPL nicht einverstanden waren⁴⁷². Ziffer 7 Eclipse Public License ist ansonsten mit der CPL identisch.

Die CPL weist große Ähnlichkeiten mit der GPL und der MPL auf, unterscheidet sich von diesen jedoch in Detailfragen. Die CPL zeichnet sich trotz ihrer hohen Regelungsdichte durch klare und leicht verständliche Formulierungen aus.

5.5.2.2. Lizenzmerkmale

Da die Lizenz – wie erwähnt – große Ähnlichkeit mit der GPL und der MPL hat, wird auf die wesentlichen Unterschiede zwischen diesen Lizenzen eingegangen.

Der Beginn der Lizenz ist missverständlich, da die CPL an dieser Stelle vom „*ACCOMPANYING PROGRAM*“ spricht. Auf diesen Ausdruck wird im Lizenztext nicht näher eingegangen, sodass es Auslegungssache bleibt, wie die CPL das Programm „begleiten“ wird oder mit ihm „verbunden“ sein soll. Nach bisherigem Stand der Technik erscheint es am sinnvollsten, den Lizenztext nach dem Urhebervermerk im Quellcode anzubringen – eine Lösung wie sie die GPL beschreibt – da er dort iSd der CPL das Programm „begleiten“ kann und mit ihm „verbunden“ bleibt⁴⁷³.

Ziffer 1 CPL enthält Definitionen der Lizenzvertragsbegriffe. Sie geht dabei genauer als die GPL auf die Problematik der Weiterentwicklung an CPL

⁴⁷² <http://www.eclipse.org/legal/eplfaq.php>.

⁴⁷³ Rosen, Open Source Licensing: Software Freedom and Intellectual Property Law (2005) 162.

lizenzierter Software ein. Hier findet man den ersten Unterschied zur GPL, da die CPL bestimmt, dass

1. eine „Contribution“ keine „*seperate modules of software*“ umfasst, die unter einer eigenen Lizenz stehen. Dieser Begriff wird durch die CPL zwar nicht weiter definiert, ist aber für jeden Softwareentwickler klar verständlich.
2. eine „Contribution“ auch kein „*derivative work*“ sein darf. Mit dieser Definition stimmt die CPL zwar mit der GPL überein, ist in ihrem Lizenztext jedoch präziser als diese, da sie Software, die keine „*derivative works*“ des CPL lizenzierten Programms sind, explizit von einer zwingenden Lizenzierung unter der CPL ausschließt.

Im Ergebnis bedeutet dies, dass Lizenznehmer den Copyleft-Effekt der CPL umgehen können, wenn sie ein eigenständiges Programm entwickeln, welches kein „*derivative work*“ darstellt. Das mag einem Außenstehenden logisch erscheinen, ist jedoch wie bei anderen Open Source Lizenzen stets im Einzelfall zu beurteilen, da vorwiegend technische Abgrenzungskriterien herangezogen werden müssen, welche teilweise schwierig festzulegen sind. Es ist daher positiv zu bewerten, dass die CPL in dieser Hinsicht eine klare Sprache in ihrem Lizenztext verwendet, aufgrund derer weitere Auslegungen vermieden werden.

In Ziffer 2a CPL folgt die Einräumung der Nutzungsrechte. Ein weiterer wesentlicher Unterschied zur GPL zeigt sich in Ziffer 2b CPL, welche die Einräumung von Patentrechten behandelt. Dieser Punkt gleicht im Wesentlichen der MPL, da hier ebenfalls die Kombination von unter der CPL lizenzierter Software mit anderer Soft- oder Hardware untersagt wird. ISd CPL bedeutet dies hinsichtlich der Patentrechte, dass folgende Situationen nicht durch die CPL gedeckt werden:

1. Jede Verbindung einer in Ziffer 1 CPL definierten „*Contribution*“ mit Software, die nicht zu dem unter der CPL lizenzierten

Programm gehört.

2. Jede Verbindung einer in Ziffer 1 CPL definierten „*Contribution*“ mit einer weiterentwickelten Version des Programms, vorausgesetzt, die eingeräumten Patentrechte sind notwendig für die aktuelle Version des Programms.
3. Jede gänzlich neu entwickelte Software, welche unter der CPL eingeräumte Patentrechte beinhaltet, selbst wenn diese Software die gleichen Merkmale und Funktionen besitzt wie ein unter der CPL lizenziertes Programm oder eine in Ziffer 1 CPL definierte „*Contribution*“ (Parallelentwicklungen).

Der Sinn dieser vielfach kritisierten⁴⁷⁴ Bestimmung wird deutlich, wenn man den Autor dieser Lizenz kennt: IBM, der weltweit größte Inhaber von Patentrechten, für den der Umgang mit Patenten daher von besonderer Bedeutung und Wichtigkeit⁴⁷⁵ ist. Daher werden durch die CPL nur solche Nutzungsrechte hinsichtlich der Patente eingeräumt, die dem ursprünglichen Programm anhaften.

Der Copyleft-Effekt der CPL befindet sich in Ziffer 3 CPL und unterscheidet sich nur wenig von den Bestimmungen der GPL und MPL. Die CPL erlaubt einem Lizenznehmer den Vertrieb eines weiterentwickelten Programms in Objektform unter einer selbst gewählten Lizenz, wobei diese Auswahl dadurch eingeschränkt wird, dass die selbst gewählte Lizenz mit den Bestimmungen der CPL übereinstimmt⁴⁷⁶ und jedem weiteren Lizenznehmer mitgeteilt wird, wo er den Quellcode des Programms beziehen kann. Für den Vertrieb der Software in Quellcodeformat sind die Bestimmungen der CPL restriktiver und stimmen mit jenen der GPL überein.

⁴⁷⁴ <http://www.gnu.org/philosophy/license-list.html>.

⁴⁷⁵ Rosen, Open Source Licensing 166.

⁴⁷⁶ In der Praxis wird es regelmäßig wenig Sinn machen, eine andere Lizenz für das Programm zu wählen.

Ziffer 4 CPL ist hervorzuheben, da es die erste Open Source Lizenz ist, die den Umgang mit Open Source Software im wirtschaftlichen Vertrieb explizit regelt. In Ziffer 4 CPL wird der kommerzielle Anbieter eines unter der CPL lizenzierten Programms darauf aufmerksam gemacht, dass sein wirtschaftlicher Umgang mit der Software keine anderen Rechteinhaber in die Pflicht nehmen sollte. Das Wort „*should*“ wird hier in Zusammenhang mit der Schadloshaltung der anderen Rechteinhaber benutzt und ist keine verpflichtende Bestimmung. Darüber hinaus fordert die CPL in Ziffer 4 CPL, dass der kommerzieller Anbieter andere Rechteinhaber vor jeglicher rechtlichen Verfolgung zu schützen habe. In Ziffer 4 zweiter Absatz befindet sich ein für Lizenztexte eher unüblicher Abschnitt. Er enthält das Beispiel eines Falls der Schutz und Schadloshaltung iSd Ziffer 4 CPL, welchem keinerlei rechtliche Relevanz zukommt. Insgesamt muss in Zusammenhang mit Ziffer 4 CPL auf die österreichische Rechtslage verwiesen werden, die eigene Regeln für haftungs- und gewährleistungsrechtliche Problemstellungen hat und auf die in Kapitel 6 eingegangen wird.

Ziffer 5 und 6 CPL behandeln den Haftungs- und Gewährleistungsausschluss. Die abschließende Ziffer 7 CPL bestimmt im ersten Absatz, dass eine Änderung der Lizenzbestimmungen nur in jenen Fällen zulässig ist, in denen die Lizenz gegen eine nationale Rechtslage verstößt. Ziffer 7 zweiter Absatz CPL regelt den Patentschutz ähnlich wie die MPL, wonach das Lizenzvertragsverhältnis automatisch aufgelöst wird, sollte es zu folgenden Situationen kommen:

1. Sollte ein Lizenznehmer patentrechtlich gegen einen Rechteinhaber vorgehen und dies in Zusammenhang mit einem zu Software gehaltenen Patent geschehen (es ist dabei unerheblich, ob das Patent auf CPL lizenzierte Software gehalten wird), werden jegliche Patenteinräumungen mit Zeitpunkt der Gerichtsanhängigkeit widerrufen⁴⁷⁷.

⁴⁷⁷ Rosen, Open Source Licensing 170.

2. Sollte ein Lizenznehmer patentrechtlich gegen das CPL lizenzierte Programm selbst vorgehen, werden ihm die in Ziffer 2b CPL eingeräumten Patentrechte entzogen⁴⁷⁸.

Der dritte Absatz Ziffer 7 CPL regelt die Lizenzhoheit von IBM über die CPL, der abschließende vierte Absatz Ziffer 7 CPL enthält eine Gerichtsstandsklausel für den US-Bundesstaat New York.

5.5.2.3. Kompatibilität

Die CPL ist mit der GPL nicht kompatibel, da die Patentklausel von der FSF als überflüssig erachtet wird⁴⁷⁹, denn die Forderungen der CPL an einen Patentinhaber, seine Rechte herauszugeben, sind der GPL unbekannt. Darüber hinaus kann es in Einzelfällen dazu kommen, dass die Definitionen der Ziffer 1 CPL nicht mit den in Ziffer 2b GPL genannten Forderungen übereinstimmen.

5.5.3. Die Nethack General Public License

Die „Nethack General Public License“ wurde speziell für das Computerspiel „NetHack“ entwickelt und basiert auf der von Richard Stallman entwickelten „BISON General Public License“.

⁴⁷⁸ Anzumerken ist hierbei, dass in der CPL Patentrechte ausschließlich in der Ziffer 2b CPL eingeräumt werden, so dass diese Formulierung etwas unverständlich ist.

⁴⁷⁹ <http://www.gnu.org/philosophy/license-list.html>.

5.5.3.1. Hintergrund

„NetHack“ ist die weiterentwickelte Version des Computerspiels „Rogue“ aus dem Jahre 1980 und damit eines der ältesten Computerspiele, die aktuell noch weiterentwickelt werden. Der Name „NetHack“ weist darauf hin, dass das Spiel mit Hilfe des Internets entwickelt wurde, obwohl es keinerlei Funktion im Internet besitzt, das heißt nicht multiplayerfähig ist. Der Begriff „Hack“ kommt aus dem Rollenspielbereich und hat mit dem Programmierbegriff „hacking“ nichts zu tun. Das auf einem Text basierende Computerrollenspiel wurde unter die „NetHack General Public License“ (NGPL) gestellt, da die Erfinder des Spiels eine freie Verbreitung sicherstellen wollten, ohne selbst haftungsrechtliche Klagen befürchten zu müssen.

5.5.3.2. Lizenzmerkmale

Die NGPL ist eine Kurzfassung der GPL, da sie die gleichen Rechte und Pflichten wie diese beinhaltet, aber aufgrund dessen, dass sie speziell für „NetHack“ entworfen wurde, keine über den Schutz des Computerspiels hinaus benötigte Rechte und Pflichten behandelt⁴⁸⁰. Das wird in der Einleitung der Lizenz deutlich, welche die Intentionen der Lizenzautoren bzw Urheber des Computerspiels erklären. Die NGPL gleicht in Umfang eher der BSD-Lizenz, erreicht jedoch mit ihrem nur 4 Ziffern umfassenden Lizenztext, den Copyleft-Effekt in strengem Ausmaß aufrecht zu erhalten. Die Lizenz wird nicht im Detail besprochen, da sie – wie erwähnt – der GPL ähnlich ist, was Rechte und Pflichten betrifft; aber dennoch deutlich macht, wie Open Source Software im Besonderen geschützt werden kann, wenn eine Lizenz nur ein bestimmtes Programm berücksichtigt.

⁴⁸⁰ Die NGPL kann als eine Urfassung der GPL gesehen werden, da sie noch vor der Version 1 der GPL herausgegeben wurde und Richard Stallman wesentlich an ihrem Inhalt beteiligt war.

5.5.3.3. Kompatibilität

Die NGPL ist mit der GPL kompatibel. Es wird allerdings empfohlen, für Weiterentwicklungen auf GPL lizenzierten Code zurückzugreifen, da dieser durch die hohe Regeldichte der GPL besser geschützt werden kann. Die NGPL hat daher mehr historischen Charakter und sollte nur noch dem Schutz des „alten“ Quellcodes dienen.

5.6. Lizenzen mit beschränktem Copyleft-Effekt – MPL-artige Lizenzen

Bei Open Source Lizenzen mit beschränktem Copyleft-Effekt muss nicht jedes abgeleitete Werk oder jede Weiterentwicklung erneut unter die Ursprungslizenz gestellt werden, sondern nur jene Veränderungen, die vom Lizenznehmer in einer bereits bestehenden Datei vorgenommen werden. Neue Dateien, die vom Lizenznehmer hinzugefügt werden, dürfen unter anderen Lizenzbestimmungen und damit auch proprietär weiterverbreitet werden. Bei Lizenzen mit beschränktem Copyleft-Effekt spricht man von „Larger Works“. Sie dienen als Kompromisslösung⁴⁸¹ zwischen den BSDartigen Lizenzen und den Lizenzen mit strengem Copyleft. Das Hauptaugenmerk liegt hierbei auf der Mozilla Public License (MPL), da die anderen Lizenzen jeweils leicht abgewandelte Varianten der MPL darstellen. Folgende Lizenzen gehören zu den MPLartigen Lizenzen:

1. Common Development and Distribution License (CDDL) (v1.0)
2. Erlang Public License (v. 1.1)
3. ICS Open Source Public License
4. Interbase Public License
5. **Mozilla Public License (v. 1.0 und 1.1)**
6. Netizen Open Source License (NOSL)

⁴⁸¹ Jaeger/Metzger, Open Source Software 61.

7. Nokia Open Source License
8. Open Telecom Public License
9. Ricoh Source Code Public License
10. Sun Public License
11. Sun Industry Standards Source License (v. 1.1)
12. Zenplex Public License
13. Common Development and distribution License (CDDL)

5.6.1. Die Mozilla Public License

Die Mozilla Public License wurde von der Firma Netscape 1998 für den damals geplanten Netscape Communicator 5 entwickelt. Netscape konnte seinen Browser nicht unter der GPL freigeben, da der ursprünglich proprietäre Browser Programmbestandteile enthielt, die von Drittanbietern stammten, die einer Veröffentlichung ihres Quellcodes nicht zustimmten⁴⁸². Außerdem benutzt Netscape kryptografische Programmbestandteile, die nach amerikanischem Recht nicht öffentlich gemacht werden dürfen. Auch eine Lizenzierung unter der BSD Lizenz kam nicht in Frage. Netscape befürchtete, dass Mitbewerber ihren Quellcode nehmen und die damit gemachten Weiterentwicklungen erneut unter eine proprietäre Lizenz stellen könnten. Haupteinsatzfeld der MPL ist die Lizenzierung des Mozilla Webbrowsers und seiner zugehörigen Software. Die Lizenz hat mittlerweile innerhalb der Open Source Gemeinde sowie in der Wirtschaft viele Anhänger gefunden und wurde daher von anderen Lizenzautoren für ihre Programme adaptiert, bspw von Sun Microsystems als „Common Development and Distribution License“ für OpenSolaris.

⁴⁸² <http://www.mozilla.org/html/MPL/Faq.html>.

5.6.1.1. Hintergrund

Da Netscape keine volle Urheberherrschaft über das Produkt besaß, war es juristisch nicht einfach, den proprietären Browser mit seinen Bestandteilen in Open Source Software zu überführen. Die Hauptaufgabe bestand darin, eine Lizenz zu entwickeln, in der proprietäre Software neben Open Source Software bestehen konnte und immer noch lauffähig war. Die Version 1.0 wurde zeitgleich mit der Netscape Public License⁴⁸³ von *Mitchell Baker*⁴⁸⁴ entwickelt und lizenzierte den Netscape Navigator-Code ab dem 1. April 1998. Danach müssen Änderungen am bestehenden Quellcode – darunter fallen größtenteils Bugfixes – wieder unter der NPL lizenziert werden. Neuentwicklungen dagegen können unter die MPL gestellt werden. Dadurch sicherte Netscape das Fortbestehen und die Weiterentwicklung des Browsers, der seit 1998 konkurrenzfähig geblieben ist. Die MPL wurde in Folge überarbeitet und liegt in der durch die Mozilla Foundation entworfenen aktuellen Version 1.1 vor.

5.6.1.2. Lizenzmerkmale

Die MPL ist geprägt durch eine hohe Regelungsdichte. Die Lizenz beginnt mit einer Reihe von Begriffsdefinitionen, die einen Einblick in ihre Systematik zulassen. Danach unterscheidet sie zwischen Rechten, die einem Lizenznehmer durch einen „Contributor“ oder einem „Initial Developer“ eingeräumt werden können. Diese werden in Ziffer 2 MPL genauer erklärt.

Ziffer 2.1 MPL regelt die Rechtseinräumung durch den „*Initial Developer*“ und damit den Umgang hinsichtlich des der MPL unterstellten „*Original Codes*“

⁴⁸³ Vgl die Browserkriege Kap 2.3.5.

⁴⁸⁴ *Mitchell Baker* ist Juristin und war bei Netscape für rechtliche Fragen, die in Zusammenhang mit der Produktentwicklung standen, sowie für den Urheberrechtsschutz zuständig. Momentan ist sie der „*Chief Lizard Wrangler*“ für mozilla.org (siehe <http://www.mozilla.org/about/staff>).

iSd Ziffer 1.10 MPL. Demnach erhält ein Lizenznehmer das weltweite, lizenzgebührenfreie, einfache urheberrechtliche Nutzungsrecht, das Programm zu benutzen, zu vervielfältigen, weiterzuentwickeln, öffentlich zugänglich zu machen, zu verbreiten und Unterlizenzen⁴⁸⁵ zu vergeben (Ziffer 2.1a MPL). Werden durch die Vervielfältigung und Verbreitung des Programms Patentrechte betroffen, wird dem Lizenznehmer auch die Benutzung des Patents gestattet (Ziffer 2.1b MPL). Diese Klausel wird dahingehend eingeschränkt, dass dies nur für jene Patentansprüche gilt, die von der Nutzung des unveränderten Codes betroffen sind (Ziffer 2.1d). Diese Regelung lässt sich einfach erklären: Der Lizenzgeber will bei der Lizenzerteilung keine allgemeine Patentlizenz an jeden einräumen, die es Dritten erlauben würde, die Patente des Lizenzgebers in einem anderen Zusammenhang als dem unveränderten Quellcode zu nutzen⁴⁸⁶. Sollte der Lizenznehmer eine veränderte Version vervielfältigen und verbreiten wollen, so braucht er die Einwilligung des Patentrechtinhabers dafür.

Ziffer 2.2 MPL regelt die Rechtseinräumung durch den „Contributor“ und damit die Rechte an den Teilen des Codes, die erst nachträglich durch Weiterentwicklungen zu dem „Original Code“ hinzugekommen sind⁴⁸⁷. Diese Rechte entsprechen weitestgehend denen des „Initial Developers“ und unterstreichen die Absicht der Lizenz, zwischen ursprünglichen Urheberrechten und Bearbeiterurheberrechten zu unterscheiden.

Ziffer 3 MPL behandelt die Pflichten, die einem Lizenznehmer auferlegt werden. Sie gleicht den Open Source Lizenzen mit strengem Copyleft-Effekt.

⁴⁸⁵ Vgl. *Jaeger/Metzger*, Open Source Software 63, die darauf hinweisen, dass die Einräumung des Rechtes zur Vergabe von Unterlizenzen unlogisch erscheint, da die MPL in Ziffer 1.12 MPL davon ausgeht, dass jeder Lizenznehmer die Nutzungsrechte direkt vom Urheber erhält.

⁴⁸⁶ <http://www.mozilla.org/MPL/MPL-1.1-annotated.html>; *Jaeger/Schulz*, NOW Gutachten 46.

⁴⁸⁷ *Jaeger/Metzger*, Open Source Software 63.

Ziffer 3.1 MPL enthält einige Hinweise auf die Anwendbarkeit der Lizenz. Dabei verweist Ziffer 3.1 MPL auf Ziffer 2.2 MPL und verpflichtet den Urheber einer Weiterentwicklung eines unter der MPL lizenzierten Programms, dieses wiederum unter die MPL zu stellen⁴⁸⁸. Außerdem legt sie fest, dass bei der Weitergabe und Vervielfältigung eine Kopie der Lizenz in unveränderter Form mitgeliefert werden muss, um weiteren Lizenznehmern die Möglichkeit zur gültigen Annahme des Lizenzvertrags zu verschaffen. Ausgenommen von dem Verbot der Veränderung sind lediglich Dokumente wie in Ziffer 3.5 MPL beschrieben.

Bei der Zugänglichmachung des Quellcodes besteht für den Lizenznehmer eine Wahlmöglichkeit (Ziffer 3.2 MPL). Er kann entweder den Quellcode gemeinsam mit dem Objektcode auf demselben Datenträger vertreiben oder den Quellcode über einen „*Electronic Distribution Mechanism*“ – wie bspw. einer Webseite oder einem FTP-Server – für mindestens 12 Monate zugänglich machen. Sollte der Lizenznehmer die Weiterentwicklung selbst verändern, wird von Ziffer 3.2 MPL ein Zeitraum von sechs Monaten als ausreichend angesehen.

Die Regeln über Dokumentationspflichten und den Lizenzvermerk befinden sich in Ziffer 3.3 MPL. Danach müssen Lizenznehmer, die eine Weiterentwicklung vorgenommen haben, diese in einer beizufügenden Datei mit einem deutlichen Hinweis dokumentieren, welche die Weiterentwicklung beschreibt und das Datum angibt, an welchem die Weiterentwicklung vorgenommen wurde. Der Systematik der Lizenz folgend findet sich in Ziffer 3.3 MPL auch eine Bestimmung den „Initial Developer“ betreffend. Dieser muss in dem beizufügenden Hinweis namentlich erwähnt werden und es muss klargestellt werden, dass die Weiterentwicklungen auf dem „Original Code“

⁴⁸⁸ Diese Klausel wird von *Jaeger/Metzger* kritisch betrachtet, siehe *Jaeger/Metzger*, Open Source Software 65.

basieren. Dadurch wird sichergestellt, dass die Urheber im weiteren Verlauf bekannt und damit auch erreichbar bleiben⁴⁸⁹.

Die MPL versucht in Ziffer 3.4 MPL ebenfalls die Rechte von Dritten zu schützen, die Urheber-, Patent- oder sonstige Rechte an Programmen oder Programmteilen innehaben, die weiterverbreitet werden. Nach Ziffer 3.4 a MPL muss der Urheber einer Weiterentwicklung bei Vervielfältigung und Weitergabe auf diese Rechte in einer eigenen Datei hinweisen, die „LEGAL“ betitelt werden muss. Sollten die Rechte Dritter erst zu einem späteren Zeitpunkt bekannt werden, muss der Lizenznehmer seine MPL dahingehend aktualisieren⁴⁹⁰. Die Mozilla Foundation will mit dieser Klausel erreichen, dass ungeklärte Rechtsverhältnisse in Bezug auf den Programmcode einer Weiterentwicklung nicht im Wege stehen⁴⁹¹. Um Lizenznehmer, die MPL-lizenzierten Code weiterentwickeln, vor patentrechtlichen Angriffen zu schützen, sieht Ziffer 3.4 b MPL vor, dass bei der Einbindung von Schnittstellen zur Anwendungsprogrammierung – so genannten APIs⁴⁹² – darauf geachtet werden muss, jeweils einen Hinweis auf mögliche oder bestehende Patentrechte Dritter anzubringen. Der Hinweis erscheint sinnvoll, da auf diese Weise Weiterentwickler von Anfang an über bestehende Schutzrechte Dritter informiert sind und nicht erst durch eine Klage gegen sich davon Kenntnis erhalten. Ziffer 3.4 c MPL ist abschließend mehr als Hinweis gedacht, um Weiterentwickler daran zu erinnern, dass sie nur jene Rechte gültig weiter einräumen können, für die sie die Berechtigung haben.

Ziffer 3.5 MPL klärt die inhaltlichen Rechte und Pflichten, die der Lizenznehmer hinsichtlich des Urhebervermerks „*Exhibit A*“ hat und welche am

⁴⁸⁹ Dies war vor allem für die Neulizenzierung von Bedeutung, siehe Kap 5.6.1.3.

⁴⁹⁰ Eine Informationspflicht anderer Lizenznehmer ist in diesem Fall ausreichen, wenn ein entsprechender Hinweis auf die Aktualisierung auf der Webseite des Distributors aufscheint und er die Aktualisierung auch in den entsprechenden Newsgroups oder über seine Mailingliste verteilt.

⁴⁹¹ <http://www.mozilla.org/MPL/MPL-1.1-annotated.html>.

⁴⁹² „Application Programming Interface“.

Ende der Lizenz zu finden sind. Über die in Ziffer 3.1 MPL geregelte Verpflichtung zur Lizenzmitgabe bei Vervielfältigung und Verbreitung findet sich in Ziffer 3.5 MPL ein erneuter Hinweis.

Die Kompatibilität mit proprietären Lizenzen ist wie erwähnt ein Hauptmerkmal der MPL. Ziffer 3.6 MPL regelt den Vertrieb des Objektcodes. Um zu gewährleisten, dass diese Klausel nicht dazu benutzt wird, den Quellcode unter Verschluss zu halten, werden in Ziffer 3.6 MPL bestimmte Anforderungen gestellt. Der Quellcode muss weiterhin - unter den in den einzelnen Abschnitten der Ziffer 3 MPL geregelten Verpflichtungen - frei verfügbar sein, wobei vor allem auf Ziffer 3.2 MPL hingewiesen wird. Wird der Objektcode unter einer anderen Lizenz als der MPL vertrieben, muss der Verreiber auch eine Haftungsfreizeichnung gegenüber allen an dem zu vertreibenden Programm beteiligten Urhebern abgeben. Da der Quellcode - durch den bei der MPL vorhandenen Copyleft-Effekt - von dem Recht ausgenommen bleibt, ihn unter einer anderen Lizenz als der MPL zu vertreiben, ist diese Regelung von geringem wirtschaftlichem Interesse⁴⁹³, weil hierbei im Endeffekt nur die Dienstleistung vom Kompilieren des Quellcodes angeboten werden kann. Wirtschaftlich interessanter wird der Support des Programms oder die Verknüpfung der Objektform mit einem anderen (proprietären) Programm sein.

Im Gegensatz zu den Open Source Lizenzen mit strengem Copyleft-Effekt erlauben die MPLartigen Lizenzen den Vertrieb ihrer Programme mit anders lizenzierten Programmen als Ganzes. Ziffer 3.7 MPL erlaubt dies unter der Voraussetzung, dass alle anderen Lizenzbestimmungen der MPL erfüllt werden. Die Genehmigung für diese „Larger Works“ erklärt sich aus der Geschichte des Mozilla Browsers, der wie erwähnt auf proprietären Code zurückgreift und es daher ohne diese Bestimmung nicht möglich wäre, den Browser zu vervielfältigen und zu verbreiten. Die Unterscheidung von

⁴⁹³ Jaeger/Metzger, Open Source Software 66.

Weiterentwicklungen, die in Ziffer 1.9 MPL unter „Modifications“ geregelt sind, von den als „Larger Works“ bezeichneten, zusammengeführten und jeweils eigenständigen Werken, findet nach der MPL nur nach formalen Gesichtspunkten statt⁴⁹⁴. Nach österreichischem Recht können hierbei die Kriterien, die eine Miturheberschaft iSd § 11 UrhG und jene, die eine Teilurheberschaft iSd § 1175 ABGB begründen, angewendet werden⁴⁹⁵.

Für den österreichischen Rechtsraum eher unbedeutend ist Ziffer 4 MPL, die jene Fälle behandelt, in denen eine wortgetreue Erfüllung der MPL, sei es aufgrund der nationalen Rechtslage oder einer gerichtlichen Entscheidung, nicht möglich ist. Danach muss die Lizenz „*to the maximum extent possible*“ erfüllt werden. Diese Bestimmung begründet sich ebenfalls aus der Entstehungsgeschichte des Browsers, wonach im Browser enthaltene Kryptographiebestandteile in den USA aufgrund eines Gesetzes nicht freigegeben werden durften.

Ziffer 6 MPL regelt die Lizenzhoheit der Mozilla Foundation, der Nachfolgesellschaft der in der Lizenz genannten „Netscape Communications Corporation“ (Ziffer 6.1 MPL) und die Anwendbarkeit von neuen Lizenzversionen (Ziffer 6.2 MPL), die danach subsidiär neben älteren Lizenzversionen stehen können.

In Ziffer 7 MPL findet man den für Open Source Software üblichen Gewährleistungsausschluss, der allerdings von dem in Ziffer 9 MPL beinhalteten Haftungsausschluss getrennt ist. Beide Ziffern werden im Rahmen einer noch folgenden Gesamtbetrachtung behandelt⁴⁹⁶.

⁴⁹⁴ Jaeger/Schulz, NOW Gutachten 49.

⁴⁹⁵ Vgl dazu Kap 3.2.2.2.

⁴⁹⁶ Siehe Kap 6.

Ziffer 8 MPL enthält eine Auflösungsklausel, die die Einhaltung der Lizenzbestimmungen durch den Lizenznehmer mittels einer auflösenden Bedingung mit dinglicher Wirkung iSd §§ 696 und 704 ABGB überwacht. Nach Ziffer 8.1 MPL beendet eine Verletzung der Lizenz das Vertragsverhältnis 30 Tage nach bekannt werden ex tunc. Die Ziffern 8.2 und 8.3 MPL gehen dabei detailliert auf die Auflösungstatbestände ein, die entstehen, wenn ein Lizenznehmer gegen einen Rechteinhaber patentrechtlich vorgeht. Im Falle einer Auflösung des Lizenzvertragsverhältnisses bleiben die Sublicenzverhältnisse aufrecht. Diese Tatsache begründet sich wie bei der GPL daraus, dass weitere Lizenznehmer in einem Kettenlizenzverhältnis ihrer gültig zustande gekommen Rechtsgrundlage beraubt würden und sich darum bemühen müssten, mit dem „Kettenglied“ vor dem „Verletzer“ neu zu kontrahieren.

Ziffer 11 MPL enthält eine Gerichtsstandsklausel, die Lizenz nach dem Recht des US-Bundesstaates New York zu bewerten, wenn zwischen Lizenzgeber und Lizenznehmer kein anderer Gerichtsstand vereinbart wurde⁴⁹⁷.

5.6.1.3. Kompatibilität

Die MPL ist mit der GPL inkompatibel⁴⁹⁸. Das ergibt sich aus den Rechten, die die MPL in Zusammenhang mit den „Larger Works“ einräumt. Die MPL ist bezüglich ihrer Kompatibilität flexibler als die GPL, da sie eine Verbindung von MPL-lizenzierten Programmen mit proprietären Programmen zu einem einheitlichen Ganzen nicht nur gestattet, wenn das Gesamtprogramm dabei wieder unter die MPL gestellt werden muss, sondern auch die „Larger Works“ toleriert. Dieses Merkmal macht sie wirtschaftlich interessant, wohingegen die Erlaubnis, den Objektcode unter einer anderen Lizenz als der

⁴⁹⁷ Zur Zuständigkeit siehe Kap 4.1.

⁴⁹⁸ Siehe auch <http://www.fsf.org/licensing/licenses/index.html> und <http://www.mozilla.org/MPL/relicensing-faq.html>.

MPL zu vertreiben zwar bequem, aber wirtschaftlich von minderm Interesse ist.

Da die MPL die GPL aber keineswegs verdrängen, sondern lediglich eine Alternative bieten will, wurde 2001 damit begonnen, den Browser ebenfalls unter einer anderen Lizenz herauszubringen. Das Ergebnis lag Anfang 2004 vor. Die Betaversion von Mozilla 1.7 konnte unter eine dreifache Lizenz, bestehend aus MPL, GPL und LGPL, gestellt werden. Zirka 450 virtuelle Unterschriften mussten dafür eingeholt werden, was einmal mehr die Vielseitigkeit und Größe einer Open Source Software demonstriert⁴⁹⁹. Ein Lizenznehmer hat nun die Wahl, unter welche Lizenz er seine Weiterentwicklung stellen möchte. Für eine proprietäre Verwendung kommen lediglich MPL und LGPL in Frage.

5.7. Sonstige Lizenzen mit beschränktem Copyleft-Effekt

Die nachfolgenden Lizenzen zeichnen sich ebenfalls durch einen beschränkten Copyleft-Effekt aus, allerdings sind nur die „GNU Library General Public License“, die „Motosoto Open Source License“, die „Sybase Open Watcom Public License“ und die „wxWindows Library License“ von der OSI approbiert worden:

1. Bremer Lizenz für freie Softwarebibliotheken (OSCI-Lizenz) (v1.0)
2. Cougaar Open Source License Agreement, (Einordnung umstritten)
3. Hi-Potent Open Source License
4. **GNU Library General Public License (LGPL) (v2.0 und 2.1)**
5. Motosoto Open Source License (v0.9.1)
6. **Sybase Open Watcom Public License (v1.0)**
7. **wxWindows Library License (v3.0)**

⁴⁹⁹ <http://www.pro-linux.de/news/2004/6344.html>.

Da die „Motosoto Open Source License“ den „Community Portal Server“⁵⁰⁰ als Hauptanwendung hat und daher sehr speziell Anwendung findet, wird in diesem Rahmen nur auf die „Sybase Open Watcom Public License“ und die „wxWindows Library License“ eingegangen:

5.7.1. Die GNU Library General Public License

Die GNU Library General Public License (LGPL), die „kleine Schwester der GPL“ wurde ebenfalls von der FSF entwickelt und soll insbesondere die Fälle abdecken, in denen ein Programm dynamisches Linken verwendet⁵⁰¹. Hierzu gehören insbesondere Programmbibliotheken, so genannte „Libraries“, deren Einbindung man auch Lizenznehmern proprietärer Programme erlauben will. Anders als bei der GPL muss das proprietäre Programm bei der Einbindung aber nicht unter die LGPL gestellt werden⁵⁰².

5.7.1.1. Hintergrund

Als 1990 deutlich wurde, dass die GPL in einigen Fällen – meist für die bereits erwähnten Programmbibliotheken – zu restriktiv war, wurde eine „gelockerte“ Lizenz mit dem Namen „GNU Library General Public License“ (LGPL) im Juni 1991 veröffentlicht. Dies geschah zeitgleich mit der zweiten Version der GPL, um deren Bedeutung zu unterstreichen. Mit der Version 2.1 wurde die Lizenz 1999 in „*GNU Lesser GPL*“ umbenannt⁵⁰³.

⁵⁰⁰ <http://sourceforge.net/projects/mtstcops/>.

⁵⁰¹ Bei der dynamischen Einbindung von Library-Routinen werden diese nur während der Laufzeit des Executables Teil des Objectcodes im Gegensatz zu statischen Einbindungen bei denen sie ständiger Bestandteil sind.

⁵⁰² Siehe auch Metzger, Freiheit den Bibliotheken, http://www.ifross.de/ifross_html/art9.html.

5.7.1.2. Lizenzmerkmale

In der Präambel der LGPL wird der Lizenznehmer darauf hingewiesen, sich im Einzelfall zu überlegen, sein Werk bzw. seine „Library“ unter die LGPL oder unter die GPL zu stellen. Dahinter steckt der Wunsch der FSF, vorrangig die GPL zu verwenden und die LGPL als Kompromisslösung anzusehen⁵⁰⁴. Ansonsten stimmt die LGPL mit der GPL textlich und inhaltlich größtenteils überein.

Die LGPL gewährt gem Ziffer 1 LGPL hinsichtlich der Nutzungsrechte die gleichen Befugnisse wie die GPL⁵⁰⁵. Ein Lizenznehmer darf daher das Programm verändern, vervielfältigen und lizenzgebührenfrei weiterverbreiten, wobei die Urhebervermerke nicht verändert werden dürfen und eine Kopie der Lizenz bei der Verbreitung stets mitgegeben werden muss.

Ziffer 2 LGPL entspricht ebenfalls Ziffer 2 GPL und gestattet dem Lizenznehmer eine unter der LGPL lizenzierte Programmbibliothek zu verändern, wobei die Veränderung in der Datei entsprechend gekennzeichnet werden muss, um diese veränderte Programmbibliothek wiederum unter der LGPL weiterzuverbreiten.

Ziffer 3 LGPL enthält eine „devote Klausel“ gegenüber der GPL, wonach eine Kopie des Programms unter die strengere GPL gestellt werden kann. Hierzu müssen alle Hinweise, die auf die LGPL verweisen, auf die GPL umgeändert werden. Dieser Vorgang ist gem Ziffer 3 Abs 2 LGPL irreversibel, sodass zukünftige Kopien dann nur noch unter der GPL lizenziert werden dürfen. Des Weiteren muss Ziffer 3 LGPL so verstanden werden, dass ein

⁵⁰³ <http://www.gnu.org/licenses/lgpl.html>.

⁵⁰⁴ <http://www.fsf.org/philosophy/why-not-lgpl.html>

⁵⁰⁵ *Jaeger/Metzger*, Open Source Software 50; *Jaeger/Schulz*, NOW Gutachten 38.

Lizenzgeber seine Einwilligung gibt, ein ursprünglich unter der LGPL stehendes Programm auch unter der GPL weiterlizenzieren zu dürfen⁵⁰⁶.

Ziffer 4 LGPL regelt korrespondierend mit Ziffer 3 GPL die Befugnis des Lizenznehmers, das Programm im Objektcode zu verbreiten. Die Ziffern 5 und 6 LGPL beinhalten das Herzstück der Lizenz, das heißt die Möglichkeit der Kombination von Bibliotheken, die unter der LGPL lizenziert sind, mit proprietären Programmen und machen den eigentlichen Sinn der LGPL im Unterschied zur GPL deutlich. Zunächst besagt Ziffer 5 Abs 1 LGPL, dass ein Programm, welches keinen abgeleiteten Teil einer unter der LGPL lizenzierten Programmbibliothek enthält, aber so geschrieben ist, dass es auf eine LGPL-Programmbibliothek zugreift, als eigenständiges Werk zu sehen ist, welches nicht von den Lizenzbestimmungen der LGPL erfasst wird. Diese Abgrenzung soll verdeutlichen, dass proprietäre Programme zwar auf die LGPL-Programmbibliotheken zugreifen dürfen⁵⁰⁷, deshalb aber nicht ihre Proprietät aufgeben müssen. Andererseits behalten die LGPL- Programmbibliotheken ihre Freiheit iSd Open Source Gedankens und dürfen ihrerseits nicht in das proprietäre Programm so eingegliedert werden, dass Ihr Quellcode dort verschlossen weiterbenutzt wird. Ziffer 5 Abs 2 LGPL sagt in diesem Zusammenhang aus, dass auch bei dynamischen Linken, soll heißen bei der Verbindung von zugreifenden Programmen mit der LGPL- Programmbibliothek zu einem Executable, ein „*executable that is derivative of the Library*“ entsteht. Die damit verbundenen Konsequenzen finden sich in Ziffer 6 LGPL.

Da die LGPL das „Copyleft“ nicht vollständig aufgeben, sondern lediglich lockern will, stellt Ziffer 6 LGPL Verpflichtungen auf, nach denen die Verbreitung des zugreifenden Programms unter einer selbst gewählten Lizenz möglich ist und damit Ausnahmen von der in Ziffer 2 c LGPL gemachten Verpflichtung bilden, das Programm als Ganzes unter der LGPL verbreiten zu

⁵⁰⁶ Jaeger/Metzger, Open Source Software 51.

⁵⁰⁷ Siehe auch Ziffer 0 Abs 4 Satz 2 LGPL „*The act of running a program using the library is not restricted.*“

müssen. Nach Ziffer 6 Abs 1 LGPL dürfen daher auch dann Kombinationen aus LGPL-Programmbibliotheken und proprietären Programmen zusammen verbreitet werden, wenn einem Lizenznehmer das Recht eingeräumt wird, Veränderungen vorzunehmen sowie das Programm zur Fehlerbehebung zu dekompile. Hinsichtlich des Dekompilierens kann gesagt werden, dass das österreichische Urheberrecht in § 40e UrhG einem Nutzer schon weitreichende Rechte für einen Dekompilierungsvorgang einräumt und diese daher nur wenig durch Ziffer 6 Abs 1 erweitert werden. Ziffer 6 Abs 2 LGPL stellt sicher, dass jede Kopie des Programms einen auffälligen Hinweis auf die Nutzung der LGPL- Programmbibliothek enthält und eine Kopie der LGPL beigefügt werden muss. Daneben muss eine der folgenden Bedingungen wahlweise erfüllt werden, damit Lizenznehmer weiterhin die LGPL-Programmbibliotheken anpassen dürfen und gleichzeitig eine Nutzung mit dem zugreifenden Programm gestattet bleibt⁵⁰⁸:

- Mitlieferung des Quellcodes der LGPL-Programmbibliothek sowie des Objektcodes des zugreifenden Programms (natürlich darf auch der Quellcode des zugefügten Programms mitgeliefert werden). Dies muss in einer Weise geschehen, die dem Lizenznehmer erlaubt, die LGPL-Programmbibliothek zu modifizieren und die veränderte Version mit dem zugreifenden Werk benutzen zu können, um ein neues funktionierendes Executable zu erhalten
- Verwendung eines geeigneten „shared-Library-Mechanismus“ von LGPL-Programmbibliothek zu dem zugreifenden Programm. Die technischen Anforderungen der Ziffer 6b LGPL an den Begriff „suitable“ bestehen darin, dass der Mechanismus
 - während der Laufzeit eine im System des Lizenznehmers bereits vorhandene Kopie der LGPL-Programmbibliothek benutzt, anstatt Funktionen der LGPL-Programmbibliothek in die Executable zu kopieren

⁵⁰⁸ Jaeger/Schulz, NOW Gutachten 40.

und

- mit einer veränderten Version der LGPL-Programmbibliothek fehlerfrei funktioniert unter der Voraussetzung, dass alte und neue LGPL-Programmbibliotheken schnittstellenkompatibel sind.
- Beifügung eines für drei Jahre gültigen schriftlichen Angebots zur Lieferung der Programme in Form der in der ersten Alternative bereits beschriebenen Art und Weise (Ziffer 6a LGPL). Dafür darf ein Entgelt verlangt werden, das aber nur die Materialkosten abdecken darf. Sollte das Programm über das Internet vertrieben worden sein, können diese Materialien auch dort zum Download zur Verfügung gestellt werden (Ziffer 6d LGPL)

Ziffer 7 LGPL beschäftigt sich mit der Kombination von einer LGPL-Programmbibliothek und einer proprietären Programmbibliothek zu einer gemeinsamen Programmbibliothek. Damit beide zwar gemeinsam vertrieben werden können, die proprietäre Programmbibliothek dabei aber nicht ebenfalls unter der LGPL lizenziert werden muss, bedarf es folgender zwei Voraussetzungen:

1. Beifügung der LGPL-Programmbibliothek in isolierter Form als Kopie zu der kombinierten Programmbibliothek. Die Verbreitung in getrennter Form muss dabei gestattet sein⁵⁰⁹.
2. Anbringung eines deutlichen Hinweises auf die Verwendung einer LGPL-Programmbibliothek und darauf, wo diese Bestandteile aufzufinden sind.

Die Ziffern 8-14 LGPL entsprechen den Ziffern 4-10 GPL. Die LGPL endet ebenfalls mit einem Haftungs- und Gewährleistungsschluss und einem Hinweis, wie die Lizenz am besten auf eine Programmbibliothek angewendet werden kann.

⁵⁰⁹ Dies ergibt sich aus Ziffer 7 Punkt 1 LGPL iVm Ziffer 7 Abs 1 LGPL.

5.7.1.3. Kompatibilität

Wie ausgeführt wurde besteht hinsichtlich der GPL Kompatibilität mit der LGPL, da diese in Ziffer 3 LGPL eine „devote Klausel“ zugunsten der GPL enthält. Die gegebene Kompatibilität mit proprietären Programmen wurde unter den Ziffern 5-7 LGPL ausreichend behandelt, sodass an dieser Stelle nicht weiter darauf eingegangen wird.

5.7.2. Die Apple Public Source License

Seit dem Jahr 2000 hat die Firma Apple das Betriebssystem „Darwin“ – auf dem Mac OS X basiert – und andere Programme⁵¹⁰ unter die eigens dafür entworfene „Apple Public Source License“ (APSL) gestellt. Erst Version 2.0 aus dem Jahr 2003 wurde auch von der FSF anerkannt⁵¹¹, obwohl Version 1.0 bereits von der OSI zertifiziert wurde. Die Lizenz wird oft fälschlicherweise den Lizenzen mit strengem Copyleft-Effekt zugeordnet⁵¹². Sie gehört jedoch zu den Open Source Lizenzen mit beschränktem Copyleft-Effekt, da die APSL eine Verbindung von unter ihr lizenziertem Code mit proprietärem Code erlaubt.

5.7.2.1. Hintergrund

Die Firma Apple sah Ende der 1990er Jahre Vorteile darin, ihre Softwareprodukte Open Source anzubieten. Nach Sichtung mehrerer Open Source Lizenzen – wie die GPL, die BSD-Lizenz, die Apache License, die NPL, die MPL und die Sun Community Source License – entschied Apple, seine

⁵¹⁰ <http://developer.apple.com/darwin/projects/>.

⁵¹¹ <http://www.heise.de/newsticker/meldung/39248-APSL.htm>.

⁵¹² http://www.ifross.de/ifross_html/lizenzcenter.html.

Software unter eine eigene Lizenz zu stellen. Diese wurde von der FSF aus unterschiedlichen Gründen nicht anerkannt⁵¹³. Da spätere Versionen ebenfalls Anlass für Kritik boten⁵¹⁴, entschied Apple die Version 2.0 mit Hilfe von *Richard Stallman* und der FSF herauszubringen. Diese wurde letztendlich von der FSF anerkannt, allerdings konnten nicht alle strittigen Punkte von Seiten der FSF beseitigt werden⁵¹⁵.

5.7.2.2. Lizenzmerkmale

Die Lizenz gleicht in ihrem Inhalt sehr der MPL und beginnt mit einer Klausel bezüglich der Annahme des Lizenzvertrags. Danach wird ein potentieller Lizenznehmer darauf hingewiesen, den Lizenztext der APSL gründlich durchzulesen bevor er unter der APSL lizenzierte Software herunterlädt. Die Klausel macht Sinn, wenn man die Diskussion zu den „*Click-Wrap*“ betrachtet.⁵¹⁶

Ziffer 1 APSL definiert unter anderem, dass die Lizenz nur auf solche Software Anwendung findet, welche von Apple veröffentlicht und darüber hinaus als „Original Code“ gekennzeichnet wurde. Damit kann die Lizenz ausschließlich für Software, die ursprünglich von Apple entwickelt worden ist, genutzt werden. Ziffer 1.5 APSL enthält eine Definition bezüglich „Larger Works“ und zeigt einem Lizenznehmer schon früh, dass die Möglichkeit besteht Software, die unter der APSL lizenziert ist, mit anders lizenzierter Software zu verbinden.

⁵¹³ <http://www.gnu.org/philosophy/historical-apsl.html>.

⁵¹⁴ <http://perens.com/Articles/APSL.html>.

⁵¹⁵ <http://www.gnu.org/philosophy/apsl.html>.

⁵¹⁶ Siehe insbesondere: *Staudegger* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 86; *Jakob*, Freiheit und Software in *Schweighofer/Menzel/Kreuzbauer* (Hrsg), Auf dem Weg zu ePerson. Aktuelle Fragestellungen der Rechtsinformatik (2001) 331 (337).

Ziffer 2 APSL räumt dem Lizenznehmer ein weltweites, kostenloses, nicht ausschließliches und alle hinsichtlich der Software von Apple gehaltenen Patent- und Urheberrechte umfassendes Nutzungsrecht ein,

1. unveränderten Quellcode gem Ziffer 2.1 APSL zu nutzen, zu kopieren, aufzuführen und zu jeglichem Zweck zu verbreiten, solange alle Urheberrechtsvermerke, die im Quellcode enthalten sind (Ziffer 2.1a APSL) und der Lizenztext unverändert (Ziffer 2.1b APSL) mitverbreitet werden.
2. unveränderten Quellcode gem Ziffer 2.2 APSL weiterzuentwickeln und diesen weiterentwickelten Code zu nutzen, zu kopieren, aufzuführen und zu jeglichem Zweck zu verbreiten, solange die in den Ziffern 2.1a und b APSL genannten Bedingungen erfüllt werden (Ziffer 2.2.a APSL) und darüber hinaus alle Weiterentwicklungen mit Datum und Art der Veränderung in „Exhibit A“ dokumentiert und mitverbreitet werden (Ziffer 2.2b APSL). Bei der Verbreitung von weiterentwickeltem Code muss sichergestellt werden, dass der Quellcode verfügbar bleibt. Dieser darf nur unter der APSL lizenziert werden und muss entweder für den Zeitraum des Vertriebs und mindestens für 12 Monate nach erstmaligem Vertrieb verfügbar sein (Ziffer 2.2c APSL).

Die Verbreitung des Programms in Objektform wird in Ziffer 2.3 APSL geregelt, wonach jedem Lizenznehmer mitgeteilt werden muss, dass er den Quellcode unter den Bedingungen der APSL erhalten kann und wo dieser zu erhalten ist. Danach folgt in Ziffer 2.4 APSL eine Freizeichnungsklausel gegenüber Dritten, wonach jeder Lizenznehmer bei der Weiterverbreitung darauf zu achten hat, dass nur jene Programmteile weitergegeben werden, für die dem Lizenznehmer die Nutzungs- und Verwertungsrechte eingeräumt wurden.

Die Ziffer 3 APSL enthält eine Bestimmung mit Copyleft-Effekt, da sie den Lizenznehmer dazu verpflichtet, bei der Weiterverbreitung die gleichen Rechte einzuräumen, die er selbst durch die APSL erhalten hat. Der Copyleft

wird nachfolgend jedoch durch Ziffer 4 APSL eingeschränkt. Demnach darf der Lizenznehmer Programme, die unter der APSL lizenziert worden sind mit anders lizenzierten Programmen verbinden und als gemeinsames Produkt vertreiben. Diese Einschränkung des Copyleft-Effekts disqualifiziert die APSL als Open Source Lizenz mit strengem Copyleft-Effekt, da dieser nicht konsequent angewendet werden muss.

Apple weist in Ziffer 5 APSL darauf hin, dass bei weiterentwickelten Programmen oder „Larger Works“ Patentrechte – die über die in der APSL eingeräumten hinausgehen – benötigt werden könnten, welche mit Apple in einer eigenen Lizenz vereinbart werden müssten. Ziffer 6 ist ebenfalls als unverbindliche Vorgabe zu sehen, da die Lizenz einem Lizenznehmer gestattet, mit einem Dritten Sonderleistungen – wie bspw Support – zu vereinbaren. Dabei muss der Lizenznehmer klarstellen, dass er auf eigene Verantwortung handelt.

Ziffer 7 APSL regelt die Lizenzhoheit von Apple und untersagt Dritten, jegliche Änderungen am Lizenztext vorzunehmen. Die folgenden Ziffern 8 und 9 APSL enthalten den Haftungs- und Gewährleistungsausschluss.

Ein besonderes Merkmal befindet sich in Ziffer 10 APSL, welches die markenrechtliche Behandlung der durch Apple eingetragenen Marken regelt. Demnach räumt die APSL einem Lizenznehmer keinerlei Rechte ein, die in Verbindung mit den in Ziffer 10 APSL genannten Marken stehen⁵¹⁷.

In Ziffer 11 APSL behält sich Apple das Recht vor, Weiterentwicklungen am Programm unter eine andere Lizenz zu stellen oder diese gar nicht zu lizenzieren. Durch diese Regelung wird ebenfalls der Copyleft-Effekt

⁵¹⁷ Zur markenrechtlichen Betrachtung siehe Kap 7.

aufgeweicht, da einmal unter der APSL lizenzierte Software nicht zwangsweise unter dieser weiter lizenziert werden muss.

Die Auflösungsbedingungen der Lizenz finden sich in Ziffer 12 APSL und sind als auflösende Bedingung mit dinglicher Wirkung iSd §§ 696 und 704 ABGB konzipiert. Nach Ziffer 12.1a APSL beendet eine Verletzung der Lizenz das Vertragsverhältnis 30 Tage nach bekannt werden ex tunc; sollte die Lizenz iSd Ziffer 13.5 APSL mit einer nationalen Rechtslage unverträglich sein, endet das Lizenzverhältnis gem Ziffer 12.1b APSL mit sofortiger Wirkung ex tunc.

Die in den Unterpunkten der Ziffer 13 APSL behandelten Punkte haben teilweise nur für das US-amerikanische Hoheitsgebiet Geltung (Ziffer 13.1 APSL). Die anderen behandeln gesellschaftsrechtliche Aspekte (Ziffer 13.2 APSL), die Entwicklung von eigenständigen Werken durch Apple, die auf unter der APSL lizenzierter Software basieren (Ziffer 13.3 APSL)⁵¹⁸, den Fall, dass die Lizenz mit einer nationalen Rechtslage unverträglich ist (Ziffer 13.5 APSL) und den Gerichtsstand (Ziffer 13.6 und 13.7 APSL). Der Lizenztext endet mit „*Exhibit A*“.

5.7.2.3. Kompatibilität

Die APSL ist mit der GPL nicht kompatibel, da sie in Ziffer 4 APSL die Verbindung von unter der APSL lizenzierter Software mit anders lizenzierter Software zu einem Gesamtwerk gestattet. Die Gründe für die Inkompatibilität decken sich dadurch mit denen der MPL.

⁵¹⁸ Damit wird auch in der APSL die Eigenständigkeit von Weiterentwicklungen behandelt, wobei die APSL speziell auf den Fall der Parallelentwicklung eingeht.

5.7.3. Die Sybase Open Watcom Public License

Das „Open Watcom“-Projekt“ hat eine lange Geschichte hinsichtlich der Entstehung seines Programms. Die Freigabe als Open Source Software geschah dagegen erst 2003⁵¹⁹.

5.7.3.1. Hintergrund

Der Compiler „Watcom“ beruht auf einem im Jahre 1965 von Studenten der University of Waterloo (Kandada) entwickelten FORTRAN Compiler (WATFOR), welcher 1988 in der Programmiersprache C weiterentwickelt und mit dem Namen „Watcom C 6.0“ versehen wurde. Der „Watcom code generator“ unterstützte von da an sowohl die in C als auch die FORTRAN geschriebenen Programme und war Plattform übergreifend einsetzbar (so bspw für DOS, Windows, OS/2, und Windows NT). Aufgrund dessen wurde Watcom in den 1990er Jahren sehr populär. Im Zeitraum 1993-1996 wurden fast alle Spiele für das Betriebssystem DOS mit dem Watcom C entwickelt, darunter so bekannte Titel wie „DOOM“, „Descent“ oder „Duke Nukem 3D“⁵²⁰. 1994 wurde Watcom von der Firma „Powersoft“ aufgekauft, welche wiederum 1995 mit „Sybase“ fusionierte⁵²¹. Der folgende Versuch, Watcom kommerziell zu vermarkten scheiterte, sodass 2003 die Watcom C, C++ und FORTRAN Compiler mit dem Namen „Open Watcom“ versehen und unter die „Sybase Open Watcom Public License“ (WPL) gestellt wurden.

⁵¹⁹ <http://www.golem.de/0302/23880.html>.

⁵²⁰ <http://www.openwatcom.com/index.php/History>.

⁵²¹ http://www.sybase.com/about_sybase/history.

5.7.3.2. Lizenzmerkmale

Die Einleitung der Lizenz beginnt mit einem Nutzungsausschluss, welcher die Nutzung ohne vorherige Zustimmung zum Lizenztext untersagt. Gleichzeitig wird festgestellt, dass eine Zustimmung konkludent durch die Nutzung erfolgt. Dieser Vorbehalt kann nur außerhalb der gesetzlich zugesicherten Nutzungsrechte iSd § 40d UrhG Geltung finden, daher wird auch im letzten Satz der Einleitung empfohlen, das Programm nicht zu benutzen, sollte man dem Lizenzinhalt nicht zustimmen können.

Ziffer 1.1 bis 1.10 WPL enthalten Begriffsdefinitionen, die jenen der MPL ähnlich sind. Dabei wird in Ziffer 1.1 WPL darauf hingewiesen, dass die Lizenz keine generelle Anwendung auf Programme findet, sondern ausschließlich für die Open Source Versionen des „Sybase Watcom C“, „C++“ und „FORTRAN Compiler“ bestimmt ist.

Aus diesem Grund tritt als Lizenzgeber Sybase selbst auf und räumt einem Lizenznehmer in Ziffer 2 WPL ein weltweites, kostenloses, nicht exklusives Nutzungsrecht für die Software ein, wobei die von Sybase gehaltenen Patent- und Urheberrechte inbegriffen sind. Danach darf der Lizenznehmer die Software benutzen, kopieren, vorführen, weiterentwickeln und verändert oder unverändert verbreiten, solange dies ausschließlich zur persönlichen Nutzung oder zum Zweck der internen Untersuchung oder Fortentwicklung dient und der Lizenznehmer darüber hinaus den Urhebervermerk sowie den Lizenztext unverändert bei Weiterverbreitung mitliefert. Ziffer 2c WPL empfiehlt zusätzlich, diese Informationen in einer „*click-wrap*“-Form bekannt zu machen. Diese Lösung ist wie bereits erwähnt⁵²² nicht

⁵²² Vgl Kap 4.2.

sehr glücklich, da die „Click-Wrap“-Verträge im Open Source Bereich nicht zur Regel werden sollten und ihre Gültigkeit überwiegend verneint wird⁵²³.

Über die in Ziffer 2.1 WPL gewährten Rechte hinaus räumt Ziffer 2.2 WPL das Recht ein, die Software zu nutzen, zu kopieren, vorzuführen, weiterzuentwickeln und auch außerhalb der persönlichen Nutzung oder zum Zweck der internen Untersuchung weiterzuverbreiten, solange die Änderungen in der Beilage „*Exhibit A*“ mit einem bedeutsamen Urhebervermerk und dem Datum der Änderung versehen werden. Der Quellcode muss gem Ziffer 2.2c WPL öffentlich zugänglich sein, solange der veränderte Code vom Lizenznehmer weiterverbreitet wird oder - falls dieser die Weiterverbreitung eingestellt hat - zumindest für einen Zeitraum von 12 Monaten. Sollte die Software nur im Objektcode ausgeliefert werden, muss der Hinweis erfolgen, dass der Quellcode unter den Lizenzbedingungen der WPL ebenfalls verfügbar ist.

Ziffer 2.2e erlaubt es einem Lizenznehmer, die Software in Objektform unter einer selbst gewählten Lizenz zu vertreiben, welche jedoch keinen geringeren Schutz für Sybase oder andere Mitlizenzgeber bieten darf. Diese Klausel hat - wie Ziffer 3.6 MPL - kaum praktischen Nutzen, da ein Vertrieb wirtschaftlich uninteressant ist, wenn die Software im Quellcode frei verfügbar ist. Allerdings gestattet sie dem Lizenznehmer den Vertrieb der Software gegebenenfalls mit einem Support-Vertrag zu verbinden, der nur den Vertreiber in die Pflicht nehmen darf.

Ziffer 2.3 WPL erklärt, dass Sybase keine Verantwortung für Patent- oder Urheberrechtsverletzungen übernimmt. Lizenznehmer sind demnach selbst dafür verantwortlich, benötigte Rechte vom Rechteinhaber eingeräumt zu bekommen. Diese Freizeichnungsklausel kann nach österreichischem Recht

⁵²³ *Wiebe/Prändl*, ÖJZ 2004, 635 mwA.

keine generelle Wirkung entfalten, da ein solcher Ausschluss der Verantwortung gegen die Bestimmungen des § 879 Abs 3 ABGB verstößt. Die Intention des Lizenzgebers Sybase kann man nur dahingehend interpretieren, dass Sybase keine Verantwortung übernehmen will, wenn ein Lizenznehmer durch eine Weiterentwicklung Patent- oder Urheberrechte von Dritten verletzt.

Ziffer 3 WPL stellt sicher, dass der Lizenznehmer Sybase und jedem Dritten die gleichen Rechte gewährt, die er selbst durch die WPL erhalten hat. Wie in der MPL sind gem Ziffer 4 WPL, „*Larger Works*“ erlaubt⁵²⁴, wobei nur die durch die WPL gewährten Patent- und Urheberrechte von Sybase eingeräumt werden. Darüber hinaus benötigte Rechte können von Sybase unabhängig (proprietär) erworben werden (Ziffer 5 WPL).

Ziffer 6 WPL räumt dem Lizenznehmer das Recht ein, bei der Weiterverbreitung der Software Haftung und Gewährleistung zu übernehmen und Ziffer 7 WPL stellt die Lizenzhoheit von Sybase sicher.

Die Ziffern 8 und 9 WPL beinhalten die Haftungs- und Gewährleistungsausschlüsse⁵²⁵.

Ziffer 11 WPL sichert jedem Miturheber an der unter der WPL stehenden Software die Urheber- und Verwertungsrechte zu. Sybase behält sich hier die Ausnahmeregelung vor, die von Sybase entwickelten Programmänderungen nicht mehr unter die WPL stellen zu müssen. Diese Regelung mag für Sybase interessant sein, erweist sich aber als Unsicherheitsfaktor für das Projekt, da Sybase signifikante Weiterentwicklungen nicht mehr freigeben muss.

⁵²⁴ Vgl Kap 5.6.1.2.

⁵²⁵ Vgl Kapitel 6.

Ziffer 12 WPL enthält eine Auflösungsklausel, ähnlich Ziffer 8 MPL, welche die Einhaltung der Lizenzbestimmungen durch den Lizenznehmer mittels einer auflösenden Bedingung mit dinglicher Wirkung iSd §§ 696 und 704 ABGB überwacht. Nach Ziffer 12.1 WPL beendet eine Verletzung der Lizenz das Vertragsverhältnis 30 Tage nach bekannt werden ex tunc.

Ziffer 13 WPL enthält unterschiedliche Regelungen, welche teilweise nur für den US-amerikanischen Rechtsraum von Interesse sein dürften (Ziffer 13.1 WPL). Ziffer 13.2 WPL bestimmt gesellschaftsrechtliche, Ziffer 13.3 WPL wettbewerbsrechtliche und Ziffer 13.4 WPL formale Auslegungsregeln der Lizenz. Ziffer 13.5 WPL erlaubt eine Einschränkung des Lizenztextes, sollten nationale Gesetze oder Rechtsprechung dies erfordern. Ziffer 13.6 WPL setzt den „Northern District of California“ als Gerichtsstand für Streitigkeiten mit Sybase fest und schließt dabei die Anwendung des UN-Kaufrechts explizit aus. Durch Ziffer 13.7 WPL wird die Lizenz dem US-amerikanischen Recht und dem Recht des US-Bundesstaates Kalifornien unterstellt. Für Quebec wird eine Ausnahme bestimmt, nach der ein Lizenznehmer bestätigen muss, dass es seine Absicht war, die Lizenz in englischer Sprache abzuschließen. Die Lizenz endet mit dem „*Exhibit A*“, welches den Urhebervermerk enthält.

5.7.3.3. Kompatibilität

Die WPL ist mit der GPL nicht kompatibel. Dies ergibt sich einerseits aus den gleichen Gründen wie bei der MPL andererseits sind die Bestimmungen der Ziffer 11 WPL mit der GPL keinesfalls zu vereinbaren. Die WPL versucht auch keine Kompatibilität mit der GPL herzustellen, da die Intentionen des Lizenzgebers Sybase andere Ziele zu verfolgen scheinen. Sybase trat – wie Netscape – eine Flucht in den Open Source Bereich an, um dem wirtschaftlichen Untergang zu entgehen. Dabei möchte man sich einen Rückzug in eine proprietäre Umgebung offen halten. Dies ist mit GPL-lizenzierter Software nicht möglich.

5.7.4. Die wxWindows Library License

Die „wxWindows Library License“ (wxWLL) ist eine Open Source Lizenz mit beschränktem Copyleft-Effekt, die zwar auf die LGPL verweist, aber dennoch dem Lizenznehmer weiterreichende Möglichkeiten einräumen will. Da das Projekt in „wxWidgets“ umbenannt wurde, wird die Lizenz ebenfalls in „wxWidgets Library License“ umbenannt werden, was allerdings keinen Einfluss auf den Lizenztext haben wird⁵²⁶.

5.7.4.1. Hintergrund

Das wxWindows-Projekt wurde 1992 an der University of Edinburgh gestartet und war eine C++-Klassenbibliothek zur Plattform übergreifenden Programmierung von Anwendungen mit grafischer Benutzeroberfläche (GUI), welches 2004 aufgrund einer „freundlichen Aufforderung“ durch Microsoft in „wxWidgets“ umbenannt wurde⁵²⁷. Durch die Möglichkeit, Plattform übergreifend Software zu entwickeln und diese dann unter unterschiedlichen Lizenzen verbreiten zu können, hat wxWidgets eine große Fangemeinde aufbauen können.

5.7.4.2. Lizenzmerkmale

Die wxWLL selbst ist sehr kurz gehalten, da sie auf die Bestimmungen der LGPL verweist und nur Ausnahmen zu dieser macht. Sie beginnt mit der Erlaubnis, den Lizenztext selbst zu kopieren und zu verbreiten, aber nicht ändern zu dürfen. Der darauf folgende Abschnitt enthält den direkten Verweis auf die LGPL, wobei die Auswahl der Lizenzversion dem Lizenznehmer

⁵²⁶ <http://www.wxwindows.org/newlicen.htm>.

überlassen bleibt. Danach folgen der Haftungsausschluss und der Versuch, die Garantie für eine fehlerfreie Nutzung zu einem bestimmten Zweck oder die Marktgängigkeit („merchantability“) der Programmbibliothek auszuschließen⁵²⁸. Details zu diesem Ausschluss sollen der LGPL entnommen werden.

Der zweite und letzte Abschnitt der Lizenz umfasst die „*Exception Notice*“, welche die besonderen Lizenzmerkmale der wxWLL enthält. Ziffer 1 wxWLL erklärt, dass die Urheber der Library den in der wxWLL gemachten Ausnahmen zustimmen. Diese Ausnahmen werden in Ziffer 2 wxWLL definiert und räumen dem Lizenznehmer das Recht ein, Objektformen von Programmbibliotheken, die auf der wxWLL basieren, zu nutzen, zu kopieren, zu verlinken, weiterzuentwickeln und zu verbreiten. Dies macht die unter der Lizenz stehenden Programmbibliotheken auch für die proprietäre Nutzung interessant, da bspw. das statische Linken modifizierter oder unmodifizierter Versionen mit kommerziellen Closed Source Anwendungen problemlos möglich ist. Damit die wxWLL mit der GPL und mit der LGPL kompatibel bleiben kann, enthält sie in Ziffer 3 wxWLL eine Demutsklausel zu deren Gunsten. Danach muss der Teil der Lizenz (und damit im Ergebnis die ganze Lizenz) entfernt werden, wenn Teile des Codes kopiert wurden, die unter der GPL oder der LGPL lizenziert wurden. Ziffer 4 wxWLL überlässt dem Lizenznehmer abschließend die Wahl, ob er eine von ihm geschaffene Weiterentwicklung unter der wxWLL verbreitet oder unter eine andere Lizenz stellt. Die Bestimmung, dass der Lizenznehmer, sollte er sich für eine andere Lizenz zur Verbreitung entscheiden, die Ausnahmeregelung der Lizenz zu entfernen hat, kann nur als Hinweis aufgefasst werden, da eine Beibehaltung unweigerlich die fortlaufende Geltung der wxWLL zur Folge hätte.

⁵²⁷ <http://www.heise.de/newsticker/meldung/44891>.

⁵²⁸ Zum Haftungsausschluss siehe Kap 6.

5.7.4.3. Kompatibilität

Die wxWLL ist einerseits mit der GPL und der LGPL andererseits mit proprietären Lizenzen kompatibel⁵²⁹. Dies wird durch die Demutsklausel zu Gunsten der GPL und LGPL erreicht, ohne die eine Kompatibilität nicht möglich wäre. Die Lizenzautoren haben auf die Kompatibilität Rücksicht genommen, um dem Lizenznehmer einen möglichst großen Spielraum zu geben, was die Nutzung weiterentwickelter Werke in Objektform anbelangt.

5.8. Lizenzen mit Wahlmöglichkeiten

Diese Lizenzen sehen unterschiedliche rechtliche Folgen vor, je nachdem wie umfangreich eine Modifikation ist. Außerdem werden dem Lizenznehmer verschiedene Wahlmöglichkeiten eingeräumt, auf welche Weise Weiterentwicklungen verbreitet werden können. Folgende Lizenzen gehören zu den Open Source Lizenzen mit Wahlmöglichkeiten:

1. Artistic License (v. 1.0), (Einordnung umstritten)
2. **Artistic License (v. 2.0)**
3. Clarified Artistic License
4. Keith Devens' Open Source License
5. LaTeX Project Public License (LPPL) (v. 1.2)
6. Microsoft Permissive License (Ms-PL)
7. Physnet Package License
8. SGI Free Software License B (v. 1.0) , (Einordnung umstritten)
9. SGI Free Software License B (v. 1.1)
10. **Sleepycat Software Product License**

⁵²⁹ <http://www.wxwindows.org/newlicen.htm>.

In diesem Bereich werden die „Artistic License“ und die „Sleepycat License“ stellvertretend behandelt.

5.8.1. Die Artistic License

Die „Artistic License“ (AL) sorgte seit ihrer Entstehung für einige Verwirrung in der Open Source Gemeinschaft. Das ging sogar soweit, dass die FSF sie nicht als Open Source Lizenz anerkannte; dies geschah erst ab ihrer aktuellen Version 2.

5.8.1.1. Hintergrund

Die AL wird hauptsächlich für die Programmiersprache „Perl“ benutzt⁵³⁰, die seit 1986 existiert und eine der meist genutzten Programmiersprachen der Welt ist⁵³¹. Die Programmiersprache wird seit 1991 im „Dual Licensing“ unter der AL und der GPL lizenziert. Da die Artistic License in der Version 1 äußerst missverständlich in ihrer Lizenzsprache war und mehrere RFCs⁵³² mit Änderungswünschen vorlagen, wurde im Jahr 2000 die „Clarified Artistic License“ herausgegeben, welche Version 1 abgelöst hat⁵³³.

⁵³⁰ Zum Schutz einer Programmiersprache nach dem UrhG siehe *Jaeger/Metzger*, Open Source Software 71.

⁵³¹ http://sourceforge.net/softwaremap/trove_list.php?form_cat=160.

⁵³² „Request for Change“ bezeichnen Änderungswünsche in Softwareprojekten.

⁵³³ <http://dev.perl.org/perl6/rfc/346.html>.

5.8.1.2. Lizenzmerkmale

Die AL erklärt zu Beginn die Absichten der Lizenzautoren und definiert nachfolgend einige in der AL benutzten Begriffe. Ziffer 1 AL räumt dem Lizenznehmer das Recht zur Vervielfältigung und Verbreitung des nicht weiterentwickelten Quellcodes ein, solange die Urhebervermerke und der Haftungsausschluss ebenfalls mitverbreitet werden.

Die Rechte zur Weiterentwicklung werden dem Lizenznehmer durch die Ziffern 2 + 3 AL eingeräumt. Sie unterscheiden dabei zwischen „Bug- und Portabilityfixes“ und jeder anderen Art von Weiterentwicklung. Während „Bug- und Portabilityfixes“ nicht als Weiterentwicklung und damit als eigenständiges Werk zählen sollen, bestehen für den Urheber jeglicher anderen Weiterentwicklung gem Ziffer 3 AL vier Verpflichtungen, aus denen der Urheber aber nur eine zu erfüllen braucht. Er hat also die Wahl, seine andere Weiterentwicklung,

1. der Public Domain zu übergeben, was aber in Österreich aufgrund der starken urheberrechtlichen Persönlichkeitsrechte nicht möglich ist⁵³⁴ oder diese frei verfügbar iSd Definition „*freely available*“ der AL im Internet anzubieten, bspw durch die Nutzung von FTP-Archivdiensten. Dadurch wird gewährleistet, dass Dritte lizenzgebührenfrei die Weiterentwicklung verwenden dürfen. Weitere Veränderungen an der Weiterentwicklung müssen dadurch nicht zwingend gestattet werden,
2. nur Behörden- oder Firmenintern zu nutzen. Die AL sieht in einem solchen Fall explizit keine Pflicht zur Veröffentlichung⁵³⁵.
3. in Objektform zu verbreiten, wobei jedes weiterentwickelte Executable umbenannt werden muss, um nicht mit denen der

⁵³⁴ Siehe Kap 3.2.2.

⁵³⁵ Siehe dazu auch die Problematik des Öffentlichkeitsbegriffes in Kap 5.4.1.2.1.

ursprünglichen Version in Konflikt zu geraten. Dabei muss deutlich dokumentiert sein, in welcher Weise sich beide Versionen unterscheiden.

4. unter einen gesondert mit dem Urheber zu vereinbarenden Lizenzvertrag zu stellen.

Ziffer 4 behandelt die Verbreitung des unter der AL lizenzierten Programms in Objektform. Dies ist einem Lizenzgeber ebenfalls gestattet, solange er eine der folgenden Auswahlmöglichkeiten berücksichtigt:

1. Mitgabe der ursprünglichen Version des Programms und zugehöriger Programmbibliotheken in Objektform sowie einer Anleitung, wo diese sonst noch zu bekommen ist.
2. Mitgabe des Quellcodes
3. Mitgabe der korrespondierenden Executables in der ursprünglichen Version. Dabei müssen die weiterentwickelten Executables umbenannt, dies mit Verweis auf die Art der Unterschiedlichkeit dokumentiert und mit dem Hinweis versehen werden, wo die ursprüngliche Version zu beziehen ist.
4. Eine gesonderte Vereinbarung mit dem Lizenznehmer getroffen wird.

Ziffer 5 AL stellt sicher, dass auch unter der AL lizenzierte Software nur lizenzgebührenfrei weitergegeben werden darf. Danach dürfen Gebühren ausschließlich für das Kopieren der Software (in angemessener Weise) und sonstige Supportleistungen verlangt werden. In Ziffer 5 AL befindet sich noch eine weitere Änderung gegenüber den Vorgängerversionen der Lizenz. Demnach ist es gestattet, unter der AL lizenzierte Software mit anders (proprietär) lizenzierter Software gemeinsam zu verbreiten. Dabei muss sichergestellt werden, dass auf den AL lizenzierten Teil als solchen hingewiesen wird und eine Lizenzgebühr für diesen ausgeschlossen bleibt.

Die Ziffern 6 und 7 AL behandeln die Zulässigkeit der Einbindung von Skripten und Programmbibliotheken in proprietäre Programme. Ziffer 8 AL untersagt die Nutzung des Namens des Urhebers hinsichtlich des Vertriebs weiterentwickelter Versionen. Die Lizenz endet mit einem kurz gehaltenen Haftungs- und Gewährleistungsausschluss.

5.8.1.3. Kompatibilität

Nachdem die alte Version der AL von der FSF wie erwähnt nicht einmal als Open Source Lizenz anerkannt wurde, ist die aktuelle Version nicht nur allgemein anerkannt, sondern auch mit der GPL kompatibel, da die vorher von der FSF als missverständlich kritisierten Passagen gestrichen oder verändert wurden. Mit anderen (proprietären) Lizenzen ist die AL ebenfalls kompatibel, solange die in Ziffer 5 AL geforderten Verpflichtungen eingehalten werden.

5.8.2. Die Sleepycat License

Die „Sleepycat License“ (SL) ist auch als „Sleepycat Public License“ oder „Berkeley Database License“ bekannt und darf nicht mit der „Sleepycat Commercial License“ verwechselt werden. Teilweise wird sie den BSD-artigen Lizenzen zugeordnet, was nachzuvollziehen ist⁵³⁶.

⁵³⁶ Hier wird sie deshalb unter den Lizenzen mit Wahlmöglichkeiten behandelt, um auf die kommerziellen Verwertungsmöglichkeiten besonders hinzuweisen.

5.8.2.1. Hintergrund

Die SL wird dazu benutzt, die Berkeley-Datenbank „Berkeley DB“ – eine eingebettete Datenbank-Bibliothek mit Programmierschnittstellen bspw zu C, C++, Java, Perl, Python und Tcl – der Firma „Sleepycat Software“ zu lizenzieren. Das Programm wurde 1991 an der Universität Berkeley entwickelt, um einen Ersatz für eine auf UNIX-Systemen eingesetzte Bibliothek zu schaffen. Diese stand ab 1992 unter der BSD-Lizenz und wurde erst im Zuge der kommerziellen Verwertung durch „Dual Licensing“ unter die SL und eine von Sleepycat Software entworfene kommerzielle Lizenz gestellt. Dafür wird die Berkeley DB als internes Unternehmensprojekt entwickelt, sodass alle Rechte am Quellcode weiterhin bei Sleepycat Software verbleiben⁵³⁷.

5.8.2.2. Lizenzmerkmale

Die SL ist eine der kürzeren Open Source Lizenzen. Sie ist in drei fast identische Abschnitte unterteilt, die auf die drei Urheber der Software: Sleepycat Software, die Universität von Kalifornien und die Universität von Harvard eingehen. Jeder Urheber stellt dabei besondere Bedingungen auf, die nur unwesentlich voneinander abweichen.

Alle Lizenzgeber räumen dem Lizenznehmer dabei das Recht ein, das Programm im Original oder in weiterentwickelter Form weiterzugeben und zu nutzen, solange

- der Urhebervermerk, der Lizenztext und der Haftungsausschluss mitgeliefert werden.

⁵³⁷ <http://www.sleepycat.com/company/licensing.html>.

- bei der Verbreitung in Objektform der Urhebervermerk, der Lizenztext und der Haftungsausschluss der Dokumentation oder sonstigen mitgelieferten Materialien beigelegt wird.

Im Fall der Firma Sleepycat Software kommt noch folgende Verpflichtung hinzu:

- In jedem Fall muss bei der Verbreitung eine Bezugsquelle für den Quellcode der „Berkeley DB“ mitgeliefert werden. Dieser muss entweder mitgegeben werden oder gegen angemessene Kopierkosten zu beziehen sein.

Im Falle der beiden oben genannten Universitäten kommen noch folgende Verpflichtungen dazu:

- Weder die Namen der Universitäten noch sonstiger Urheber dürfen im Zusammenhang mit der Bewerbung weiterentwickelter Versionen ohne vorherige schriftliche Genehmigung verwendet werden.

Die SL endet mit dem Haftungs- und Gewährleistungsausschluss, der sich bei den drei Varianten nur darin unterscheidet, dass der jeweilige Lizenzgeber schadlos gehalten werden soll.

5.8.2.3. Kompatibilität

Die Lizenz ist mit der GPL kompatibel. Dies hat aber nur geringen praktischen Wert, da die „Berkeley DB“ nicht durch die Open Source Community weiterentwickelt wird und daher hauptsächlich unter den beiden Sleepycat Lizenzen weitergegeben wird.

5.9. Lizenzen mit Sonderrechten

Die Lizenzen mit Sonderrechten gewähren den Lizenznehmern zwar alle Rechte, die Freie Software ausmachen, sehen aber zugleich besondere Privilegien für den Lizenzgeber bei Weiterentwicklungen durch den Lizenznehmer vor. Diese Lizenzen werden meist bei Programmen verwendet, die ursprünglich proprietär vertrieben wurden. Folgende Lizenzen gewähren dem Lizenznehmer Sonderrechte:

1. Apple Public Source License (v. 1.2)
2. Macromedia Open Source License Agreement (v. 1.0)
3. Netscape Public License (NPL) (v. 1.0 und v 1.1)
4. OCLC Research Public License (v. 1.0)
5. **OCLC Research Public License (v. 2.0)**
6. Open Map Software License Agreement
7. **Q Public License (QPL)**
8. Real Networks Community Source License (v 1.2)
9. RealNetworks Public Source License (RPSL) (v. 1.0)

Da die Lizenzen häufig nur auf eine Software Anwendung finden, wird ausschließlich die „Q Public License“ stellvertretend für die anderen Lizenzen mit Sonderrechten behandelt.

5.9.1. Die Q Public License

Die „Q Public License“ (QPL) entstand aus der Konkurrenz zwischen den beiden Linux-Oberflächen KDE und GNOME. Die Firma „Troll Tech“ hat ihr „Qt-Toolkit“ ab Version 2.0 unter die QPL gestellt.

5.9.1.1. Hintergrund

„Qt“ war bis 1998 ein rein proprietäres Produkt. Da von vielen Anhängern der Software befürchtet wurde, die Monopolstellung des Programms könnte dazu führen, die nicht lizenzrechtlich abgesicherte Freiheit zur Weiterentwicklung zu gefährden, drängte die FSF Troll Tech darauf, „Qt“ unter eine Open Source Lizenz zu stellen⁵³⁸. Troll Tech kam der Forderung nicht zuletzt deshalb nach, weil durch die Open Source Gemeinschaft „GNOME“ als Alternative zu „Qt“ entwickelt worden war. Durch die QPL wurde daher sichergestellt, „Qt“ als Open Source Software weiterentwickeln zu können. Die Popularität zeigt sich bspw anhand von KDE, das „Qt“ als Bibliothek für die graphische Darstellung benutzt.

5.9.1.2. Lizenzmerkmale

Ziffer 1 QPL macht die Lizenzvertragsannahme zur Bedingung, dass ein Lizenznehmer die in der QPL genannten Rechte nicht ausschließlich erhält. Dies ist insofern unnötig, da ein Lizenznehmer ohne Annahme der QPL keine über die gesetzlichen Mindestrechte des § 40d UrhG hinausgehenden Nutzungsrechte erhalten kann.

Ziffer 2 QPL gestattet dem Lizenznehmer die noch nicht weiterentwickelte Software zu kopieren und zu verbreiten, solange der Urheber- und Markenrechtsvermerk sowie der Haftungsausschluss in unveränderter Form mitgegeben wird.

Die Ziffer 3 QPL enthält die umstrittene Regelung, dass jede Weiterentwicklung nur in Form eines „Patches“ weiterverbreitet werden darf.

⁵³⁸ <http://slashdot.org/article.pl?sid=98/11/22/1029225&tid=117>.

Die QPL überlässt dem Lizenznehmer die Wahl, seine „Patches“ wieder unter die QPL oder eine andere Lizenz zu stellen⁵³⁹. Dabei ist es ihm untersagt, die bestehenden Urhebervermerke zu entfernen. Sollte er seinen „Patch“ wiederum unter die QPL stellen, muss er gem Ziffer 3b QPL dem ursprünglichen Programmentwickler (für „Qt“ also Troll Tech) das Recht einräumen, die Weiterentwicklung auch in proprietärer Form verbreiten zu dürfen⁵⁴⁰.

Die Verbreitung des QPL lizenzierten Programms in Objektform gestattet Ziffer 4 QPL unter den Bedingungen, dass der Lizenztext mitgeliefert (Ziffer 4a QPL) und auf eine Bezugsquelle für den lizenzgebührenfreien Erhalt des Quellcodes deutlich hingewiesen wird (Ziffer 4b QPL). Darüber hinaus muss die Objektform des Programms ebenfalls wieder unter die QPL gestellt werden (Ziffer 4c QPL).

Auf die Tatsache, dass „Qt“ als Programmbibliothek mit anderer Software zwingend verlinkt werden muss⁵⁴¹, damit „Qt“ genutzt werden kann, geht Ziffer 6 QPL ein. Die andere Software muss dabei folgende Bedingungen erfüllen:

1. Jegliche Verbreitung der Software in Objektform ist erlaubt, solange der Quellcode dafür lizenzgebührenfrei oder gegen angemessene Kopierkosten zu beziehen ist.
2. Sollte eine Weiterentwicklung nicht veröffentlicht werden, muss der Quellcode auf Verlangen des ursprünglichen Entwicklers herausgegeben werden.

Dies bedeutet im Ergebnis, dass ein Programmpaket zwar proprietär vertrieben werden darf, es sich dabei aber trotzdem um Open Source Software handelt, da der Quellcode in jedem Fall frei verfügbar sein muss⁵⁴². Die Verpflichtung zur Herausgabe soll sicherstellen, dass der Quellcode in jedem

⁵³⁹ Jaeger/Metzger, Open Source Software 77.

⁵⁴⁰ Jaeger/Metzger, Open Source Software 76.

⁵⁴¹ Es macht in diesem Fall keinen Unterschied, ob dies dynamisch oder statisch geschieht.

⁵⁴² <http://www.trolltech.com/licenses/gpl-annotated.html>.

Fall verfügbar ist, da die QPL einen direkten Herausgabeanspruch formuliert. Die QPL geht dabei weiter als andere Open Source Lizenzen, bei denen ein Lizenznehmer in einem solchen Fall nur seine Rechte aus der Lizenz verliert⁵⁴³.

Die QPL endet mit einem kurzen Haftungs- und Gewährleistungsausschluss und einer Rechts- und Gerichtsstandsklausel, nach der die Lizenz unter dem Recht von Norwegen steht und Oslo als Gerichtsstand festgelegt wird.

5.9.1.3. Kompatibilität

Die QPL ist mit der GPL inkompatibel und wird von der FSF kritisiert⁵⁴⁴, da Weiterentwicklungen nur als "Patchfiles" verbreitet werden dürfen. Da die FSF aber die Popularität von „Qt“ anerkennt und Programmierern die Möglichkeit zur Verwendung in GPL lizenzierten Programmen nicht verweigern will, hat sie eine Ausnahmeregelung aufgestellt. Danach muss dem Quellcode der Weiterentwicklung folgende Anmerkung beigefügt werden: "*As a special exception, you have permission to link this program with the FOO library and distribute executables, as long as you follow the requirements of the GNU GPL in regard to all of the software in the executable aside from FOO.*"⁵⁴⁵ Dadurch würdigt die FSF das Entgegenkommen von Troll Tech, "Qt" unter der QPL zur Verfügung zu stellen. Troll Tech selbst bietet "Qt" im "Dual Licensing" an⁵⁴⁶ und dient als gutes Beispiel in diesem Zusammenhang.

⁵⁴³ Jaeger/Metzger, Open Source Software 77.

⁵⁴⁴ <http://www.gnu.org/philosophy/license-list.html>.

⁵⁴⁵ <http://www.gnu.org/philosophy/license-list.html>.

⁵⁴⁶ <http://www.trolltech.com/products/qt/licensing.html>.

5.10. Das Dual Licensing

„Dual licensing is simply the licensing of one software product under two different licenses, where one license allows for free distribution and the other allows for license revenues.“

- *Philip H. Albert*⁵⁴⁷

Open Source Software hat sich nicht nur wegen ihrer eigenständigen Philosophie durchgesetzt, sondern wurde auch von der Wirtschaft aufgrund des „Dual Licensing“ anerkannt. Dual Licensing bedeutet, dass ein Programm unter zwei unterschiedliche Lizenzen - meist eine Open Source und eine proprietäre - gestellt wurde. Diese Lösung erlaubt Firmen, den Quellcode unter der Open Source Lizenz frei verfügbar zu machen und gleichzeitig die Möglichkeit zu haben, zusätzliche Leistungen wie „Add-Ons“ oder „Bundles“ unter einer proprietären Lizenz kommerziell zu vertreiben. Firmen wie MySQL, Sleepycat oder Trolltech betreiben die Strategie des Dual Licensing sehr erfolgreich und zeigen damit, dass Open Source Software auch Open Source bleiben kann, ohne damit wirtschaftsfremd oder gar wirtschaftsfeindlich zu sein.

Das Dual Licensing ist rechtlich anerkannt, da jeder Urheber über seine Rechte als Inhaber frei verfügen und selbst entscheiden kann, unter welche Art von Lizenz er sein Programm stellt. Da die unterschiedlichen Arten der Open Source Lizenzen – solche mit Copyleft-Effekt gegenüber solchen ohne Copyleft-Effekt – darüber entscheiden, welche Möglichkeiten Lizenzgeber und –nehmer durch das Dual Licensing haben, müssen diese Möglichkeiten differenziert dargestellt werden.

⁵⁴⁷ Aus seinem Artikel „Dual Licensing: Having Your Cake and Eating It Too“, <http://www.linuxinsider.com/story/38172.html>.

5.10.1. Dual Licensing: Proprietäre Lizenz und Open Source Lizenz mit Copyleft Effekt

Das Paradebeispiel des Dual Licensing betrifft die GPL⁵⁴⁸ in Verbindung mit einer proprietären Lizenz, da diese Art von den meisten kommerziellen Softwareanbietern gewählt wird. Die Grundvoraussetzungen für das Dual Licensing in diesem Fall sind:

1. Ein Lizenzgeber muss die ausschließlichen Nutzungsrechte an dem Programm besitzen⁵⁴⁹.
2. Ein Lizenznehmer darf nicht die ausschließlichen Nutzungsrechte an dem Programm erhalten.

Für das Programm bedeutet dies, dass es nicht wieder in eine rein proprietäre Lizenz überführt werden kann, wenn es einmal weiterverbreitet wurde, da dies die Bestimmungen der Lizenz mit strengem Copyleft verhindern⁵⁵⁰. Solange das Programm nur durch den ursprünglichen Rechteinhaber weiterentwickelt wird, ändert dies nichts an den Möglichkeiten, das Programm unter zwei oder mehrere Lizenzen zu stellen. Erst im Falle einer Weiterentwicklung des Programms durch Dritte benötigt der Dual Licensing-Lizenzgeber das Einverständnis des Weiterentwicklers, da ihm dann selbst keine ausschließlichen Verwertungsbefugnisse am Gesamtwerk mehr zustehen⁵⁵¹. Sollte dennoch der Wunsch bestehen, die Weiterentwicklung durch Dual Licensing verfügbar machen zu wollen, muss mit dem Rechteinhaber der Weiterentwicklung eine Sondervereinbarung getroffen werden. Dazu ist es wichtig, einen Überblick über die Weiterentwicklungen zu behalten, da es ansonsten schwierig bis unmöglich wird, die Urheber der Weiterentwicklungen

⁵⁴⁸ Siehe Ziffer 10 GPL, die zwar nicht direkt auf das Dual Licensing eingeht, aber dennoch die Möglichkeit dazu anspricht.

⁵⁴⁹ Jaeger/Metzger, Open Source Software 80; <http://www.linuxinsider.com/story/38172.html>.

⁵⁵⁰ Metzger, Nebeneinander von Lizenztypen, Einmal GPL, immer GPL? <http://www.linux-magazin.de/Artikel/ausgabe/2001/01/recht/recht.html>.

⁵⁵¹ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 135.

ausfindig zu machen⁵⁵². In den meisten Fällen werden die Programme daher intern weiterentwickelt wie bspw im Falle von Sleepycat und MySQL⁵⁵³, damit die Strategie des Dual Licensing konsequent verfolgt werden kann.

Der Copyleft-Effekt verhindert auch, dass die unter Nutzung der Open Source Lizenz weiterentwickelten Programme nicht mehr in das unter der proprietären Lizenz stehende Programm eingebunden werden können. Allerdings ist es möglich, ein eigenständiges Werk, das den gleichen Zweck erfüllt, in den proprietären Teil zu integrieren⁵⁵⁴.

5.10.2. Dual Licensing: Proprietäre Lizenz und Open Source Lizenz ohne Copyleft Effekt

Einige Softwareanbieter haben eine andere Art des Dual Licensing gewählt. Sie stellen ihre Software unter eine Open Source Lizenz ohne Copyleft-Effekt, vorrangig die BSD-Lizenz und eine proprietäre Lizenz. Die Gründe hierfür liegen häufig darin, dass sie weiterhin die Möglichkeit haben wollen, Weiterentwicklungen von der unter der Open Source Lizenz stehenden Software in den proprietären Teil überführen zu können. Die Grundvoraussetzungen in diesem Fall sind:

⁵⁵² Man bedenke nur den Fall des mozilla-Browsers, als die mozilla Foundation nachträglich 450 Einwilligungen einholen musste, um mozilla unter eine Triple-Lizenz zu stellen. Dafür musste eine eigene Kampagne gestartet werden, um alle Miturheber ausfindig zu machen, siehe <http://www.mozilla.org/MPL/missing.html>.

⁵⁵³ <http://www.linuxinsider.com/story/38172.html>.

⁵⁵⁴ Vgl dazu die Problematik der Eigenständigkeit in Kap 5.4.1.2.2.

1. Der Lizenzgeber muss nicht die ausschließlichen Nutzungsrechte am Programm besitzen.
2. Der Lizenznehmer muss vom Lizenzgeber beliebig viele einfache Nutzungsrechte zur Weitergabe erhalten⁵⁵⁵.

Für das Programm bedeutet dies, dass es aus dem Open Source Bereich in den proprietären Bereich überführt werden kann, auch wenn es bereits weiterverbreitet wurde. Es kommt auch nicht darauf an, ob das Programm von Dritten weiterentwickelt wurde, da ein Lizenznehmer durch die Open Source Lizenz ohne Copyleft-Effekt die Möglichkeit erhält, über den weiteren Verwendungszweck selbst zu entscheiden.

5.11. Ergebnis

Das Kapitel zeigt, dass Open Source Lizenzen eine breite Vielfalt an Möglichkeiten für Lizenzgeber bieten, ihre Software Open Source anzubieten. Die größte Unterscheidung muss zwischen Open Source Lizenzen ohne Copyleft-Effekt gegenüber denen mit Copyleft-Effekt gemacht werden. Daher ist es für Lizenzgeber positiv zu bewerten, in den jeweiligen Bereichen nicht nur auf eine einzige Lizenz zurückgreifen zu müssen, sondern eine Wahl treffen zu können. Die wesentlichen Lizenzmerkmale werden an dieser Stelle in der Reihenfolge ihrer Behandlung zusammengefasst und bewertet.

Das wichtigste Merkmal an Non-Copyleft-Lizenzen ist wie bereits erwähnt⁵⁵⁶, dass Benutzern die Wahlmöglichkeit offen steht, Weiterentwicklungen

1. ebenfalls unter die Ursprungslizenz zu stellen,
2. unter eine andere Open Source Lizenz zu stellen oder

⁵⁵⁵ Jaeger/Metzger, Open Source Software 81.

⁵⁵⁶ Vgl Kap 5.2.

3. nicht mehr unter eine Open Source Lizenz zu stellen und somit den Quellcode der Weiterentwicklung unfrei werden zu lassen.

Zu letzterer Variante kommt es häufig, wenn die Weiterentwicklung in Folge proprietär vertrieben werden soll. Die FSF bevorzugt Copyleft-Lizenzen⁵⁵⁷, da diese die Freiheit der Software schützt. Andere in der Open Source Gemeinschaft werden in den Non-Copyleft-Lizenzen eine größere „Freiheit“ sehen, da diese für den Benutzer mit weniger Pflichten verbunden sind. Damit bleibt es dem Einzelnen überlassen, Copyleft- oder Non-Copyleft-Lizenzen als „freier“ anzusehen.⁵⁵⁸

Die BSD-artigen Lizenzen haben in ihrer Entwicklung die Kompatibilität mit der GPL als wichtigstes Ziel. Die Lizenztexte selbst ähneln einander sehr stark und unterscheiden sich oftmals nur in kleinen Details. Die Kürze der Lizenztexte führt bei Lizenzgebern und –nehmern in der Regel zu wenig Verständnisschwierigkeiten und sorgt dadurch für große Rechtssicherheit bei den Lizenzvertragspartnern. Der Umgang mit der BSD-Lizenz – als eine der ersten Open Source Lizenzen überhaupt – führt kaum noch zu Auslegungsfragen. Der Bereich der sonstigen Lizenzen ohne Copyleft-Effekt bedarf keiner genaueren Betrachtung, da in diesem Bereich nur eine einzige Lizenz, die OGTSL, von der OSI approbiert wurde und diese keine generelle Anwendungsmöglichkeit bietet. Für kommerzielle Vertrieber und Dienstleister sind die Open Source Lizenzen ohne Copyleft-Effekt empfehlenswert, da die Pflichten im Umgang mit den Verwertungsrechten kaum Komplikationen bereiten dürften.

Die Lizenzgeber von Open Source Software haben sich mehrheitlich für die GPL entschieden und unterstellen ihre Programme damit einem strengen Copyleft-Effekt. Dies kann zum einen darauf zurückgeführt werden, dass der Linux-Kernel unter der GPL steht und daher jedes auf diesen aufbauende

⁵⁵⁷ <http://www.fsf.org/philosophy/bsd.html>.

⁵⁵⁸ So auch *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 3.

Programm als „*derivative Work*“ zwingend wieder unter die GPL gestellt werden muss und zum anderen bieten Lizenzen mit strengem Copyleft-Effekt einen weitreichenden Schutz für Open Source Software und gewährleisten, dass ein Lizenzgeber die Kontrolle über sein Werk behält, auch wenn er Nutzungsrechte an Lizenznehmer einräumt. Darüber hinaus ist die GPL im wirtschaftlichen Verkehr anerkannt und der Umgang mit der Lizenz ist aufgrund ihrer Popularität zur Gewohnheit geworden. Durch die deutsche Rechtsprechung ist die GPL 2 bestätigt worden⁵⁵⁹ und damit die einzige Open Source Lizenz, die gerichtlich verhandelt wurde. 2007 wurde die Version 3 der GPL herausgebracht, bei der einige sinnvolle Adaptionen gegenüber der GPL 2 vorgenommen wurden; dass deren Endergebnis dennoch wenig überzeugt, macht die Tatsache deutlich, dass prominente Urheber, bspw. Linus Torvalds, nicht bereit sind, ihren Anteil vom Linuxkernel unter die GPL 3 zu stellen. Als Groteske muss angesehen werden, dass die GPL 3 erst zwei Monate nach ihrer Veröffentlichung von der OSI zertifiziert wurde⁵⁶⁰. Eine Aufforderung zur Vorlage durch die OSI wurde von der FSF mit Befremden aufgenommen und zeigt den immer noch bestehenden Konflikt der beiden Parteien im Kampf um eine Vormachtstellung im Freien/Open Source Software Bereich.

Bezüglich der Open Source Lizenzen, die sich neben der GPL für einen strengen Copyleft-Effekt entschieden haben, bleibt festzustellen, dass deren Lizenzautoren meist sehr bewusst eigene Lizenzen herausgegeben haben, nur zu dem Zweck – wie das Beispiel IBM zeigt – ihre Software zwar Open Source zu lizenzieren, aber trotzdem größtmögliche Kontrolle darüber zu behalten. Dazu gehört die Kontrolle über die Lizenz, welche man sich durch die Lizenzhoheit sichert. Wie man bei der CPL sehen kann, werden vorrangig wirtschaftliche Güter wie Patentrechte geschützt, auf die noch besonders eingegangen wird. Oft endet es damit, dass diese Lizenzen dann nicht mehr mit

⁵⁵⁹ LG München I, 19.05.2004, Az 21 O 6123/04; LG München I, 24.07.2007, Az 7 O 5245/07.

⁵⁶⁰ http://www.cbronline.com/article_news.asp?guid=A4BE06FA-EF8F-4D1E-8C62-B3DEF66F893F.

der GPL kompatibel sind, obwohl sie ebenfalls einen strengen Copyleft-Effekt beinhalten.

Wenn man die GPL-artigen Lizenzen mit den MPL-artigen Lizenzen vergleicht, fallen einem vor allem wirtschaftliche Aspekte auf, die die MPL „freier“ erscheinen lassen. Sollte einem Urheber, der sein Programm unter der GPL lizenziert hat, zu einem späteren Zeitpunkt einfallen, das Programm proprietär zu vertreiben, könnte er damit aufhören, sein Programm unter der GPL zu lizenzieren und es unter eine proprietäre Lizenz stellen. Der ursprünglich freie Code wäre damit nur noch für jene frei, die eine Kopie unter der GPL erhalten haben. Dieses „Unfrei-werden“ ist unter der MPL nicht in diesem Ausmaß möglich, da hier jeder das Recht hat, das Programm zu kommerzialisieren. Eine Monopolstellung ist hierbei nicht möglich, da potentielle Mitbewerber in den wirtschaftlichen Wettbewerb eintreten könnten, da sie diese Berechtigung ebenfalls aus der MPL erhalten haben. Dies muss im Ergebnis dazu führen, dass ein unter der MPL lizenziertes Programm seine Freiheit behält, da eine Vereinnahmung letztendlich keine Vorteile bringen würde⁵⁶¹. Dieser Argumentation ist entgegenzuhalten, dass GPL-lizenzierte Programme meist nicht von einer Person allein geschrieben worden sind und eine solche Situation daher nur realistisch erscheint, wenn einer oder nur wenige die Urheberrechte am Quellcode innehaben. Man konnte sehen, dass es im Fall von Mozilla schwierig und zeitraubend war, die erforderlichen Genehmigungen einzuholen, den Programmcode unter eine andere Lizenz zu stellen. Daher ist davon auszugehen, dass ein unter der GPL freigegebener Code „freier“ bleibt, je länger der Zeitraum ist und je mehr er verbreitet wird.

Wesentlich ist ebenfalls, dass die MPL mit weit mehr Lizenzen kompatibel ist als die GPL. Sollte jemand versuchen, ein Programm aus mehreren Programmteilen, die unter verschiedenen Lizenzen stehen (bspw. BSD, MPL und GPL), zusammenzutragen, um sie gemeinsam und einheitlich

⁵⁶¹ <http://www.croftsoft.com/library/tutorials/gplmpl/>.

zu vertreiben, wird er Probleme mit der GPL bekommen, die von ihm verlangt, dass er das gemeinsame Produkt unter der GPL vertreibt. Die MPLartigen Lizenzen erlauben ihm hier größere Freiheiten, da nur der unter der MPL lizenzierte Teil auch darunter lizenziert bleiben muss.

Letztendlich bleibt das Argument betreffend der Open Source „Lizenzrevolution“⁵⁶², wonach unter der MPL stehende Programmteile weiterhin verwendet werden dürften, sollte eine neue und populäre Open Source Lizenz für Neuentwicklungen Verwendung finden. Dies wäre bei der GPL nicht möglich, da wiederum das Gesamtwerk nur unter der GPL vervielfältigt und verbreitet werden darf. Dazu bleibt anzumerken, dass der Großteil der Open Source Software unter der GPL steht und es daher wenig realistisch erscheint, dass alle diese Programme durch Nicht-GPL-Programme ersetzt werden⁵⁶³.

Die Open Source Lizenzen mit beschränktem Copyleft-Effekt finden vorrangig auf Programmbibliotheken Anwendung wie die LGPL und die wxWLL beweisen. Hier kann die Absicht der Lizenzautoren darin gesehen werden, Kompatibilitätsschranken zu entfernen, um so den Einsatz der Software in kommerziellen Produkten zu erleichtern.

Sowohl die Open Source Lizenzen mit Wahlmöglichkeiten als auch jene mit Sonderrechten haben einen starken Bezug zu proprietären Programmen bzw kommerzieller Verwertung. Open Source Lizenzen mit Wahlmöglichkeiten nehmen besondere Rücksicht darauf, dass der Vertrieb von Software in Objektform ohne Mitgabe des Quellcodes möglich ist, solange wenigstens eine Bezugsquelle für diesen angegeben wird. Dies ist besonders für einen reinen Nutzer der Software positiv zu bewerten, der durch ein mitgeliefertes SDK unsicher werden könnte. Solche Nutzer, die darüber hinaus Lizenznehmer

⁵⁶² <http://www.croftsoft.com/library/tutorials/gplmpl/>.

⁵⁶³ Eine Tatsache, der sich *Richard Stallmann* und die FSF bei der Entwicklung der GPL sicher bewusst waren. Die Popularität der GPL zeigt die positive Aufnahme dieser Entscheidung in der Open Source Gemeinschaft.

werden wollen, haben immer noch die Möglichkeit, die aktuellste Version der Software über das Internet zu beziehen und dadurch Informationen über Anlaufstellen zu erhalten, an denen sie weiterentwickelte Versionen oder Bugfixes abgeben können. Diese Kontaktstellen sind meist für Entwickler unabdingbar, da dort wesentliche Informationen über Newsgroups zu erhalten sind, die für eine effektive Weiterentwicklung nützlich sind. Software, welche unter einer Open Source Lizenz mit Sonderrechten steht, kommt meist aus dem proprietären Bereich. Die Lizenzen bewegen sich daher oft am Rande der Zuordnung zum Open Source Bereich. In vielen Fällen wird deshalb ein Programm im Dual Licensing unter zwei verschiedene Open Source Lizenzen gestellt, da dass Dual Licensing vor allem kommerziellen Anbietern von Open Source Software die Möglichkeit bietet, ihre wirtschaftlichen Ziele trotz kostenlos lizenzierter Software erreichen zu können.

Die Auswahl des Dual Licensing-Modells (Verbindung der proprietären Lizenz mit einer Open Source Lizenz mit oder ohne Copyleft-Effekt) bleibt dem ersten Rechteinhaber bzw Urheber vorbehalten. Dieser hat bei der Verbindung mit einer Open Source Lizenz mit Copyleft-Effekt die Möglichkeit, weitreichende Kontrolle über sein Programm zu behalten. Er kann allerdings keine Verbindungen von unter der Open Source Lizenz gestelltem Code mit proprietärem Code – bspw durch statisches Linken – herstellen, ohne in Folge das Gesamtwerk unter die Open Source Lizenz stellen zu müssen⁵⁶⁴. Dies ist im Falle einer Verbindung mit einer Open Source Lizenz zwar möglich, allerdings sind die Kontrollmöglichkeiten beschränkt. Der Lizenzgeber muss hierbei befürchten, dass die Weiterentwicklungen nicht mehr Open Source verfügbar sein werden und er durch die Überführung in proprietäre Produkte von Drittanbietern Konkurrenz bekommt. Einem Lizenznehmer bietet das Dual Licensing ebenfalls eine größere Auswahlmöglichkeit. Er kann wählen, ob er die Open Source Variante einsetzen möchte – dann aber wenig bis keinen Support erhalten wird und er jede Weiterentwicklung wiederum unter die Open Source Lizenz stellen muss – oder ob er die proprietäre Lizenz wählt, bei der

⁵⁶⁴ Siehe bspw: <http://www.cheiron.org/misc/license.htm>.

Weiterentwicklungen durch den Lizenznehmer nicht möglich sind, er aber je nach Lizenzvereinbarung, bspw Support und die Übernahme von Haftung und Gewährleistung erhält.

6. HAFTUNG UND GEWÄHRLEISTUNG

„Wie kann ich diesen langhaarigen Leuten trauen, die mir einen Bug Fix schicken?“

- *Volker Grassmucks*⁵⁶⁵

Die Haftungs- und Gewährleistungsausschlüsse der Open Source Lizenzen „schreien“⁵⁶⁶ geradezu nach Aufmerksamkeit, da man sie in den US-amerikanischen Lizenzen mit Großbuchstaben geschrieben sieht. Diese wollen aber nicht den Leser „anschreien“, ihre Schreibweise beruht auf den §§ 2 – 316 des US-amerikanischen „Uniform Commercial Code“ (UCC), die unter anderem die formale Gestaltung von Gewährleistungs- und Haftungsausschlüssen regeln und für diese vorsieht, dass sie in optisch auffallender Form, also durch die Schreibweise mit Großbuchstaben oder fett gedruckt im Lizenztext untergebracht sein müssen. Die übliche Anbringung der Gewährleistungs- und Haftungsausschlüsse in Open Source Lizenzen erscheint logisch, wenn man bedenkt, dass ein Lizenznehmer ohne Bezahlung von Lizenzkosten an die Software und die damit verbundenen Rechte und Pflichten gelangt. Ob diese Lizenzteile nach österreichischem Recht Bestand haben, erscheint zweifelhaft und wird im folgenden Abschnitt behandelt.

Da bei Vervielfältigung und Verbreitung von Open Source Software unterschiedliche Vertragskonstellationen möglich sind, wird zuerst die Einordnung des Erwerbs der Open Source Software unter einen bestimmten Vertragstyp vorgenommen und danach die gewährleistungs- und haftungsrechtlichen Folgen für diese erörtert.

⁵⁶⁵ Grassmuck, Freie Software Zwischen Privat- und Gemeineigentum.

⁵⁶⁶ Im Internet – vor allem beim Chatten – wird das Schreiben mit Großbuchstaben als unhöflich angesehen und bedeutet, dass man jemanden anschreit. Das Schreiben mit Großbuchstaben ist meist nur erlaubt, wenn man ein Wort oder einen Satz besonders hervorheben will.

6.1. Die einzelnen Vertragskonstellationen

Zunächst werden die populärsten Vertragskonstellationen behandelt, ihre Verträglichkeit mit den Haftungs- und Gewährleistungsausschlüssen der Lizenzen überprüft und mögliche Folgen nach österreichischem Recht erörtert. Dabei muss darauf Rücksicht genommen werden, dass beim Erwerb von Open Source Software je zwei Vertragsgegenstände rechtsgeschäftlicher Bestandteil werden⁵⁶⁷:

1. Die Überlassung der Open Source Software selbst und
2. die im Lizenzvertrag geregelten Nutzungsrechte.

Beide Vertragsgegenstände werden deshalb in Folge bei der Erörterung unterschieden.

6.1.1. Kostenloser Erwerb durch das Herunterladen der Software im Internet direkt vom Lizenzgeber

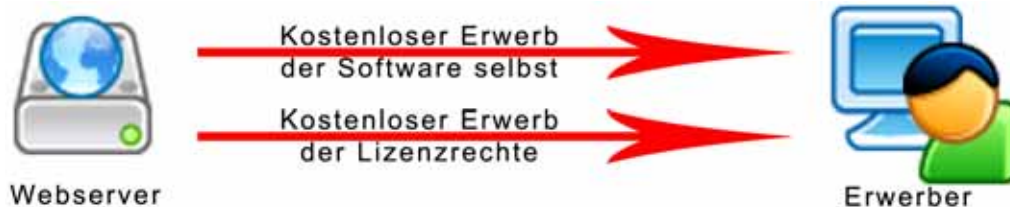


Abb 8

Der „Urfall“ des Erwerbs von Open Source Software betrifft das kostenlose Herunterladen der Software vom Lizenzgeber. In dieser Fallkonstellation ist kein fremder Code weiterentwickelt worden. Es wird davon ausgegangen, dass die Open Source Software vollständig vom Lizenzgeber entwickelt wurde. Der erste Schritt muss daher sein, den kostenlosen Erwerb auf seine Unentgeltlichkeit iSd § 917 ABGB hin zu überprüfen.

⁵⁶⁷ Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 137.

Von der Unentgeltlichkeit kann im vorliegenden Vertragstyp unbestritten ausgegangen werden, solange mit dem Lizenznehmer keine gesonderte Gegenleistung – bspw Weiterentwicklung eines bestimmten Programmteils oder umfangreiches „Debugging“ – vereinbart wurde⁵⁶⁸. Sollte eine solche Gegenleistung trotz Kostenlosigkeit des Erwerbs vereinbart sein, kann im Einzelfall Entgeltlichkeit iSd § 917 ABGB (eventuell iVm § 942 ABGB) vorliegen. Ohne eine gesonderte Vereinbarung ist jedenfalls von Unentgeltlichkeit auszugehen, weshalb zu erörtern ist, ob eine nahe liegende Schenkung iSd §§ 938 ff ABGB oder ein unentgeltlicher Vertrag sui generis vorliegt.

6.1.1.1. Schenkung

In Deutschland wird die Einordnung unter die Schenkung mehrheitlich bejaht⁵⁶⁹. Auch nach österreichischem Recht ist die Annahme einer Schenkung iSd § 938 ABGB zur Diskussion gestellt⁵⁷⁰. Um den Lizenzvertrag als Schenkung qualifizieren zu können, bedarf es zunächst einer Zustimmung durch den Beschenkten⁵⁷¹, welche regelmäßig durch Zustimmung zum Open Source Lizenzvertrag vom Lizenznehmer gegeben sein wird. Während die Schenkung der Werknutzungsbewilligung iSd des § 24 UrhG keine Probleme birgt, da diese als Schenkungsgut anerkannt sind⁵⁷², ist dies hinsichtlich der Software schwieriger. Die Erfordernis des § 938 ABGB, dass das Schenkungsgut eine Sache im weiten Sinne des § 285 ABGB⁵⁷³ sein muss, ist nicht einwandfrei geklärt, da die Sacheigenschaft von Software umstritten ist.

⁵⁶⁸ *Wiebe/Prändl*, ÖJZ 2004, 635.

⁵⁶⁹ *Jaeger/Metzger*, Open Source Software 144; *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 140 f; *Spindler*, Rechtsfragen der Open Source Software 73; *Siepmann*, Lizenz- und haftungsrechtliche Fragen, Abs 136; *Sujecki*, MR 2005, 40, wohl auch: *Omsels*, in FS Hertin 151; kritisch: *Schneider*, Handbuch des EDV-Rechts Rz 19 ff; eher verneinend: *Koch*, CR 6/2000, 335.

⁵⁷⁰ *Wiebe/Prändl*, ÖJZ 2004, 635.

⁵⁷¹ *Koziol/Welser*, Bürgerliches Recht II¹² (2001) 178.

⁵⁷² *Schubert* in *Rummel*, Kommentar I³ (2000) § 938 Rz 2.

⁵⁷³ *Schubert* in *Rummel*, Kommentar I³ § 938 Rz 2.

Der OGH hat bereits unkörperliche Sachen als Gegenstand eines Kaufvertrags anerkannt⁵⁷⁴, die Unentgeltlichkeit in dieser Vertragskonstellation kann daher analog zu dieser Entscheidung als Schenkung qualifiziert werden⁵⁷⁵.

Schwierig ist ebenfalls die geforderte „Freigiebigkeit“⁵⁷⁶ des Lizenzgebers zu beurteilen. Dem Lizenzgeber die Freigiebigkeit von vorneherein abzusprechen⁵⁷⁷, ist mE falsch. Viele kleine Open Source Projekte sind Resultate einer selbstlosen Einstellung ihrer Entwickler, sei es, dass sich diese über eine nichtvorhandene Anwendungsmöglichkeit geärgert haben und deshalb eine selbst entwickelte Open Source Lösung anbieten – um anderen dieses Ärgernis zu ersparen⁵⁷⁸ – oder schlicht ein ihrer Meinung nach nützliches „Plug-In“ entwickelt haben. An dieser ursprünglichen Freigiebigkeit ändert sich auch dann nichts, wenn diese „Hilfsprogramme“ aufgrund wachsender Popularität oder durch Weiterentwicklungen eine Größe annehmen können, in deren Folge für den Lizenzgeber ökonomische Interessen im Vordergrund stehen. Ähnliches gilt für von öffentlichen Verwaltungsstellen entwickelte Open Source Software, die ihre entwickelten Programme als Bürgerservice anbieten⁵⁷⁹ und bei denen es ebenfalls an einem ökonomischen Interesse fehlt. An Freigiebigkeit wird es jenen Open Source Programmen mangeln, die eine bestimmte Entwicklungsgröße überschritten haben und wo der überwiegende Teil eines an der Entwicklung einer Open Source Software beteiligten Teams seinen Lebensunterhalt mit den mit der Open Source Software verbundenen Leistungen bestreitet. Daher ist es wichtig, bei Open Source Software-Projekten auf die Größe des Teams und die Vielfalt der mit

⁵⁷⁴ OGH 14.10.1997, 5 Ob 504/96 = JBI 1998, 577; siehe auch *Staudegger*, Zur Qualifikation von Verträgen, die der Überlassung von Computersoftware dienen, JBI 1998, 604.

⁵⁷⁵ *Wiebe/Prändl*, ÖJZ 2004, 635.

⁵⁷⁶ OGH 16.01.1986, 6 Ob 808/83, JBI 1986, 526 (*Pfersmann*) = RdW 1986, 174; *Kulka*, Unentgeltlichkeit und Freigiebigkeit, ÖJZ 1969, 477 f.

⁵⁷⁷ *Wiebe/Prändl*, ÖJZ 2004, 635.

⁵⁷⁸ Als Bsp mag „fli4l“ dienen, eine Open Source Software, die dazu eingesetzt werden kann, einen nur aus zwei Netzwerkkarten, Motherboard, Laufwerk und Netzteil bestehenden PC als Linux-basierten ISDN-, DSL- und Ethernet-Router einzusetzen.

⁵⁷⁹ Bspw <http://www.justiz.nrw.de/IndexSeite/Organisation/projekte/lisa.html>.

der Open Source Software verbundenen Hilfsangebote (Support, Schulung, etc) zu achten, um die Frage nach der Freizügigkeit beantworten zu können.

Die Formpflichten der Schenkung iSd § 938 iVm § 943 ABGB werden beim Herunterladen der Open Source Software erfüllt, da es zu einer tatsächlichen Übergabe von Software und Lizenzvertrag kommt, welche nach außen hin erkennbar ist⁵⁸⁰. Dabei spielt es im Ergebnis keine Rolle, dass die tatsächliche Übergabe der Open Source Software der unentgeltlichen Einräumung der Nutzungsrechte zeitlich vorausgeht⁵⁸¹. Die in den Open Source Lizenzen enthaltenen Pflichten des Lizenznehmers sind als Auflagen zu sehen⁵⁸², die im strengsten Fall eine Minderung der Zuwendung bedeuten können, keinesfalls aber eine Gegenleistung iSd § 917 ABGB⁵⁸³ darstellen, da die Einhaltung der Lizenzpflichten nicht im Interesse des Lizenzgebers liegt, das ohne die Zuwendung nicht befriedigt werden kann. Bei den BSDartigen Lizenzen sind die Auflagen nicht so strikt wie bei den Lizenzen mit Copyleft-Effekt, weshalb bei diesen auf gar keinen Fall eine Gegenleistung angenommen werden kann.

Für Open Source Lizenzen in vieler Hinsicht positiv zu bewerten ist, dass die Einhaltung der Auflagen mittels Klage durchgesetzt werden können und ihnen damit zu einer stärkeren Stellung verholfen wird.

⁵⁸⁰ OGH 17.09.1992, 7 Ob 579/92 in WBI 1993, 95 = ecolex 1993, 18.

⁵⁸¹ Siehe dazu auch Kap 4.2.; *Jaeger/Metzger*, Open Source Software 138.

⁵⁸² *Wiebe/Prändl*, ÖJZ 2004, 635. Im Gegensatz zum dt Recht (siehe § 518 BGB) müssen Auflagen nicht notariell beurkundet werden.

⁵⁸³ Kritisch dazu *Koch*, CR 2000, 335.

6.1.1.2. Unentgeltlicher Vertrag sui generis

Die Open Source Lizenzverträge lassen sich durch die in ihnen enthaltenen AGBs unter dem Vertragstypus der zweiseitigen atypischen Verträge einordnen. Dieser Vertragstypus könnte zusammen mit dem Rechtsgeschäft, welches durch das Herunterladen der Open Source Software zustande kommt, als unentgeltlicher Vertrag sui generis qualifiziert werden⁵⁸⁴. Dies ist abzulehnen, wenn man bedenkt, dass die von Lizenzgeber und –nehmer zu erbringenden Leistungen überwiegend dem Schenkungsvertragsrecht zugeordnet werden können⁵⁸⁵. Das Rechtsgeschäft beim unentgeltlichen Download ist daher nach österreichischem Recht als Schenkung zu qualifizieren⁵⁸⁶.

6.1.1.3. Rechtsfolgen

Die zivilrechtlichen Rechtsfolgen treten unabhängig davon ein, ob man eine Schenkung oder einen Vertrag sui generis annimmt. Sie knüpfen in jedem Fall an die Unentgeltlichkeit des Geschäfts an. Die möglichen Rechtsfolgen betreffend Gewährleistung und Haftung werden daher in dieser Vertragskonstellation erörtert.

6.1.1.3.1. Gewährleistung

Über die Anwendbarkeit des Gewährleistungsrechts auf Software wurde viel diskutiert. Ein für Open Source Software interessantes Argument betrifft

⁵⁸⁴ Für das dt Recht zustimmend *Koch*, CR 2000, 335.

⁵⁸⁵ *Spindler*, Rechtsfragen 75.

⁵⁸⁶ Im Ergebnis gleich *Heidinger*, Von Copyright zu „Copyleft“: Wenn Programme freigegeben werden, *Die Presse*, 22.12.2003.

dabei die Aussage, dass Software gewährleistungsrechtlich nicht relevant sei⁵⁸⁷, da der Beweis der Fehlerhaftigkeit von Software nie geführt werden könne. Open Source Software ist aber gerade deshalb „offen“, um anderen Entwicklern die Möglichkeit zu geben, das Programm zu „debuggen“, also Fehler auszubessern. Es kann daher kaum behauptet werden, dass eine Gewährleistung für Sach- oder Rechtsmängel, welche der Open Source Software geradezu immanent bei der Übergabe anhängen, nicht zur Anwendung kommen darf. Noch stichhaltiger dürfte der Einwand sein, der die Unentgeltlichkeit der Überlassung betrifft. Hier gibt es im ABGB klare Aussagen über die Anwendbarkeit des Gewährleistungsrechts. Schon im ersten Satz des § 922 ABGB wird deutlich, dass eine Gewährleistung im Falle der unentgeltlichen Überlassung nicht durch das Gesetz gegeben ist. Auch Fälle, bei denen das KSchG Anwendbarkeit findet, genießen keinen gewährleistungsrechtlichen Schutz, da § 9 KSchG an das ABGB anknüpft⁵⁸⁸.

Dies führt im Ergebnis dazu, dass bei dieser Vertragskonstellation die Gewährleistungsausschlüsse der Open Source Lizenzen gültig sind. Für den Lizenzgeber hat dies insbesondere den Vorteil, dass er mehr Rechtssicherheit bei Vervielfältigung und Verbreitung genießt, es ihm aber freisteht, einem Lizenzgeber über den Vertrag hinausgehende Sicherheiten bezüglich der Gewährleistung anzubieten. Des Weiteren ist nicht einzusehen, dass Programmierer, die aufgrund ihrer selbstlosen Einstellung, Open Source Software unentgeltlich zum Herunterladen bereitstellen, für diese gewährleistungsrechtliche Ansprüche übernehmen sollen. Eine solche Bewertung würde unweigerlich in einer weiteren, auch vom OGH geführten⁵⁸⁹, Diskussion enden, ab wann ein Fehler im Programm als Mangel bewertet werden muss.

⁵⁸⁷ Staudegger in *Jahnel/Schramm/Staudegger*, Informatikrecht², 101.

⁵⁸⁸ *Wiebe/Prändl*, ÖJZ 2004, 636.

⁵⁸⁹ *Staudegger* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 114.

Dieses Ergebnis bedeutet auch für den Lizenznehmer eine annehmbare Lösung, da er die Wahl hat, Open Source Software unentgeltlich, dafür aber ohne Gewährleistung zu erhalten oder gegen Entgelt die Gewährleistung „hinzuzukaufen“.

6.1.1.3.2. Haftung

Nachdem die Gewährleistung aufgrund der Unentgeltlichkeit ausgeschlossen und die entsprechenden Klauseln der Open Source Lizenzen bestätigt werden konnten, lässt sich diese Frage bei der Haftung nicht ganz so einfach beantworten, da sich die haftungsrechtlichen Bestimmungen aus dem Schadenersatzrecht des ABGB ergeben, wonach jeder schuldhaft und rechtswidrig verursachte Schaden zu einer Ersatzpflicht führt. Demnach spielt die Unentgeltlichkeit auf den ersten Blick keine allzu wichtige Rolle⁵⁹⁰. Eine Haftung kann ex contractu oder ex delicto begründet werden, wobei im Bereich von Open Source Software die Haftung ex contractu eine wesentlichere Rolle spielt⁵⁹¹. Im Folgenden wird daher die vorliegende Vertragskonstellation auf ihre haftungsrechtlichen Rechtsfolgen untersucht, die sich aus dem Lizenzvertrag ergeben oder die durch die Produkthaftung bestehen könnten.

6.1.1.3.3. Vertragshaftung

Die Open Source Lizenzen enthalten in Vielzahl gänzliche Haftungsausschlüsse, die teilweise nur durch einen Vorbehalt eingeschränkt werden, der auf die geltenden nationalen oder regionalen Rechtsordnungen

⁵⁹⁰ Der OGH hat in 2 Ob 300/97z JBI 2000, 249 = ecolex 1999, 821 die Haftung bei einer Schenkung bejaht.

⁵⁹¹ Eine Haftung ex delicto wird in dieser Arbeit nicht weiter behandelt, da es allzu logisch erscheint, dass jemand, der bspw. wissentlich schädliche Computerviren mit der Open Source Software verbreitet, für daraus entstandene Schäden zur Haftung herangezogen werden kann.

verweist⁵⁹². Die MPL erklärt aber bspw auch, dass der in der MPL enthaltene Haftungsausschluss dann nicht anwendbar ist, wenn nationale Rechtsprechung oder Gesetzgebung diesen Haftungsausschluss für ungültig halten⁵⁹³. Diese Freizeichnungsklauseln wurden vom OGH schon des Öfters bewertet⁵⁹⁴ mit dem Ergebnis, dass Vereinbarungen über den Ausschluss oder die Beschränkung der Haftung nur insoweit wirksam sind, als dass ihr Abschluss und ihre Anwendung im Einzelfall nicht gegen die guten Sitten verstoßen. Dabei kommt es darauf an, ob es sich um einen Schaden aus den für das Rechtsverhältnis typischen oder im Einzelfall nach dessen besonderen Verhältnissen voraussehbaren Gefahren handelt⁵⁹⁵. Schon früher wurden vom OGH nur voraussehbare und kalkulierbare Risiken als verzichtbar angesehen⁵⁹⁶. Des Weiteren ist auf die Bestimmungen des § 915 ABGB Rücksicht zu nehmen, da aufgrund der Schenkung nur ein einseitig verbindlicher Vertrag vorliegt⁵⁹⁷. An der Geltungskontrolle des § 864a ABGB iVm § 879 Abs 3 ABGB werden die Freizeichnungsklauseln der Open Source Lizenzen hinsichtlich eines gänzlichen Haftungsausschlusses regelmäßig scheitern. Bei Verbrauchergeschäften iSd KSchG muss zusätzlich § 6 Abs 3 KSchG einbezogen werden⁵⁹⁸. Eine geltungserhaltende Reduktion der Haftungsklauseln ist allerdings möglich⁵⁹⁹.

⁵⁹² Die GPL spricht dabei bspw von „*unless required by applicable law*“.

⁵⁹³ Siehe Ziffer 9 letzter Satz MPL: „*SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES; SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.*“

⁵⁹⁴ OGH 03.12.1981, 8 Ob 530/81, JBI 1983, 255; OGH 09.05.1985, 6 Ob 836/83, HS 16.609 = RdW 1985, 306 = JBI 1986, 172; OGH 29.05.1996, 3 Ob 2004/96v, HS 27.422 = ecolex 1996, 743 = JBI 1997, 458 (Staudegger) = RdW 1997, 73 = ÖBA 1997, 214

⁵⁹⁵ OGH 14.01.2004, 7 Ob 259/03v, immolex 2004/93.

⁵⁹⁶ OGH 03.12.1981, 8 Ob 530/81, JBI 1983, 255.

⁵⁹⁷ Gschnitzer in Klang (Hrsg), Kommentar zum allgemeinen bürgerlichen Gesetzbuch IV/1² (1968) 415; Binder in Schwimann, Praxiskommentar ABGB IV³ § 915 Rz 5; Rummel in Rummel, Kommentar I³ § 915 Rz 2.

⁵⁹⁸ Wiebe/Prändl, ÖJZ 2004, 636.

⁵⁹⁹ Staudegger in Jahnel/Schramm/Staudegger (Hrsg), Informatikrecht², 115.

Die Haftung des Lizenzgebers bleibt daher, mit besonderer Rücksichtnahme auf § 915 ABGB, auf Vorsatz und grobe Fahrlässigkeit beschränkt, für leichte Fahrlässigkeit haftet er nicht⁶⁰⁰. Dieses Ergebnis ist gerechtfertigt, wenn man bedenkt, dass die Weitergabe der Open Source Software unentgeltlich erfolgt und daher relativ gesehen einen größeren Nutzerkreis hat, als dies bei kommerziell vertriebener (Open Source) Software der Fall ist. Die Nutzungsmöglichkeiten sind daher für einen Lizenzgeber weitgehend unüberschaubar, sodass eine Haftung für nur leichte Fahrlässigkeit zu weit gehen würde. Für die Praxis bedeutet dies, dass der Lizenzgeber dann haftbar gemacht werden kann, wenn er vorsätzlich schädigenden Code⁶⁰¹ – bspw durch einen Virus oder einen Trojaner – zum Herunterladen bereitgestellt hat oder er von den schädigenden Auswirkungen der Software Kenntnis hatte oder Kenntnis haben musste.

6.1.1.3.3.1. Produkthaftung

Die Produkthaftung in Bezug auf Software, insbesondere was die Eigenschaft der Software als Produkt iSd § 4 PHG angeht, ist nicht nur in Österreich umstritten⁶⁰². An dieser Stelle wird diese Diskussion⁶⁰³ jedoch nicht aufgegriffen, sondern die Auswirkungen einer möglichen Produkthaftung

⁶⁰⁰ Ebenso: *Wiebe/Prändl*, ÖJZ 2004, 636.

⁶⁰¹ *Wiebe/Prändl*, ÖJZ 2004, 636 sprechen hier von "Fehlern im Programm". Der Begriff „Fehler“ ist in diesem Zusammenhang allerdings ungeeignet, da ein „Fehler“ nicht unbedingt eine haftungsrechtlich relevante schädigende Auswirkung haben muss, und wie bereits erwähnt, Open Source Software gerade aus dem Grund unentgeltlich verbreitet wird, um „Fehler“ ausbessern zu lassen.

⁶⁰² Für Deutschland siehe bspw *Jaeger/Metzger*, Open Source Software 152; *Schmitz*, Die vertraglichen Pflichten und die Haftung der Informationsanbieter im Internet: nationale und internationale Haftungsgrundlagen (2000) 71 f.

⁶⁰³ Siehe dazu *Andréewitch*, Zur Anwendung des Produkthaftungsgesetzes für Softwarefehler, EDVuR 2/1990, 50; *Welser*, Produkthaftungsgesetz² (1988) § 4 Rz 11 f.; *Preslmayer*, Handbuch der Produkthaftung² (2002) 53 f; *Staudegger* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 115; und insbesondere *Horwath*, Software – ein Produkt? *ecolex* 2000, 784.

aufgezeigt, da die Aussagen der EG⁶⁰⁴ zu diesem Thema richtungweisend sind. Geht man dementsprechend von einer Anwendbarkeit des PHG in Zusammenhang mit Software aus, gilt es zu klären, ob auch die vorliegende Vertragskonstellation, das Herunterladen der Software vom Lizenzgeber, ebenfalls unter die von § 4 PHG geforderte Produkteigenschaft fällt. Dazu wird auf das bereits in Kap 3.2.3. Gesagte verwiesen, wonach es im Endeffekt unerheblich ist, auf welche Art die Open Source Software auf den Computer des Lizenznehmers gelangt, da der Zweck der gleiche ist und nur ein Zwischenschritt ausgelassen wurde. § 6 PHG verlangt korrespondierend zu dieser Problematik nur, dass ein Produkt vom Unternehmer einem anderen, hier dem Lizenznehmer, in dessen Verfügungsmacht oder zu dessen Gebrauch übergeben wird. Dies alles sind Kriterien, die durch das Herunterladen erfüllt werden. Der letzte Satz des § 6 PHG gibt zu Denken, da hier von der Versendung an den Abnehmer gesprochen wird, wobei es selbstverständlich scheint, dass damit der Postweg gemeint ist. Schaut man sich den Postweg analog zum Herunterladen an, muss man zu dem Ergebnis kommen, dass das Produkt im Endeffekt beim Ab- bzw. Lizenznehmer ankommt, sei es durch tatsächliche Übergabe oder durch Post- bzw Online-Sendung.

Eine Freizeichnung ist nach den Bestimmungen des PHG nur eingeschränkt möglich, da die auf dem PHG beruhenden Ersatzpflichten gem § 9 PHG im Voraus weder beschränkt noch ausgeschlossen werden können⁶⁰⁵. Für Open Source Software spielt hierbei § 8 Abs 2 PHG eine wichtige Rolle, der eine Haftung für das „Entwicklungsrisiko“⁶⁰⁶ ausschließt. Demnach ist ein Haftungsausschluss in jenen Fällen möglich, in denen der Nachweis erbracht werden kann, dass die Eigenschaft des Produkts nach dem Stand der Wissenschaft und Technik zur Zeit des Inverkehrbringens nicht als Fehler erkannt werden konnte. Da Open Source Software wie Software im Allgemeinen immer versucht, auf dem aktuellsten Stand zu sein, wird man

⁶⁰⁴ ABI 1989 C 144, 11.

⁶⁰⁵ *Koziol/Welser*, Bürgerliches Recht II¹², 358.

⁶⁰⁶ *Welser*, Produkthaftungsgesetz² § 8 Rz 11 ff.

besonders im Fall eines „Software Development Kits“ (SDK) im Open Source Bereich davon ausgehen müssen, dass hier Programme auf dem aktuellsten Stand zum Herunterladen angeboten werden.

Die haftungsbegründenden Fehler des Produkts sind in § 5 PHG aufgezählt. Danach ist Open Source Software fehlerbehaftet, wenn die Software nicht die Sicherheit bietet, die man bei Gebrauch billigerweise zu erwarten hat. Dabei spielt auch der Zeitpunkt des Inverkehrbringens eine Rolle. Es darf gem § 6 Abs 2 PHG nicht als Fehler - und dies ist für Open Source Software besonders zu betonen - angesehen werden, wenn zu einem späteren Zeitpunkt ein besseres Produkt (Update) in den Verkehr gebracht wird.

Abschließend bleibt zu betonen, dass es unwahrscheinlich erscheint, dass ein Lizenzgeber von Open Source Software haftungsrechtlich iSd PHG zur Rechenschaft gezogen wird. Software kann nur mittelbar für eine Schädigung ursächlich sein, da die Software selbst keine Schädigung verursachen kann. Darüber hinaus wird Software, die Einsatz in Geräten und Computern findet, welche eine Gefährdung für den Menschen darstellen könnten - bspw Software im Flugbereich (Flugverkehrskontrolle, Navigation, im Flugzeug selbst, etc), im Gesundheitsbereich (Messgeräte in Krankenhäusern, Dosierungshelfer, etc.) oder in der Industrie (Steuerung von Kraftwerken, Maschinen, etc) - nicht einfach installiert und eingesetzt, sondern vorher aufwendig auf ihre Lauffähigkeit überprüft.

6.1.2. Erwerb der Software gegen Entgelt direkt vom Lizenzgeber



Abb 9

Die zweite Vertragskonstellation behandelt den Erwerb der Open Source Software direkt vom Lizenzgeber gegen Entgelt. Die Einordnung des Rechtsgeschäfts unter einen bestimmten Vertragstyp muss in diesem Fall einmal hinsichtlich der Überlassung der Software als solche überprüft werden und zum anderen hinsichtlich der Lizenzierung nach der Open Source Lizenz. Des Weiteren muss Rücksicht darauf genommen werden, auf welche Weise die Software erworben wird. Open Source Software wurde in den letzten Jahren hauptsächlich per Datenträger gegen Entgelt vertrieben⁶⁰⁷. Mittlerweile ist die Infrastruktur des Internet aber soweit fortgeschritten, dass es für Anbieter von Open Source Software effektiv ist, auch Programme mit größerem Datenvolumen zum Herunterladen bspw per FTP anzubieten⁶⁰⁸.

6.1.2.1. Schenkung

Eine Schenkung iSd §§ 938ff ABGB ist bei dieser Fallkonstellation hinsichtlich der Nutzungsrechte zu überprüfen, da die Open Source Lizenzen keine Lizenzgebühren zulassen. Hierbei wird auf das in Kap 6.1.1.1. Ausgeführte verwiesen. Das Entgelt bei dieser Variante bezieht sich nur auf Kosten, welche durch andere Leistungen – wie bspw das Herstellen der Programmkopien oder einer weiterführenden Dokumentation – entstanden sind.

⁶⁰⁷ Vgl auch *Jaeger/Metzger*, Open Source Software 155.

⁶⁰⁸ Vorrangig ist hierbei auf die Möglichkeit zum Herunterladen diverser Linuxdistributionen hinzuweisen.

Aus rechtlicher Sicht unwesentlich ist hierbei die Art der Übertragung (auf Datenträger oder Herunterladen über Datennetz), da dies keine Auswirkungen auf die Einordnung als Schenkung hat.

6.1.2.2. Kauf

Hinsichtlich des Kaufvertrages iSd §§ 1053 ff ABGB entstehen verschiedene Problemfelder, die es zu erörtern gilt. Zuerst werden die unterschiedlichen Arten der Übertragung bewertet, da diese verschiedene Auswirkungen auf die Rechtsfolgen haben können.

6.1.2.2.1. Vertrieb auf Datenträger

Die Einordnung des Vertriebes von Open Source Software auf einem Datenträger gegen Entgelt stellt aus rechtlicher Sicht kein Problem dar, weil es sich im Fall der Open Source Software erübrigt zu diskutieren, ob Software als körperlich oder unkörperlich zu bewerten ist. Lehre und Rsp haben sich darüber hinaus für eine Bewertung als körperliche Sache ausgesprochen.⁶⁰⁹ Die Open Source Software kann – was den Quellcode betrifft – vom Datenträger getrennt behandelt werden, da die Open Source Lizenzen das Verlangen von Lizenzgebühren für den Quellcode selbst verbieten und nur der Datenträger selbst kostenpflichtig sein darf. Daher wird in diesem Fall von den Open Source Lizenzen selbst eine Trennung zwischen Software und Hardware vollzogen, die schon deshalb Sinn macht, da die Open Source Software andere Fehler als der

⁶⁰⁹ *Staudegger* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 82; *Ertl*, Allgemeine Geschäftsbedingungen der Softwareverträge, EDVuR 1994, 19 (20); *Holzinger*, Bedienerhandbuch – Schlüssel für EDV-Entscheidungen? EDVuR 1994, 54 (55); *Blocher*, Die Rechtsstellung des Software-Anwenders nach österreichischem und deutschem Urheberrecht, EDVuR 1994, 5 (13 f unter Berufung auf *Ertl*). Der OGH hat in 5 Ob 504/96, HS 28.417 = SZ 70/202 = ecolex 1998, 127 (*Wilhelm*) ebenfalls ausgesprochen, dass die „Übertragung fertiger Standardsoftware auf Datenträgern gegen einmaliges Entgelt [...] als Kauf einer körperlichen Sache zu qualifizieren“ sei.

Datenträger aufweisen kann, was für die weitere haftungs- und gewährleistungsrechtliche Betrachtung ausschlaggebend ist. Bei der Weitergabe der Open Source Software in Objektform wird diese Trennung nicht so deutlich, da hierfür auch für die Software Entgelt verlangt werden kann. Hierbei ist analog auf den Umgang mit proprietärer Standardsoftware zu verweisen, bei der wie erwähnt, die Überlassung auf einem Datenträger gegen einmaliges Entgelt als Kauf einer körperlichen Sache iSd § 1053 ABGB gewertet wurde⁶¹⁰.

6.1.2.2.2. Vertrieb über Download

Bei dem Vertrieb der Open Source Software durch Herunterladen hat der OGH festgestellt, dass zwar der Begriff "Kaufvertrag" in Zusammenhang mit dem Rechtsgeschäft in aller Regel vermieden wird und durch andere Vertragsbezeichnungen – wie bspw Nutzungsüberlassung, Nutzungskauf oder Lizenzvertrag – ersetzt wird, aber bei wirtschaftlicher Betrachtung des Geschäftsfalls dem Erwerber die zeitlich unbegrenzte Verfügungsmacht über die Software eingeräumt wird. Der OGH kommt daher zu dem Ergebnis, dass ein derartiger Vertrag durchaus als "Sachkauf" beurteilt werden kann⁶¹¹.

Auch hierbei ist wieder die Trennung zwischen Software und Hardware zu erwähnen. Vergleicht man die beiden unterschiedlichen Vertriebsvarianten stellt man fest, dass die Software in beiden Fällen auf gleiche Weise beim Lizenznehmer zum Einsatz kommen wird, die genutzte Hardware sich aber unterscheidet. Der bei der ersten Variante benutzte Datenträger wird in diesem Fall nicht benötigt und kann daher auch keine haftungs- und

⁶¹⁰ Siehe auch *Staudegger*, JBI 1998, 604.

⁶¹¹ OGH 23.05.2000, 4 Ob 30/00 – Handwerkerpaket WIN 2.3 – ecolex 2000/291 = MR 2000, 249 = RdW 2000/651 = ÖBI 2001, 141; OGH 14.10.1997, 5 Ob 504, 505/96, ecolex 1998, 127.

gewährleistungsrechtlichen Folgen auslösen, die entstehen könnten, wenn der Datenträger bspw selbst schadhaft wäre.

6.1.2.2.3. Gesamtbeurteilung des Rechtsgeschäfts

Eine Gesamtbetrachtung des Rechtsgeschäfts muss an dieser Stelle zunächst hinsichtlich des Vertriebs auf einem Datenträger vorgenommen werden, da es fraglich ist, ob die auf dem Datenträger verkörperte Open Source Software mit diesem zusammen verkauft wird oder ob in diesem Fall eine gemischte Schenkung vorliegt. Außerdem gilt es bei dieser Fallvariante die durch die Open Source Lizenzen eingeräumten Nutzungsrechte in Zusammenhang mit dem Kauf der Software zu bewerten.

Beim Erwerb von Open Source Software auf einem Datenträger gegen Entgelt muss von einem einheitlichen Kaufvertrag hinsichtlich des Datenträgers und Open Source Software ausgegangen werden⁶¹². Hier ist der deutschen Rsp zu folgen, die auf die Verkehrsauffassung des jeweiligen Verkehrskreises abstellt⁶¹³. Es erscheint verständlich, dass der Käufer eines Datenträgers, auf welchem die Software gespeichert ist, von einem einheitlichen Kaufvertrag ausgeht⁶¹⁴. Deshalb ist es unwichtig herauszufinden, ob hinsichtlich der Open Source Software ein Schenkungswille seitens des Lizenzgebers bestand. Dieser müsste dem Käufer klar erkennbar gemacht werden, um eine andere rechtliche Einordnung vornehmen zu können⁶¹⁵.

⁶¹² *Omsels* in FS *Hertin* 151 Fn 24; *Jaeger/Metzger*, Open Source Software 157.

⁶¹³ BGH 4.11.1987 – VIII ZR 314/86, NJW 1988, 406 (409).

⁶¹⁴ *Jaeger/Metzger*, Open Source Software 156.

⁶¹⁵ Denkbar wäre bspw das Verlangen einer geringen Gebühr, welche dem Käufer bewusst macht, dass das Entgelt nur als Aufwandsentschädigung für das Bereitstellen des Datenträgers bzw für diesen selbst zu entrichten ist.

In Zusammenhang mit dem Kaufvertrag über die Software ist in Folge die Einräumung der Nutzungsrechte zu bewerten, welche soweit sie über die gesetzlich eingeräumten hinausgehen – wie bereits besprochen⁶¹⁶ – als Schenkung zu qualifizieren sind⁶¹⁷. Am nahe liegendsten ist die Einordnung des Rechtsgeschäfts als gemischte Schenkung, da im Falle des Erwerbs von Open Source Software gegen Entgelt der Parteiwille iSd § 935 ABGB zu interpretieren ist, wonach aus dem Verhältnis der am Rechtsgeschäft beteiligten Personen vermutet werden darf, dass sie einen, „aus einem entgeltlichen und unentgeltlichen vermischten Vertrag schließen wollten“. Auch das Erfordernis, dass sich beide Parteien dieses doppelten Charakters des Geschäfts bewusst sein müssen, ist in diesem Fall als gegeben zu erachten⁶¹⁸.

Interessanter als die Beurteilung des Rechtsgeschäfts ist die rechtliche Behandlung dieser gemischten Schenkung. Kaufvertrag und Schenkung folgen im Falle von Open Source Software einander zeitlich nach⁶¹⁹, da der Kauf der Software noch keine Schenkung begründet. Diese kommt erst durch Abschluss des Open Source Lizenzvertrags zustande. Es ist daher schwierig zu entscheiden, ob die gemischte Schenkung nach der Trennungs- oder der Einheitstheorie⁶²⁰ behandelt werden muss, da auch der OGH in dieser Hinsicht keine klare Meinung vertritt⁶²¹. Für den Erwerb der Open Source Software auf einem Datenträger gegen Entgelt wird eine Trennung bzw eine Aufspaltung ohne Zerreiung des inneren Zusammenhangs auch in Rücksichtnahme auf § 878 Satz 2 ABGB möglich sein⁶²², weshalb der Abschluss des Lizenzvertrags

⁶¹⁶ Siehe Kap 4.2.

⁶¹⁷ Die folgende Besprechung bezieht sich sowohl auf den Erwerb der Open Source Software gegen Entgelt, als auch auf jenen durch Download.

⁶¹⁸ *Binder* in *Schwimmann*, Praxiskommentar ABGB IV³ § 938 Rz 50; *Wiebe/Prändl*, ÖJZ 2004, 636.

⁶¹⁹ Vgl Kap 4.2.

⁶²⁰ *Binder* in *Schwimmann*, Praxiskommentar ABGB IV³ § 938 Rz 52 f.

⁶²¹ Der Trennungstheorie folgend: OGH 02.03.1955, 1 Ob 81/55 = SZ 28/60: Wiederruf/laesio enormis; OGH 08.03.1966, 8 Ob 26/66 = 39/46; Wiederruf; Der Einheitstheorie folgend: 19.01.1969, 5 Ob 260/68 = SZ 5/42, Widerruf; OGH 14.09.1983, 3 Ob 574/83 = RdW 1984, 43; OGH 27.10.1992, 5 Ob 124/92 = JBI 1993, 312.

⁶²² *Binder* in *Schwimmann*, Praxiskommentar ABGB IV³ § 938 Rz 50.

dem Schenkungsrecht iSd §§ 938 ff ABGB unterliegt. Der Kauf der Software mit oder ohne Datenträger muss dagegen dem Kaufrecht zugeordnet werden, solange dem Käufer nicht explizit bewusst war oder sein musste, dass er nur den Datenträger oder eine Dienstleistung, die in Zusammenhang mit dem Erwerb steht, bezahlt und die Open Source Software selbst geschenkt bekommt. Die aus dieser Gesamtbetrachtung resultierenden Rechtsfolgen werden nachfolgend geprüft.

6.1.2.3. Rechtsfolgen

Im vorliegenden Fall treten die Rechtsfolgen je nach Rechtsgeschäft ein, da die Unterscheidung beim Erwerbsvorgang – Unentgeltlichkeit bei der Schenkung und Entgeltlichkeit beim Kaufvertrag – maßgeblich für die möglichen Rechtsfolgen sind.

6.1.2.3.1. Gewährleistung

Während bei der Schenkung in der vorhergehenden Fallvariante die Gewährleistung ausgeschlossen werden kann und damit die Gewährleistungsausschlüsse der Open Source Lizenzen ihre Gültigkeit behalten, spricht die Entgeltlichkeit in der vorliegenden Fallvariante für die Anwendbarkeit von Gewährleistungsrecht iSd §§ 922 ff ABGB. Ein Ausschluss ist nach den Bestimmungen des § 9 KSchG bei Verbraucherverträgen auch nicht möglich, da dieser für den Verbraucher nachteilige Lizenzbestimmungen untersagt. Die Gewährleistung beim Erwerb der Open Source Software gegen Entgelt direkt vom Lizenzgeber wird sowohl in Bezug auf den Datenträger überprüft als auch wird untersucht, in welcher Form der Lizenzgeber die Mängelfreiheit der Open Source Software zu gewährleisten hat.

6.1.2.3.1.1. *Der Erwerbsvorgang*

Das vom OGH herausgearbeitete Kriterium der „Abnahme des IT-Systems“⁶²³ – um zwischen Nichterfüllung und Gewährleistung entscheiden zu können – muss bei Open Source Software unterschiedlich bewertet werden. Open Source Software, welche vom Lizenzgeber direkt erworben wird, kann nicht zwangsläufig mit großen proprietären IT-Systemen gleichgesetzt werden, weshalb oft kein Abnahmeverfahren wie bei komplexen IT-Systemen vorgenommen wird. Bei dem Erwerb von Open Source Software wird man daher von einer stillschweigenden Annahmeerklärung ausgehen können, die darin besteht, dass der Erwerber die Open Source Software vorbehaltlos bezahlt und kommentarlos in Betrieb nimmt⁶²⁴. Sollten größere IT-Projekte durch den Einsatz von Open Source Software verwirklicht werden, kann es zu einem umfangreichen Abnahmeverfahren kommen. Beim Erwerb von Open Source Software direkt vom Lizenzgeber erscheint dies allerdings eher unwahrscheinlich.

6.1.2.3.1.2. *Der Mängelbegriff*

Ein Mangel – ein quantitatives, qualitatives oder rechtliches Zurückbleiben des Schuldeten hinter dem vertraglich vereinbarten iSd § 922 Abs 1 Satz 1 ABGB – ist hinsichtlich des Datenträgers recht einfach zu bewerten, da ein Mangel hier leichter festzustellen sein wird. Auf mögliche Mängel selbst einzugehen, würde an dieser Stelle zu weit führen und ist im Einzelfall zu beurteilen.

Fehler bzw Mängel bei der Open Source Software sind nach dem Stand der Technik im jeweiligen Bereich und daher auch hier immer im Einzelfall zu

⁶²³ Staudegger in *Jahnel/Schramm/Staudegger*, Informatikrecht², 102 f.

⁶²⁴ Staudegger in *Jahnel/Schramm/Staudegger*, Informatikrecht², 103.

beurteilen⁶²⁵. Da § 922 ABGB auf die im Vertrag zugesicherte Leistung abzielt, gilt es festzustellen, ob der Lizenzgeber mit dem Erwerber eventuell über einen Open Source Lizenzvertrag hinausgehende Leistungen vereinbart hat oder ihm bestimmte Leistungsmerkmale zugesichert hat. Hat der Lizenzgeber mit dem Erwerber nichts vereinbart, gelten die gleichen Grundsätze wie beim gewährleistungsrechtlichen Umgang mit proprietärer Standardsoftware. Dabei ist auf die Besonderheiten der Entwicklung von Open Source Software Rücksicht zu nehmen, da es einen Unterschied macht, ob man Open Source Software im Quellcode oder in Objektform auf einem Datenträger vertreibt. Während bei dem (seltenen, weil wirtschaftlich uninteressanten) Vertrieb in Quellcodeformat die Intention der Weiterentwicklung im Vordergrund steht und es daher zum Vertrieb von Betaversionen (nicht vollständig fertig entwickelte Versionen) kommen kann, die nicht selten mangelanfälliger sind, ist beim Vertrieb in Objektform der Anwendungsgedanke vorrangig. In diesem Fall wird es regelmäßig zu Verbrauchergeschäften iSd KSchG kommen, bei welchem dem Erwerber keine allzu großen Fachkenntnisse im Umgang mit Open Source Software zugemutet werden dürfen, da es für diesen kaum zum Abschluss eines Open Source Lizenzvertrags kommt. Die Gewährleistungsausschlüsse der Open Source Lizenzen sind in diesem Fall unwesentlich, da diese nicht Bestandteil des Kaufvertrags werden.

Sollte es im Zuge der Weiterentwicklung zum Abschluss eines Open Source Lizenzvertrags kommen, werden die Gewährleistungsausschlüsse der Open Source Lizenzen unwirksam, wenn es sich bei dem Lizenznehmer, wie eingangs erwähnt, um einen Verbraucher iSd KSchG handelt. Bei Rechtsgeschäften unter Unternehmern sind die Ausschlüsse nicht pauschal unanwendbar, da diese Gewährleistungsausschlüsse im Gegensatz zu Verbrauchergeschäften dispositiv regeln können. Die Gewährleistungsausschlüsse der Open Source Lizenzen können aber je nach

⁶²⁵ *Staudegger in Jahnel/Schramm/Staudegger, Informatikrecht*², 106.

Fall die Grenzen der Sittenwidrigkeit iSd § 879 Abs 3 ABGB überschreiten⁶²⁶. Es muss hierbei darauf geachtet werden, welche Angaben der Lizenzgeber hinsichtlich der Software gegenüber dem Lizenznehmer gemacht hat.

Bezüglich der Beweislast für Mängel an der Software ist Folgendes festzustellen: Derjenige Vertragspartner, der sich auf einen bestimmten Vertragsinhalt beruft, muss diesen auch beweisen⁶²⁷. Allerdings lässt sich aus § 924 ABGB schließen, dass Mängel, die innerhalb der ersten sechs Monate nach Erwerb der Software auftreten, schon bei der Übergabe vorhanden gewesen sein müssen. In diesem Fall liegt es am Lizenzgeber, sich freizubeweisen. Sollte der Lizenznehmer Mängel zur Gewährleistung heranziehen, welche sich nicht aus dem normalen Gebrauch der Software heraus ergeben, trifft den Lizenznehmer die Beweislast, dass diese „besonderen Eigenschaften“⁶²⁸ vom Lizenzgeber vertraglich zugesichert wurden.

Sollten Mängel an der Software auftreten, muss bei Open Source Software die Herausgabe des Quellcodes nicht erst (gerichtlich) verlangt werden⁶²⁹. Der Erwerber ist theoretisch selbst in der Lage, eine notwendige Verbesserung vorzunehmen. Sollte ihm das aber entweder nicht möglich sein oder er keine Notwendigkeit darin sehen, eine eigenständige Verbesserung vorzunehmen, kann er sich der nachfolgenden Gewährleistungsbehelfe bedienen.

⁶²⁶ *Wiebe/Prändl*, ÖJZ 2004, 636; *Staudegger* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 112.

⁶²⁷ *Reischauer*, Das neue Gewährleistungsrecht und seine schadensersatzrechtlichen Folgen, JBI 2002, 137 (141).

⁶²⁸ *Reischauer*, JBI 2002, 141.

⁶²⁹ Siehe bspw: *Wolff*, Die Schlichtung von EDV – Streitigkeiten EDVuR 1988, 2; OGH 03.08.2005, 9 Ob 81/04h = MR 2005, 480.

6.1.2.3.1.3. Die Gewährleistungsbehelfe

§ 932 ABGB zählt die Arten der Gewährleistungsbehelfe auf, die darin bestehen,

1. zunächst eine Verbesserung der Software oder den Austausch des mangelhaften Produkts durch den Lizenzgeber zu verlangen (§ 932 Abs 2 ABGB)
2. sollten sowohl Verbesserung als auch Austausch nicht möglich oder vom Lizenzgeber nicht in angemessener Frist vorgenommen worden sein, kann der Erwerber eine Preisminderung oder Wandlung des Vertrags fordern (§ 932 Abs 4 ABGB).

Das ABGB gibt dem Lizenzgeber zunächst das Recht zur Fehlerbehebung. Dieser Gewährleistungsbehelf wird vorrangig für Softwaremängel in Frage kommen, da bei Fehlern an der Hardware es in der Regel zum Austausch kommen wird. Eine Verbesserung kann der Lizenzgeber bspw durch die Lieferung eines Bugfixes oder Patches vornehmen. Außerdem sollte es vor allem beim Erwerb der Open Source Software durch Herunterladen gestattet sein, die Verbesserung dadurch vorzunehmen, dass die Bugfixes oder Patches auf einer Webseite zum Download angeboten werden wie es auch in der Praxis überwiegend üblich ist. Beim Erwerb der Open Source Software auf einem Datenträger wird diese Möglichkeit zwar häufig in der Praxis angewendet, hier muss aber auf Umfang und Art des Programms Rücksicht genommen werden. Bei datenmäßig umfangreichen Open Source Programmen wird dem Erwerber unter Umständen erlaubt sein, Verbesserungen auf einem Datenträger fordern zu dürfen. Diese Verbesserungen müssen innerhalb einer angemessenen Frist vorgenommen werden⁶³⁰. Bei der Bereitstellung eines Bugfixes bzw Patches auf der Webseite ist im Einzelfall zu überprüfen, ob es dem Erwerber zugemutet werden kann, eine Verbesserung auf diese Art hinnehmen zu müssen.

⁶³⁰ OGH 24.06.1998, 3 Ob 2427/96z, ecolex 1999, 16.

Ein Austausch wird in der Regel nur bei Mängeln am Datenträger sinnvoll sein, da es unwirtschaftlich ist, einen solchen zu reparieren. Der Erwerber kann in diesem Fall den Austausch des Datenträgers gegen einen fehlerfreien fordern. Der Austausch der Software wird nur in jenen Fällen Sinn machen, in denen dem Erwerber eine falsche Software geliefert wurde⁶³¹ oder die Fehler so umfangreich sind, dass eine Verbesserung nicht sinnvoll möglich ist und dem Lizenzgeber auch nicht iSd § 932 Abs 4 ABGB zugemutet werden kann.

Sollten eine Verbesserung oder der Austausch nicht möglich sein - sei es, dass sie für den Lizenzgeber mit einem unverhältnismäßig hohen Aufwand verbunden sind oder der Lizenzgeber diesen Ansprüchen nicht in angemessener Frist nachkommt - stehen dem Erwerber als sekundäre Gewährleistungsbehelfe die Preisminderung oder Wandlung des Vertrags zu. Eine Preisminderung kann unabhängig von der Art des Mangels vorgenommen werden⁶³², wird allerdings beim Erwerb von Software nur dann Sinn machen, wenn die Mängel geringfügig sind (schon eine zu 40% schadhafte Software wird nicht mehr nutzbar sein). Letztendlich bleibt noch die Möglichkeit der Wandlung, wonach der Erwerbsvorgang ex tunc rückgängig gemacht wird⁶³³. Während beim Erwerb des Programms auf einem Datenträger dieser wieder zurückgestellt werden muss, reicht beim Erwerb durch Herunterladen die vollständige Deinstallation der Software.

6.1.2.3.2. Haftung

Die Haftung wurde bereits im vorherigen Fall besprochen⁶³⁴. Das dort Ausgeführte findet auch auf den Erwerb von Open Source Software gegen

⁶³¹ Vorstellbar wäre hier vorrangig die Lieferung einer falschen Programmversion, welche nicht durch ein Update auf die vertraglich vereinbarte Version gebracht werden kann.

⁶³² *Reischauer*, JBI 2002, 150.

⁶³³ *Staudegger* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 108.

⁶³⁴ Vgl daher Kap 6.1.1.

Entgelt direkt vom Lizenzgeber Anwendung, da das Merkmal der Entgeltlichkeit bzw Unentgeltlichkeit auf die haftungsrechtlichen Folgen keine Auswirkung hat. Der Lizenzgeber kann daher mit Rücksichtnahme auf § 915 ABGB weder vorsätzliches Handeln noch grobe Fahrlässigkeit ausschließen. Eine besondere Betrachtung der Vertragshaftung ist für diesen Fall daher nicht notwendig. Hinsichtlich der Produkthaftung muss festgestellt werden, dass eine Produkteigenschaft iSd § 4 PHG beim Erwerb der Software auf einem Datenträger unbedingt bejaht werden muss und daher die relevanten Bestimmungen des PHG zu beachten sind. Hierbei wird erneut auf das zum vorherigen Fall Erörterte verwiesen⁶³⁵.

6.1.3. Erwerb der Software gegen Entgelt im Einzelhandel oder vom Distributor

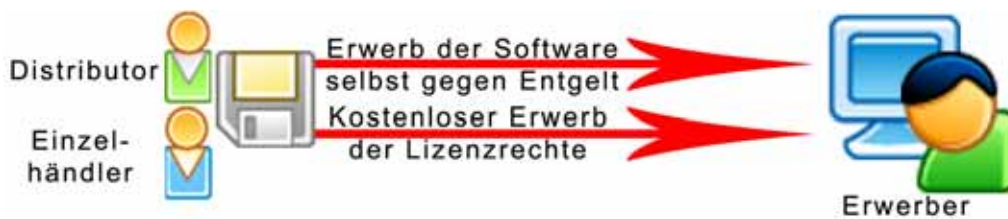


Abb 10

Durch den zunehmenden Einsatz von Open Source Software im privaten Bereich kommt dem Erwerb der Software gegen Entgelt im Einzelhandel oder vom Distributor wohl der größte Stellenwert zu. Eine Distribution ist die Zusammenstellung von Software, die als Komplettpaket weitergegeben wird und vor allem im Open Source Bereich häufig genutzt wird, da viele Open Source Programme zwar auf den gleichen Quelltexten basieren, aber unterschiedliche Eigenheiten bspw bei der Installation, im Erscheinungsbild oder in der konkreten Programmauswahl haben können.

⁶³⁵ Vgl Kap 6.1.1.

Des Weiteren bestehen diese Distributionen nicht ausschließlich aus Open Source Software wie die folgende Grafik verdeutlicht.

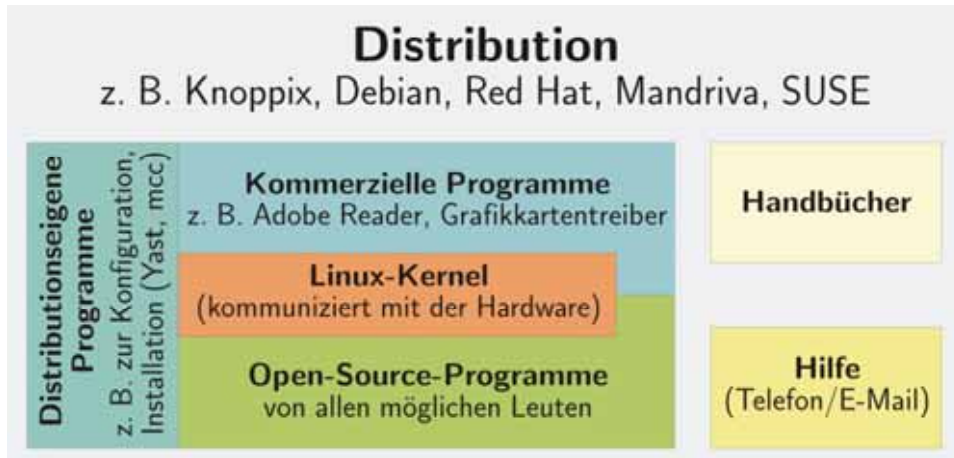


Abb 11⁶³⁶

Die Distributionen können entweder vom Distributor selbst erworben werden oder durch Versand der Distribution. Häufig steht aber ein weiteres Kettenglied in der Reihe, da der Erwerb einer solchen Distribution oft durch den Einzelhandel geschieht, welcher mit der Zusammenstellung und Entwicklung nichts zu tun hat.

Der Unterschied zu den vorherigen Fallkonstruktionen besteht darin, dass die Distributoren bzw der Einzelhandel nicht selbst Lizenzgeber der Software sind. Auch hier wird wieder die vorher gemachte Trennung – Erwerb der Software an sich auf der einen und Einräumung bzw Abschluss des Open Source Lizenzvertrags auf der anderen Seite – deutlich. Diese Trennung wird im Folgenden rechtlich beurteilt.

6.1.3.1. Schenkung und Kauf

Beim Erwerb der Software von einem Distributor oder im Einzelhandel gegen Entgelt gilt das Gleiche wie beim Erwerb der Software gegen Entgelt

⁶³⁶ <http://de.wikipedia.org/wiki/Bild:Linux-Distribution.svg>.

direkt vom Lizenzgeber⁶³⁷. Der Unterschied zwischen den beiden Fallvarianten ist der, dass die Einräumung der Nutzungsrechte bzw der Abschluss des Open Source Lizenzvertrags mit dem Lizenzgeber vereinbart werden, während der Erwerb der Software selbst mit dem Distributor oder Einzelhändler ein eigenständiges Rechtsgeschäft bildet. Während die Einräumung der Nutzungsrechte dem Schenkungsrecht iSd § 938 ABGB unterliegen, muss auf den Erwerbsvorgang Kaufrecht iSd § 1052 ABGB angewendet werden.

Während der Erwerb der Software gegen Entgelt vom Lizenzgeber nicht als gemischter Vertrag eingestuft werden musste, ist die Bewertung für diese Fallkonstruktion problematisch, da sowohl das Rechtsgeschäft der Schenkung nicht zwingend ist als auch die Unterschiedlichkeit der Vertragspartner beachtet werden muss. Die Einordnung des Erwerbsvorgangs als gemischten Vertrag⁶³⁸ scheint daher wenig nachvollziehbar. Beide Rechtsgeschäfte sind in diesem Fall deutlich von einander zu trennen⁶³⁹.

Die Aufspaltung der Rechtsgeschäfte beim Erwerbsvorgang in Schenkung iSd § 938 ABGB und Kauf iSd § 1052 ABGB wird an dieser Stelle auf mögliche Rechtsfolgen überprüft.

6.1.3.2. Rechtsfolgen

Die Rechtsfolgen bei dieser Fallkonstruktion entsprechen in etwa jenen des Erwerbs der Software gegen Entgelt vom Lizenzgeber, da es sich aber um

⁶³⁷ Siehe Kap 6.1.2.

⁶³⁸ So *Wiebe/Prändl*, ÖJZ 2004, 636, die es als schwierig erachten, eine Grenze zu ziehen, „zwischen der Haftung für die entgeltliche Leistung der Zusammenstellung der Software und der Haftung für die unentgeltliche Software selbst“, dabei aber übersehen, dass die Software selbst sehr wohl entgeltlich erworben werden kann da „Free Software“ eben nicht „Free Beer“ bedeutet.

⁶³⁹ Für Deutschland siehe *Jaeger/Koglin/Kreutzer/Metzger/Schulz*, Die GPL kommentiert und erklärt 147.

unterschiedliche Vertragspartner handelt, muss gewährleistungs- und haftungsrechtlich zwischen diesen unterschieden werden.

6.1.3.2.1. Gewährleistung

Bei dem Erwerb einer Distribution gegen Entgelt wird es in der Regel schwierig sein, einen Mangel einem bestimmten Lizenzgeber bzw Urheber zurechnen zu können, da die Fehleranalyse im Softwarebereich sehr aufwendig sein kann und keine eindeutigen Ergebnisse liefert. Des Weiteren ist darauf zu achten, dass gewährleistungsrechtliche Ansprüche gegen den Lizenzgeber bzw Urheber der Open Source Software nur dann gerichtet werden können, wenn sich die Entgeltlichkeit des Erwerbs auch auf das Programm erstreckt, was nicht zwingend der Fall sein muss.

Mängel werden dem Lizenzgeber meist als Rechtsmängel zuzuweisen sein, wenn er einem Lizenznehmer nicht die geschuldete Rechtsposition verschafft. Dies kann dann der Fall sein, wenn der Wahrnehmung der Rechte aus dem Open Source Lizenzvertrag Rechte Dritter entgegenstehen, die der Lizenzgeber dem Lizenznehmer verschwiegen hat. Denkbar wäre bspw der Fall, dass ein Programmierer ein für seinen Arbeitgeber entwickeltes Programm eigenmächtig unter eine Open Source Lizenz gestellt hat und weiterverbreitet.

Diese Rechtsmängel können ebenso beim Distributor oder Einzelhändler auftreten, wenn dieser ebenfalls nicht zur Weitergabe berechtigt war und dem Erwerber deshalb die rechtmäßige Nutzung untersagt ist. Ein gutgläubiger Erwerb von Urheberrechten kann nach österreichischem Recht wie besprochen nicht stattfinden.⁶⁴⁰

⁶⁴⁰ Siehe Kap. 4.5.

Bei der Beurteilung der Gewährleistung ist es wichtig, auf die Vertragsbeziehung zwischen Distributor bzw Einzelhändler und dem Erwerber einzugehen, da diese gesonderte Vereinbarungen hinsichtlich der Gewährleistung getroffen haben können und ebenfalls entscheidend ist, ob die verbraucherrechtlichen Vorschriften des KSchG zur Anwendung kommen.

Die Open Source Lizenzen bestimmen in Zusammenhang mit diesen zusätzlichen Gewährleistungsvereinbarungen, dass der Distributor bzw Einzelhändler dafür Sorge tragen muss, den Lizenzgeber dabei von jeglichen Ansprüchen schadlos zu halten. Dies ist zwar eine legitime Forderung, findet aber nur innerhalb der Grenzen des § 915 ABGB Rückhalt. Der Distributor bzw Einzelhändler kann allerdings nicht die Gewährleistung für die Software auf den Lizenzgeber „abschieben“.

Er muss für Mängel am Datenträger eintreten, da diese auch von ihm bereitgestellt wurden. Das Gleiche gilt aber auch für Mängel an der Software. Da der Urheber den Kauf der Software betreffend nicht Vertragspartner geworden ist, muss der Distributor bzw Einzelhändler für die gewährleistungsrechtlichen Folgen eintreten. Die Gewährleistungsbehelfe sind dabei dieselben wie in der vorherigen Fallkonstruktion.

6.1.3.2.2. Haftung

Betreffend der Haftung ist festzustellen, dass die haftungsrechtlichen Folgen dem zur ersten Fallkonstruktion Erörterten entsprechen und denjenigen in die Pflicht nehmen, der den Schaden zu verantworten hat, was im Streitfall nicht einfach zu beurteilen sein wird. Für die Haftung nach dem PHG kann nur der Distributor bzw Einzelhändler zur Rechenschaft gezogen werden, da das PHG auf denjenigen verweist, der das Produkt in Verkehr gebracht hat.

6.2. Ergebnis

Die Haftungs- und Gewährleistungsausschlüsse sind nach dem bisher Ausgeführten nicht grundsätzlich unanwendbar. Die Aufspaltung der Rechtsgeschäfte ist zwingend notwendig und erleichtert eine systematische Betrachtungsweise des Problemfeldes. Klare Antworten auf die gewährleistungs- und haftungsrechtlichen Fragen sind das wichtigste Kriterium beim Erwerb von Open Source Software, da es für einen Erwerber bzw Lizenznehmer wichtig ist, in diesen Punkten Rechtssicherheit zu genießen. Darüber hinaus ist es für die Open Source Gemeinschaft von großer Bedeutung zu wissen, welche Risiken mit der Entwicklung und Verbreitung von Open Source Software verbunden sind. Man stelle sich bspw die Folgen vor, wenn eine Open Source Serversoftware versagt und daraufhin lebenswichtige Systeme der Flugsicherung ausfallen. Der Einsatz von Open Source Software in diesen sensiblen Bereichen soll aber dennoch möglich sein und das Entstehen für Gewährleistung und Haftung dabei gerecht verteilt werden.

Die Zuordnung der Einräumung der Nutzungsrechte zur Schenkung iSd § 938 ABGB erscheint momentan die einzig Logische zu sein. Die Merkmale, die im Moment dazu dienen, die Einräumung als Schenkung zu qualifizieren, können sich selbstverständlich ändern, wenn man die Entwicklung verschiedener Open Source Projekte betrachtet. Dabei erscheint der Schenkungscharakter bei großen Open Source Firmen wie MySQL grenzwertig zu sein, sich jedoch dadurch rechtfertigen zu lassen, dass die kommerzielle Vermarktung der Software nicht unter einer Open Source Lizenz vorgenommen wird. Das Dual Licensing dient als weiteres Indiz, dass es im Fall der Open Source Lizenzierung zur Schenkung kommt. Bei der kostenlosen Weitergabe der Software wird dies auf jeden Fall anzunehmen sein, wobei neben der Einräumung der Nutzungsrechte die Software selbst ebenfalls geschenkt wird.

Die Gewährleistungsbehelfe des § 932 ABGB sind in Bezug auf Software als gelungen zu bewerten. Da Software nicht fehlerfrei sein kann, der technische Einsatz aber dem Zweck der Software entsprechend möglich und daher gewährleistet sein soll, ist der Verbesserungsanspruch des Erwerbers bei Mängeln an der Software als erste Maßnahme gegen den Verkäufer zu begreifen. Auch dem Verkäufer gegenüber erscheint diese vorrangig zu wählende Möglichkeit der Gewährleistung angemessen, bekommt dieser doch die Chance, Fehler an der Software ohne Aufhebung des Vertragsverhältnisses zu beheben. Sollte Open Source Software durch Herunterladen bezogen worden sein, wird man dem Anbieter die Möglichkeit einräumen müssen, einem Verbesserungsanspruch dadurch zu entsprechen, die notwendige Software zur Fehlerbehebung auf seiner Webseite zum Herunterladen bereitzustellen. Beim Erwerb der Software auf einem Datenträger wird diese Art der Verbesserung anerkannt werden müssen, solange es dem Erwerber zugemutet werden kann, die Software ohne großen Aufwand herunterladen zu können. Als Kriterien für den Aufwand müssen das Datenvolumen und die Art der Installation berücksichtigt werden. Ein ausgebesserter Quellcode, welcher vom Erwerber zu kompilieren ist, wird daher nicht ausreichen, um einem Verbesserungsanspruch nachzukommen. Dies kann nur dann der Fall sein, wenn die Art des Rechtsgeschäfts zwischen Verkäufer und Erwerber darauf hinweist, dass eine Ausbesserung des Quellcodes als ausreichend angesehen werden muss⁶⁴¹ oder diese Art der Verbesserung zwischen den beiden Vertragspartnern explizit vereinbart worden ist.

Die Haftung für Schäden, die aus dem Einsatz der Open Source Software entstehen, wirft kaum rechtliche Probleme auf. Der OGH sieht nur voraussehbare und kalkulierbare Risiken als verzichtbar an⁶⁴², wonach im Einzelfall darauf Rücksicht genommen werden muss, ob ein Haftungsausschluss rechtmäßig vereinbart werden kann. Die

⁶⁴¹ Bspw wenn die Software zum Zweck der Weiterentwicklung erworben wurde, der Kaufpreis sich daher nicht auf die Bereitstellung der Software in Objektform bezieht und der Erwerber ausreichende Kenntnisse zum Einsatz des verbesserten Quellcodes besitzt.

⁶⁴² OGH 03.12.1981, 8 Ob 530/81, JBI 1983, 255.

Haftungsausschlüsse der Open Source Lizenzen können ebenfalls nicht pauschal ausgeschlossen werden. Die Haftung nach dem PHG bezüglich Software wurde in der Lehre diskutiert und überwiegend bejaht. In Bezug auf Open Source Software ist vor allem auf den Bereich der Distributionen zu verweisen, da diese unweigerlich als Produkt iSd PHG angesehen werden müssen, da es sich um eine Zusammenstellung verschiedener Komponenten – Datenträger und Begleitmaterial – handelt, die zusammengefasst als Ganzes vertrieben werden. Die Ansprüche nach dem PHG müssen sich daher nicht ausschließlich aus dem Einsatz der Software ergeben, da eine falsche Einschulung bzw Dokumentation für einen Schaden verantwortlich sein kann.

An dieser Stelle auch auf die arbeitsrechtlichen Problemfelder (bspw Erstellung von Individualsoftware als Neuentwicklung oder Weiterentwicklung von Open Source Software als Auftragsarbeit) Haftung und Gewährleistung betreffend einzugehen, würde den Rahmen der Arbeit sprengen.

7. MARKENRECHT

Mit der zunehmenden wirtschaftlichen Bedeutung von Open Source Software ist auch das Markenrecht von immer größer werdender Relevanz, da die ständig wachsende Anzahl an Marken im Open Source Bereich - wie bereits erwähnt - eine dynamische Entwicklung zeigt. Häufig führen deren Entstehung zu Diskussionen innerhalb der Gemeinschaft von Nutzern freier und Open Source Software, da die Einräumung von Markennutzungsrechten meist direkt in der jeweiligen Open Source Lizenz geregelt ist und die Restriktion der Nutzung bestimmter Marken vielen zu weit geht. Um den Namen „Apache“ bspw zu schützen, bestimmt Punkt 5 des Apache-Lizenzvertrags, dass der Name „Apache“ für Weiterentwicklungen nur mit schriftlicher Genehmigung der ASF verwendet werden darf. Dieser markenrechtliche Aspekt ist wie auch *Jaeger/Metzger*⁶⁴³ anmerken nicht unproblematisch, da die Marke „Apache“ nicht entsprechend eingetragen wurde. Deshalb bleibt zunächst zu klären, welches Recht bei Markeneintragungen Anwendung findet und was durch das Markenrecht überhaupt geschützt wird, bevor dann auf die Auswirkung hinsichtlich von Marken im Open Source Softwarebereich eingegangen wird.

7.1. Die Rechtsquellen des Markenrechts

Die MarkenRL hat die markenrechtlichen Bestimmungen weitgehend vereinheitlicht und die nationalen Gesetze an die Vorgaben des Gemeinschaftsrechts angeglichen. In Österreich ist das Markenrecht wesentlich durch das Markenschutzgesetz 1970 (MSchG) geregelt, welches in den §§ 21, 28, 35, 39, 41, 42, 55, 60b, 68g, 68i und 72 MSchG auf Bestimmungen des PatG verweist. Das Markenrecht steht dabei subsidiär neben den anderen

⁶⁴³ *Jaeger/Metzger*, Open Source Software 56.

Schutzrechten (bspw jene des gewerblichen Rechtsschutzes). Weitere markenrechtlich relevante Bestimmungen befinden sich im ABGB (Namensschutz iSd § 43 ABGB), im UWG (firmen- und titelrechtlicher Schutz iSd § 9 UWG) und im UrhG (Titelschutz iSd § 80 UrhG). Einzelne Verfahrensangelegenheiten des Markenrechts sind auch in Verordnungen (bspw die Patent-, Gebrauchsmuster-, Marken- und Musterverordnung) zu finden.

Neben der Gesetzgebung haben der EuGH und das EuG wesentliche Rechtsgrundsätze in ihrer Rechtsprechung herausgearbeitet. Darüber hinaus bietet die Gemeinschaftsmarke ein eigenständiges, einheitliches Schutzrecht, welches neben den nationalen Schutzrechten existiert und im gesamten EU-Raum Geltung hat⁶⁴⁴. Die Relevanz des Gemeinschaftsschutzrechts wird nachfolgend kurz erläutert.

Eine Gemeinschaftsmarke ist ein in der gesamten Europäischen Gemeinschaft gültiges Zeichen zur Kennzeichnung und Unterscheidung von Waren und Dienstleistungen. Die Gemeinschaftsmarke wird beim Harmonisierungsamt für den Binnenmarkt (HABM) gemäß den in der Gemeinschaftsmarkenverordnung⁶⁴⁵ festgelegten Bedingungen eingetragen. Die Anmeldung einer Gemeinschaftsmarke und eine eingetragene Gemeinschaftsmarke sind in der Europäischen Gemeinschaft als Ganzes gültig. Sowohl die Anmeldung als auch die nachfolgende Eintragung erstrecken sich automatisch und gleichzeitig auf alle 27 Mitgliedstaaten der Europäischen Gemeinschaft. Es ist nicht möglich, die geographische Ausdehnung des Schutzes auf bestimmte Mitgliedstaaten zu beschränken. Es besteht außerdem ein einheitliches und vom HABM zentral verwaltetes Eintragungsverfahren. Anträge bei den nationalen Ämtern für gewerblichen Rechtsschutz sind nicht erforderlich. Nichtigkeit, Zurückweisung oder Ablauf einer Gemeinschaftsmarke

⁶⁴⁴ *Oliva in Kucsko*, Geistiges Eigentum 567.

⁶⁴⁵ <http://oami.eu.int/de/mark/aspects/reg/reg4094.htm>.

gelten zwangsläufig in der gesamten Europäischen Gemeinschaft. Des Weiteren stellt die Gemeinschaftsmarke ein alleiniges Eigentumsrecht dar. Übertragungen können nur für die gesamte Gemeinschaft und nicht für einzelne Länder erfolgen; die Vergabe von Lizenzen mit einer geographischen oder sonstigen Beschränkung, einschließlich der Beschränkung auf einen bestimmten Mitgliedstaat, ist jedoch möglich.

Das System der Gemeinschaftsmarke berührt nicht die nationalen Markensysteme der Mitgliedstaaten (bzw. das Benelux-Markenamt für Belgien, Luxemburg und die Niederlande). Den Unternehmen steht es frei, eine Marke entweder nur auf nationaler Ebene, auf Gemeinschaftsebene oder auf beiden Ebenen schützen zu lassen. Die in den Mitgliedstaaten eingetragenen zahlreichen nationalen Marken behalten weiterhin ihre Gültigkeit. Der Rückgriff auf den Schutz der Gemeinschaftsmarke, ausschließlich oder in Verbindung mit einem nationalen Schutzrecht, hängt ganz von der Strategie des Markenanmelders bzw. -inhabers ab. Der Vorrang eines älteren nationalen oder gemeinschaftlichen Schutzrechts gilt gleichermaßen für nationale und Gemeinschaftsmarken.

7.2. Was wird durch das Markenrecht geschützt?

Eine Marke dient dazu, Ware(n) oder Dienstleistung(en) eines Inhabers von gleichartigen eines Mitbewerbers abgrenzen und die Marke dadurch einer bestimmten Herkunft zuordnen zu können⁶⁴⁶. Sie erfüllt damit eine „Herkunftsfunktion“. Ein weiterer Effekt einer Marke sollte darin bestehen, mit der Marke eine bestimmte Qualität verbinden zu können, um dem Kunden dadurch Vertrauen in die Marke zu vermitteln. Eine Marke genießt daher eine Garantie- und Vertrauensfunktion, aus welcher zwar keine rechtliche

⁶⁴⁶ *Oliva in Kucsko*, Geistiges Eigentum 201 mwA.

Verpflichtung zur Einhaltung eines Qualitätsstandards folgt⁶⁴⁷, dennoch sieht der OGH⁶⁴⁸ in der Herkunftsfunktion der Marke eine Garantie der Produktidentität, wobei der Markeninhaber die von ihm verantwortete Ursprungsidentität kontrollieren muss. Dies bedeutet, dass die Gewährleistung der Herkunftsfunktion als vorrangiges Kriterium für die Einräumung eines Markenschutzes gesehen werden kann⁶⁴⁹. Gerade für Open Source Software ist die Herkunftsfunktion von herausragender Bedeutung. Nur durch die Eintragung der Marke kann eine Zuordnung einer bestimmten Software zu einem bestimmten Entwickler oder Entwicklungsteam vorgenommen werden. Dies zeigt sich vor allem im Bereich der Linuxdistributionen, denen durch die unterschiedlichen Markenbezeichnungen unterschiedliche Attribute zugeordnet werden können. Trotzdem gibt es viele private Adaptionen und Anpassungen, welche erst nach einer Überprüfung durch den Markeninhaber den Markennamen zur Weiterverbreitung annehmen dürfen.

7.3. Die Entstehung des Markenschutzes

Markenrechte entstehen nicht wie beim Urheberrecht bspw schon mit Schaffung des Werks bzw der Marke, sondern großteils erst durch die Eintragung der Marke in das Markenregister. Nachfolgend wird ein kurzer Einblick in die formalen und materiellen Anforderungen der Markenmeldung gegeben.

⁶⁴⁷ *Oliva in Kucsko*, Geistiges Eigentum 203.

⁶⁴⁸ OGH 04.02.1999, 4 Ob 305/98a – Red Puma – ÖBI 1999, 191.

⁶⁴⁹ Erwägungsgrund 10 der Ersten Richtlinie 89/104/EWG des Rates vom 21. Dezember 1988 zur Angleichung der Rechtsvorschriften der Mitgliedstaaten über die Marken, ABI 1989 L 40, 1; EuGH 18.06.2002, Rs C-299/99, *Philips vs Remington*, Slg. 2002, I-5475 = GRUR Int. 2002, 842.

7.3.1. Die Registrierung von Marken von Open Source Software

Wer in Österreich eine Marke anmelden will, muss die Anmeldung beim Patentamt in Wien schriftlich einbringen, da gem § 2 Abs 1 MSchG für den Erwerb eines Markenrechts das Eintragungsprinzip gilt. Das Patentamt führt ebenfalls gem § 16 Abs 1 MSchG das österreichische Markenregister. Markenabbildungen wie Zahlen-, Buchstaben oder Wortmarken ohne bildmäßige Ausgestaltung müssen nicht besonders dargestellt werden. Dies geht e contrario aus § 16 Abs 2 MSchG hervor, der bestimmt, dass bei Bildmarken eine Darstellung der Marke zu überreichen ist. Für den Open Source Bereich ist – neben der Registrierung von graphischen Zeichen – die Bezeichnung der Waren und Dienstleistungen wichtig, verdienen doch viele Firmen im Open Source Bereich durch den Support ihr Geld. Gem § 16 Abs 3 MSchG ist daher auf der Anmeldung anzugeben, für welche Waren und Dienstleistungen die Marke bestimmt ist⁶⁵⁰. Dabei muss der Antragsteller gem § 11 PAV Begriffe verwenden, die eine Beurteilung des Schutzzumfangs ermöglichen. Diese Begriffe sollen vorzugsweise aus der Liste der Waren und Dienstleistungen des Abkommens von Nizza⁶⁵¹ entnommen werden. In dieser Liste sind die verschiedenen Waren- und Dienstleistungsbegriffe in verschiedene Klassen unterteilt. Für den Open Source Bereich sind besonders die Klassen 35 und 42 von Bedeutung, welche die Erbringung von Dienstleistungen und den Entwurf und die Entwicklung von Computersoftware behandeln.

⁶⁵⁰ Schweiger in *Kucsko*, Geistiges Eigentum 358.

⁶⁵¹ <http://www.dpma.de/suche/klass/wd/abkommen.html>.

7.3.2. Die Priorität der Anmeldung

Gem § 23 Abs 1 MSchG wird mit dem Tag der Anmeldung der Vorrang gegenüber später angemeldeten Marken – die Priorität – erlangt⁶⁵². An dieser Stelle kann der internationale Aspekt für den Open Source Bereich wichtig werden. In Österreich ist es einem Markeninhaber unter bestimmten Voraussetzungen möglich, das nationale Recht auf Priorität zu beseitigen, indem er sich darauf beruft, dieselbe Marke in einem anderen Land bereits angemeldet zu haben und aufgrund dessen beansprucht, eine Priorität der ausländischen Anmeldung gegenüber der österreichischen Nachanmeldung zu haben⁶⁵³. Diese Möglichkeit ist in § 24 MSchG genauer geregelt. Nach der Prüfung des Antrags auf seine Gesetzmäßigkeit iSd §§ 20 u 21 MSchG durch das Patentamt wird die Marke, solange keine Registrierungshindernisse vorliegen, eingetragen.

7.3.3. Der Markeninhaber im Open Source Bereich

Der Markeninhaber muss rechtsfähig sein. Neben natürlichen Personen kommen bspw eine GmbH, AG, OG oder ein Verein⁶⁵⁴ in Frage. Im Gegensatz dazu ist eine GesbR iSd § 1175 ABGB mangels Rechtspersönlichkeit nicht fähig, als Markeninhaber aufzutreten⁶⁵⁵. In diesem Fall können die Gesellschafter die Marke als Einzelpersonen gemeinsam innehaben⁶⁵⁶. Diesbezüglich besteht zwar keine gesetzliche Grundlage, da eine Rechtseinräumung an Dritte bspw durch Übertragung oder Lizenzierung im Zweifel aber die Zustimmung aller Inhaber bedarf, erscheint es logisch, dass in

⁶⁵² Vgl *Schönherr/Thaler*, Entscheidungen zum Markenrecht (1985) E 1 zu § 23.

⁶⁵³ *Schweiger in Kucsko*, Geistiges Eigentum 380.

⁶⁵⁴ OGH 14.12.1993, 4 Ob 157/93 – TÜV I – ÖBI 1994 85.

⁶⁵⁵ *Schweiger in Kucsko*, Geistiges Eigentum 341.

⁶⁵⁶ *Hämmerle/Wünsch*, Handelsrecht systematisch dargestellt I⁴, 283.

einem solchen Fall immer von einem gesamthandschaftlichen Eigentum ausgegangen werden kann⁶⁵⁷. Die Auswirkungen für eine gemeinschaftliche Entwicklung von Open Source Software bestehen darin, dass Entwickler nur gemeinsam entscheiden können, in welcher Form sie die Markenrechte wahrnehmen wollen. Die Durchsetzung dieses gemeinschaftlichen Rechts kommt jedoch jedem Markeninhaber alleine zu. Diese analoge Behandlung zu § 9 MSchG und § 27 Abs 3 PatG erscheint sinnvoll, da nur so bei einer Markenverletzung schnell reagiert werden kann, ohne die Einwilligung aller Inhaber einholen zu müssen⁶⁵⁸.

Eine Aufteilung der Markenrechte in bestimmte den Entwicklern zugewiesene Anteile kann immer nur im Innenverhältnis Wirkung entfalten, die Eintragung ins Markenregister hat ohne eine solche Angabe zu erfolgen⁶⁵⁹. Das auch ausländische Anmelder in Österreich unbeschränkt und ohne Reziprozitätsgrundsatz Inhaber von Markenrechten werden können, entspricht dem grenzüberschreitenden Aspekt der Entwicklung von Open Source Software. Es bestehen diesbezüglich also keine Hindernisse, bei einer Einzel- oder gemeinschaftlichen Anmeldung. In diesem Fall ist es ausreichend, einen (oder mehrere) Vertreter zu bestellen, eine persönliche Anwesenheit der Markeninhaber bei der Anmeldung ist nicht nötig.

7.3.4. Die Grenzen der Registrierbarkeit von Open Source Software an Hand konkreter Beispiele

Solange gegen eine Marke keine absoluten oder relativen Registrierungshindernisse iSd § 4 MSchG bestehen, kann gem § 1 MSchG als

⁶⁵⁷ So auch: *Grünzweig*, Österreichisches, europäisches und internationales Markenrecht (2006) §§ 16, 17 Rz 6. Kritisch: *Gamerith*, ÖBl 1996, 63.

⁶⁵⁸ Vgl Kap 3.2.3.

⁶⁵⁹ *Grünzweig*, Markenrecht §§ 16, 17 Rz 6.

Marke gelten, was sich in Form von Zeichen darstellen lässt. Für den Open Source Bereich haben vor allem Wörter (Projektbezeichnung als klassische Wortmarke), Abbildungen (bspw der Pinguin „Tux“ als Bildmarke) und Buchstaben – rekursive Akronyme (vgl Fußnote 12) sind hier besonders erwähnenswert⁶⁶⁰ – eine wichtige Bedeutung. Zahlen, die Form der Ware wie auch das Warendesign spielen nur eine sehr untergeordnete Rolle, da sich die „Produkte“ meist im Internet befinden und ein Kunde überwiegend über die Kombination Wort-Bild-Marke (bspw „Deamon“ und Schrift „FreeBSD“) angesprochen wird. Da es keine taxative Aufzählung der Markenarten gibt, kann jede denkbare Form von Zeichen benutzt werden. Auch zukünftige, bislang unbekannte technische Darstellungsformen können daher von der aktuellen Rechtslage umfasst werden, solange sie den formalen und materiellen Anforderungen entsprechen.

Die graphische Darstellbarkeit betreffend erlaubt Art 15 Abs 1 des TRIPS-Abkommens den Mitgliedern, „*die visuelle Wahrnehmbarkeit als Eintragungsvoraussetzung*“ festlegen zu dürfen. Dieser Aspekt ist im Open Source Bereich hinsichtlich Geruchs- oder Hörzeichen von minderer Bedeutung, da diese mE dort keine Verwendung finden. Das Gemeinschaftsrecht schränkt den Schutz auf jene Kennzeichen, die sich graphisch darstellen lassen, im Bereich der eingetragenen Marken als zwingend ein⁶⁶¹. Die Bestimmungen des MSchG folgen diesen Vorgaben, was zwar nach *Dunkl* bei den Farbmarken zu Problemen führen kann, aber dies war für Open Source Software bislang ebenfalls ohne große Bedeutung.

⁶⁶⁰ Diese könnten auch ein Registrierungshindernis iSd § 4 MSchG darstellen, da diese Form der Produktbezeichnung ja ein beschreibendes Zeichen darstellt. Vgl OGH 13.04.1999, 4 Ob 17/99z, ÖBl 1999, 283.

⁶⁶¹ *Dunkl* in *Kucsko*, Geistiges Eigentum 262.

Nachfolgend Beispiele für Markendarstellungen im Open Source Bereich:



Abb 12

Wort-Bild-Marke

Linux



Abb 13

Bildmarke

7.3.4.1. Fehlende Unterscheidungskraft als Registrierungs- hindernis am Beispiel „kllustrator“ und „samba“

Die Unterscheidungsfunktion einer Marke ist eine Grundvoraussetzung des Markenrechtsschutzes und soll der Verwechslungsgefahr durch den Kunden vorbeugen. Erfüllt die Marke nicht diese Anforderung, ist sie gem § 4 Abs 1 Z 3 MSchG nicht eintragungsfähig. Die Marke muss daher „*etwas Besonderes, Individuelles an sich haben, das sich schon nach ihrer Art dazu eignet, ihren Träger von anderen Personen (bzw dessen Waren von fremden Waren) zu unterscheiden.*“⁶⁶² Die fehlende Unterscheidungsfunktion hat im Open Source Bereich bereits zu einem Gerichtsverfahren geführt, da Entwickler von Open Source Software oftmals nicht nur Aufbau und Idee eines proprietären Programms verwenden, um es Open Source-gemäß umzusetzen, man schaue sich bspw OpenOffice an, sondern auch die dafür gewählten Namen an das proprietäre Ursprungsprodukt anlehnen. Dieser Fall wird nachfolgend kurz geschildert und analysiert.

„kllustrator“ war eine Grafiksoftware für das K Desktop Enviroment (KDE). Aufgrund des Namens "kllustrator" mahnte die Softwarefirma Adobe in einer Mail das „kllustrator“-Entwicklerteam ab. Adobe sah sein Markenrecht auf den

⁶⁶² OGH 13.04.1999, 4 Ob 17/99z – fremde Waren – ÖBl 1999, 283.

Namen seines Produkts "Adobe Illustrator" aufgrund fehlender Unterscheidungskraft verletzt und forderte Schadensersatz in Höhe von EUR 2500⁶⁶³. Der Entwickler *Kai-Uwe Sattler* musste daraufhin den „kllustrator“ von seiner Webseite nehmen. Das Verfahren endete damit, dass sich „Adobe“ und das „kllustrator“-Entwicklerteam darauf einigten, den „kllustrator“ in „Kontour“ umzubenennen, wobei „Adobe“ die Kosten des Rechtsstreits vollständig übernahm⁶⁶⁴.

Für *Jaeger/Metzger*⁶⁶⁵ ist eine Markenrechtsverletzung in diesem Fall äußerst fraglich, da ihrer Meinung nach keine Verwechslungsgefahr des „kllustrator“ mit dem „Adobe Illustrator“ für die Allgemeinheit gegeben sei. Dies insbesondere aufgrund der Tatsache, dass der Name „Adobe“, der für die Kennzeichnungskraft für zusammengesetzte Begriffe unterscheidungskräftigere Teil sei.

Dieser Meinung ist insofern zuzustimmen, da Adobe zum damaligen Zeitpunkt keine eigene Grafiksoftware für KDE besaß und der „kllustrator“ wiederum auf keinem Betriebssystemen funktionsfähig war, auf denen Produkte von „Adobe“ verwendet werden konnten. Betrachtet man aber die vom OGH entwickelten Grundsätze zu einem solchen Fall, muss man zu dem Ergebnis kommen, dass der Name „kllustrator“ alleine gegenüber „Illustrator“ gestellt werden muss, da Adobe eine so vielfältige Produktpalette hat, dass im allgemeinen Sprachgebrauch der Vorsatz „Adobe“ überwiegend weggelassen wird. Der Begriff „kllustrator“ hat mE gegenüber der Bezeichnung „Illustrator“ wenig bis nichts Besonderes oder Individuelles, zumal das Programm seiner Funktion nach dem „Illustrator“ wesentlich angenähert ist. Auch an anderen für die Bestimmung der Unterscheidungskraft wichtigen und von der Judikatur herausgearbeiteten Grundsätzen scheitert der „kllustrator“ bei einer näheren

⁶⁶³ <http://lists.kde.org/?l=koffice-devel&m=99405898605976&w=2>.

⁶⁶⁴ <http://www.linux-community.de/Neues/story?storyid=1651>. Zum Kostenersatz siehe 7.2.2.

⁶⁶⁵ *Jaeger/Metzger*, Open Source Software 131.

Überprüfung. Die Marke „kllustrator“ ist nicht geeignet, ihre Herkunft so zu kennzeichnen, dass ein Kunde die Herkunft problemlos zuordnen kann⁶⁶⁶. Dieser Fall legt sogar die Vermutung nahe, Adobe hätte den „kllustrator“ selbst für KDE herausgegeben. Auch bei der Argumentation, dass in diesem speziellen Fall den Benutzern von „kllustrator“ gute Branchenkenntnisse unterstellt werden könnten⁶⁶⁷, ist die Herkunft schwierig zuzuordnen⁶⁶⁸ und eine Irreführung iSd § 33c MSchG anzunehmen. Da es durch die zunehmende Popularität von Open Source Software zu einer großen Anzahl an potentiellen Anwendern gekommen ist, darf der Bereich der Open Source Anwendungen, welche im beruflichen Alltag verwendet werden, nicht mehr auf einen bestimmten Verkehrskreis eingeschränkt werden.⁶⁶⁹ Die Unterscheidungsfunktion der Marken untereinander ist in einem solchen Fall also nicht gegeben.

Die Unterscheidungskraft betreffend ist des Weiteren auszuführen, dass dem OGH nach *„grundsätzlich nur frei erfundene, keiner Sprache angehörende Phantasiewörter (ieS) oder solche Wörter, die zwar dem allgemeinen Sprachgebrauch angehören, jedoch mit der Ware (Dienstleistung), für die sie bestimmt sind, in keinem Zusammenhang stehen (Phantasiewörter iwS) Unterscheidungskraft besitzen. Dabei ist entscheidend, ob die Wörter im Verkehr als Phantasiebezeichnungen aufgefasst werden. Beschreibende Angaben – insbesondere solche über Beschaffenheit und Bestimmung der Ware – sind in aller Regel nicht unterscheidungskräftig, können den Schutz nach § 9 Abs 3 UWG aber bei entsprechender Verkehrsgeltung erlangen.“*⁶⁷⁰

⁶⁶⁶ EuGH 18.06.2002, Rs C-299/99 (*Philips vs Remington*), Slg. 2002, I-5475 = GRUR Int. 2002, 842 = ÖBI 2003, 55 (*Gamerith*) = GRUR 2002, 804.

⁶⁶⁷ Siehe dazu: EuG 27.02.2002, Rs T-34/00 (*EUROCOOL*), Slg 2002, II-683 = wbl 2002, 218.

⁶⁶⁸ EuG 07.02.2002, Rs T-88/00 (*Stabtaschenlampen*), Slg 2002, II-467 = GRUR Int 2002, 288 und 531.

⁶⁶⁹ Ausgenommen davon sollten solche Anwendungen bleiben, welche nur an einen kleinen Kreis von Anwendern gerichtet sind, wie bspw Entwicklertools.

⁶⁷⁰ OGH 13.04.1999, 4 Ob 17/99z – fremde Waren – ÖBI 1999, 283.

Ein solches Beispiel ist „samba“. Im Fall „samba“⁶⁷¹ kam es zur Markenrechtsverletzung, da die Beratungsgesellschaft CMG diesen Begriff bereits eingetragen hatte. Diese mahnte daraufhin die Anbieter der Software ab. Der Streit um die Nutzung wurde beigelegt⁶⁷², indem CMG der Service Network GmbH als kommerziellem Anbieter von „samba“ stellvertretend für die Open Source Gemeinde eine Nutzungslizenz für den symbolischen Betrag von damals 1,- DM übertrug. Alle, die nun die Open Source Software Samba vertreiben oder dafür Support anbieten, können gegen eine Beteiligung am Abwicklungsaufwand von 10,- Euro eine Unterlizenz erwerben.

„Samba“ ist ohne Zweifel ein Wort, das dem allgemeinen Sprachgebrauch angehört (Tanz, musikalische Stilrichtung). Dennoch verwenden verschiedene Firmen den Begriff „samba“ für Ihre Produkte. Diese stehen – wie vom OGH gefordert – in keinem Zusammenhang mit der alltagsgebräuchlichen Verwendung. Die Luxusversion eines VW-Busses wird im Alltagsgebrauch „Samba“ genannt, ebenso gab es in den 1980er Jahren einen Fußball der Marke Adidas, der Samba hieß. Im Softwarebereich findet man gleich drei verschiedene Programme mit dem Namen Samba. So hieß der erste Browser des Apple Macintosh „samba“. Die deutschen Banken benutzen eine Software „samba“ im Bereich des Meldewesens und letztendlich ist es die bereits beschriebene beliebte Server-Software aus dem Open Source Bereich. Im Softwarebereich kann man von einer ausreichenden Unterscheidungskraft zwischen den verschiedenen „sambas“ ausgehen. Den Browser „samba“ gibt es nicht mehr, die anderen beiden sind spezielle Anwendungen, welche vom Verkehrskreis klar unterschieden werden können. Darüber hinaus hat die Open Source Software „samba“ eine einprägsame Wort-Bildmarke, welche ein deutliches Unterscheidungsmerkmal darstellt. Dieses Beispiel zeigt, dass das Registrierungshindernis der fehlenden Unterscheidungskraft nicht allein deshalb eintritt, weil zwei Produkte, welche im gleichen Bereich angesiedelt sind, eine

⁶⁷¹ Samba ist ein Open Source Programmpaket, mit dem jeder Unixrechner Datei- und Druckdienste für SMB und CIFS Clients zur Verfügung stellen kann.

⁶⁷² <http://samba.sernet.de/old.news.html>, sowie <http://samba.sernet.de/lizenz/presse.html>.

Namensgleichheit oder zumindest Ähnlichkeit haben. Es sollte aber bei der Namensgebung auf ein prägnantes Unterscheidungsmerkmal geachtet werden, um das Registrierungshindernis der fehlenden Unterscheidungskraft abzuwenden.

7.3.4.2. Beschreibende Zeichen als Registrierungshindernis im Open Source Bereich am Beispiel „*Konsole*“

Das relative Registrierungshindernis gem § 4 Abs 1 Z 4 MSchG für Zeichen, die ausschließlich aus beschreibenden Zeichen bestehen, ist für Open Source Software nur ein bedingtes Problem. Viele Open Source Projekte suchen sich bewusst eigenständige Namen, um ihr Projekt von anderen unterscheiden zu können. Jene Namen, welche aus einem rekursiven Akronym bestehen, wie bspw „GNU“, „Zope“ oder „WINE“⁶⁷³ beschreiben ihre Herkunft (GNU = **G**nu is **N**ot **U**nix, geht aber aus diesem hervor), ihr Anwendungsgebiet (Zope = **Z**ope **O**bject **P**ublishing **E**nvironment, also eine Anwendung für die Desktopumgebung) oder ihre Intention (WINE = **W**ine **I**s **N**ot an **E**mulator, das Programm ist nicht als Emulator zu verstehen, da es eine Windows-kompatible Laufzeitumgebung darstellt). Diese Fälle sind in der rechtlichen Beurteilung jedoch nicht als kritisch einzustufen.

Problematisch wird es im Softwarebereich, wenn der Markenname eine allgemeingebrauchliche Anwendung beschreibt. Dies soll am Bsp der Terminalemulationssoftware „Konsole“ stellvertretend demonstriert werden.

„Konsole“⁶⁷⁴ ist eine Terminalemulationssoftware, welche auf Unix-artigen Betriebssystemen läuft. Der Begriff „Konsole“⁶⁷⁵ wird im IT-Bereich dazu

⁶⁷³ Siehe dazu auch Kap 2.2.

⁶⁷⁴ <http://konsole.kde.org>.

⁶⁷⁵ Gleichbedeutend mit: „Kommandozeile“, „Command Line“, „Terminal“ oder „Shell“.

benutzt, den Eingabebereich für die Steuerung eines Programms, insbesondere eines Betriebssystems, der normalerweise (aber nicht zwingend) im Textmodus abläuft, zu beschreiben. Das Computerprogramm „Konsole“ kann demzufolge als „*Bezeichnung der Art und sonstiger Merkmale*“ iSd § 4 Abs 1 Z 4 MSchG angesehen werden, da die wesentliche Eigenschaft des Programms die Merkmale umfasst, die unter den Konsolenbegriff allgemein fallen.

Man wird in diesem Fall einen Verkehrsgeltungsnachweis iSd § 4 Abs 2 MSchG nicht beweisen und daher die geforderte Unterscheidungskraft im Inland⁶⁷⁶ als nicht erbracht ablehnen können. Vielmehr wird man das Freihaltebedürfnis zu Gunsten der IT-Anwender auslegen müssen, wonach es untersagt ist, dass eine beschreibende Marke einem einzigen Unternehmen vorbehalten ist. Ebenso wesentlich ist, dass der Begriff „Konsole“ im normalen Sprachgebrauch nach dem „Verständnis der angesprochenen Verkehrskreise“ die wesentlichen Merkmale der Software „Konsole“ umfasst⁶⁷⁷.

Das Beispiel „Konsole“ zeigt neben der angeführten Übereinstimmung zwischen Begriff und beschreibender Marke auch noch, dass hier eine besondere Bewertung vorgenommen werden muss, da die Wahl des Programmnamens von den Entwicklern nicht ohne Phantasie vorgenommen wurde und man dies im Open Source Bereich häufiger findet⁶⁷⁸.

Zum einen bleibt zu bemerken, dass die Wahl des Buchstabens „K“ am Anfang des Wortes auf den Einsatz des Programms in einer KDE-Umgebung verweist⁶⁷⁹ und daher nicht ausschließlich für die Konsolenprogramm-

⁶⁷⁶ Siehe dazu bspw: OGH 12.07.2006, 4 Ob 38/06a – Shopping City – wbl 2006, 253.

⁶⁷⁷ Siehe dazu bspw: EuG 20.03.2002, Rs T-355/00 (*TELE AID*), Slg 2002, II-1939= wbl 2002, 271 = GRUR Int 2002, 747.

⁶⁷⁸ Siehe nur „*Sourceforge*“ und die dort aufgelisteten Projektnamen.

⁶⁷⁹ Dies ist für den relevanten Verbraucherkreis in solchen Fällen leicht erkennbar, da die Verbindung des Buchstabens „K“ in Bezug auf Software, welche unter KDE lauffähig ist, häufig benutzt wird, siehe bspw „*kubuntu*“ für die Linuxdistribution „*ubuntu*“ (welche eigentlich „*Gnome*“ benutzt) oder das in Kap 7.3.4.1 benutzte Bsp „*killustrator*“.

eigenschaft beschreibend ist und zum anderen handelt es sich in diesem Fall um einen Projektnamen, der bislang nur als reine Wortmarke genutzt wird und für welchen noch kein Markenschutz beantragt wurde.

Abschließend bleibt festzustellen, dass in einem solchen Fall die Eintragung der Wortmarke „Konsole“ am Hindernis des § 4 Abs 1 Z 4 MSchG scheitern muss, da es sich – auch in Betrachtnahme des Verweises auf die KDE-Oberfläche – um ein beschreibendes Zeichen handelt und die Nutzung des Anfangsbuchstabens „K“ nicht nur als eine „*bloße Andeutung einer bestimmten Beschaffenheit*“⁶⁸⁰ zu bewerten ist. Während der Versuch, die Wortmarke „*Konsole*“ anzumelden keinen Erfolg haben dürfte, könnte dies im Falle einer Wort-Bildmarke „*Konsole*“ anders bewertet werden. Sollte bspw das „K“ eine optisch herausragende Position einnehmen, so kann dies eine Unterscheidungskraft bewirken, welche die Marke eintragungsfähig machen würde. Zu beachten ist dabei allerdings, dass im Fall einer kombinierten Marke jene Bestandteile nicht registrierbar sind, die für sich allein keine Kennzeichnungskraft besitzen⁶⁸¹. In diesem Fall müsste die Verkehrsgeltung des Programms durch ein Kammergutachten, einen Sachverständigenbeweis oder allenfalls ein demoskopisches Gutachten beurteilt werden, um eine umfassende Bewertung für die Eintragung als Wort-Bildmarke vornehmen zu können. Open Source Software Entwickler könnten durch die Kombination von Wort und Bild eine Unterscheidungskraft kreieren, welche eventuell nicht von einem Anfangs gewählten (Arbeits-)Titel erbracht werden kann, was am vorher erörterten Bsp „*samba*“ zu sehen war.

⁶⁸⁰ Siehe dazu bspw: OGH 20. 03. 2007, 17 Ob 3/07a – *Immoeast* - JusGuide 2007/21/4736; OGH 18.02.2003, 4 Ob 10/03d – *More II* – ÖBI-LS 2003/60.

⁶⁸¹ Siehe dazu bspw: OGH 11.07.1995, 4 Ob 59/95 – *New Yorker* – ÖBI 1996, 141 = *ecolex* 1995, 817 = *wbl* 1995, 511; so auch; *Oliva* in *Kucsko*, Geistiges Eigentum 310.

7.3.4.3. Gattungsbezeichnungen als Registrierungshindernis im Open Source Bereich am Beispiel des Pinguins „Tux“⁶⁸²

„The Problem of Success: when a Trademark becomes a Generic Term“⁶⁸³.

Aus den **„Markenbusiness News“**.

Das Registrierungshindernis der Gattungsbezeichnung gem § 4 Abs 1 Z 5 MSchG schließt jene Zeichen von der Registrierung aus, die *„ausschließlich aus Zeichen oder Angaben bestehen, die im allgemeinen Sprachgebrauch oder in den redlichen und ständigen Verkehrsgepflogenheiten zur Bezeichnung der Ware oder Dienstleistung üblich sind.“* Wie man aus dem Wortlaut erkennen kann, ist eine klare Abgrenzung zum Registrierungshindernis der beschreibenden Zeichen nicht möglich, da es zwangsläufig zu Überschneidungen kommen muss⁶⁸⁴. Im Open Source Bereich sind Gattungsbezeichnungen als Projektname ungewöhnlich, da Bezeichnungen wie bspw „Computer“, „Platine“ oder „Monitor“ keine Anziehungskraft für Open Source Projekte besitzen. Eine tierische Gattung hat es dennoch zu Ehren im Open Source Bereich gebracht: „Tux“ ist der Name für einen Pinguin, der 1996 von dem Graphiker *Larry Ewing* auf Wunsch von Linus Torvalds gezeichnet wurde⁶⁸⁵. Der Pinguin ist als Maskottchen der gesamten GNU/Linux-Gemeinschaft deshalb so berühmt, weil ein Großteil des Merchandisings im Linux-Bereich mit Tux gemacht wird.

Abb 14



Links die Abbildung des offiziellen Tux [dt.: [tuxs](#); engl.: [tuxs](#)]. Der Gattungsbegriff des Pinguins assoziiert heute schon stark mit Linux und deshalb auch allgemein mit Open Source Software. Ein Spruch dazu lautet: *„Da Pinguine nicht fliegen können, können sie auch nicht abstürzen“*.

⁶⁸² <http://www.isc.tamu.edu/~lewing/linux/>.

⁶⁸³ <http://www.markenbusiness.com/en/news.php?newsid=1685>.

⁶⁸⁴ Siehe bspw: EuGH 04.10.2001, Rs C-517/99 (*Bravo*), Slg 2001, I-6959 = ÖBl 2002, 105 = ecolex 2002, 35 (*Schanda*) = MarkenR 2001, 403 = GRUR Int 2002, 145.

⁶⁸⁵ Der Name zeigt einerseits wieder die starke Betonung des „X“ bei Softwareprodukten die auf Unix basieren und kann als Abkürzung für „Torvald’s Unix“ gesehen werden. Weitere Informationen auch unter: <http://www.sjbaker.org/tux/>.

Die markenrechtliche Frage, die sich aus der Darstellung der Bildmarke „Tux“ ergibt, stellt sich vorrangig hinsichtlich des Schutzes des Pinguins als solcher für Linux, der Wortmarke „Tux“ als Markenbezeichnung und der Wort-Bildmarke „Tux“ (Pinguin + Name).

Die Antworten darauf sind rechtlich gesehen von unterschiedlichem Gewicht. Die versuchte Anmeldung der Wortmarke „Tux“ beim europäischen Harmonisierungsamt für den Binnenmarkt führte damals zu wüsten Beschimpfungen der Antrag stellenden Schweizer Firma⁶⁸⁶. Die ablehnende Entscheidung des Amts noch in der Anmeldephase im Oktober 2005 beantwortete jedoch die Frage des Markenschutzes für „Tux“. Auch wenn die genauen Gründe für die Ablehnung aus den zur Verfügung gestellten Dokumenten des europäischen Harmonisierungsamts für den Binnenmarkt nicht ersichtlich sind, spricht einiges dafür, dass die Wortmarke „Tux“ schon 2004 eine solche Verkehrsgeltung erreicht hatte, dass sie auch ohne Anmeldung markenrechtlichen Schutz genoss.

Der Schutz der (Wort-)Bildmarke „Tux“ muss ebenfalls als gegeben erachtet werden. Zwar kann das Bildnis eines Pinguins allgemein nicht mit Linux im Besonderen oder Open Source Produkten im Allgemeinen assoziiert werden, dennoch gilt dies für das besondere Bildnis, welches vorher dargestellt wurde. Das comichafte Bild des „wohlgenährten“ und „glücklichen“ Pinguins erbringt klar den Verkehrsgeltungsnachweis – die Verwendung des Tux-Bildes als Bezeichnung für GNU/Linux und für Dienstleistungen rund um GNU/Linux ist absolut üblich und erlangt somit Verkehrsgeltung iSd § 9 UWG – und genießt daher auch ohne Eintragung Markenschutz. Ob die verschiedenen auf dem Bild des „Tux“ basierenden Abwandlungen, die mannigfaltig im Internet zu finden sind und bei denen meist klar zu erkennen ist, dass sie ein „Derivat“ von diesem

⁶⁸⁶ Die schweizer Firma „tux LLC“ hatte 2004 eine Anmeldung unter der Nr 005323399 vorgenommen.

darstellen, ebenfalls diesen Schutz genießen oder sich als eigenständige Marke eintragen lassen müssen, ist umstritten.

Nach *Jaeger/Metzger*⁶⁸⁷ ist dies dann möglich, wenn die neue Marke die erforderliche Unterscheidungskraft besitzt und der „*Verkehr die Marke aufgrund ihrer Abwandlung als ein produktidentifizierendes Unterscheidungszeichen auffasst.*“ In Österreich muss man zur Beantwortung dieser Frage § 9 Abs 3 UWG heranziehen. Dort heißt es: „*Der besonderen Bezeichnung eines Unternehmens stehen Geschäftsabzeichen und sonstige zur Unterscheidung des Unternehmens von anderen Unternehmen bestimmte Einrichtungen, insbesondere auch Ausstattungen von Waren, ihrer Verpackung oder Umhüllung und von Geschäftspapieren, gleich, die innerhalb beteiligter Verkehrskreise als Kennzeichen des Unternehmens gelten.*“ Auch hier ist es deshalb möglich, eine Marke einzutragen, welche der Gattungsbezeichnung „Tux“ nur ähnlich sieht, die aber die im MSchG geforderte Unterscheidungskraft besitzt. So kann bspw ein Pinguin als Bildmarke in einer anderen Klasse als zumindest 9 und 42 eingetragen werden. Für die Klassen 9 und 42 besteht mE eine herausragende Verkehrsgeltung des „Tux“, da dieser – wie erwähnt – nicht nur für Linux insbesondere, sondern mittlerweile für den gesamten Open Source Bereich stellvertretend steht und daher kein Produkt in diesem Bereich die erforderliche „produktidentifizierende“ Unterscheidung erreichen könnte. Das Registrierungshindernis entfällt schon allein wegen des „Entstehens einer Verkehrsauffassung zugunsten der Individualzeicheneigenschaft der Bezeichnung“⁶⁸⁸. Die Derivate des Bildes sind daher dem „Original-Tux“ untergeordnet und unterliegen der ausdrücklichen Erlaubnis bzw Duldung durch den Inhaber des „Original-Tux“. Die Urheberrechte und die damit einhergehenden Markenrechte liegen bei *Larry Ewing* und werden von diesem frei lizenziert⁶⁸⁹.

⁶⁸⁷ *Jaeger/Metzger*, Open Source Software², 203 f.

⁶⁸⁸ *Kucsko*, Geistiges Eigentum 328.

⁶⁸⁹ Siehe: <http://www.isc.tamu.edu/~lewing/linux/>. Bislang wurde auch keine Bildmarke eingetragen, die Tux als Bildzeichen verwendetet, siehe auch: *False*, Markenrecht und Open Source – Sinn oder Unsinn? www.ifross.de/Fremdartikel/VortragLinuxTag2003.pdf.

7.4. Die Lizenzierbarkeit von Open Source Software-Marken

In verschiedenen Open Source Lizenzen tauchen eigene Klauseln auf, die den Umgang mit Markenrechten regeln. Häufig langt daher ein Blick in die rechtlichen Hinweise, bzw in die Lizenzbestimmungen oder FAQs bei der Nutzung von Namen und Logos von Open Source Projekten bzw Software, um Aufschluss darüber zu bekommen, in welcher Form diese verwendet werden dürfen. So findet man bspw für die Verwendung der Software „Zope“ der Firma Zope Corporation in Punkt 4 der Zope Public License (ZPL) Version 2.0⁶⁹⁰ markenrechtliche Bestimmungen. Die Nutzung von „Servicemarks (sm)“ oder „Trademarks (tm)“⁶⁹¹ der Zope Cooperation sind demnach in einem besonderen Vertrag geregelt⁶⁹². Zum Teil finden sich in den Lizenzen ausdrückliche Verbote bezüglich der Benutzung von Marken bei der Weitergabe der Software. So enthält die Apple Public Source License (APSL) in § 10 einen ausdrücklichen Vorbehalt für alle von Apple gehaltenen Marken. Wer also Software entsprechend den Vorschriften der APSL verbreitet, darf hierbei die Marken „Apple“ und „QuickTime“ nicht benutzen. Diese Regelungen wurden notwendig, um Missbrauch vorzubeugen, da bei steigendem Erfolg eines Open Source Projekts oftmals „Markengrabbing“ stattfand⁶⁹³. Inzwischen haben viele Open Source Firmen, seien es Verlage, Entwickler, Distributoren oder Dienstleistungsunternehmen die wirtschaftliche Wichtigkeit von Marken entdeckt, sodass bspw der Verlag *O’Reilly* seine Bücher im Open Source Bereich mit schwarz-weiß Szenen aus dem Wilden Westen versieht, die *Apache Foundation* eine Indianerfeder im Logo führt und der „*Deamon*“ des FreeBSD-Logos sogar einem Redesign-Wettbewerb⁶⁹⁴ ausgesetzt war, da sich

⁶⁹⁰ <http://www.zope.com/license/>.

⁶⁹¹ Diese Bestimmungen gelten vorrangig für den amerikanischen Raum.

⁶⁹² Siehe <http://www.zope.com/Marks>. Die Einräumung der Nutzung der Markenrechte soll kostenfrei sein, die Lizenzstrategie befindet sich aber zum jetzigen Zeitpunkt noch in der Testphase.

⁶⁹³ *Jaeger/Metzger*, Open Source Software 130.

⁶⁹⁴ <http://logo-contest.freebsd.org/announce.txt>.

viele Amerikaner von dem Bild des Teufelchens gestört fühlten. Wie sieht aber die rechtliche Einordnung dieser Lizenzbestimmungen aus?

7.4.1. Die Einordnung der Lizenzbestimmungen unter § 14 MSchG

§ 14 MSchG regelt den Markenlizenzvertrag und besagt, dass eine Marke für alle oder nur einen bestimmten Teil der Open Source Software, für die sie eingetragen ist, Gegenstand sein kann. Dabei kann der Lizenzgeber eine territoriale Begrenzung vornehmen, indem er die Markenlizenz für das ganze Bundesgebiet oder nur einem Teil davon erteilt. Hinsichtlich der mittels der Markenlizenz übertragenen Rechte kann der Lizenzgeber dem Lizenznehmer dabei eine ausschließliche oder nicht ausschließliche Lizenz einräumen. Diese Regelung entspricht den markenrechtlichen Bestimmungen und den Intentionen der Lizenzgeber in den betroffenen Open Source Lizenzen, weshalb diese als Markenlizenzvertrag zu qualifizieren sind. Die Markenklauseln räumen dem Lizenznehmer dezidierte Rechte ein und regeln die Rechtseinräumung, welche meist nicht ausschließlich erteilt wird⁶⁹⁵. Beim Abschluss des Open Source Lizenzvertrags schließt der Lizenznehmer in der Regel einen Markenlizenzvertrag iSd § 14 MSchG mit ab. Die häufige Unentgeltlichkeit der Lizenzierung könnte zwar den Anschein einer einseitig empfangsbedürftigen Willenserklärung durch den Markeninhaber erwecken⁶⁹⁶, mE gehen die Bestimmungen in den Open Source Lizenzen aber über einen reinen Verzicht gegenüber dem Lizenznehmer auf die Nichtgeltendmachung von Ansprüchen, die sich aus dem Markenrecht ergeben, hinaus. Die Unentgeltlichkeit kann in diesem Fall nicht als Indiz für Einseitigkeit dienen, was sich aus der Eigenständigkeit der Open Source Software ergibt.

⁶⁹⁵ Vgl bspw § 10 APSL.

⁶⁹⁶ Grünzweig, Markenrecht §§ 14 Rz 2.

7.4.1.1. Die Einschränkung des Lizenzgebietes für Open Source Software-Marken

Die in § 14 MSchG mögliche territoriale Einschränkung bestimmt sich ausschließlich nach dem Inhalt des Markenlizenzvertragstextes. Schränkt dieser das Schutzgebiet nicht ein oder macht er überhaupt keine Angaben darüber, so muss davon ausgegangen werden, dass die Marke im gesamten Bundesgebiet gebraucht werden darf. Open Source Lizenzgebern ist es also möglich, so den Gebrauch ihrer Open Source Software-Marke einzuschränken. Auch wenn bislang ein solcher Fall bei einer Open Source Lizenz nicht bekannt ist, könnte das doch weitreichende Konsequenzen haben, worauf in Kap 7.1.2. näher eingegangen wird.

7.4.1.2. Die sachliche Beschränkung der Lizenz für Open Source Software-Marken

Die sachliche Beschränkung bezieht sich auf den Umfang der Rechtseinräumung hinsichtlich der Open Source Software selbst. Der Lizenzgeber kann daher Teile seiner Software vom Umfang der Rechtseinräumung ausnehmen. So kann er bspw einzelne Applikationen herausnehmen oder nur bestimmte Versionen der Software unter den Markenschutz stellen. Hier tritt erneut ein Spannungsverhältnis zu den Bestimmungen der OSD auf, welches in Kap 7.1.2. besprochen wird.

7.4.1.3. Die Rechteeinräumung für Open Source Software-Marken am Beispiel von „Linux“ und dem „Linux Sublicense Agreement“

Ähnlich der urheberrechtlichen Möglichkeit der Einräumung einer Werknutzungsbewilligung oder eines Werknutzungsrechts, hat auch im Markenrecht der Markenzugeber die Wahl zwischen der Einräumung einer ausschließlichen oder nicht ausschließlichen Markenzugebung. Während die Einräumung einer ausschließlichen Lizenz dem Lizenznehmer ein exklusives Recht – ähnlich dem Werknutzungsrecht – verleiht, welches den Markeninhaber selbst von der Markenverwertung ausschließt, erlaubt das nicht ausschließliche Recht dem Lizenznehmer die Nutzung der Lizenz gemäß des Markenzugebungsvertrags und behält dem Lizenzgeber eine eigene Verwertung vor. Wie bereits erwähnt kommt es in Open Source Software Lizenzen zur Einräumung einer nicht ausschließlichen Markenzugebung. Dies erscheint verständlich, da die Weitergabe des Markenrechts als „quasi absolutes“⁶⁹⁷ Recht, nur in den Fällen sinnvoll erscheint, in denen für den Markeninhaber sichergestellt wird, dass seine Marke nicht missbräuchlich verwendet wird. Die unterschiedliche Rechteeinräumung hat natürlich Einfluss auf die Aktivlegitimation im Falle einer Verletzung des Markenrechts⁶⁹⁸. Nachfolgend wird die Rechteeinräumung am Beispiel von Linux erörtert.

Die ersten Distributoren verwendeten Linux, ohne die Marke „Linux“ jemals angemeldet zu haben. Im Jahre 1995 kam es dann in den USA zu einem Rechtsstreit⁶⁹⁹, bei dem *William R Della Croce Jr.*, der die Marke „Linux“ im gleichen Jahr auf seinen Namen angemeldet hatte, nun von jedem, der „Linux“ in seinem Produktnamen führte, zunächst 5% und später 10% der Einkünfte

⁶⁹⁷ Grünzweig, Markenrecht §§ 14 Rz 8.

⁶⁹⁸ Siehe dazu Kap 7.3.1.

⁶⁹⁹ <http://www.linux10.org/history/>.

einforderte⁷⁰⁰. Die Empörung der Linux-Anwender und -Nutzer war verständlich. Dieser Rechtsstreit⁷⁰¹ endete dank des Zusammenhalts der Linux-Gemeinschaft schließlich damit, dass die Marke an Linus Torvalds in einer außergerichtlichen Einigung übertragen wurde⁷⁰². Auch in Deutschland kam es zu einer solchen missbräuchlichen Eintragung, die mit dem gleichen Ergebnis endete⁷⁰³. „Linux“ ist als Gemeinschaftsmarke auf Linus Torvalds in den Klassen 9, 16 und 42 Nizzaer Klassifikation angemeldet und gilt somit für das Gebiet der Mitgliedstaaten.

Die Verwendung der Wortmarke ist von der Einwilligung von Linus Torvalds abhängig, der die Markenrechte für die Wortmarke „Linux“ innehat und der dem Linux Mark Institute (LMI)⁷⁰⁴ die exklusiven Rechte eingeräumt hat, die Markenrechte zu unterlizenzieren. Das LMI hat ein so genanntes „Linux Sublicense Agreement“⁷⁰⁵ (LSA) herausgegeben, welches aktuell in der Version 2.0 vorliegt und die kostenlose⁷⁰⁶ Vergabe der Markenrechte an „Linux“ weltweit regelt. Das LSA regelt die rechtlichen Befugnisse äußerst umfangreich, weshalb an dieser Stelle kurz darauf⁷⁰⁷ eingegangen wird. – Der Lizenztext beginnt mit einer Präambel, die über die oben genannten Rechte von Linus Torvalds und des LMI informiert.

⁷⁰⁰ <http://www2.linuxjournal.com/article/2098>.

⁷⁰¹ <http://linuxgazette.net/issue13/tmark.html>.

⁷⁰² <http://www2.linuxjournal.com/article/2559>.

⁷⁰³ Jaeger/Metzger, Open Source Software 130.

⁷⁰⁴ www.linuxmark.org/.

⁷⁰⁵ http://www.linuxmark.org/linux_sublicense.php.

⁷⁰⁶ Bis 2007 konnten Kosten zwischen \$ 200 und \$ 5000, je nach Nutzungsart entstehen. Die Markenlizenzkosten wurden aber 2007 vom LMI abgeschafft.

⁷⁰⁷ Der Abschluss eines LSA mit dem LMI muss nur in den Fällen geschehen, in denen die Marke „Linux“ kommerziell und mit nachhaltiger Gewinnabsicht genutzt wird. Ausrichter von Open Source Veranstaltungen, welche bspw Kugelschreiber oder Ähnliches mit der Aufschrift „Linux“ versehen, fallen unter die „*fair-use-policy*“ des LMI und brauchen keinen Markenlizenzvertrag abzuschließen.

Dort werden zwei Ansprüche des LMI besonders hervorgehoben:

- Der Schutz der Markenrechte für den kommerziellen Gebrauch, ausgenommen die Fälle, die unter die „fair-use-policy“ fallen und
- den Schutz der Markenrechte im Sinne der „Linux community“ auszuüben.

Der zweite Punkt zeigt, dass die freie Nutzung der Marke propagiert wird, solange dies im Rahmen – sprich im Sinne der Linux-Gemeinschaft geschieht – und nicht von einzelnen versucht wird, diese parasitär auszubeuten. Der Rahmen, in welchem die Nutzung eingeräumt wird, folgt im dritten und letzten Abschnitt des LSA. Danach sind Linux-basierte Produkte oder Dienstleistungen:

- Computersysteme (samt Hardware) und Programme, welche Linux ausführen oder die Programme ausführen, welche auf dem Linux-Kernel⁷⁰⁸ basieren und
- jene Dienstleistungen, die Linux-basierte Produkte verwenden, dokumentieren oder beim Einsatz unterstützen.

Der Rahmen ist damit weitgefasst und betrifft nicht nur das Betriebssystem selbst, sondern bspw auch Handbücher, Support-Anbieter oder Entwickler. Die Grenze muss bei jenen Gütern und Dienstleistungen gezogen werden, bei der eine Interessenabwägung zwischen Linux-basierten Gütern und dem Interesse der Linux-Gemeinschaft zu Ungunsten der Linux-basierten Güter ausfällt. Der Open Source Gedanke spielt bei dieser Interessenabwägung eine zentrale Rolle. Dies bedeutet zwangsläufig, dass eine einschneidende Einschränkung des Open Source Gedankens – wie bspw das Vertreiben von proprietärer Software unter der Marke Linux – nicht möglich und daher gegen die Lizenzbestimmungen des LSA verstößt.

⁷⁰⁸ Den Linux-Kernel findet man unter: <http://www.kernel.org>.

Werden die grundlegenden Anforderungen der Präambel eingehalten, räumt § 1.1 LSA dem Markenlizenznehmer ein weltweites, nicht ausschließliches und nicht übertragbares Recht ein, die Marke und dem damit verbundenen Goodwill⁷⁰⁹ zu nutzen. Ausdrücklich untersagt wird durch § 1.3 LSA die Möglichkeit, die Marke als Lizenznehmer eigenständig bei einer Markenregistrierungsstelle anzumelden. Darüber hinaus weist das LSI in § 1.4 LSA jede Miteinbeziehung des LSI oder von *Linus Torvalds* bei einer Markenrechtstreitigkeit von sich, bei denen Dritte gegeneinander um die Rechtmäßigkeit der jeweiligen Markenlizenz streiten. Diese Gefahr war vor dem nun durchgeführten Antragsprozess und der damit verbundenen persönlichen Bindung durch die weitestgehende Anonymität der Lizenznehmer vorhanden, da sich zwei Unternehmer bspw auf die Markenlizenz berufen und dieses Ausschließlichkeitsrecht dazu benutzen könnten, gegen den jeweils anderen rechtlich vorzugehen. Das LSA macht durch diese Aussage deutlich, dass es selbst oder *Linus Torvalds* nicht als Mediationsstelle oder Schiedsgericht auftreten wird und darüber hinaus auch nicht in solche Streitigkeiten einbezogen werden will.

Gem § 2 LSA kann der Markenlizenznehmer oder das LMI die Lizenz mit einer 30-tägigen Frist – ausgenommen individuelle Vereinbarungen mit dem LMI – schriftlich kündigen. Bei einer Verletzung der Markenlizenz kann diese Frist auf 21 Tage nach Kenntnis durch das LMI verkürzt werden. § 2.4 LSA enthält eine automatische Beendigungsklausel, wonach der Markenlizenzvertrag automatisch endet, wenn die in § 2.4 LSA aufgezählten Bedingungen eintreten. Diese Bedingungen stellen mE allesamt wichtige Gründe dar, weshalb diese als vorzeitige Kündigungsmöglichkeiten – der Markenlizenzvertrag ist ein Dauerschuldverhältnis⁷¹⁰ – gültig sind.

⁷⁰⁹ Der Goodwill einer Marke bezieht sich auf den mit ihr verbundenen Wert bzw Ruf in der Öffentlichkeit.

⁷¹⁰ *Schönherr/Thaler*, Entscheidungen zum Markenrecht (1985) E 27 zu § 11.

§ 3 LSA behandelt zusätzliche Rechtseinräumungen und Pflichten des Lizenznehmers. Der Lizenznehmer muss bei seinen Produkten oder Dienstleistungen an prominenter Stelle folgenden Markenhinweis anbringen: *“The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.”* Das LSI ist diesbezüglich sehr großzügig und erlaubt einem Lizenznehmer sinngemäße Hinweise sowie die Möglichkeit einen Hinweis anzubringen, der vorher mit dem LSI abgesprochen wurde. Gem § 3.1 LSA ist es aber nicht möglich, in dem Markenhinweis eine eigene Anmerkung zu platzieren.

§ 4 LSA folgt den Haftungs- und Gewährleistungsausschlüssen, die in den überwiegenden Teilen der Open Source Lizenz zu finden sind und zu dem das bereits in Kap 6 Ausgeführte gilt. Die abschließenden Punkte unter § 5 LSA betreffen formale Aspekte des Markenlizenzvertrags, wobei hier zu erwähnen ist, dass der Markenlizenzvertrag amerikanischem Markenrecht unterliegt und der Gerichtsstand mit Multnomah County in Oregon festgelegt ist. Diese Rechtswahl hat auch hier Gültigkeit, da in einem Streitfall aus dem Markenlizenzvertrag das EVÜ zur Anwendung kommt. Hierbei ist vorrangig auf die Rechtswahl der Parteien Rücksicht zu nehmen, welche sich im Falle des LSA aus § 5.5 LSA ergibt. Ist im Markenlizenzvertrag keine Regelung zu finden, gilt gem Art 4 EVÜ das Recht jenes Staates, zu dem der stärkste Bezug besteht.

Letztendlich ist festzustellen, dass das LSA trotz seiner Ausführlichkeit im Rahmen des § 14 MSchG gehalten ist, wonach das LSA auch in Österreich ohne Abzüge gilt. Der § 14 MSchG birgt dennoch die Möglichkeit, Spannungsfelder für Markenlizenzen im Open Source Bereich aufzubauen. Daher werden diese in Folge erörtert.

7.4.2. Mögliche Spannungsfelder zwischen § 14 MSchG und der OSD

Die OSD gibt als Standard Vorgaben, die herangezogen werden, um ein Programm, ein Produkt oder eine Dienstleistung als Open Source zu bezeichnen. Daher macht es Sinn, bestimmte – in der OSD festgehaltene – Ansprüche auf ihre Verträglichkeit mit § 14 MSchG hin zu überprüfen.

Die in § 1 OSD geforderte freie Weiterverbreitung bezieht sich ieS nur auf die Software selbst. Die Forderung für die Nutzung der Software keinerlei Lizenz- oder andersartige Gebühren zu verlangen, betrifft aber iwS auch die Markenlizenz. Demnach ist es nicht möglich, die Weitergabe der Software an den entgeltlichen Erwerb einer Markenlizenz zu knüpfen, es sei denn, die Software wird frei weitergegeben und kann unter einer eigenen Marke oder einem eigenen Namen verwendet werden. Der Abschluss einer Markenlizenz zur Verwendung der Marke für die Software ist in diesem Fall möglich und kann mE vor allem mit der Qualitätsfunktion der Marke argumentiert werden. Die Qualitätsfunktion ist für Open Source Marken von wesentlicher Bedeutung. Einerseits, um sich von proprietären Produkten abzugrenzen, andererseits um im kommerziellen Bereich Vertrauen zum Produkt herzustellen. Durch die Möglichkeit der freien Weiterentwicklung könnte die Qualität leiden, (unfertige Software kommt auf den Markt, schlechtere oder keine Supportleistungen werden angeboten) was unweigerlich den Ruf des Ursprungsprodukts in Gefahr bringen würde. Festzuhalten bleibt daher: Solange die Software als solche frei weiterverbreitet werden kann, verletzt eine erhobene Markenlizenz nicht §1 OSD.

§ 3 OSD ist ebenfalls in diesem Sinne auszulegen. Demnach darf die Markenlizenz nicht so konstruiert sein, dass ein auf dem Ursprungsprogramm basierendes Werk nur unter Beibehaltung des Markennamens und einem damit verbundenen obligatorischen Abschluss einer Markenlizenz weiterentwickelt

bzw bearbeitet werden darf. Diese Bestimmung muss unabhängig von den Bestimmungen über die Software geschehen.

Der Tatbestand der Diskriminierung durch eine Marke wird seit „Uncle Ben’s“⁷¹¹ einer heftigen Debatte unterzogen. Ohne sich an dieser inhaltlich beteiligen zu wollen, untersagt § 5 OSD die Diskriminierung von einzelnen Personen oder Gruppen. Programme oder Produkte, die einen diskriminierenden oder gar erniedrigenden Namen haben, können daher nicht als Open Source angesehen werden, da sie gegen § 5 OSD verstoßen. Diese Interpretation ist eine weite Auslegung des § 5 OSD, da dieser i.e.S nur den eigentlichen Lizenztext umfasst. Da aber der Projektname meist Teil des Lizenzinhaltes ist oder zumindest bezugnehmend dort behandelt wird, ist er als Bestandteil des Lizenztextes aufzufassen. Ein Projektname, der bspw die Markenbezeichnung „Microsux“ trägt, verstößt – unabhängig seiner markenrechtlichen Behandlung bei der Eintragung oder seiner wettbewerbsrechtlichen Beurteilung – gegen § 5 OSD und scheidet daher als Open Source Software-Produktbezeichnung aus.

§ 6 OSD bezieht sich auf die Einschränkung für bestimmte Anwendungsbereiche, wonach die kommerzielle Nutzung eines Open Source Programms nicht untersagt werden kann. Verbietet eine Markenlizenzklausel die Nutzung der Software im kommerziellen Bereich – ein Grund könnte bspw darin liegen, dass ein von einem Unternehmen für alternative Energien entwickeltes Open Source Programm überwiegend von Kernkraftwerken oder Gentechnikfirmen eingesetzt würde – muss dies differenziert betrachtet werden. Der Markeninhaber kann verlangen, dass seine Software eine andere Markenbezeichnung im Außenverhältnis tragen muss, weil er sich sonst der Gefahr aussetzen würde, von seinen Geschäftspartnern als unseriös eingestuft zu werden. Die Verwendung des Programms kann gem § 6 OSD nicht

⁷¹¹ http://www.marke.at/content/knowledgebase/brand_news/brandnews_detail.asp?bnid=116.

untersagt werden, im Gegensatz zur Verwendung des Markennamens im Rahmen des § 14 MSchG.

Das Verknüpfungsverbot des § 7 OSD bezieht sich auf die Umgehung der Open Source typischen Nutzungsrechte, ist in diesem Fall aber als zentrale Klausel zu sehen, die auch eine Verknüpfung mit einer Markenlizenz untersagt, welche gegen die 10 Punkte der OSD verstößt. Die Verwendung indirekter Mittel wird an dieser Stelle explizit erwähnt und untersagt. Sie kann daher als Ausgangspunkt für eine Kompatibilitätsprüfung hinsichtlich der Markenlizenz mit der OSD gesehen werden.

Das Verbot der Einschränkung der Lizenz für ein bestimmtes Produkt iSd § 8 OSD muss als rein auf die Software bezogen betrachtet werden. Während die Markenlizenzklausel logischerweise die Zuordnung der Marke zu einem bestimmten Programm vorschreiben kann, gilt das für die Software nicht. Diese darf gem § 8 OSD nicht durch die Lizenz so eingeschränkt werden, dass sie Teil einer bestimmten Distribution ist. Wird die Software von einem Lizenznehmer weiterentwickelt, kann dieser durch die Markenlizenz dazu gezwungen werden, im kommerziellen Gebrauch einen anderen Markennamen zu verwenden.

Die erörterten Punkte zeigen, dass die OSD mit der Vorgabe der Maßstäbe, mit denen sich Open Source Software messen muss, einen starken Einfluss auf die rechtliche Beurteilung hat. Dabei bleiben sie im Rahmen des § 14 MSchG und geben dem Lizenzgeber von Open Source Lizenzen klare Vorgaben über die Rechtmäßigkeit der Lizenzinhalte. Ebenfalls deutlich wurde, dass damit ein starker Einfluss auf die Markenlizenzen genommen wird.

7.4.3. Die OSD, die Marke „Open Source“ und der Fall „Sugar CRM“

“When is the OSI going to stand up to companies who are flagrantly abusing the term “open source”?” The answer is: starting today.”⁷¹²

- aus dem Blog von *Michael Tiemann*

Der Begriff „Open Source“ hat als Marke mittlerweile große Attraktivität erlangt. Da der Begriff aber nur die OSI als Markenverwalter⁷¹³ hat und der Begriff an sich mit keinem bestimmten Produkt assoziiert, sondern als Label bestimmten Produkten zugeordnet werden kann, benutzen einige Firmen den Begriff Open Source für ihre Software, obwohl die Lizenzbedingungen, unter der die Produkte vertrieben werden, nicht als Open Source Software iSd OSD angesehen werden dürfen. Die Gründe dafür liegen zum einen darin, dass die Lizenzen nicht bei der OSI zur Überprüfung und Bestätigung vorgelegt wurden und daher formal nicht zu den „approved licenses“ gehören, zum anderen darin, dass die Lizenzen inhaltlich gegen die Vorgaben der OSD verstoßen. Ein prominenter Fall, der die OSI dazu veranlasst hat, gegen solche „Trittbrettfahrer“ vorzugehen, die den guten Ruf von Open Source nutzen, ohne jedoch die Vorgaben erfüllen zu wollen, ist „SugarCRM“. Der Markt für CRM-Lösungen ist momentan gesättigt, weshalb der einzige Zugang nur mittels Open Source CRM-Software möglich scheint. Diese Markteinstiegsstrategie mag die Vertreter von SugarCRM und Socialtext dazu bewogen haben, ihre CRM-Lösungen als Open Source Software zu bezeichnen, obwohl die dazugehörigen Lizenzen SugarCRM Public License und Socialtext Public License bisher nicht von der OSI in die Liste der „approved licenses“ aufgenommen wurden und die Lizenzinhalte auch keine Open Source Lizenzen iSd OSI sind.

⁷¹² <http://www.opensource.org/node/163>.

⁷¹³ Die Marke „Open Source“ wurde in den USA auf die OSI angemeldet, siehe: <http://www.opensource.org/pressreleases/osi-launch.php>.

Aus diesem Verhalten ergeben sich eine Reihe von Fragen für den markenrechtlichen Umgang mit der Wortmarke „Open Source“. Es gilt – unter dem Gesichtspunkt, dass „Open Source“ mittlerweile eine Gattungsbezeichnung darstellt und daher aus den Gründen des § 4 Abs 1 Z 5 MSchG nicht eintragungsfähig ist – zu klären, welche Rechtswahrnehmung der Markeninhaber von „Open Source“ gegenüber anderen hat. Der OGH hat dazu festgestellt⁷¹⁴, dass es ausreiche, *„dass das Wort in den beteiligten Verkehrskreisen als Gattungsbegriff zur Bezeichnung von Waren allgemein verwendet wird. Handle es sich dabei um seltene Waren oder hochspezialisierte Dienstleistungen, an denen nur ein sehr kleiner Abnehmerkreis interessiert sei, komme es eben nur auf die Auffassung dieses kleinen und fachlich gebildeten Personenkreises an.“* Open Source Dienstleistungen sind schon lange nicht mehr hochspezialisiert, weswegen diese erweiterte Qualifikation nicht einmal zur Bewertung des Begriffs „Open Source“ als Gattungsbezeichnung herangezogen werden muss. Außerdem besteht das Problem, dass Firmen, die Computerprogramme vertreiben, nicht sicher sein können, ob sie den Begriff „Open Source“ für Ihre Produkte verwenden dürfen.

Betrachtet man das grundlegende Problem des Registrierungs-hindernisses der Marke „Open Source“ als Gattungsbezeichnung, muss dennoch die Möglichkeit bestehen, die Bezeichnung gegenüber anderen Softwarevarianten⁷¹⁵ abzugrenzen. Eine freie Verwendung ist klar abzulehnen – eine Schallplatte kann nicht als CD bezeichnet werden – die Gattungsbezeichnung „Open Source“ muss sich daher nach den Bestimmungen der OSD richten. Die OSI hat in der OSD den Begriff geprägt und - wie bereits erwähnt – jene Vorgaben festgelegt, die eingehalten werden müssen, um eine Software bzw Lizenz als Open Source anzusehen. Dieses Ergebnis muss zwangsläufig Einfluss auf die Beantwortung des nächsten sich

⁷¹⁴ OGH 24.11.1998, 4 Ob 266/98s – Tabasco – ecolex 1999, 134 = RdW 1999, 78 = ÖBI 1999, 124.

⁷¹⁵ Vgl Kap 2.4.

ergebenden Problems haben. Kann aufgrund des absoluten Freihaltebedürfnisses der Gattungsbezeichnung kein Markeninhaber gegen eine missbräuchliche Verwendung vorgehen, würde dies eine große Gefahr für den Begriff „Open Source“ darstellen. Dieses Problem wird aber durch das Lauterkeitsrecht, mit seinem – wie *Wiebe* richtig betont⁷¹⁶ – „*primär kommunikationsregelnden Charakter*“ gelöst. Das Lauterkeitsrecht sichert über die Generalklausel des § 1 UWG auch dann den Schutz vor Ausbeutung fremder Leistung, wo ein immaterialgüterrechtlicher Schutz – wie in diesem Fall durch das Markenrecht – nicht vorhanden ist⁷¹⁷. Die Anbieter von „SugarCRM“ können daher aufgefordert werden, den missbräuchlichen Nutzen des Begriffs „Open Source“ einzustellen, da ihr Produkt nicht den Anforderungen des Gattungsbegriffs entspricht.

Eine andere Möglichkeit den Begriff „Open Source“ markenrechtlich zu schützen besteht darin, ihn als Verbandsmarke iSd §§ 6w ff MSchG anzumelden. Diese Variante ist allerdings in der Praxis noch schwer durchsetzbar, da mit dem Verein „Open Source Austria“ ein noch junger Verband existiert, der einerseits noch nicht über die ausreichenden finanziellen Mittel zur Anmeldung der Verbandsmarke verfügt – die Anmeldegebühr beträgt immerhin das Vierfache der in § 18 Abs 1 MSchG festgelegten Anmeldegebühr, die Erneuerungsgebühr das Zehnfache der in § 18 Abs 2 MSchG festgelegten Schutzdauergebühr – und der andererseits noch nicht die Größe erreicht hat, um als repräsentativer Verband angesehen zu werden.

Es wird ersichtlich, dass die Durchsetzung von markenrechtlichen Ansprüchen für die Entwickler und Dienstleister im Open Source Bereich einen wichtigen Aspekt darstellen, weswegen die Durchsetzung des zivilrechtlichen Schutzes nachfolgend erörtert wird.

⁷¹⁶ *Wiebe*, Umsetzung der Geschäftspraktikenrichtlinie und Perspektiven für eine UWG-Reform, JBI 2007, 69 (70).

⁷¹⁷ *Kucsko*, Geistiges Eigentum 102.

7.5. Der zivilrechtliche Schutz von Open Source Software-Marken

Marken haben – wie ausgeführt – für Open Source Software Produkte mit zunehmender wirtschaftlicher Bedeutung eine immer wichtigere Stellung eingenommen. Durch das Open Source Lizenzmodell und die offene Gestaltung der Lizenzannahme muss besonderes Augenmerk auf die Aktivlegitimation und die Sanktionsmöglichkeiten gelegt werden.

Die zivilrechtlichen Schutzmöglichkeiten sind ähnlich wie beim UrhG ausgeprägt. Die in Zusammenhang mit Open Source Software-Marken wichtigsten sind dabei, neben den Schutzmöglichkeiten des Lauterkeitsrechts im UWG, der Unterlassungsanspruch gem § 51 MSchG und der Beseitigungsanspruch gem § 52 MSchG⁷¹⁸. Diese werden in Folge hinsichtlich des Gebrauchs von Open Source Marken besprochen. Ein wichtiger Aspekt in diesem Zusammenhang ist die Klärung der Aktivlegitimation, da die Markenlizenz oftmals an ein anonymes Publikum vergeben wird und dies zu ungewollten Klagen führen kann.

7.5.1. Die Aktivlegitimation bei Open Source Marken

Um die Frage nach der Aktivlegitimation beantworten zu können, muss man die Markenlizenz in Hinblick auf die Rechteeinräumung überprüfen. Durch das Open Source Lizenzmodell kommt es in der Praxis häufig zu dem Problem, dass sich nicht der eigentliche Markeninhaber mit einer Markenschutzverletzung auseinandersetzen muss, sondern ein Lizenznehmer. Aus diesem Grund befasst sich dieser Abschnitt vorrangig mit den rechtlichen

⁷¹⁸ Der Zahlungsanspruch gem § 53 MSchG ist zwar für die Praxis bedeutend, wirft aber im Zusammenhang mit Open Source Marken keine besonderen rechtlichen Probleme auf.

Rahmenbedingungen, die die Aktivlegitimation des Markenlizenznehmers regeln.

Die Gesetzgebung erwähnt die Markenlizenzen in den §§ 11, 14 und 28 MSchG ohne dabei Dispositivnormen aufzustellen, welche die Klagebefugnis des Lizenznehmers regeln könnten. Diese Problematik wurde von der Rsp geklärt⁷¹⁹, die dazu ausführte, dass der Lizenzgeber sich darauf beschränken könne, dem Lizenznehmer lediglich den Gebrauch zu gestatten. Eine andere Möglichkeit besteht darin, dem Lizenznehmer eine Lizenz mit absoluter Wirkung gegenüber Dritten einzuräumen. Die sich daraus ergebenden Konsequenzen für die Open Source Markenlizenzen werden nachfolgend erörtert.

7.5.1.1. Die Gebrauchsüberlassung einer Open Source Marke

Hat der Markenlizenzgeber einer Open Source Marke dem Lizenznehmer lediglich den Gebrauch eingeräumt, entsteht dadurch weder eine dingliche Rechtsübertragung noch die Einräumung einer echten Nutzungsbefugnis⁷²⁰. *Schanda* kritisierte 2001⁷²¹, dass die vom OGH gefundene Lösung die Begriffe „ausschließlich“ und „absolut“ in einer Art verknüpfe, die keine klare Interpretation zulasse, sondern Rechtsunsicherheit schaffe. Ausschließliche Markenlizenzen müssten als Exklusivitätsvereinbarung mit dem Lizenzgeber verstanden werden. Der OGH reagierte in seiner Entscheidung „Brühl“⁷²² auf die Kritik und stellte klar, dass auch der Lizenznehmer, der eine Gebrauchsüberlassung der Marke eingeräumt bekommen hat, über eine Aktivlegitimation verfügen kann, gegen Markenschutzverletzungen Dritter mittels Klage vorzugehen, wenn der

⁷¹⁹ OGH 17.08.2000, 4 Ob 178/00f – BOSS-Brillen II. – ecolex 2000/352 = WBI 2001/34 = ÖBI 2001/89.

⁷²⁰ Man spricht in diesem Zusammenhang auch vom „einfachen“ Markenlizenznehmer.

⁷²¹ *Schanda*, Kritisches zur Aktivlegitimation des Markenlizenznehmers ÖBI 2001, 151.

⁷²² OGH 15.10.2002, 4 Ob 209/02t – Brühl – WBI 2003/95 = ÖBI 2003/87.

Lizenzgeber dem Lizenznehmer über die Gebrauchsüberlassung hinaus eine Ermächtigung dazu eingeräumt hat. Dies ergibt sich für den OGH aus dem Wortlaut des § 51 MSchG und darf nicht als bloße Prozessstandschaft gewertet werden. Für Open Source Markenlizenznehmer muss daher festgestellt werden inwiefern die Markenlizenz die Aktivlegitimation einräumt. Punkt 6 APL bspw räumt lediglich einen eingeschränkten Gebrauch der Marke „*Apache*“ ein. In diesem Fall kann keinesfalls von einer Aktivlegitimation des Lizenznehmers ausgegangen werden. Solange Open Source Markenlizenzen keine ausdrückliche Ermächtigung erteilen, kann in keinem Fall von einer Aktivlegitimation ausgegangen werden. Die Ermächtigung einer Markenlizenz durch Interpretation zu erlangen, ist mE nicht möglich, da die offen gestaltete Lizenzannahme durch jeden einen restriktiven Umgang mit Markenlizenzen fordert und daher nicht dazu führen darf, dass eine Marke missbräuchlich in einen Prozess verwickelt wird. Problematisch wird es in den Fällen, in denen der „*einfache*“ Markenlizenznehmer, der keine Ermächtigung erhalten hat, mit einer Markenschutzverletzung konfrontiert wird, die ihn in seinem Wettbewerb hindert. In einem solchen Fall muss er den Markeninhaber davon überzeugen, entweder selbst seine Rechte durchzusetzen oder den Markenlizenznehmer nachträglich dazu ermächtigen. Letztere Variante erscheint in der Praxis am sinnvollsten, da die Markeninhaber oftmals in anderen Ländern sitzen und das wirtschaftliche Risiko einer Gerichtsverhandlung scheuen werden. Eine andere Möglichkeit ist, den Markenlizenznehmer mit „absoluten“ Rechten auszustatten.

7.5.1.2. Die Einräumung einer Open Source Markenlizenz mit absoluter Wirkung

Die Einräumung einer Open Source Markenlizenz mit absoluter Wirkung ist gegeben, wenn der Lizenzgeber dem Lizenznehmer die Befugnis zur Abwehr von Markenverletzungen einräumt. Im Unterschied zum dt Markenschutzrecht – § 30 Abs 3 dt MSchG bestimmt, dass der Lizenznehmer trotz ausschließlicher Lizenz nur mit Zustimmung des Markeninhabers tätig

werden kann – ist die Aktivlegitimation in Österreich im Falle einer Lizenz mit absoluter Wirkung nicht von der Zustimmung des Markeninhabers abhängig. *Schanda* betont⁷²³, dass dies e contrario gefolgert werden müsse, da das österreichische MSchG keine dem dt MSchG äquivalente Bestimmung aufweise. Ein Lizenzgeber muss daher die Klagebefugnis explizit ausschließen, – wie dies bspw in der „Nokia Open Source License“ der Fall ist⁷²⁴ – wenn er dem Lizenznehmer keine Aktivlegitimation zukommen lassen will. Wichtig anzumerken ist, dass die Einräumung der Klagebefugnis des Lizenznehmers, die des Lizenzgebers nicht ausschließt⁷²⁵. Diese Drittwirkung ist für Open Source Lizenzen sehr wichtig, da kein Open Source Projekt als Markeninhaber eine ausschließliche Markenlizenz erteilen würde, wenn dadurch der Schutz der eigenen Marke von Dritten abhängig wäre. Durch die Möglichkeit der alternativen Aktivlegitimation kann der Markenlizenzgeber jedoch dem Lizenznehmer die Möglichkeit geben, schon frühzeitig gegen Markenrechtsverletzungen vorzugehen, wenn ihm dies selbst nur schwer – der Markeninhaber sitzt in einem anderen Land, der Markeninhaber hat keine ausreichenden Ressourcen – möglich ist.

7.5.2. Der Unterlassungsanspruch von Open Source Software-Marken am Bsp der Marke „Open Source“

Die Voraussetzung für eine Unterlassungsklage gem § 51 MSchG ist das Bestehen der Wiederholungsgefahr, welche von der Rsp definiert wurde. Demnach ist von einer Wiederholungsgefahr dann auszugehen, wenn eine Wiederholung ernstlich zu besorgen ist⁷²⁶. Diese Gefahr ist im IT-Bereich auf

⁷²³ *Schanda*, ÖBI 2001, 151.

⁷²⁴ Dort wird in Punkt 10 bestimmt: *“Nothing in this License shall grant You any rights to use any of the trademarks of Nokia or any of its Affiliates, even if any of such trademarks are included in any part of Covered Software and/or documentation to it.”*

⁷²⁵ OGH 17.08.2000, 4 Ob 178/00f – BOSS-Brillen II – ecolex 2000/352 = WBI 2001/34 = ÖBI 2001/89.

⁷²⁶ *Karmasin* in *Kucsko*, Geistiges Eigentum 522.

Grund des vorwiegenden Einsatzes des Internet besonders groß und dies gilt für den internetlastigen Vertrieb von Open Source Software um so mehr, weswegen hier ein besonders strenger Maßstab angelegt werden muss⁷²⁷. Benutzt jemand den Ausdruck „Open Source“ missbräuchlich, kann er bislang nur vom Markeninhaber der OSI geklagt werden, da die OSI als Markeninhaber keine Unterlizenzen – weder einfache Markenlizenzen mit Ermächtigung noch ausschließliche Markenlizenzen – vergeben hat. In der Praxis führte dies dazu, dass Firmen die Marke „Open Source“ teilweise missbräuchlich verwendeten, ohne Sanktionen befürchten zu müssen, da die OSI bis Ende 2006 äußerst lethargisch geführt wurde. Erst der Beitrag von *Blankenhorn*⁷²⁸ führte dazu, dass die OSI ihre Rolle als Hüter der Marke „Open Source“ anerkannte und inzwischen versucht, die damit verbundenen Markenrechte zukünftig durchzusetzen. Sollte jemand den Ausdruck „Open Source“ nutzen, ohne dass sein Produkt den Regeln der OSD entspricht, wird zunächst ein Unterlassungsanspruch gerechtfertigt sein. Da es vom Verhalten des Beklagten abhängt, ob von einer Wiederholungsgefahr ausgegangen werden muss, kann bei der Vermarktung eines Nicht-Open Source Produkts als Open Source Produkt die Wiederholungsgefahr nur dadurch ausgeschlossen werden, wenn der unrechtmäßige Vertreiber

- seinen Wettbewerbsverstoß zugibt und sich zu dem Verstoß bekennt⁷²⁹ oder
- bei der Berufung auf den Wegfall der Wiederholungsgefahr daran festhält, durch die beanstandete Handlung keinen Gesetzesverstoß begangen zu haben, aber dennoch einen

⁷²⁷ Der OGH hat dieser Ansicht in seinem Urteil OGH 13.03.2002, 4 Ob 39/02t – kunstNET – ÖBI-LS 2002/125 = ecollex 2002, 597 (*Schanda*) Ausdruck verliehen.

⁷²⁸ <http://blogs.zdnet.com/open-source/?p=1123>.

⁷²⁹ OGH 04.02.1999, 4 Ob 348/98z – Ford-KG – ÖBI 1999, 295; OGH 22.04.1997, 4 Ob 96/97i – Ramtha – ÖBI 1998, 53 = ecollex 1997, 681 (*Schanda*).

den ganzen Unterlassungsanspruch umfassenden und an keinerlei Bedingungen geknüpften Vergleich anbietet⁷³⁰;

Wie angeführt wird der Beweis des Wegfalls der Wiederholungsgefahr im Internetbereich aber nur schwer zu führen sein, weshalb der Beklagte den Beweis für den Wegfall in der Praxis nur im seltensten Fall wird erbringen können.

Ein Vorteil für die Durchsetzung des Unterlassungsanspruchs ist die Möglichkeit, das Unterlassungsgebot allgemeiner zu gestalten. Dies kann schon bei Begehungsgefahr als vorbeugende Unterlassungsklage eingebracht werden⁷³¹, wenn bspw die Ankündigung eines Open Source Produkts bekannt wird, von dem mit überwiegender Wahrscheinlichkeit davon ausgegangen werden kann, dass das Produkt nicht den Regeln der OSD entsprechen wird. Für die überwiegende Wahrscheinlichkeit spricht bspw, dass das als Open Source angekündigte Produkt in seiner Produktbeschreibung eine proprietäre Lizenz anführt oder von vorneherein Lizenzgebühren für den Quellcode verlangt werden.

Ein anderer Aspekt zum Unterlassungsbegehren findet sich im UWG. Im Gegensatz zu Deutschland ist es nicht nötig, vor der Klagseinreichung den Verletzer abzumahnern. Dies hat für den Kostenersatz weitreichende Folgen. Während § 12 dt UWG eine solche Abmahnung dezidiert vorsieht, hat der korrespondierende § 14 österreichisches UWG die Entscheidung darüber – durch das Auslassen einer entsprechenden Bestimmung – der Judikatur überlassen. Der OGH hat dazu ausgeführt, dass ein Kläger nicht verpflichtet ist, den Beklagten vor der Einleitung des Rechtsstreits abzumahnern⁷³² und hat

⁷³⁰ OGH 04.02.1999, 4 Ob 348/98z – Ford-KG – ÖBI 1999, 295; OGH 22.04.1997, 4 Ob 96/97i – Ramtha – ÖBI 1998, 53 = ecolex 1997, 681 (*Schanda*). Bejaht das Erstgericht den Verstoß, so ist das Vergleichsangebot unzureichend: OGH 29.09.1992, 4 Ob 66 und 67/92 – MERCEDES – Teyrosowsky – ÖBI 1992, 273 = MR 1992, 252 = ecolex 1993, 99.

⁷³¹ Vgl dazu: OGH 12.06.2001, 4 Ob 139/01x – täglichalles.at – MR 2001, 245 (*Korn*) = ecolex 2001, 923 (*Schanda*) = wbl 2001, 540.

⁷³² OGH 31.05.1983, 4 Ob 311/83 – Testkäufer – ÖBI 1983, 129.

diese Entscheidung im Jahre 2000 präzisiert⁷³³, indem er ausgesprochen hat, dass der Beklagte die Klage nicht veranlasst hat, wenn eine vorherige Abmahnung vereinbart wurde, der Beklagte aber tatsächlich nie abgemahnt wurde. Diese Tatsache knüpft der OGH allerdings an eine Bedingung. Der Beklagte kann sich nur dann auf die Abmahnvereinbarung berufen, „*wenn er den Anspruch des Klägers sofort anerkennt; bestreitet er ihn hingegen, so ist der Verstoß gegen die Abmahnvereinbarung ebenso sanktionslos wie ein Verstoß gegen die nach deutschem Handelsbrauch bestehende Pflicht, den Verletzer zur Vermeidung einer prozessualen Auseinandersetzung und zur Abwendung von Kostennachteilen zunächst abzumahnern.*“ Die Anwendung des § 45 ZPO ist damit unter bestimmten Voraussetzungen möglich, auch wenn diese auf Unterlassungsklagen für das UWG allgemein verneint wurde⁷³⁴. Durch das fehlende Erfordernis der Abmahnung sind vor allem Open Source Anbieter im KMU-Bereich im Falle einer – auch unabsichtlich begangenen – Verletzung zur Leistung des Kostenersatzes verpflichtet, sollte der Markeninhaber ohne Vorwarnung eine gerichtliche Durchsetzung beantragen.

Ein anderes wesentliches Hilfsmittel für den Schutz der eigenen Marke ist die einstweilige Verfügung, da mit dieser schnellst möglich auf eine drohende Markenverletzung reagiert werden und zudem die Einhaltung der einstweiligen Verfügung mit massiven Beugestrafen sanktioniert werden kann. Da das Verfahren der einstweiligen Verfügung bis zum OGH gehen kann, besteht die Möglichkeit, dass durch dieses ein allfälliges Hauptverfahren unter Umständen vermieden wird. Es gilt jedoch das Prozessrisiko gründlich abzuwägen, da eine ungerechtfertigte einstweilige Verfügung zu Schadenersatzansprüchen gem § 394 EO führen kann, die in der Praxis recht hoch ausfallen dürften.

⁷³³ OGH 12.04.2000, 4 Ob 106/00t – Tiroler Tageszeitung – MR 2000, 173 (*Frauenberger*).

⁷³⁴ OLG Wien 24.05.1976, 1 R 118/76 – MusiCassetten – ÖBl 1976, 167; siehe dazu auch: *Schimanko*, Die Klage auf Unterlassung und § 45 ZPO, ÖJZ 2002, 127 der sich sehr kritisch mit der Ablehnung des § 45 ZPO für das UWG auseinandersetzt.

Der Unterlassungsanspruch ist damit der erste und mittels einstweiliger Verfügung auch wirksamste Rechtsbehelf, um (s)eine Marke zivilrechtlich schützen zu können. Der Aktivlegitimation kommt dabei eine wichtige Rolle zu und es kann durch die bislang noch nicht allzu effektiv genutzten Markenlizenzverträge dazu führen, dass Markenverletzungen erst relativ spät gerichtlich geahndet werden können, da die Markeninhaber in den USA sitzen und nur die wenigsten eine Gemeinschaftsmarke oder gar nationale Marken angemeldet haben.

7.5.3. Der Beseitigungsanspruch von Open Source Software-Marken am Bsp der Marke „Open Source“

Dem Unterlassungsanspruch nachfolgend, gilt der Beseitigungsanspruch gem § 52 Abs 1 MSchG, wonach ein Markenverletzer zur Beseitigung des dem Gesetz widerstrebenden Zustands verpflichtet ist. Die Beseitigung richtet sich im Open Source Bereich vorrangig auf die Eingriffsgegenstände. Handelt es sich bei dem Eingriffsgegenstand um einen Werbeprospekt oder einen physikalischen Datenträger – bspw CD-ROM, DVD-ROM oder USB-Stick – können diese ohne Probleme vernichtet werden. Problematischer wird die Beseitigung im Internetbereich. Hier muss der Markenverletzer zumindest die in seinem Einflussbereich – eigene Webseite, Webspace, Webseite von Vertriebspartnern – liegenden Eingriffsgegenstände unverzüglich durch Löschung beseitigen. Außerdem muss er dafür sorgen, dass Suchmaschinen die markenverletzenden Inhalte nicht mehr indizieren können. Auch wenn die direkte Abrufbarkeit nicht mehr gegeben ist, besteht weiterhin die Möglichkeit über Suchmaschinen nicht direkt verlinkte Seiten aufzufinden und den Inhalt abzurufen. Soweit es dem Markenverletzer zumutbar ist, muss er dafür sorgen, dass die wichtigsten Suchmaschinen die markenverletzenden Inhalte aus ihren Caches löschen. Bezüglich der Eingriffsmittel muss betont werden, dass das Unbrauchbarmachen derselben nicht Bestandteil des Klagebegehrens sein kann, da dies über den Zweck der Beseitigung hinausgehen und ein zu

drastisches Mittel darstellen würde. Während bei der klassischen Produktpiraterie die Vernichtung bzw das Unbrauchbarmachen der Eingriffsmittel nachvollziehbar erscheint, kann dies bei der Markenverletzung im Internetbereich kein geeignetes Mittel sein. Als Eingriffsmittel muss hier bspw jeder Computer und Server angesehen werden. Diese können – müssen aber nicht – ausschließlich oder vorzugsweise dazu eingesetzt werden, die markenverletzenden Produkte herzustellen. Es wird daher ausreichen, wenn die markenverletzenden Produkte von den Speichermedien - wie bspw Festplatten der Rechner und Server - des Markenverletzers nachweisbar gelöscht werden.

7.6. Die Situation von Open Source Gemeinschaftsmarken in Bezug auf die EU-Erweiterung.

Ein wichtiger internationaler Aspekt für Open Source Marken ergibt sich aus der Erweiterung der EU und die damit verbundenen kollisionsrechtlichen Fragen. Wurden auch durch die immer stärker ausgeprägte Harmonisierung des gewerblichen Rechtsschutzes in der EU die rechtlichen Probleme, die sich beim grenzüberschreitenden Einsatz von Marken ergeben, weitestgehend gelöst, so haben sich durch die EU-Erweiterung dennoch neue Rechtsfragen in Zusammenhang mit der Erstreckung der Gemeinschaftsmarke ergeben. Diese Problematik ist für Open Source Marken aufgrund der internationalen Verbreitung immanent.

Abhilfe schafft da der seit Anfang 2004 gültige und für die damals durch die 10 neuen Mitgliedstaaten entstandene EU-Erweiterung eingefügte Art 142a GMV. Eine wichtige rechtliche Auswirkung für Inhaber von Gemeinschaftsmarken – zum damaligen Zeitpunkt gab es mehr als 300.000 Gemeinschaftsmarken⁷³⁵ – hatte, dass der Schutz für die vor dem Tag des

⁷³⁵ <http://oami.europa.eu/de/office/aspects/communications/05-03.htm>.

Beitritts angemeldeten Marken auf das Gebiet der neuen Mitgliedstaaten erstreckt wurde und seitdem vor den Gemeinschaftsmarkengerichten aller Mitgliedstaaten durchgesetzt werden kann. Zu dieser automatischen Erstreckung kam es ebenfalls 2007 im Zuge der Beitritte von Rumänien und Bulgarien, womit die Gemeinschaftsmarke nun in 27 Ländern unmittelbar Schutz genießt. Eine Besonderheit im Zusammenhang der Erweiterung war und ist es, dass eine erstreckte Gemeinschaftsmarke nicht durch das Bestehen absoluter Schutzverweigerungsgründe, die durch den Beitritt neuer Mitgliedstaaten entstanden sind, ihren Schutz verliert. Damit kann eine erstreckte Marke auch in den neuen Mitgliedstaaten gegen die öffentliche Ordnung oder die guten Sitten verstoßen, wobei nach Meinung von *de Boer*⁷³⁶ die Möglichkeit besteht, die Benutzung einer solchen Marke gem Art 106 Abs 2 GMV zu untersagen. Für Open Source Marken ist hierbei der Vorteil zu erwähnen, dass eine als Gemeinschaftsmarke anerkannte Marke keiner neuerlichen Prüfung standhalten muss, was wesentlich zur Rechtssicherheit beiträgt.

Der Widerspruch gegen erstreckte Gemeinschaftsmarken aufgrund relativer Schutzverweigerungsgründe kann nur dann erhoben werden, wenn diese während der letzten sechs Monate vor dem Beitrittstag angemeldet wurden. Zu beachten ist daher, dass Gemeinschaftsmarken, die zwischen dem 1. Juli 2006 und dem 31. Dezember 2006⁷³⁷ angemeldet wurden, nur unter folgenden Voraussetzungen angegriffen werden können⁷³⁸ und zwar: Wenn das Prioritätsdatum der nationalen Marke absolut gesehen älter ist als jenes der erstreckten Gemeinschaftsmarke und darüber hinaus gutgläubig erworben wurde. Der Widerspruch muss gem Art 42 GMV binnen drei Monaten ab Veröffentlichung der Gemeinschaftsmarke eingebracht werden und steht daher einer Veröffentlichung nach dem Beitrittsdatum nicht entgegen.

⁷³⁶ <http://oami.europa.eu/de/office/aspects/communications/05-03.htm>.

⁷³⁷ Die Spanne bei der Erweiterung 2004 betrug: 1. November 2003 – 30. April 2004.

⁷³⁸ Siehe dazu auch: *Boesch*, Die Auswirkung der Erweiterung der EU auf die Gemeinschaftsmarke, ÖBI 2005/20, 100.

Neben dem Widerspruch gegen erstreckte Gemeinschaftsmarken aufgrund relativer Schutzverweigerungsgründe besteht für den Inhaber eines national erworbenen Markenrechts die Möglichkeit, die Benutzung der Gemeinschaftsmarke gem Art 106 und 107 GMV zu untersagen, wenn er seine älteren Rechte iSd Art 8 und 52 Abs 2 GMV nachweisen kann. Dieser Anspruch verfällt nach fünfjähriger Nutzung der jüngeren Gemeinschaftsmarke, wenn der Inhaber des älteren Rechts diese Nutzung wissentlich duldet⁷³⁹. Besteht daher bspw in Rumänien der Markenschutz für eine Software namens „SUSE“, welcher älter ist als der, der Gemeinschaftsmarke „SUSE LINUX“, ist es dem rumänischen Markeninhaber möglich, durch ein rumänisches Gericht die Nutzung der erstreckten Gemeinschaftsmarke untersagen zu lassen. Dieses Nutzungsverbot gilt dann für die Gemeinschaftsmarke in Rumänien, in den übrigen 26 Mitgliedstaaten kann sie verwendet werden, solange dort nicht ebenfalls nationale Hinderungsgründe bestehen.

Letztlich bleiben praktische Probleme zu untersuchen, die bei der Erstreckung der Gemeinschaftsmarke auftauchen könnten. Der Nachweis der Bösgläubigkeit – dies betrifft jene nationalen Marken, die vor dem 1. Januar 2007 zum Zweck der Markenerschleichung, der Errichtung einer Sperr- oder Hinterhaltmarke angemeldet wurden - wird in der Regel schwierig sein und somit Missbrauch fördern. Diese Gefahr ist im Open Source Bereich groß, da die missbräuchliche Anmeldung vor allem Marken aus dem KMU-Bereich treffen wird. Große Unternehmen dagegen haben ihre Marke in der Regel bereits national registriert. Die Beweislast für die Bösgläubigkeit liegt beim Inhaber der erstreckten Gemeinschaftsmarke und erschwert dieses Problem zusätzlich. Bösgläubigkeit muss immer in jenen Fällen angenommen werden, bei denen die Umstände darauf hinweisen, dass die nationale Marke in der Absicht registriert wurde, diese an den Inhaber der erstreckten Gemeinschaftsmarke oder einen seiner Wettbewerber zu verkaufen, zu

⁷³⁹ Boesch, ÖBl 2005/20, 102 mwA.

lizenzieren oder auf sonstige Weise zu veräußern, um damit Gewinne zu erzielen, die über jene, mit der Anmeldung der nationalen Marke zusammenhängenden Kosten, hinausgehen. Die Bösgläubigkeit bemisst sich darüber hinaus nach nationalen Vorschriften⁷⁴⁰, weshalb hier eine weitere Hürde, die Auseinandersetzung mit nationalen Rechtsvorschriften, entsteht. Dem folgend kann die meist lange Dauer des Verfahrens zur Rechtsdurchsetzung in den neuen Mitgliedsstaaten als Problem angesehen werden wie *Boesch* richtig betont⁷⁴¹.

Den Inhabern einer Open Source Gemeinschaftsmarke muss daher geraten werden, trotz der automatischen Erstreckung ihrer Markenrechte, sich auf mögliche Konflikte vorzubereiten. Sollte eine nationale Projektgruppe bestehen, welche ein globales Open Source Projekt betreut und die dadurch unter Umständen einen älteren Markenschutz aufgrund der Verkehrsgeltung – die Überprüfung der Verkehrsgeltung erfolgt jeweils national, in Österreich wäre § 31 MSchG maßgeblich – erlangt hat, sollte das Gespräch gesucht werden, um die Inhaber der älteren nationalen Marke zu einer möglichen Partnerschaft bzw einem möglichen gemeinschaftlichen Vertrieb zu überreden. Bezüglich der Bösgläubigkeit bleibt zu raten, einen national ansässigen Rechtsbeistand mit dem Fall zu betrauen und die Durchsetzung mittels einstweiliger Verfügung zu versuchen, da diese in der Praxis das effektivste Mittel sein wird, einen weiteren Missbrauch der Marke zu verhindern.

7.7. Ergebnis

Der Begriff „Open Source“ ist mittlerweile so populär geworden, dass er selbst in eine Markenrechtsdebatte verwickelt ist. Der Schutz des Begriffs als

⁷⁴⁰ Siehe dazu: *Fottner*, EU-Marken im Zeichen der Osterweiterung, <http://www.ipmunich.de/Grafik/mfs-02-2004.pdf>.

⁷⁴¹ *Boesch*, ÖBl 2005/20, 103.

Qualitätsmerkmal iSd OSD sollte dabei im Vordergrund stehen, um die Open Source Gemeinschaft und ihre Lizenzen nachhaltig zu schützen. Dazu wird ein höheres Engagement der OSI notwendig sein, die es bisher leider verabsäumt hat, einen effektiven Markenrechtsschutz für „Open Source“ zu erwirken. Dieses Verhalten verwundert, da mit der Einführung des Open Source Begriffs die wirtschaftliche Verwertung freier Software gefördert und herausgestrichen werden sollte. Dabei jedoch das wesentlichste Merkmal ungeschützt zu lassen, ist mE schwer nachvollziehbar, zeigt aber für wie unbedeutend das Markenrecht in der Open Source Gemeinschaft gehalten wurde. Erst in den letzten Jahren – mit der zunehmenden Bekanntheit und Verbreitung von Open Source Software – hat auch das Markenrecht eine immer wichtigere Bedeutung für Open Source Produkte bekommen. Die diskutierten Fälle der Registrierungshindernisse hinsichtlich der Anmeldung werden in Zukunft, durch die immer professionellere Vorgangsweise bei der Entwicklung und Verbreitung von Open Source Projekten, kaum noch zu ungewollten Markenrechtsverletzungen führen. Aktuelle Beobachtungen zeigen, dass kommerzielle Projekte den Marketingansatz immer mehr in den Vordergrund stellen und die Software selbst an diesen angepasst wird und nicht umgekehrt.

Der zunehmende Stellenwert des Markenrechts für den Open Source Bereich führt dazu, dass Open Source Projekte sich zu Beginn mit den markenrechtlichen Rahmenbedingungen auseinandersetzen müssen, um eine Marke erfolgreich anmelden zu können. Dabei muss der Sinn einer Anmeldung als Gemeinschaftsmarke überdacht werden, je nach Größe, Umfang und grenzüberschreitender Verbreitung des Projekts. Ebenso wichtig ist es, an die Nutzung der Marke iSd § 33a MSchG zu denken, da die Anmeldung der Marke zukünftig oftmals noch vor Projektbeginn stattfinden wird. Hierbei sollte von Beginn überlegt werden, wer als Markeninhaber eingetragen wird und in welcher Form eine Markenlizenzierung iSd § 14 MSchG vorgenommen wird. Die Markenlizenz sollte dabei neben den Nutzungs- und Verwertungsrechten der Lizenz vereinbart werden. Einerseits kann dadurch vermieden werden, dass neue Projekte weitere Lizenzen produzieren, nur um eine Lizenz mit dem

eigenen Markennamen zu haben, andererseits kann eine eigene Markenlizenz unabhängig von der Open Source Lizenz adaptiert werden. Die Markenlizenz muss neben den gesetzlichen Vorgaben auch die Regeln der OSD befolgen, wenn sie dafür genutzt wird, ein Produkt als Open Source zu vermarkten.

Durch die Zunahme der Bedeutung des Markenschutzes wächst die Gefahr einer gerichtlichen Auseinandersetzung wegen der Verletzung von Markenrechten, die – wie erwähnt – im Gegensatz zu Deutschland keiner Abmahnung bedarf und daher hohe Kosten verursachen kann. Bei Open Source Projekten kann dies vor allem aus folgenden Gründen vorkommen:

1. Die bewusste Anlehnung an bekannte Marken bzw. Namen und Logos,
2. keine Überprüfung älterer Rechte,
3. Übernahme von Projektverantwortung ohne die rechtlichen Konsequenzen zu bedenken,
4. Vermischung von kommerziellen und freien Entwicklungsprojekten,
5. Nutzung des Begriffs „*Open Source*“ ohne Berücksichtigung der OSD,
6. Nutzung von Projektnamen, deren markenrechtlichen Eintragung absolute Eintragungshindernisse entgegenstehen.

Bisher konnten diese Konflikte meist außergerichtlich beigelegt werden. Mit wachsendem Konkurrenzdruck werden diese Auseinandersetzungen zunehmend vor Gericht verhandelt werden müssen. Die klagenden Parteien spekulieren dabei häufig mit der organisatorischen Schwäche kleinerer Open Source Projekte, die sich meist nur unzureichend gegen die juristischen Angriffe wehren können und dann den Forderungen – seien sie zu recht oder zu unrecht gestellt – nachgeben. Bei der Frage der Berechtigung ist es wichtig zu klären,

ob das streitgegenständliche Open Source Projekt überhaupt im geschäftlichen Verkehr steht. Dies ist dann der Fall, wenn es kommerziell genutzt wird oder im direkten Wettbewerb zu anderen kommerziell tätigen Unternehmen steht. Oft reicht schon die Schaltung von Werbung (bspw durch ein Werbebanner auf einer Webseite), um iSd Markenrechts im geschäftlichen Verkehr tätig zu sein. Handelt es sich bei dem Open Source Projekt um ein Forschungsprojekt (bspw an einer Universität), bei dem ausschließlich die wissenschaftliche Erkenntnisgewinnung vorangetrieben werden soll, muss das Projekt als nicht im geschäftlichen Verkehr tätig angesehen werden und kann somit keine markenrechtlichen Bestimmungen verletzen. Die angeführten Fälle machen deutlich, dass viele proprietäre Software-Anbieter die Konkurrenz von Open Source Lösungen fürchten und mit allen Mitteln versuchen, diese vom Markt zu verdrängen. Dabei kommt in Österreich dem UWG neben dem MSchG die wichtigste Bedeutung zu. Auf der anderen Seite nutzen Firmen, die im Open Source Bereich tätig sind, immer stärker Marken zur Produktwerbung für Marketingzwecke oder als Qualitätsmerkmal, da viele der Projekte die „Amateurphase“ hinter sich gelassen haben und nun im direkten Wettbewerb versuchen, wirtschaftlich erfolgreich zu sein. Den konkurrierenden proprietären Produktanbietern sollte bewusst sein, dass Klagen, die aufgrund von markenrechtlichen Streitigkeiten eingebracht werden, nicht mehr ausreichen, um Konkurrenz durch Open Source Software loszuwerden. Die neuesten Tendenzen, mit Immaterialgüterrechten die Entwicklung und Verbreitung von Open Source Software anzugreifen, zeigen sich deutlich in der Patentrechtsdebatte auf EU-Ebene.

Positiv anzumerken sind die Harmonisierungsbestrebungen, welche im internationalen Bereich wie auf Gemeinschaftsebene vorangetrieben werden. Der wirtschaftliche Wert einer Marke wird hierdurch effektiver geschützt. Für ein Open Source Projekt wird durch die Möglichkeit einer Markenrecherche im Internet kostengünstig überprüft, ob das Projekt eventuell bestehende Markenrechte eines Dritten verletzt oder zumindest als Beeinträchtigung gewertet werden kann.

8. VERGABERECHT

„*Never change a running system*“! Dieser Slogan begegnet einem regelmäßig, wenn bei einer Firma oder Einrichtung öffentlichen Rechts zur Debatte steht, welche Software auf den jeweiligen Computern zukünftig betrieben werden soll. Erfüllt ein System die Erwartungen, besteht sicher kein Grund, neue Software einzusetzen. Auch bei Open Source Software können teilweise erhebliche Kosten anfallen, da Installation und Wartung zwar selbst durchgeführt werden können, das benötigte Know How dafür jedoch in der Regel nicht vorhanden sein wird. Darüber hinaus fallen Einschulungskosten an, um Mitarbeitern die für den Einsatz nötigen Informationen zu vermitteln. Während Behörden meist aufgrund leerer Haushaltskassen bei Anschaffung, Installation, Wartung und Einschulung auf die Kosten achten müssen, können Firmen höhere Anschaffungskosten nicht nur meist steuerlich absetzen, sondern diese auch damit rechtfertigen, dass sie einen besseren praktischen Nutzen aus der teureren Software ziehen könnten.

Die Entscheidung für ein neues System sollte gut überdacht werden. Bei diesen Überlegungen wird Open Source Software immer interessanter. Großstädte wie München, Paris und Wien⁷⁴² haben sich dazu entschlossen, ihre Computer auf Linux umzustellen. Aus diesem Grund wird eine vergaberechtliche Betrachtung wichtig, um beurteilen zu können, ob die Entscheidungsmöglichkeit zwischen Open Source und proprietärer Software in einem rechtmäßigen Verfahren getroffen werden kann und in welchem Fall der Einsatz von Open Source Software in einer Behörde vergaberechtliche Relevanz hat.

⁷⁴² Siehe <http://www.wien.gv.at/ma14/wienux.html>; anschaulich auch *Lutz*, WIENUX und OpenOffice – Die freiwillige Migration im Magistrat der Stadt Wien, in *Lutterbeck/Bärwolff/Gehring*, Open Source Jahrbuch 2006 (2006) 73 ff.

Das Vergaberecht macht einem öffentlichen Auftraggeber strikte Vorschriften, welche sicherstellen sollen, dass alle Interessenten gleich behandelt und die Auswahl des Anbieters nach objektiven und nachvollziehbaren Kriterien getroffen werden⁷⁴³. In Österreich legt das BVergG 2006 eine einheitliche gesetzliche Grundlage vor, die nachfolgend kurz skizziert wird.

8.1. Gemeinschaftsrechtliche Grundlagen

Aufgrund seiner wirtschaftlichen Bedeutung unterliegt das Vergaberecht strengen europarechtlichen Vorgaben. Das Interesse der EU, für alle Mitgliedstaaten gleichermaßen bindende Regelungen aufzustellen, scheint nachvollziehbar, wenn man bedenkt, dass die öffentlichen Beschaffungsmärkte der EU ungefähr 1 Billion Euro betragen⁷⁴⁴, wobei das Auftragsvolumen Österreichs bei ca 35 Milliarden Euro liegt. Da diese hohen Beträge dazu geeignet sind, Monopolstellungen von Firmen zu begründen oder zu verfestigen – die Auftragsvolumen können, wenn man das Gesamtvolumen betrachtet durchaus volkswirtschaftlichen Einfluss haben – versucht die EU schon seit den 1970er Jahren die Auftragsvergabe durch Richtlinien zu reglementieren, um dadurch die Verwirklichung des Binnenmarktes iSd Art 14 EG sicherstellen zu können⁷⁴⁵.

⁷⁴³ *Steiner/Windisch*, Einführung in das Vergaberecht (2004) 13.

⁷⁴⁴ Die Angaben schwanken hier sehr stark. Während *Kaufmann/Schnabl*, Vergaberecht in der Praxis (2004) 11, von einem Gesamtauftragsvolumen von 700 Milliarden Euro ausgehen, schätzen *Holoubek/Fuchs/Weinhandl*, Vergaberecht (2005) 17, dieses auf immerhin 1500 Milliarden.

⁷⁴⁵ *Holoubek/Fuchs/Weinhandl*, Vergaberecht 18.

Die EU hat zwar letztlich davon abgesehen, eine explizite Regelung des Vergaberechts im EG vorzunehmen⁷⁴⁶, dennoch sind dem gemeinschaftsrechtlichen Primärrecht die Grundlagen zu entnehmen, welche für das Vergaberecht von essentieller Bedeutung sind. Vorrangig sind das allgemeine Diskriminierungsverbot iSd Art 12 EG, das Gebot der Warenverkehrsfreiheit iSd Art 28 ff EG, der Niederlassungsfreiheit iSd Art 43 ff EG und der Dienstleistungsfreiheit iSd Art 49 ff EG zu nennen.⁷⁴⁷ Auch der EuGH hat in seiner Rsp in Zusammenhang mit dem Vergaberecht auf die Grundregeln des EG im Allgemeinen und dem Verbot der Diskriminierung aus Gründen der Staatsangehörigkeit im Besonderen verwiesen⁷⁴⁸.

Die nähere Ausgestaltung des Vergaberechts obliegt daher dem sekundären Gemeinschaftsrecht. Derzeit gibt es zwei "materielle" Vergaberichtlinien, die durch Normierung von nachprüfbaren Verpflichtungen für den Auftraggeber gewährleisten sollen, dass es hinsichtlich der Bewerber und Bieter zur Gleichbehandlung kommt. Diese sind:

1. Die „Klassischen Vergabe-Richtlinie“ (2004/18/EG) und
2. die „Sektorenrichtlinie“ (2004/17/EG).

Mit Datum vom 1.2.2006 haben die oben genannten die bis dahin geltenden Richtlinien – Dienstleistungskoordinerungs-Richtlinie (92/50/EWG), Lieferkordinierungs-Richtlinie (93/36/EWG), Baukordinierungs-Richtlinie (93/37/EWG) sowie Sektoren-Richtlinie (93/38/EWG) – abgelöst.

Darüber hinaus bestehen zur Verstärkung dieser Richtlinien zwei Rechtsmittelrichtlinien, die als Instrumentarium für eine schnelle und wirksame Nachprüfung von Vergabeentscheidungen dienen und eine effektive

⁷⁴⁶ Der Begriff „öffentliches Auftragswesen“ wird zwar in Art 163 Abs 4 EG erwähnt, enthält aber keine weiteren Anordnungen. Siehe dazu auch: *Estermann in Heid/Preslmayr* (Hrsg), *Handbuch Vergaberecht*² (2005) 2.

⁷⁴⁷ *Elsner/Keisler/Hahn*, *Vergaberechtsschutz in Österreich* (2004) Rz 2.

⁷⁴⁸ Siehe: EuGH 07.12.2000, Rs C-324/98 (*Teleaustria*); EuGH 30.05.2002, Rs C-358/00 (*Buchhändler*); EuGH 23.01.2003, Rs C-57/01 (*Makedoniko*) Slg. 2003, I-1091.

Durchsetzung der materiellen Vergaberechtsrichtlinien unterstützen sollen⁷⁴⁹.

Diese sind:

1. Die „Rechtsmittelrichtlinie“ (89/665/EWG) und
2. die „Sektorenrechtsmittel-Richtlinie (92/13/EWG)

Diese gelangen allerdings nur dann zur Anwendung, wenn die Voraussetzungen des persönlichen und sachlichen Geltungsbereichs einer der materiellen Vergaberechtsrichtlinien erfüllt sind.⁷⁵⁰

Besonders beachtet werden muss die mit 1. Jänner 2006 in Kraft getretene Verordnung Nr. 2083/2005. Durch diesen unmittelbar geltenden Rechtsakt wurden die Schwellenwerte des BVergG 2006 adaptiert⁷⁵¹.

8.2. Grundlagen des österreichischen Rechts

Das österreichische Vergaberecht wird durch das Bundesvergabegesetz (BVergG) geregelt, das erstmals mit dem BVergG 1993⁷⁵² auf Bundesebene in Kraft trat und welches zuletzt durch das BVergG 2006⁷⁵³ die Vorgaben der bereits erwähnten Richtlinien umgesetzt hat. Die nachfolgenden Ausführungen beziehen sich direkt auf die Auswirkungen der gesetzlichen Vorgaben auf den Open Source Bereich und erörtern jeweils die rechtlichen Fragestellungen.

⁷⁴⁹ *Holoubek/Fuchs/Weinhandl*, Vergaberecht 23.

⁷⁵⁰ Siehe dazu ausführlich *Elsner/Keisler/Hahn*, Vergaberechtsschutz in Österreich Rz 6.

⁷⁵¹ Zur vereinfachten Übersicht siehe <http://www.bva.gv.at/BVA/Rechtsgrundlagen/EURichtlinien/default.htm>.

⁷⁵² BGBl 1993/462.

⁷⁵³ BGBl I 2006/17.

8.2.1. Das Problem des persönlichen und sachlichen Geltungsbereichs nach dem BVergG in Bezug auf Open Source Software

Das BVergG definiert in § 3 BVergG den persönlichen Geltungsbereich, durch welchen festgelegt wird, welche öffentlichen Auftraggeber dazu verpflichtet sind, im Zuge ihrer Beschaffungstätigkeit auf die Bestimmungen des BVergG Rücksicht zu nehmen. Auf die Vielzahl der Möglichkeiten öffentlicher Auftraggeber auf den persönlichen Geltungsbereich soll an dieser Stelle nicht näher eingegangen werden, da dies zu erörtern aus dem Rahmen dieser Arbeit fallen würde. Dennoch muss in Bezug auf Open Source Software überprüft werden, ob das BVergG beim Erwerb zur Anwendung kommen kann⁷⁵⁴. Da der persönliche Geltungsbereich nur mögliche Lizenznehmer, bzw Erwerber behandelt und Open Source Lizenzen niemanden von der Nutzung ausschließen, muss im weiteren Verlauf der sachliche Geltungsbereich betrachtet werden.

Eine Übersicht für den sachlichen Geltungsbereich des BVergG findet man in § 1 BVergG, der die einzelnen Auftragsvergaben benennt. Eine genauere Beschreibung dieser Auftragsarten nimmt das BVergG dann in den §§ 4 - 9 BVergG vor, wobei für Open Source Software nur Lieferaufträge iSd § 5 BVergG, Dienstleistungsaufträge iSd § 6 BVergG und Dienstleistungskonzessionsverträge zur Anwendung kommen können. Da für jede Art der im BVergG genannten Aufträge besondere Bestimmungen gelten, muss eine Abgrenzung der einzelnen Arten voneinander vorgenommen werden, um einen möglichen Erwerbsvorgang der Open Source Software zuordnen und auf seine Rechtmäßigkeit überprüfen zu können.

⁷⁵⁴ Zur näheren Betrachtung des persönlichen Geltungsbereiches siehe bspw *Holoubek/Fuchs/Weinhandl*, Vergaberecht 37 ff.

8.2.2. Die Einordnung von Open Source Software unter den sachlichen Geltungsbereich

Bezüglich des Erwerbs von Open Source Software ist eine Abgrenzung hauptsächlich zwischen einem Lieferauftrag iSd § 5 BVergG und einem Dienstleistungsauftrag iSd § 6 BVergG vorzunehmen. Während Lieferaufträge gem § 5 BVergG entgeltliche Aufträge sind, deren Vertragsgegenstand Kauf, Ratenkauf, Miete, Pacht oder Leasing – mit oder ohne Kaufoption – einschließlich der Nebenarbeiten beinhaltet, sind Dienstleistungsaufträge gem § 6 BVergG entgeltliche Aufträge, deren Vertragsgegenstand Dienstleistungen iSd Anhänge III BVergG (prioritäre Dienstleistungsaufträge) und IV BVergG (nichtprioritäre Dienstleistungsaufträge) sind. Die Abgrenzung zwischen beiden Arten legt das BVergG selbst in § 9 Abs 1 BVergG fest, wonach ein Auftrag, der beide Elemente umfasst, dann als Dienstleistungsauftrag zu qualifizieren ist, wenn der Wert der vom Auftrag umfassten Dienstleistungen über dem Gesamtwert der zu liefernden Waren liegt. Liegt der Wert darunter, ist e contrario § 9 Abs 1 BVergG der Vertrag als Lieferauftrag zu sehen.

Im Falle von Software ist daher festzuhalten, dass ein Lieferauftrag angenommen werden muss, wenn der Wert der Lieferungen – dazu zählen neben der Hard- und Software auch der Einbau und die Installation⁷⁵⁵ - höher als der Wert der „immateriellen Leistungen“⁷⁵⁶ ist. Der Erwerb von Open Source Software vom Distributor oder im Einzelhandel wird, sollte er vergaberechtlich relevant sein, daher als Lieferauftrag anzusehen sein, wenn mit diesem nicht weiterführende Dienstleistungen vereinbart werden.

Die prioritären Dienstleistungen werden in Anhang III BVergG taxativ aufgezählt und sehen in Kategorie 7 vor, dass „*Datenverarbeitung und*

⁷⁵⁵ Heid, Geltungsbereich des Bundesvergabegesetzes, in *Heid/Preslmayr* (Hrsg), Handbuch Vergaberecht² (2005) 58.

⁷⁵⁶ EuGH 19.04.1994, Rs C-331/92 (*Gestion Hotelera Internacional*) Slg 1994, I-1329.

verbundene Tätigkeiten“ als Dienstleistungen zu qualifizieren sind. Kategorie 7 verweist zu einer genaueren Einordnung auf die Gütersystematik der Central Product Classification (CPC), deren Referenznummer 84 die Bezeichnung „*Computer and related services*“ trägt. Danach umfasst der Begriff der Dienstleistung die Wartung der Hardware, die Entwicklung und Implementierung von Software – dazu zählen Beratung, Analyse, Design, Programmierung und Wartung der Software – und Datenbankdienste⁷⁵⁷.

Das Bundesvergabeamt (BVA) hat im Falle der Einführung des „*elektronischen Bibliothekenverbundsystemes*“⁷⁵⁸ die erworbenen Leistungen dem Dienstleistungsauftrag zugeordnet, da das BVA der Ansicht war, dass die benötigte Anpassung der Software über das übliche Maß einer Installation hinausgehen würde. Für Open Source Software ist dieses Merkmal der Anpassung bzw Weiterentwicklung wesentlich, kommt der freien Verfügbarkeit des Quellcodes doch eine herausragende Stellung zu, die bei der Auswahl der Anforderungskriterien der Software oft hoch bewertet wird. Im Bereich der Open Source Standardsoftware (wie bspw Apache, Linux und MySQL), deren Erwerb im Rahmen des Vergaberechts als Lieferauftrag anzusehen ist, muss daher darauf geachtet werden, ob mit der Auftragsvergabe Aufgaben verbunden sind, die über das übliche Ausmaß eines Systemerwerbs hinausgehen. Im Fall von Open Source Individualsoftware wird man in der Regel im Rahmen des Vergaberechts von einem Dienstleistungsauftrag ausgehen können, da hierbei die „*immateriellen Leistungen*“ überwiegen und es meist über die Installation hinaus zu weiter reichenden Einschulungen bezüglich Nutzung und Wartung kommen wird.

⁷⁵⁷ Siehe dazu CPC 84 (bspw unter <http://www.census.gov/eos/www/napcs/papers/cpcintro.pdf>), sowie: *Heid in Heid/Preslmayr* (Hrsg), Handbuch Vergaberecht², 58 f.

⁷⁵⁸ BVA 01.12.1999, F-33/97-24 = TCA Newsletter 2000/6 7.

8.2.3. Die Einordnung der Entgeltlichkeit im Vergaberecht in Bezug auf Open Source Software

Das Vergaberecht knüpft bei den Definitionen der verschiedenen Auftragsmöglichkeiten an die Entgeltlichkeit an⁷⁵⁹, weshalb der Erwerb von Open Source Software nicht zwingend vergaberechtlich relevant sein muss. Sollte Open Source Software kostenlos bezogen werden – wie es bspw bei Browsern wie Opera oder dem Mozilla Firefox der Fall ist – kann es zu keinem Erwerb im vergaberechtlichen Sinn kommen⁷⁶⁰. In diesem Fall ist auch die Schenkung der Lizenzrechte unerheblich, da die Software kostenlos bezogen wird und damit eine weitere vergaberechtliche Prüfung unterbleiben kann. So kann das Herunterladen und Installieren von Open Source Software durch eine Behörde zur Benutzung demnach ohne Ausschreibung auch in großem Ausmaß durchgeführt werden.

Ein unter das Vergaberecht fallender Erwerbsvorgang von Open Source Software oder ein Auftragsverhältnis mit Open Source Bezug ist dann denkbar, wenn einerseits die Entgeltlichkeit gegeben ist – bspw durch Bezug der Open Source Software vom Distributor oder Anpassung und Einschulung auf (eventuell sogar kostenlos) bezogene Open Source Software – und andererseits die Schwellenwerte iSd § 12 BVergG iVm § 37 BVergG erreicht werden. Dieser untere Schwellenwert beträgt für Liefer- und Dienstleistungsaufträge gem § 37 Punkt 2 BVergG momentan 80.000 Euro, weshalb nur umfangreiche Migrationen vergaberechtlich relevant sein werden. Beim kostenlosen Erwerb ist anzumerken, dass eine Eignungsprüfung iSd §§ 69ff BVergG stattfinden kann, da eine Wirtschaftlichkeitsberechnung trotz eventueller Kostenlosigkeit der Software selbst eine vergaberechtliche Ausschreibung notwendig machen kann. Dabei wird es dann aufgrund der vorher erwähnten Überwiegensregel normaler Weise zu

⁷⁵⁹ *Holoubek/Fuchs/Weinhandl*, Vergaberecht 60.

⁷⁶⁰ *Müller/Gerlach*, Open Source-Software und Vergaberecht, CR 2/2005, 87 (88).

Dienstleistungsaufträgen kommen, die für den Aufbau der Systeminfrastruktur zu vergeben sind. Für die Implementation von „WIENUX“ bspw wurden über 1 Million Euro an internen Kosten berechnet⁷⁶¹ und es scheint nachvollziehbar, dass in den überwiegenden Fällen die Anpassung der Software und die Einschulung der Mitarbeiter nicht ausschließlich In-House durchgeführt werden kann.

Die zeitliche Trennung der Rechtsgeschäfte beim Erwerbsvorgang⁷⁶² muss unterschiedlich beurteilt werden⁷⁶³, dennoch wird erneut darauf hingewiesen, dass der Erwerb der Open Source Software mit dem Abschluss eines Open Source Lizenzvertrags zusammenfallen kann, wenn dies so beabsichtigt war. Beide Rechtsgeschäfte können zeitlich auseinander fallen oder es kommt – wenn auch beim behördlichen Einsatz eher unwahrscheinlich – nie zum Abschluss eines Open Source Lizenzvertrags. Bei der Bewertung des Erwerbsvorgangs kann es nur dann zu einer einheitlichen Betrachtung kommen, wenn eine Weiterentwicklung bzw Anpassung bei der ursprünglichen Vergabe beabsichtigt war und dies in der Ausschreibung berücksichtigt wurde⁷⁶⁴. Diese Lösung lässt sich vergaberechtlich dadurch rechtfertigen, dass das BVergG nicht nur eine Auftragsart kennt, sondern die Möglichkeit zur Auswahl bietet. Der Auftraggeber kann dadurch die Rechtsgeschäfte schon bei der Ausschreibung trennen. Für den Erwerbsvorgang selbst kann er einen Lieferauftrag ausschreiben, für eventuelle Anpassungsarbeiten einen Dienstleistungsauftrag. In der Praxis wird es zwar meist zu einer einheitlichen Ausschreibung kommen, dennoch muss diese Möglichkeit der Trennung erlaubt sein. Gesetzlich findet eine Trennung der Aufträge seine vergaberechtliche Legitimation in § 22 Abs 1 BVergG.

⁷⁶¹ Lutz in *Lutterbeck/Bärwolff/Gehring*, Open Source Jahrbuch 2006, 76.

⁷⁶² Vgl Kap 6.

⁷⁶³ Für eine einheitliche Sichtweise *Heckmann*, IT-Vergabe, Open Source Software und Vergaberecht, CR 6/2004, 401 (402); differenzierend *Müller/Gerlach*, CR 2/2005, 88.

⁷⁶⁴ *Müller/Gerlach*, CR 2/2005, 88.

Die Tatsache, dass eine solche Trennung bei der Ausschreibung zur Ungleichbehandlung mit proprietärer Software führen kann, zeigt erneut die Flexibilität von Open Source Software gegenüber proprietärer Software. Bei letzterer wird es meist nicht möglich sein, eine Trennung der Rechtsgeschäfte vorzunehmen, da eine benötigte Anpassung nur durch den Lizenzgeber (der überwiegend auch der Veräußerer ist) selbst möglich ist. Dritte dürfen nur im Rahmen des § 40e UrhG den Quellcode einsehen und werden über die gesetzlich eingeräumten Rechte des § 40d UrhG hinaus Bewilligungen zur Weiterentwicklung des Programms benötigen, welche ihnen der Lizenzgeber des proprietären Programms aus verständlichen Gründen sicher nicht erteilen wird. Letztlich muss eine einheitliche Vergabe dahingehend kritisiert werden, mit einer einheitlichen Leistungsanforderung gerade kleine und mittelständige Unternehmen von einem Auftrag fernzuhalten⁷⁶⁵, da sie deren Leistungspotential überfordern kann.

8.3. Die Grundsätze des Vergabeverfahrens in Bezug auf Open Source Software

Auch wenn die europarechtlichen Grundlagen, die das BVergG beeinflussen, für ein hohes Maß an Regelungsdichte gesorgt haben, stellt das BVergG in den §§ 19 - 24 BVergG Grundsätze und allgemeine Bestimmungen für das Vergabeverfahren auf, die immer dann herangezogen werden, wenn es zu unregelmäßigten Vergabesituationen kommt. Auf diese Grundsätze wird hier nur kurz eingegangen, um die in Bezug auf Open Source Software relevanten Kriterien darzustellen.

⁷⁶⁵ Müller/Gerlach, CR 2/2005, 89 mwA.

Folgende Vergabegrundsätze sind dabei als vorrangig zu betrachten:

1. Freier, lauterer und fairer Wettbewerb
2. Gleichbehandlung aller Bewerber und Bieter
3. Vergabe an befugte und leistungsfähige Bieter
4. Vergabe zu angemessenen Preisen

8.3.1. Open Source Software und freier, lauterer und fairer Wettbewerb

Am Maßstab der Grundsätze des freien, lauterer und fairen Wettbewerbs sind die Entscheidungen des Auftraggebers und die korrespondierenden Verhaltensweisen der Bieter – auch untereinander – zu bewerten⁷⁶⁶. Das Gebot des freien Wettbewerbs iSd § 19 BVergG bezieht sich dabei auf jegliche Form unzulässiger Zugangs- oder Ausübungsbeschränkungen. Damit sind sowohl gebietsmäßige Beschränkungen als auch Beschränkungen auf einzelne Berufsstände unzulässig⁷⁶⁷. In Bezug auf Open Source Software sind hier einige Problemstellungen denkbar. Die Forderung nach einer bestimmten Zertifizierung des Anbieters – bspw einer Microsoft- oder SAP-Zertifizierung – muss als unzulässige Beschränkung angesehen werden. Ebenso kann eine eng gefasste Leistungsbeschreibung – die Bildschirmauflösung eines Notebook wird bspw so gefordert, dass diese Anforderung nur von einem bestimmten Hersteller oder Betriebssystem erfüllt werden kann – zu einer unzulässigen Verengung führen. Die Forderung, nur Software einsetzen zu wollen, bei dem der Quellcode eingesehen werden kann, muss jedenfalls begründet werden. Dies kann dadurch gerechtfertigt werden, dass eine Behörde eine eigene Entwicklungsabteilung hat, welche die Software für den eigenen Bedarf

⁷⁶⁶ *Holoubek/Fuchs/Weinhandl*, Vergaberecht 77.

⁷⁶⁷ *Fink/Heid*, Ablauf des Vergabeverfahrens, in *Heid/Preslmayr* (Hrsg), Handbuch Vergaberecht², 178.

anpassen und weiterentwickeln wird. Diese Anforderung führt mE nicht automatisch zu einer Verzerrung des Wettbewerbs, da es auch einem Anbieter von proprietärer Software möglich ist, den Quellcode zwar nicht allgemein, aber für die Behörde offen zu legen und die Offenlegung mittels einer Geheimhaltungsvereinbarung zu schützen. Wird in der Ausschreibung aber bspw ein „Linux-Betriebssystem“⁷⁶⁸ gefordert, schließt dies Anbieter von proprietären Betriebssystemen oder anderen Open Source Betriebssystemen aus und verstößt daher gegen das Gebot des fairen Wettbewerbs iSd § 19 BVergG.

Der lautere Wettbewerb zielt auf das Verhältnis unter den Bewerbern ab⁷⁶⁹. Dieses Gebot ist auf dem Softwaremarkt besonders zu überwachen, da einzelne Softwarehersteller durch ihre Monopolstellung versuchen könnten, einen ungerechtfertigten Wettbewerbsvorteil auszunutzen. Ein solcher Versuch kann gem § 168b StGB sogar strafrechtliche Konsequenzen nach sich ziehen. Preisabsprachen unter den Anbietern sind ebenso unzulässig, wobei diese im Softwarebereich kaum vorkommen werden, da die Unterschiedlichkeit der Leistungsmerkmale der einzelnen Systeme dies in der Regel verhindern wird. Der faire Wettbewerb letztendlich betrifft das Verhältnis vom Auftraggeber zu seinen Bewerbern. Dieser darf keinem Bewerber – bspw durch Informationsaustausch – einen Wettbewerbsvorteil verschaffen.

8.3.2. Open Source Software und die Gleichbehandlung aller Bewerber und Bieter

Das zu Beginn des Kapitels erwähnte gemeinschaftsrechtliche Diskriminierungsverbot findet im Grundsatz der Gleichbehandlung aller Bieter

⁷⁶⁸ Heckmann, Vergaberecht in *Spindler*, Rechtsfragen bei Open Source 293.

⁷⁶⁹ Fink/Heid in *Heid/Preslmayr* (Hrsg), Handbuch Vergaberecht², 178.

und Bewerber iSd § 19 BVergG seine vergaberechtliche Berücksichtigung⁷⁷⁰. Der unabhängig von der Auftragshöhe geltende Grundsatz ist nicht direkt im Gesetz zu finden, wurde aber durch die Rsp des EuGH⁷⁷¹ als wesentlich betont. Bieter sollen daher die gleichen Chancen auf den Auftrag geboten bekommen. Im Fall von Open Source Software, als geistig-schöpferischer Dienstleistung⁷⁷², hat das BVA festgehalten, dass es zur Bewertung der Leistungsspezifikation zwingend zu Verhandlungen kommen muss⁷⁷³. Diese Vorgabe ist im IT-Bereich sinnvoll, da die Leistungsbeschreibung dadurch im Vorfeld gemeinsam erörtert werden kann. Der Bieter hat die Chance, seinerseits Vorschläge zu unterbreiten, welche im Zuge einer Nachverhandlung nicht mehr berücksichtigt werden können, da Nachbesserungsmöglichkeiten ausdrücklich verboten sind und sogar zur Nichtigerklärung der Zuschlagserteilung führen können⁷⁷⁴. Das Nachverhandlungsverbot macht insofern Sinn, da alle Anbieter ihre Angebote „blind“ abgeben müssen, wodurch jedoch die Gefahr von Preisabsprachen steigt. Die Gewährleistung der Gleichbehandlung wird durch Verpflichtung des Auftraggebers zur Transparenz erreicht, welche in engem Zusammenhang mit dem Wettbewerbsverbot steht. Der EuGH definiert dies, indem er festlegt, „*dass die Zuschlagskriterien in den Verdingungsunterlagen⁷⁷⁵ oder in der Bekanntmachung so gefasst werden müssen, dass alle durchschnittlich fachkundigen Bieter sie bei der Anwendung der üblichen Sorgfalt in gleicher Weise auslegen können.*“⁷⁷⁶ Diese Bestimmung ist für den ganzen technischen Bereich von großer Bedeutung. Die objektive Überprüfung der Einhaltung des Transparenzgebots wird durch diese Vorgabe erheblich erleichtert, da eine Ausschreibung keine Vorgaben enthalten darf, die nur einem bestimmten

⁷⁷⁰ Kaufmann/Schnabl, Vergaberecht in der Praxis 54; Holoubek/Fuchs/Weinhandl, Vergaberecht 79.

⁷⁷¹ EuGH 17.09.2002, Rs C-513/99 (*Concordia Bus*), Slg 2002, I-07213.

⁷⁷² Heid in *Heid/Preslmayr* (Hrsg), Handbuch Vergaberecht², 178 mwA.

⁷⁷³ Siehe *Fink/Heid* in *Heid/Preslmayr* (Hrsg), Handbuch Vergaberecht², 178 mwA.

⁷⁷⁴ Siehe bspw: BVA 07.11.2003, 14N-91/03-29 (A2 Tunnelkette Pack).

⁷⁷⁵ Die Verdingungsunterlagen – diese bestehen aus der Leistungsbeschreibung und den allgemeinen und besonderen Vertragsbedingungen für die Leistungsdurchführung – sind ein Teil der Vergabeunterlagen, die zusätzlich noch aus dem Anschreiben bestehen.

⁷⁷⁶ EuGH 18.10.2001, Rs C-19/00 (*SIAC*), Slg 2001, I-07725.

Verkehrskreis verständlich sind. Ebenso dürfen keine diskriminierenden technischen Spezifikationen iSd § 75 BVerG gefordert werden. Auf diese Problematik wird nachfolgend, im Zuge der Leistungsspezifikationen, gesondert eingegangen. Der Grundsatz der Gleichbehandlung stellt sich als einer der wichtigsten dar, seine Einhaltung indes wird schwer zu überprüfen sein, da die Möglichkeiten der Ungleichbehandlung nahezu grenzenlos zu sein scheinen und die verschiedenen Arten von Software es nicht ermöglichen, absolute Gesamtbeurteilungskriterien aufzustellen.

8.3.3. Open Source Software und die Vergabe an befugte und leistungsfähige Bieter

Dieser Grundsatz dient dem öffentlichen Auftraggeber als Schutzinstrument. Er kann danach Bewerber ablehnen, die vorgeben die geforderten Leistungen zu erbringen, deren Befähigung jedoch angezweifelt werden darf. Die §§ 69 – 77 BVerG geben dem Auftraggeber die Möglichkeit, die Befugnis, Leistungsfähigkeit und Zuverlässigkeit eines Bewerbers durch Referenzen nachweisen zu lassen und ihn nach dieser Eignungsüberprüfung gegebenenfalls vom Vergabeverfahren auszuschließen. Einem Distributor von Software muss demnach zugetraut werden, die Leistungsfähigkeit der Hardware gewährleisten zu können. Einem Einzelunternehmer bspw wird es unmöglich sein, den Aufbau und Erhalt eines Netzwerkes durchführen zu können, mit dem in der Regel 40 Personen betraut sind. Ein Bewerber muss demnach vor Einreichung seines Angebots überzeugt sein, den Auftrag nachvollziehbar erfüllen zu können. Dieser Vorbehalt ist eine hohe Hürde für die vielen KMUs im Open Source Bereich. Um die erforderlichen Nachweise auch für größere Aufträge erbringen zu können, haben sich einige Open Source Dienstleister zur OSalliance⁷⁷⁷ zusammengeschlossen. Die OSalliance kann nach außen hin als Genossenschaft auftreten und gegebenenfalls an

⁷⁷⁷ <http://osalliance.com/kooperation/allianz>.

Vergabeverfahren teilnehmen. Dem Auftraggeber steht es dennoch frei, den Nachweis der Eignung der Genossenschaftsmitglieder einzufordern und auf Ausschlusskriterien hin zu überprüfen.

8.3.4. Open Source Software und die Vergabe zu einem angemessenen Preis

Der Grundsatz der Vergabe zu einem angemessenen Preis soll den Auftraggeber dazu bewegen, vergleichbare Preisangebote einzuholen und die relevanten Marktverhältnisse zu überprüfen⁷⁷⁸. Einen Bewerber schützt dieser Grundsatz vor „Preisdumping“, da ein möglicher Dumpingpreis eines Mitbewerbers zu einer vertiefenden Angebotsprüfung führt⁷⁷⁹. Ein Bewerber, der auf die kostenlose Lizenzierung seiner Open Source Software im Gegensatz zu einem Mitanbieter proprietärer Software verweist, wird nach diesem Grundsatz vorgehalten werden müssen, dass der kostenlose Erhalt der Lizenzgebühren nicht allein der ausschlaggebende Marktpreis sein wird. Eventuell durchzuführende Nebenleistungen wie Installation und Einschulung müssen ja ebenfalls entgeltlich abgegolten werden, weshalb die gesamte Preiskalkulation der einzelnen Anbieter zu bewerten ist und nicht nur einzelne Posten einander gegenübergestellt werden dürfen. Ein Problem für die Open Source Software ergibt sich aus den bisher nur wenig vorhandenen Open Source Anwendungen im öffentlichen Sektor. Ein Vergleich hinsichtlich der Eignung, Nachhaltigkeit und Kosteneffizienz ist daher nur schwer bis gar nicht möglich wie *Heckmann*⁷⁸⁰ richtig ausführt. Sollten diese Kriterien nicht zu ermitteln sein, weil bspw keine Open Source Software beim Auftraggeber genutzt wird, darf dies mE nicht dazu führen, dass ein Anbieter von Open Source Software vom Verfahren ausgeschlossen wird. Der Vergleich muss

⁷⁷⁸ *Holoubek/Fuchs/Weinhandl*, Vergaberecht 83.

⁷⁷⁹ *Holoubek/Fuchs/Weinhandl*, Vergaberecht 118.

⁷⁸⁰ *Heckmann* in *Spindler*, Rechtsfragen bei Open Source 296.

dann gegebenenfalls andere Vergleichsmittel heranziehen – bspw Einsatz der Software im relevanten Marktsegment, Referenzinstallationen bei Dritten – die objektiv dazu beitragen können, die Preisgestaltung gegenüberzustellen.

8.4. Die Leistungsbeschreibung und die technische Spezifikation in Bezug auf Open Source Software

Das BVergG gibt dem Auftraggeber die Möglichkeit, die Leistungsbeschreibung gem § 95 Abs 1 BVergG konstruktiv oder funktional auszuschreiben. Die funktionale Leistungsbeschreibung, bei der die Leistungen gem § 95 Abs 3 BVergG als Aufgabenstellungen durch die Festlegung von Leistungs- oder Funktionsanforderungen beschrieben werden, sind nach hL nur dann zulässig, wenn eine detaillierte Leistungsbeschreibung nicht möglich ist⁷⁸¹. Da diese Meinung keine gesetzliche Deckung findet, wird die Wahl der Art der Leistungsbeschreibung im Ermessen des Auftraggebers liegen, wobei er sich an den im Rahmen der in § 96 BVergG normierten Grundsätze für die Leistungsbeschreibung orientieren muss. Die funktionale Leistungsbeschreibung ist für den Softwarebereich von großer Bedeutung, da der Auftraggeber in der Beschreibung die gewünschten Funktionen beschreiben kann, dem Bewerber aber selbst überlassen bleibt, wie er diese Aufgabe erfüllen will. Die konstruktive oder konkrete Leistungsbeschreibung dagegen erleichtert den Vergleich zwischen verschiedenen Angeboten und dokumentiert meist eine klare Vorstellung des Auftraggebers über das zu erwartende Ergebnis. Beide Leistungsbeschreibungen finden sich im Softwarebereich im Lastenheft, welches ebenfalls die gewünschten Anforderungen enthält, dabei jedoch die Umsetzung – welche im Pflichtenheft abgedeckt wird – dem Auftragnehmer überlässt. Die Leistungsbeschreibung für Software wird in der Regel funktional geschehen, es ist jedoch durchaus rechtmäßig, Teile der Beschreibung – wie bspw Anforderungen an Schnittstellen – konstruktiv zu

⁷⁸¹ *Estermann in Heid/Preslmayr (Hrsg), Handbuch Vergaberecht², 270 mwA.*

gestalten. Die Leistungsbeschreibung insgesamt muss gem § 96 Abs 1 BVergG eindeutig, vollständig und neutral sein. Die technischen Spezifikationen iSd § 98 BVergG müssen ebenfalls Bestandteil der Leistungsbeschreibung sein, daher kommt ihnen in Bezug auf Open Source Software eine wichtige Bedeutung zu.

Die technischen Spezifikationen iSd § 98 BVergG betreffen die technischen Anforderungen an einen in Bezug auf Softwareausschreibungen möglichen Liefer- oder Dienstleistungsauftrag. Diese beschreiben den Auftragsgegenstand genauer und sollen außerdem sicherstellen, dass dieser den vom Auftraggeber festgelegten Verwendungszweck erfüllt⁷⁸². Die Neutralität der Leistungsbeschreibung selbst zeigt sich meist in ihrer Beschreibung der technischen Spezifikationen. Marken-, typen- oder herkunftbezogene Angaben sind gem § 98 Abs 7 Satz 1 BVergG unzulässig. Bei Software ist diese Bestimmung besonders zu berücksichtigen, da hier Programme durch ihre vorherrschende Monopolstellung dazu verleiten können, im Sprachgebrauch der Leistungsbeschreibung stellvertretend zu erscheinen. Denkbar wäre bspw, dass „Windows“ als Betriebssystem, „SAP“ als Buchhaltungsapplikation oder „Apache“ als Serversoftware verwendet werden. Eine Ausnahme von dem Verbot marken-, typen- oder herkunftbezogene Angaben zu verwenden, besteht gem § 98 Abs 7 Satz 2 BVergG nur dann, wenn der Auftragsgegenstand nicht auf andere Weise durch hinreichend genaue und allgemein verständliche Spezifikationen beschrieben werden kann. Dabei muss die Beschreibung der Spezifikation mit dem Zusatz „*oder gleichwertiger Art*“ versehen werden. Im Softwarebereich ist ein solcher Fall schwer denkbar, da es inzwischen kein Programm mehr gibt, welchem eine solch herausragende Stellung ohne Alternative zukommt. Software wie bspw „Apache“ hat zwar einen herausragenden Bekanntheitsgrad aufgrund seiner Beliebtheit, dennoch sind Alternativen bekannt und es kann daher auf die Verwendung des Markennamens in einer Leistungsbeschreibung verzichtet werden.

⁷⁸² Estermann in Heid/Preslmayr (Hrsg), Handbuch Vergaberecht², 273.

Der EuGH hat bereits zu einem solchen Fall Stellung nehmen müssen, da ein niederländischer Auftraggeber bei der Ausschreibung für die Lieferung und Instandhaltung einer Wetterstation gefordert hatte, dass das Mehrbenutzer-Betriebssystem „UNIX“, welches bereits im Verbund der Wetterstationen eingesetzt wurde, auch in diesem Fall verwendet werden müsse⁷⁸³. Der EuGH entschied, dass diese Vorgabe einen Verstoß gegen das Vergaberecht darstelle, da die Frage der Kompatibilität der Teilleistung die Verwendung des Markennamens nicht rechtfertigen würde.

Die technischen Spezifikationen der Leistungsbeschreibung müssen möglichst offen gestaltet werden, um den Bewerbern die Chance zu geben, individuelle Lösungswege anbieten zu können. Bezüglich der Lizenzbestimmungen müssen die Anforderungen ebenfalls weit gefasst sein. Die Forderung nach einem bestimmten Lizenztyp ist vergaberechtlich unrechtmäßig wie die Forderung nach Nutzungs- bzw Verwertungsrechten, die nur einzelne Lizenzen einräumen. Die Forderung nach der Offenlegung des Quelltextes in der Leistungsbeschreibung wurde vergaberechtlich noch nicht durch die Judikatur behandelt⁷⁸⁴. Klar scheint jedoch, dass eine solche Forderung zu heftigen Diskussionen führen wird, wenn man bedenkt, dass Open Source Software hierdurch einen beträchtlichen Vorteil gegenüber proprietärer Software haben würde. Das Ansinnen des Auftraggebers, den Quelltext zu erhalten, muss in die Wirtschaftlichkeitsbetrachtung einbezogen werden, da die Möglichkeit, die Software nach eigenen Vorgaben weiterzuentwickeln, ein wichtiges Entscheidungskriterium bei der Auswahl des Anbieters sein kann. Die Forderung nach der Offenlegung scheint gerechtfertigt, wenn die Leistungsanforderung an die Software nur durch eine Anpassung oder Weiterentwicklung erreicht werden kann und die Kosten dafür bei einer Auftragsvergabe ohne Herausgabe des Quelltextes wesentlich teurer wären. Unterscheiden sich die Gesamtkosten hierfür kaum, wird die Forderung

⁷⁸³ EuGH 24.01.1995, Rs C-359/93, *UNIX*, Slg 1995, I-157.

⁷⁸⁴ Für eine Sichtweise der Lehre siehe bspw: *Müller/Gerlach*, CR 2/2005, 90 f.

nach Offenlegung des Quelltextes durch die Ausschreibung vergaberechtlich unrechtmäßig sein.

8.5. Ergebnis

Auch wenn sich die wesentlichen Änderungen, die das BVergG 2006 gebracht hat, auf die Festlegung neuer Schwellenwerte beziehen, so hatte der Gesetzgeber bereits in der Vergangenheit die Pflicht, seine vergaberechtlichen Bestimmungen hinsichtlich der Ausschreibungskriterien für Software zu überprüfen, um eine Chancengleichheit unter allen Anbietern gewährleisten zu können. Die Eigenständigkeit von Open Source- gegenüber proprietärer Software einerseits sowie das wachsende Interesse öffentlicher Einrichtungen zur Nutzung von Open Source Software andererseits stellen für das Vergaberecht Herausforderungen dar, die bis jetzt ohne große Schwierigkeiten gemeistert wurden. Gerade die Möglichkeit der Trennung der Ausschreibung iSd § 22 Abs 1 BVergG ist sinnvoll, berücksichtigt man die Vielzahl an Anbietern von Hard- und Software. Ein öffentlicher Auftraggeber kann dadurch nicht nur Kosten einsparen, da er die einzelnen Komponenten separat bewerten kann, er kann auch für beide Bereiche eine qualitative Auswahl treffen und so eine effizientere Erwerbspolitik betreiben.

Des Weiteren wird dies durch die Möglichkeit der funktionalen Ausschreibung unterstützt, die sehr allgemein gehalten werden kann, aber dennoch geeignet ist, das Know-How der Bieter zu überprüfen. Aus Sicht der Bieter ist eine solche Trennung ebenfalls als sinnvoll zu erachten, da Entwickler von Software nicht zwangsläufig auch Hardware anbieten und umgekehrt. Der Bieter kann sich dabei ganz auf seinen Wirkungsbereich konzentrieren, ohne Dritte mit in die Anbotslegung einbeziehen zu müssen.

Sollte eine Ausschreibung, welche den Erwerb von Hard- und Software betrifft, nicht getrennt werden, muss überprüft werden, ob einerseits dem Ausschreiber eine Trennung nicht zugemutet werden kann – dies wäre vom Ausschreiber näher zu begründen – und ob andererseits die Ausschreibung dadurch den Kreis der möglichen Anbieter nicht einengt. Die Gefahr einer einheitlichen Ausschreibung besteht in der Möglichkeit, die Anforderungskriterien so zu gestalten, dass es nur einem oder ganz wenigen Anbietern möglich sein wird, diese erfüllen zu können.

Open Source Software bietet durch die kostenlose Lizenzierung und die Möglichkeit, ein Programm In-House weiterentwickeln und anpassen zu können, einen großen Anreiz für öffentliche Einrichtungen wie der Umstieg der Stadt Wien auf „WIENUX“ beweist. Das Erfordernis einer Offenlegung des Quellcodes durch die Ausschreibung sollte nur in begründeten Fällen vergaberechtlich rechtmäßig sein. Auch für Anbieter proprietärer Software besteht die Möglichkeit, den Quelltext im Rahmen der Vergabe lizenzrechtlich geschützt weiterzugeben. Dies kann allerdings nur Anbietern von kleinen Standardprogrammen oder von Individualsoftware zugemutet werden. Von Microsoft bspw kann nicht erwartet werden, den Quellcode von „Windows“ herauszugeben, um an einer Ausschreibung teilnehmen zu können.

Der Anreiz sowohl für Unternehmen als auch für kleine Firmen, Aufträge öffentlicher Einrichtungen zu erhalten, ist sicher groß. Dabei muss das Vergaberecht die Gleichbehandlung aller Anbieter garantieren und das Interesse der Auftraggeber berücksichtigen, die Befähigung der Anbieter zu einer qualitativ hochwertigen Ausführung der Leistungsbeschreibung überprüfen zu können, da der Anreiz vor allem für kleine Firmen sehr groß ist, hier ein Angebot abzugeben, das nicht der eigentlichen Leistungsfähigkeit entspricht. Die Problematik des Nachverhandlungsverbots kann – wie ausgeführt – ebenfalls zu praktischen Problemen führen, da dieses in der IT-Branche durchaus üblich ist und im Rahmen der Vergleiche der Angebote Sinn macht.

Abschließend ist festzuhalten, dass der Einsatz von Open Source Software im Rahmen einer öffentlichen Vergabe immer wieder an Bedenken der öffentlichen Auftraggeber scheitert. Häufig werden Sicherheitsbedenken gegenüber Open Source Software angeführt, so wird bspw die Offenheit des Quellcodes mit der Offenlegung von sicherheitsrelevanten Merkmalen gleichgesetzt. *Heckmann*⁷⁸⁵ weist in diesem Zusammenhang darauf hin, dass es immer auf das Know-How des Auftraggebers ankommt, ob Sicherheitslücken zu befürchten sind. Ein Vorteil ist mE, dass bei ausreichendem Kenntnisstand, Umgang und ausreichender Weiterentwicklung was Open Source betrifft, Sicherheitslücken schnell selbst oder durch die Open Source Gemeinschaft behoben werden können. Bei proprietären Produkten besteht die Möglichkeit der schnellen Selbsthilfe dagegen nicht. Betriebssysteme und Anwendungsprogramme – wie Textverarbeitung und Tabellenkalkulation – können durch eine Lücke leicht global angegriffen werden, da die Software durch die unmögliche individuelle Anpassung stets gleich ist. Der Reiz, in ein geschlossenes System einzudringen und Schadcode auszuführen ist darüber hinaus viel höher, als bei Open Source Software, weshalb die Sicherheitsbedenken mindestens denen der proprietären Software gleichwertig sind.

Weitere Bedenken bestehen in der ungesicherten Nachhaltigkeit der Software, da geeignete Open Source Software selten durch eine dahinterstehende Firma betreut wird. Diese Problematik entschärft sich jedoch zunehmend, da viele kleine und mittelständische Softwaredienstleister die Betreuung und Anpassung von bestimmten Open Source Produkten anbieten und verstärkt die Projektleitung übernehmen. Der Auftraggeber muss daher nicht mehr auf die Open Source Gemeinde verwiesen werden, wenn er Probleme mit seinem eingesetzten Programm hat.

⁷⁸⁵ Siehe dazu auch: *Heckmann* in *Spindler*, Rechtsfragen bei Open Source 298.

Die Zukunft wird zeigen, ob die erfolgten Entscheidungen, Open Source Software in der öffentlichen Verwaltung vorrangig einzusetzen, richtig waren. Die oft politisch motivierten Wechsel waren nicht immer von Vorteil. In jenen Fällen, in denen die „neuen“ Nutzer von Open Source Software mit dem Wechsel nicht zufrieden waren, weil eine Migration zu schnell oder ohne ausreichende Schulung stattgefunden hat, wurde der Open Source Gemeinde ein „Bärendienst“ erwiesen. Die dabei festgestellte fehlende Nachhaltigkeit kann daher – ohne die Hintergründe des Scheiterns miteinzubeziehen – zu negativen Entscheidungen bei zukünftigen Vergabeverfahren führen. Für Auftraggeber bleibt zu bedenken, dass Erfahrungen aus vorangegangenen Vergabeverfahren von den Open Source Anbietern evaluiert wurden und versucht wird, fehlerhafte Prozesse zu korrigieren bzw für die Zukunft auszuschließen.

9. PATENTRECHT

“The software plutocracy has attempted to draw more borders around knowledge that would destroy Open Source using software patents.”⁷⁸⁶

- *Bruce Perens*

Bei der bloßen Erwähnung des Begriffs „Patentrecht“ geht stets ein Aufschrei durch die Open Source Gemeinde, begleitet von heftigen Diskussionen. Ursächlich hierfür ist ua, dass bspw das amerikanische Patentrecht seit der Entscheidung „*Diamond vs. Diehr and Lutton*“⁷⁸⁷ aus dem Jahr 1980 die Möglichkeit zulässt, Computerprogramme unter Patentrechtsschutz zu stellen. In Amerika kommt es deshalb – auch wegen des dort herrschenden „first-to-invent“-Prinzips⁷⁸⁸ – immer wieder zu Patentrechtsstreitigkeiten, die in der Regel sehr kostspielig sind und daher einen privaten Entwickler von Open Source Software finanziell ruinieren können⁷⁸⁹. Darüber hinaus stellt die Ausgestaltung des Patentrechts in Bezug auf Schutzgegenstand und –umfang (im Gegensatz zum Urheberrecht) eine große Gefahr für das gesamte Open Source Modell dar, sollte es dazu kommen, Computerprogrammen Patentschutz zu gewähren⁷⁹⁰.

Bereits Ende der 1990er Jahre begann in Europa die Debatte um den Patentschutz für Software und damit um die Richtlinie über "computerimplementierte Erfindungen", wie sie der Rat der EU nach

⁷⁸⁶ <http://perens.sourcelabs.com/articles/2005/11/18/friday-november-18-my-speech-at-the-un-world-summit-on-the-information-society>.

⁷⁸⁷ *Diehr*, 450 U.S. 175, 209 U.S.P.Q. (BNA) 1 (1981) = GRUR Int 1981, 646; abrufbar bspw auch unter <http://www.sensortime.com/Diehr81-de.html>. Zu einem vertiefenden Einblick in die Geschichte der US-amerikanischen Softwarepatentierung siehe *Maier/Mattson*, State Street Bank ist kein Ausreißer: Die Geschichte der Softwarepatentierung im US-amerikanischen Recht, GRUR Int 2001, 677.

⁷⁸⁸ <http://www.usip.com/articles/1st2fil.htm>.

⁷⁸⁹ http://news.com.com/The+open-source+patent+conundrum/2010-1071_3-5557340.html.

⁷⁹⁰ AA *Holzer*, Einheitlicher Patentschutz für Computerprogramme in Europa? ÖBl 2003, 59.

anfänglichen Konsultationen beschloss⁷⁹¹. Aber bereits damals formierte sich der Widerstand von Kritikern, welche die vorher erwähnten "amerikanischen Verhältnisse" mit Trivialpatenten und Patentierung von Geschäftsprozessen befürchteten. Da die „Gefahr“ des Patentrechtsschutzes für Software im Moment nicht allzu konkret ist – der Versuch eine Richtlinie für "computerimplementierte Erfindungen" zu erlassen wurde im Sommer 2005 vom EU-Parlament verhindert – diskutiert dieses Kapitel den Patentschutz in Hinblick auf Open Source Software und die damit verbundenen Auswirkungen nach geltendem österreichischen Recht abstrakt und hinsichtlich aktueller Entwicklungen. Eine vollständige Erörterung des Themas würde zu weit führen, da Nischenaspekte und Einzelüberlegungen dargestellt werden müssten, um eine umfassende Darstellung zu ermöglichen.

9.1. Einführung

„We have also heard about abusive litigation practices of “patent trolls”, described to me as companies that neither invent anything new nor produce anything at all, but simply acquire patent rights and then push the bounds of their patents, suing patent holders who actually do innovate. I am interested in hearing more about these concerns, as well as proposed solutions from our witnesses.“

- Patrick Leahy

Die vier bekanntesten Grundprinzipien, die „Naturrechtstheorie“, die „Belohnungstheorie“, die „Anspornungstheorie“ und die „Vertragstheorie“ dienen den Befürwortern des Patentschutzes als Rechtfertigung. Sie werden daher zum besseren Verständnis kurz erläutert und danach dem Open Source Gedanken gegenübergestellt. Dadurch soll gezeigt werden, dass der Open Source Gedanke nicht zwangsläufig diesen Grundprinzipien entgegensteht.

⁷⁹¹ http://europa.eu.int/comm/internal_market/en/indprop/com02-92de.pdf#search='computerimplementierte%20Erfindungen'.

9.1.1. Die vier Grundprinzipien des Patentrechts

Die Naturrechtstheorie⁷⁹² geht davon aus, dass der Mensch ein unbedingtes Eigentumsrecht an seinen eigenen Ideen hat. Deren Aneignung durch andere, also ihre Verwendung ohne Erlaubnis, ist als Diebstahl zu verwerfen. Die Allgemeinheit ist moralisch dazu verpflichtet, dieses Eigentumsrecht anzuerkennen und zu schützen. Eigentum ist seinem Wesen nach ausschließlich. Daher wäre die Durchsetzung der Ausschließlichkeit beim Gebrauch patentierter Erfindungen die einzig richtige Art, mit der die Allgemeinheit dieses Eigentumsrecht begründen könnte.

Nach der Belohnungstheorie⁷⁹³ fordert es die Gerechtigkeit, jeden für die Allgemeinheit geleisteten Dienst nach seiner Nützlichkeit zu belohnen. Notfalls muss die Allgemeinheit zur Sicherung dieser Belohnung eingreifen. Erfinder leisten solche nützlichen Dienste und um ihnen eine angemessene Belohnung zu sichern, ist es angebracht, ihnen zeitlich befristete Monopolstellungen in Form ausschließlicher Patentrechte für ihre Erfindungen einzuräumen.

Die Anspornungstheorie⁷⁹⁴ geht von der Annahme aus, dass industrieller Fortschritt wünschenswert ist, dass für diesen Fortschritt Erfindungen und deren industrielle Verwertung notwendig sind, dass aber ihre Verwertung unzureichend ist, wenn Erfinder und Geldgeber keine größeren Gewinne erwarten dürfen, als sie sich aus der wettbewerblichen Ausnutzung allen technischen Wissens ergeben. Um Erfinder und deren Geldgeber zu motivieren und ihre Bereitschaft zu steigern, möglicherweise ihr Geld aufs Spiel zu setzen, muss die Gesellschaft eingreifen und die Gewinnerwartungen erhöhen. Die einfachste, billigste und wirkungsvollste Weise, einen solchen Ansporn zu

⁷⁹² Rafeiner in Kucsko, Geistiges Eigentum 809 mwA.

⁷⁹³ Rafeiner in Kucsko, Geistiges Eigentum 809.

⁷⁹⁴ Lutterbeck, Expertengespräch „Softwarepatente und Open Source“ (2001) 2, abrufbar unter:<http://ig.cs.tu-berlin.de/oldstatic/bl/061a/Lutterbeck-BT-6-2001.pdf>.

erreichen, ist nach dieser Theorie die Verleihung zeitlich befristeter Monopolstellungen in Form ausschließlicher Patentrechte an Erfindungen.

Die Vertrags- oder Offenbarungstheorie⁷⁹⁵ unterstellt, dass zwischen dem Erfinder und der Allgemeinheit ein Austauschvertrag abgeschlossen wird, aufgrund dessen der Erfinder seinen Besitz an geheimem Wissen aufgibt, um dafür den zeitlich befristeten Ausschließlichkeitsschutz für die gewerbliche Verwertung dieses Wissens einzutauschen. Auch hier wird stillschweigend vorausgesetzt, dass fortwährender industrieller Fortschritt wünschenswert ist, jedoch nicht erreicht werden kann, wenn Erfinder und Neuerer ihre Erfindungen geheim halten. In einem solchen Fall würde das neue technische Wissen erst sehr viel später dem allgemeinen Gebrauch zugänglich gemacht werden; technische Geheimnisse könnten sogar mit dem Tod ihrer Erfinder der Allgemeinheit für immer verloren gehen. Daher liegt es im Interesse der Allgemeinheit, mit dem Erfinder einen Vertrag abzuschließen und ihn so zur Offenbarung seines Geheimnisses und dessen Nutzung durch künftige Generationen zu veranlassen. Dies geschieht am besten, indem man ihm ausschließliche Patentrechte als Gegenleistung für die Offenbarung der Erfindung einräumt.

9.1.2. Die vier Grundprinzipien des Patentrechts und Open Source

Betrachtet man die angeführten Theorien unter dem Open Source Gesichtspunkt, sieht man, dass der Open Source Gedanke diese vier Grundprinzipien aufgreift. Die Naturrechtstheorie findet ihren Ausdruck im Copyleft-Gedanken. Das „Eigentumsrecht“ wird dadurch nicht nur ausdrücklich anerkannt, sondern findet sich in der Ausgestaltung der dinglichen Schuldrechte in den Lizenzen wieder. Der Schutz wird allerdings nicht vom Patentrecht

⁷⁹⁵ Rafeiner in Kucsko, Geistiges Eigentum 809 f.

beansprucht, sondern vom Urheberrecht, welches in § 1 UrhG die eigentümliche geistige Schöpfung schützt. Die Lizenzgeber von Open Source Lizenzen – wie auch jene proprietärer Lizenzen – drücken in den jeweiligen Lizenzen ihren freien Willen aus, ihr geistiges Eigentum, nach den in den Lizenzen definierten Rechten und Pflichten, zu nutzen. Eine Verletzung der Lizenzrechte führt – wie veranschaulicht⁷⁹⁶ – zu der Möglichkeit, seine Ausschließlichkeitsrechte gerichtlich durchzusetzen. Der Naturrechtstheorie wird durch das UrhG Genüge getan, ein weiterreichender patentrechtlicher Schutz kann daher nicht gefordert werden. Dieser ist als unbegründet und zu weit führend abzulehnen.

Open Source Lizenzen gehen auch auf die Belohnungstheorie ein, da kein Lizenzgeber gezwungen wird, sein Open Source lizenziertes Produkt ohne korrespondierende Belohnung zu verbreiten. Die kostenlose Lizenzierung könnte allerdings als Kriterium gewertet werden, dem Lizenzgeber würde eine Belohnung vorenthalten. Dieser entscheidet sich jedoch freiwillig zur Open Source Lizenzierung und hat die Möglichkeit, bspw durch Support, Verkauf, Weiterentwicklung und Anpassung, seine Belohnung zu erhalten. Die freiwillige Aufgabe der Monopolstellung kann hier keinesfalls als Argument dienen, Open Source Lizenzen würden eine Belohnung des geistigen Eigentümers verhindern. Eine Monopolisierung des geistigen Gedankens ist noch aus weiteren Gründen abzulehnen, welche im folgenden Kapitel 9.2. dargelegt werden.

Die Anspornungstheorie wird in den Open Source Lizenzen nicht aufgegriffen, da diese einen anderen Zugang gewählt haben, um industriellen wie wirtschaftlichen Fortschritt zu ermöglichen. Viele Open Source Projekte sind aus dem Grund entstanden, dass Benutzer von Software mit den Anwendungsmöglichkeiten, die sie hatten, unzufrieden waren und nach alternativen Lösungen gesucht haben, einen Arbeitsablauf oder einen

⁷⁹⁶ Vgl Kap 3, 7 und 9.

Arbeitsschritt effektiver gestalten zu können. Durch die offene Gestaltung der Problemlösung können Schwierigkeiten, die in vielen parallelen Arbeitsprozessen auftauchen, gleichzeitig gelöst werden. Der offene Austausch fördert dabei die Einbeziehung vieler Lösungswege und Möglichkeiten und trägt so wesentlich zu einem effektiven industriellen Fortschritt bei⁷⁹⁷. Den Gründen für die Anspornungstheorie wird also wesentlich Rechnung getragen. Ein weiteres Argument ist, dass Firmen, die bspw ein Programm unter einer BSDartigen Lizenz vertreiben, ein Produkt testen und bei Bedarf wieder in eine proprietäre Lizenz überführen können⁷⁹⁸. Gerade im schnelllebigen IT-Bereich macht es keinen Sinn, den industriellen Fortschritt durch die Bereitstellung eines Monopolschutzes zu erreichen. Dieser stellt eher eine Behinderung des Fortschritts dar, auch wenn viele Unternehmen diesen Schutz aus nachvollziehbaren Gründen gerne in Anspruch nehmen würden.

Das letzte Argument greift die Vertrags- oder Offenbarungstheorie auf. Auch hier findet sich keine ausreichende Argumentation, die Herbeiführung eines monopolartigen Zustands für Software zu erlauben. Ein Austauschvertrag nach der Vertrags- oder Offenbarungstheorie würde zu einer Zweiklassengesellschaft beim Entwicklungsprozess führen. Diejenigen, die sich die Patentgebühren leisten können, können am Entwicklungsprozess teilnehmen, die anderen nicht. Im Softwarebereich kann ein solcher Zustand nicht wünschenswert sein. Die Ausgrenzung von Entwicklern an ganzen Prozessen würde dazu führen, dass Entwicklung nur noch in wenigen Bereichen stattfinden könnte. Daher ist hier eher der urheberrechtlichen Lösung zu folgen, durch die Möglichkeit, für seine geistige Leistung eine angemessene Bezahlung durch – proprietäre oder Open Source – Lizenzierung zu erhalten, ohne auf der Idee aufbauende Entwicklungen verhindern zu können.

⁷⁹⁷ Vgl Einführung.

⁷⁹⁸ Mir ist durchaus bewusst, dass diesem Argument damit begegnet werden kann, dass in der Praxis kein Unternehmen dazu bereit wäre, seine Produkte derart zu öffnen, sodass Mitbewerber diese „kapern“ könnten, dennoch findet diese Vorgangsweise täglich statt, durch die Lizenzierung eines Projektes unter einer Open Source Lizenz.

Im Fazit ist festzuhalten: Die erwähnten Theorien verdeutlichen, dass die wirtschaftlichen Interessen eines Erfinders bzw Entwicklers ein hochwertiges und zu schützendes Gut sind, welches es zu verteidigen gilt. Diese Verteidigungshaltung zeigt sich besonders in den USA⁷⁹⁹, wo Patentrechtsstreitigkeiten meist dazu benutzt werden, Konkurrenten vom wirtschaftlichen Wettbewerb fernzuhalten⁸⁰⁰. Dabei stehen eindeutig wirtschaftliche Interessen im Vordergrund, sodass die Schutzgüter nur noch als Mittel zum Zweck angesehen werden müssen. Ebenso kann kein hinreichendes Argument gefunden werden, Software Patentschutz zu gewähren, da keine der Theorien für sich beanspruchen kann, einen über andere bestehende Schutzrechte hinausgehenden sinnvollen Schutz bieten zu können. Die materiellen Bedenken der Open Source Gemeinde gegenüber dem Patentschutz werden nachfolgend konkret dargestellt.

9.2. Open Source – die technischen und patentrechtlichen Grundlagen, Bedenken und Schutzmöglichkeiten

Software erfährt – wie bereits erwähnt⁸⁰¹ – grundlegenden Schutz durch das Urheberrecht. Ein zusätzlicher Schutz durch Patentrecht erfordert eine Abgrenzung zwischen beiden Rechtsgebieten und eine besondere Betrachtung der Software an sich.

⁷⁹⁹ Siehe bspw *Lutterbeck*, Expertengespräch 6.

⁸⁰⁰ Siehe dazu insbesondere die Rede von *Senator Leahy*:
http://judiciary.senate.gov/member_statement.cfm?id=1475&wit_id=2629.

⁸⁰¹ Vgl Kap 3.

9.2.1. Die „Doppelnatur“ von Software

Die Unterscheidung zwischen Quelltext und Objektform, die auch immer wieder in den Open Source Lizenzen deutlich wird, zeigt die „Doppelnatur“⁸⁰² der Software. Die Arbeiten und Weiterentwicklungen am Quellcode selbst genießen als Ausdrucksform geistigen Schaffens den Schutz des Urheberrechts. Die Funktionalität des Programms, welches auf dem Quellcode basiert, aber zum zweckdienlichen Einsatz in Objektformat kompiliert werden muss, kann dem Patentrecht zugeordnet werden⁸⁰³. Interessant ist in diesem Zusammenhang § 1 der Mustervorschriften der WIPO⁸⁰⁴, der ein Computerprogramm als *„eine Folge von Befehlen, die nach Aufnahme in einen maschinenlesbaren Träger fähig sind, zu bewirken, dass eine Maschine mit informationsverarbeitenden Fähigkeiten eine bestimmte Funktion oder Aufgabe oder ein bestimmtes Ergebnis anzeigt, ausführt oder erzielt“* definiert.

Die Festlegung, ob dem Quellcode oder der Funktion mehr Beachtung beigemessen werden soll, ist eine Frage des persönlichen Standpunkts.⁸⁰⁵ Festzuhalten ist, dass Urheber- und Patentrecht subsidiär nebeneinander bestehen können. Der wesentliche Unterschied zwischen den beiden Schutzrechten besteht darin, dass der Urheberrechtsschutz ab Schaffung des Werks entsteht, ohne dass es einer Anmeldung des Rechts bedarf wie es das Patentrecht vorsieht.

⁸⁰² *Wiebe*, Softwarepatente – das Ende von Open Source? MR 2004, 195.

⁸⁰³ *Horns*, Anmerkungen zu begrifflichen Fragen des Softwareschutzes, GRUR 2001, 1 (2).

⁸⁰⁴ Übersetzt abgedruckt in GRUR Int 1978, 286.

⁸⁰⁵ Eine Überlegung zu dieser Problematik findet sich bei: *Wiebe*, MR 2004, 195 f; oder *Pierson*, Softwarepatente – technische und patentrechtliche Grundlagen, JurPC Web-Dok. 184/2004, abrufbar unter: <http://www.jurpc.de/aufsatz/20040181.htm>.

9.2.2. Das PatG und die Problematik der Technizität

Der Patentrechtsschutz für Computerprogramme ist auch deshalb so umstritten, weil das PatG den Begriff der Erfindung nicht definiert⁸⁰⁶. § 1 PatG fordert von einer Erfindung, dass diese sich für einen Fachmann nicht in nahe liegender Weise ergeben darf und neuartig und gewerblich anwendbar sein muss. Nach der hL und Rsp ist eine Erfindung eine „*qualifizierte Regel zum technischen Handeln*“⁸⁰⁷, das heißt die Erfindung muss dem Bereich der Technik zugeordnet werden können, weshalb alle Produkte des menschlichen Geistes, die nicht in diesen Bereich fallen iSd § 1 Abs 2 PatG nicht patentierbar sind⁸⁰⁸. Die Erfordernis der Neuartigkeit der Erfindung bemisst sich gem § 3 Satz 1 PatG am „Stand der Technik“.

Dieser Begriff umfasst gem § 3 Satz 2 PatG alle Erfindungen, die vor dem Prioritätstag der Anmeldung zum Patent durch schriftliche oder mündliche Beschreibung, durch Benützung oder in sonstiger Weise zugänglich gemacht worden sind⁸⁰⁹. § 1 Abs 3 PatG enthält einen Negativkatalog, welcher in Z 3 (entsprechend § 52 Abs 2 EPÜ) die Patentierbarkeit von Plänen, Regeln und Verfahren für Programme für Datenverarbeitungsanlagen untersagt. Die im Negativkatalog des § 1 Abs 3 PatG deklatorisch aufgezählten Gegenstände und Tätigkeiten sind „als solche“ nicht-technischer Natur⁸¹⁰. Die Einschränkung des Ausschlusses der Patentierfähigkeit durch den Begriff „als solche“ in § 1 Abs 4 PatG hat zu einer umfassenden Diskussion geführt, ab welchem Grad der Technizität eine Software nicht mehr „als solche“ angesehen werden darf bzw

⁸⁰⁶ *Flammer in Kucsko*, Geistiges Eigentum 838.

⁸⁰⁷ *Blocher in Jahnel/Schramm/Staudegger*, Informatikrecht², 137.

⁸⁰⁸ *Flammer in Kucsko*, Geistiges Eigentum 841.

⁸⁰⁹ Dieses „Schlupfloch“ führt in der Praxis zu teils kuriosen Rechercheergebnissen. So werden unter anderem „Clever & Smart“ Hefte herangezogen, um den Beweis dafür führen zu können, dass eine Erfindung, die zum Patent angemeldet werden soll (meist durch einen wirtschaftlichen Konkurrenten), schon vorveröffentlicht worden ist und dadurch zum Stand der Technik geworden ist.

⁸¹⁰ *Pierson*, JurPC Web-Dok. 184/2004, abrufbar unter: <http://www.jurpc.de/aufsatz/20040181.htm>.

kann⁸¹¹. Diese Diskussion wird an dieser Stelle nicht weiter aufgegriffen. Es scheint jedoch verständlich, dass vorrangig die Zweckbestimmung des Programms nach der Kompilierung in ein maschinenlesbares Format für die Qualifizierung herangezogen werden sollte. Die Firma Siemens hat einen Leitfaden⁸¹² entwickelt, anhand dessen die Patentfähigkeit von Software überprüft werden kann. Demnach ist Software patentfähig, wenn eine der nachfolgenden Fragen bejaht werden kann:

1. Wird mit der Erfindung eine physikalische Eigenschaft bspw durch Funktionen, Zustände oder physikalisch messbare Werte einer Einrichtung beeinflusst?
2. Wird mit der Erfindung ein technischer Effekt wie bspw der Wirkungsgrad oder die Störsicherheit bewirkt?
3. Liegt eine technische Aufgabe vor, die bsp in einer Einrichtung eine bestimmte Fähigkeit verbessert oder eine optimale Funktionsverteilung ermöglicht?
4. Waren technische Überlegungen erforderlich, das heißt mussten bspw anlagenbedingte Beschränkungen beachtet oder überwunden werden?

⁸¹¹ Sehr umfassend *Horns*, GRUR 2001, 9 f; aber auch *Blocher* in *Jahnel/Schramm/Staudegger*, Informatikrecht², 137. *Mellulis*, Zur Patentfähigkeit von Programmen für Datenverarbeitungsanlagen, GRUR 1998, 843; *Tauchert*, Zur Patentierbarkeit von Programmen für Datenverarbeitungsanlagen, GRUR 1999, 965.

⁸¹² <http://www.patente.bmbf.de/de/pdf/software.pdf>.

Siemens hat diese Aussagen graphisch sehr gut veranschaulicht wie nachfolgend zu sehen ist:

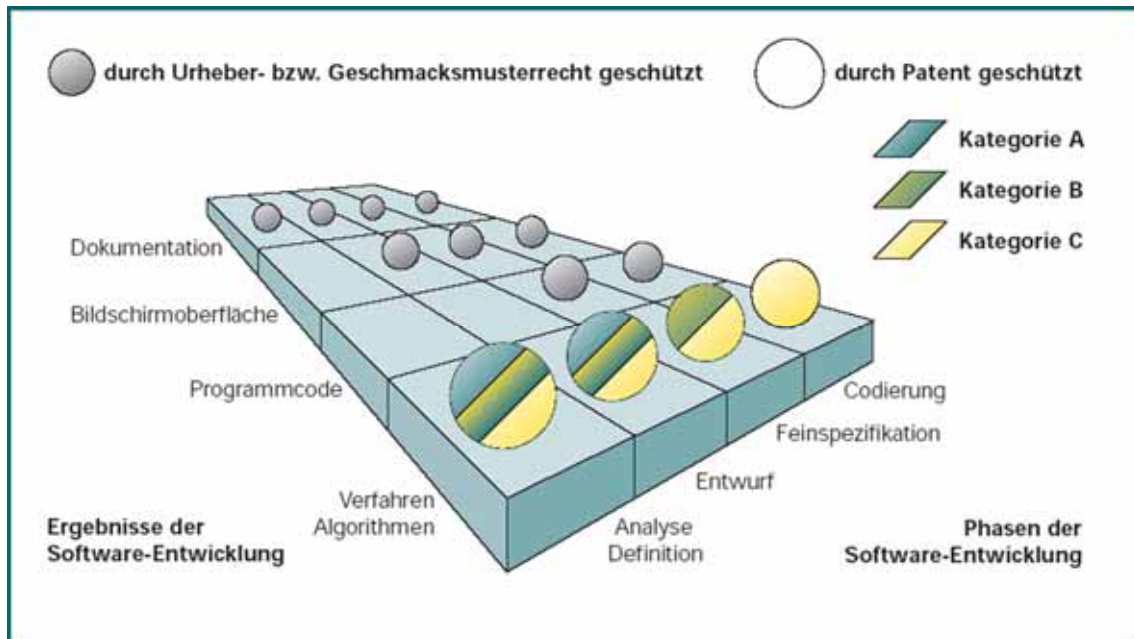


Abb 15⁸¹³

„Kategorie A“ bezeichnet:

- „Erfindungen, die unabhängig von ihrer Realisierung durch Software auf einer davon abstrahierten Ebene durch technische Abläufe beschreibbar sind.“

„Kategorie B“ bezeichnet:

- „Erfindungen, die sich nicht mehr so abstrakt beschreiben lassen, dass die Problemlösung ganz ohne Software-Elemente angebar ist.“

„Kategorie C“ bezeichnet:

- „Erfindungen, bei denen das zu lösende Problem und die Lösungsmittel ausschließlich in der Welt der Software liegen; deren weitere Verallgemeinerung zu abstrakten Ideen ohne Bezug zur Technik führen.“

Die Möglichkeit einen Patentrechtsschutz für Software zu erhalten besteht demnach, auch wenn er nur in Randbereichen stattfindet.

⁸¹³ <http://www.patente.bmbf.de/de/pdf/software.pdf>.

9.2.3. Die patentrechtlichen Bedenken der Open Source Gemeinschaft

Die Bedenken der Open Source Gemeinschaft gegen einen Patentschutz für Open Source Software bestehen aus unterschiedlichen Gründen. Einer der wesentlichsten ist die Befürchtung der Entwickler von Open Source Software, dass Dritte Patente auf Programme anmelden, die von ihnen entwickelt wurden. Diese Befürchtung ist nur nachvollziehbar, wenn man einen genaueren Blick auf das Umfeld eines Open Source Software-Entwicklers wirft. Der Entwickler hat normalerweise weder das Interesse an einer Patentierung noch die nötigen finanziellen Mittel – man bedenke, dass eine Patentanmeldung mit Kosten von ca 2000 – 5000 Euro verbunden ist⁸¹⁴ - um seine Entwicklung anzumelden. Da § 4 Satz 1 PatG zwar normiert, dass der Erfinder oder seinem Rechtsnachfolger das alleinige Recht zusteht, eine Erfindung anzumelden, § 4 Satz 2 PatG jedoch bestimmt, dass der Erstanmelder als der Erfinder anzusehen ist, bedeutet dies für den Fall der unrechtmäßigen Anmeldung, dass der wirkliche Erfinder seinen Patentanspruch erst im Beschwerdeverfahren beweisen müsste. Die Open Source Gemeinschaft weist hier auf erhebliche Nachteile hin, die entstehen, wenn ein einzelner Entwickler genötigt wäre, seinen Anspruch gegen ein Unternehmen mit professioneller Patentabteilung durchzusetzen⁸¹⁵. Auch die Möglichkeit durch „Kreuzlizenzen“ Zugang zum „Know-How“ Dritter zu bekommen und dadurch eine weitere Verwertungsmöglichkeit zu erhalten, würde für Open Source Entwickler in der Regel unmöglich sein⁸¹⁶.

Es muss ebenfalls mit der Gefahr gerechnet werden, bestehende Patente unwissentlich bei der Entwicklung von Open Source Software zu verletzen, da die Patentrecherche mit hohem Aufwand verbunden ist⁸¹⁷. Diese

⁸¹⁴ http://info.tuwien.ac.at/ai/tt_pt.htm.

⁸¹⁵ Jaeger/Metzger, Open Source Software 112.

⁸¹⁶ Wiebe, MR 2004, 200.

⁸¹⁷ Horns, GRUR 2001, 12.

Gefahr besteht für Open Source Entwickler besonders, da sie - wie bereits erwähnt - entweder alleine agieren oder nur einer kleinen Firma angehören. Ein möglicher Patentrechtsstreit, der aus einer ungewollten Verletzung resultiert, bedeutet für diese Entwickler meist den wirtschaftlichen Ruin, da sie neben den Kosten, die durch den Streit entstehen (Verfahrenskosten und mögliche Schadensersatzansprüche) auch ihr entwickeltes Werk verlieren und wieder von vorne anfangen müssen⁸¹⁸. Das Argument, Patentrechtsverletzungen durch den offenen Quellcode leichter erkennen zu können⁸¹⁹, ist nur so zu begründen, dass zwischen Open Source- und proprietären Software-Entwicklern keine „Waffengleichheit“ besteht. Denn der Nachweis einer Patentrechtsverletzung bei Software, die nur im Binärcode bzw in Objektform vorliegt, ist beschwerlich bis unmöglich. Letztlich bleibt die Sorge einer Anmeldung von Patenten auf Software durch Entwickler von Open Source Software selbst; diese Sorge beruht auf einem tatsächlichen Fall. 1999⁸²⁰ wurde bekannt, dass der Entwickler *Victor Yodaiken* in den USA ein Patent⁸²¹ auf das von ihm zusammen mit seinem Studenten *Michael Barabanov* an der Universität von New Mexico entwickelte Echtzeitbetriebssystem „RTLinux“ angemeldet hatte. Dieser Fall wird nachfolgend kurz dargestellt.

9.2.3.1. RTLinux – Der Open Source Patentrechtsfall

RTLinux basiert auf GNU/Linux und wurde zu einem Echtzeitbetriebssystem weiterentwickelt. Das Besondere an einem Echtzeitbetriebssystem ist, dass das System ein Ergebnis innerhalb eines vorher fest definierten Zeitintervalles garantiert berechnet, bevor eine bestimmte Zeitschranke erreicht ist. Diese Eigenschaft, die ein handelsübliches Betriebssystem normalerweise nicht hat, ist vor allem für den industriellen

⁸¹⁸ *Jaeger/Metzger*, Open Source Software 113.

⁸¹⁹ *Horns*, GRUR 2001, 9; *Wiebe*, MR 2004, 197.

⁸²⁰ *Metzger*, Risiko Softwarepatente, abrufbar unter: <http://www.linux-magazin.de/Artikel/ausgabe/2001/07/recht/recht.html>.

⁸²¹ Dieses Patent wurde unter der Nummer „U.S.-Patent 5,995,745“ angemeldet.

Einsatz (bspw für Roboter, Mess- oder Präzisionsgeräte) oder „Embedded Systems“ von Bedeutung, da diese auf eine feste Antwortzeit angewiesen sind. RTLinux wird heute unter einer freien und einer proprietären Lizenz⁸²² von der Firma FSMLabs vertrieben. Die Problemstellung, die dieser Fall der Patentanmeldung verdeutlicht, ist, dass hier ein Derivat von Open Source Software zum Patent angemeldet wurde und damit der urheberrechtliche Schutzbereich der GPL überschritten wurde⁸²³. Open Source Lizenzen wie die GPL versuchen die proprietäre Nutzung von Softwarepatenten zu verhindern. Sollte jedoch durch die Weiterentwicklung eines unter einer Open Source Lizenz stehenden Programms eine patentfähige Erfindung geschaffen werden, hindert den Erfinder nichts, diese Erfindung zum Patent anmelden zu lassen⁸²⁴, da das Patentrecht im Gegensatz zum Urheberrecht nicht den entwickelten Quellcode der Software schützt, sondern die technische Lehre zur Problemlösung⁸²⁵. Dies bedeutet im Ergebnis, dass die Durchsetzung der Lizenzbedingungen der Open Source Software nur dann möglich sind, wenn für das Patent Software genutzt wird, welche unter einer Open Source Lizenz stehen. Da dies aber nicht zwingend der Fall sein muss – es ist durchaus möglich, das Patent mit Software zu nutzen, welche einen anderen Ursprung hat als jene, auf der die Erfindung ursprünglich basiert – kann der Erfinder sein Patent unabhängig von den Bestimmungen der Open Source Lizenz verwerten, unter der die Software steht, auf der seine Erfindung basiert.

Im Falle von RTLinux kam es deswegen zu lang anhaltenden Diskussionen, die darin endeten, dass die Entwickler von RTLinux die Patentrechte Anfang 2001 in einer eigenen Lizenz, der „Open RTLinux Patent License“, regelten. Diese Lizenz wurde von Seiten der FSF⁸²⁶ heftig kritisiert, da

⁸²² <http://www.fsmlabs.com/rtlinuxfree.html>.

⁸²³ *Wiebe*, MR 2004, 199; *Metzger*, abrufbar unter <http://www.linux-magazin.de/Artikel/ausgabe/2001/07/recht/recht.html>.

⁸²⁴ *Jaeger/Metzger*, Open Source Software 122; *Metzger*, abrufbar unter <http://www.linux-magazin.de/Artikel/ausgabe/2001/07/recht/recht.html>.

⁸²⁵ *Horns*, GRUR 2001, 2.

⁸²⁶ <http://www.gnu.org/philosophy/rtlinux-patent.html>.

sie gegen die Bestimmungen der GPL verstieß, sodass sie durch die momentan aktuelle „Version 2“⁸²⁷ abgelöst wurde. Die Lizenz nimmt auf die vorher erwähnte Problemstellung insofern Rücksicht, da sie die lizenzgebührenfreie Nutzung des Patents in jenen Fällen erlaubt, in denen der Anwender Software verwendet, die unter der GPL lizenziert ist. In allen anderen Fällen ist mit FSMLabs eine gesonderte Vereinbarung zu treffen⁸²⁸.

9.2.3.2. Aktuelle Bedenken - Anmerkungen zum EPLA

Das Europäische Übereinkommen über Patentstreitigkeiten – engl: „European Patent Litigation Agreement“ (EPLA) – ist der momentane Entwurf eines Übereinkommens über die Schaffung eines Streitregelungssystems für europäische Patente. Durch das EPLA wird versucht, ein Zusatzprotokoll zum Europäischen Patentübereinkommen (EPÜ)⁸²⁹ abzuschließen, das zum Ziel hat, die Unterzeichnerstaaten zu einem neugeschaffenen integrierten Gerichtswesen mit einheitlichen Verfahrensregeln sowie auf ein gemeinsames Berufungsgericht zu verpflichten.

Zunächst gilt zu bemerken, dass das EPLA ein Schritt in die richtige Richtung ist, Rechtssicherheit in den Umgang mit Software in Zusammenhang mit Patentrechten zu bringen. Es darf jedoch bezweifelt werden, dass dessen Instrumente genügen, dieses Ziel auch tatsächlich zu gewährleisten. Die Gründe hierfür liegen vor allem darin, dass die anzustrebende Gleichberechtigung aller Betroffenen durch den vorliegenden Entwurf der EPLA nicht erreicht wird.

⁸²⁷ <http://www.fsmlabs.com/openpatentlicense.html>.

⁸²⁸ Siehe die Gegenüberstellung von „RTLlinuxPro“ und „RTLlinuxFree“ unter: <http://www.fsmlabs.com/rtllinuxfree.html>.

⁸²⁹ http://www.epo.org/patents/law/legislative-initiatives/epla_de.html.

Um die einzelnen Argumente besser nachvollziehen zu können und um eine sachliche Diskussion zu fördern, werden die wesentlichen Kritikpunkte in Folge kurz dargestellt:

1. **Die rechtliche Einordnung des European Patent Court (EP Ct):** Das EPLA würde den EP Ct außerhalb der EU statuieren und ihn so von der gemeinschaftlichen Kontrolle lösen, gleichwohl käme ihm nach Art 6 EPLA die Höchstgerichtsbarkeit in Bezug auf Patentstreitigkeiten zu. Die Gefahren dieser Eigenständigkeiten liegen auf der Hand, auch wenn sie nicht abgeschätzt werden können. Auf jeden Fall wären die Kontrollinstrumente nur schwach ausgebildet und bedürften einer besseren Anbindung an die EU. Eine mögliche Lösung wäre, das Gericht – ähnlich dem EUG I und dem EUGH – in die Säulenstruktur der EU einzubinden. Gerade ein so brisantes Thema wie das Patentrecht sollte nicht ausgegliedert existieren können, um jeglichen Spekulationen den Nährboden zu entziehen, welche eine sachliche Diskussion jetzt schon schwierig machen.
2. **Kosten:** Die Kosten des Verfahrens sollen gem Art. 21 EPLA beiden Prozessparteien einen fairen Zugang zum Gericht ermöglichen. Dieser Vorschlag erscheint zunächst vernünftig, bei näherer Betrachtung jedoch zu wagen, da nicht deutlich wird, was „faire“ Prozesskosten sind. Hier wäre eine genauere Definition bzw. Erläuterung wünschenswert. Die „echten“ Kosten findet man in Art. 53ff des EPLA. Hier wird auf die Entscheidungsphase des Gerichts eingegangen und die Möglichkeiten genannt, die das Gericht hat, um an Informationen zu gelangen. Das Gericht kann gem Art 53 EPLA die Parteien anhören, Informationen erfragen, sich relevante Dokumente vorlegen lassen und Zeugen sowie Sachverständige hören. Dies alles ist in der Regel mit hohen Kosten verbunden und kann schon am Anfang eines Prozesses dazu führen, dass eine Partei vom Prozess ausgeschlossen wird und damit den Fall verliert, denn Art 57 EPLA bestimmt dies, sobald eine Partei es verabsäumt, die geforderte Gerichtsgebühr zu zahlen.

Das EPLA sollte weiterhin beobachtet werden, gleichzeitig sollte bedacht werden, dass es bei der Diskussion nicht vordergründig um die Problematik der Softwarepatentierung im Speziellen, sondern um das Patentrecht im Allgemeinen geht. Diese Meinung wird auch in einem offenen Brief⁸³⁰ zum Ausdruck gebracht, den achtzehn international agierende Patentrechtsexperten, darunter drei Österreicher, unterschrieben haben und in dem dafür eingetreten wird, mehr Rechtssicherheit im Patentwesen durch effizientere und bessere Prüfmethoden zu etablieren.

9.2.4. Open Source Software und die Schutzmöglichkeiten bei einer erfolgten Anmeldung

Es besteht die Möglichkeit, dass das Patentamt auf Software ein Patent erteilt, da es im Prüfungsverfahren unter Umständen nicht erkennt oder nicht erkennen kann, dass ein Mangel der Patentfähigkeit iSd §§ 1 – 3 PatG vorliegt⁸³¹. Dritte müssen daher die Möglichkeit haben, gegen diese fälschlicherweise ergangenen Patente vorzugehen. Hierzu steht jedem bereits im Anmeldeverfahren der Einspruch iSd § 102 PatG zu, der binnen vier Monaten, ab dem Tag der Bekanntmachung der Erteilung des Patents, beim Patentamt eingebracht werden muss. Im Falle der Anmeldung von Software muss der Antragssteller – dies kann jede Person sein – seinen Einspruch auf § 102 Abs 1 PatG beziehen und anführen, dass der Gegenstand des Patents den §§ 1 bis 3 PatG nicht entspricht. Wichtig dabei ist, den Einspruch inhaltlich ausreichend zu begründen, da der OGH eine Substantiierungspflicht statuiert hat⁸³². Wird der Einspruch gegen eine Patentanmeldung von Software rechtzeitig erhoben und die erste Jahresgebühr gem § 166 Abs 6 PatG eingezahlt, kommt es gem § 107 PatG zur automatischen Patenterteilung. Die

⁸³⁰ <http://www.popa.org/pdf/misc/epocipo2007.pdf>.

⁸³¹ Siehe auch den "Open Letter From a Coalition of Patent Examiner Representatives"; www.popa.org/pdf/misc/epocipo2007.pdf.

⁸³² BA 08.09.1983, B 31/82 – Isolierglas – ÖBI 1984, 114.

Möglichkeiten, gegen eine Patenterteilung auf Software als solche vorzugehen, besteht vorrangig in der Nichtigkeitsklärung gem § 48 Abs 1 Z 1 PatG, mit dessen Hilfe eine Patenterteilungen bekämpft werden kann, bei der der Gegenstand des Patentess den §§ 1 bis 3 PatG nicht entspricht. Die Nichtigkeitsklärung wirkt in diesen Fällen ex tunc und muss daher als effektives Schutzinstrument gegen fälschlich erteilte Patente angesehen werden.

Die Durchsetzung von zivilrechtlichen Schutzansprüchen muss im Fall von Patentverletzungen negativ betrachtet werden, da Software an sich – wie ausgeführt – nicht unter den Schutz des PatG fällt. Ansprüche – wie Unterlassung iSd § 147 PatG, Beseitigung iSd § 148 PatG oder Ansprüche in Geld iSd § 150 PatG – brauchen an dieser Stelle daher nicht untersucht werden.

9.3. Ergebnis

Die patentrechtliche Behandlung von Software stellt aktuell keine große Gefahr für Software dar. Allerdings hat die WIPO im März 2006 einen Vertrag zwischen Europa, Japan und den USA zur Diskussion gestellt⁸³³. Dieses "Abkommen zur Harmonisierung materieller Fragen zum Patentrecht" („Substantive Patent Law Treaty“, „SPLT“)⁸³⁴ könnte die Möglichkeit eröffnen, Software unter Patentschutz zu stellen. Die Sorge der Open Source Gemeinschaft hinsichtlich einer Patentierung von Software scheint auch aus anderen Gründen verständlich.

Eines ihrer wichtigsten Argumente dabei ist, dass die Grundsätze der Open Source Gemeinschaft sinnvolle und effektive Alternativen zum

⁸³³ http://www.wipo.int/meetings/2006/scp_of_ge_06/en/.

⁸³⁴ <http://www.wipo.int/patent/law/en/harmonization.htm>.

Patentwesen darstellen können. Die Entwicklungen verschiedener Programme in kürzester Zeit haben gezeigt, dass Entwickler motiviert sind, Wissen nicht nur weiterzuentwickeln und mitzugestalten, sondern auch weiterzugeben⁸³⁵.

Die Schaffung von Patentpools durch Kreuzlizenzierung von Patenten – wie man sie im wirtschaftlichen Bereich findet – benachteiligt die Open Source Gemeinschaft erheblich, da sie noch nicht in der Lage ist, ebenbürtige Schutzinstrumente aufweisen zu können. Vor allem in den USA zeigt sich dieser Nachteil deutlich. Patente werden dort taktisch verwendet. Diese werden nur deshalb angemeldet, um gegen Konkurrenten bei Bedarf Klage einreichen zu können. Das führt bei großen Firmen oft zu Pattsituationen – da alle die gleiche Strategie verfolgen – in denen keiner den anderen verklagt. Einzig jene Konkurrenten, die keine Patente besitzen, haben diesen „Schutz“ nicht und das sind in der Regel die kleinen und mittleren Unternehmen, die Open Source Software entwickeln. Der Patentschutz behindert hier den Fortschritt der Entwicklung. Der britische Richter *Hugh Laddie* führt dazu aus, dass die Förderung der Monopolisierung durch Patente dazu führen würde, dass Firmen, die eine Monopolstellung innehaben, nur wenig Interesse am technologischen Fortschritt zeigen würden. Software per Copyright zu schützen, habe sich bewährt, so *Laddie* und weiter: *"Ich kenne niemanden der glaubt, dass wir heute bei Word 2003 wären, würde es Softwarepatente geben. Wir wären dann doch eher bei Wordperfect 1.2, denn warum sollte man weiterentwickeln, wenn man einen geschützten Markt hat."*⁸³⁶

Software als solche kann in Österreich wie in der übrigen EU noch keinen Patentschutz erhalten; dass Software ab einem gewissen Grad der Technizität dennoch patentierbar ist, hat zu einer Diskussion in Europa geführt, die sich von jener in den USA unterscheidet. Während in den USA die allgemeine Anwendung von Patentrecht auf Software zur Debatte steht, findet

⁸³⁵ *Perens*, Software Patents vs. Free Software, abrufbar unter:
<http://perens.com/Articles/patents.html>.

⁸³⁶ <http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/70291&words=WIPO>.

man hier eine – wie *Lutterbeck* richtig anmerkt⁸³⁷ – meist von Juristen geführte Diskussion über die Definition von „*Technizität*“. Bisher hat diese Diskussion keine zufrieden stellende Lösung hervorgebracht, auch wenn die verschiedenen Definitionsversuche nachvollziehbar sind.

Die Open Source Gemeinschaft steht dennoch einer gesetzlichen Möglichkeit der Patentierung von Software nicht völlig schutzlos gegenüber. Dies zeigen Überlegungen und Vorschläge, die von Seiten der Open Source Gemeinschaft unterbreitet wurden⁸³⁸. Ein denkbarer Schutz wäre demnach die Vorveröffentlichung nach dem Entwickler-Prinzip „*Release early, release often*“, der vorbeugend eingesetzt werden könnte⁸³⁹. Eine Veröffentlichung bspw auf einer Webseite oder in Newsgroups und -foren würde dazu führen, dass ein Programm zum „Stand der Technik“ und somit von einer Anmeldung ausgeschlossen werden würde.

Bezüglich der Gefahr der unwissentlichen Verletzung von bestehenden Patenten kann einerseits auf die Hoffnung der Verbesserung von Recherchedatenbanken verwiesen werden⁸⁴⁰ andererseits kann diese Gefahr dadurch entschärft werden, indem privaten Entwicklern ein gesetzliches Lizenzrecht ähnlich den urheberrechtlichen Bestimmungen des § 40d UrhG eingeräumt wird⁸⁴¹. Eine für die Wirtschaft annehmbare Kompromisslösung bietet das Dual Licensing an⁸⁴², welches von den „FSMLabs“ für RTLinux praktiziert wird.

⁸³⁷ <http://ig.cs.tu-berlin.de/oldstatic/bl/061a/Lutterbeck-BT-6-2001.pdf>.

⁸³⁸ Der etwas unsachliche Beitrag von *Holzer*, Einheitlicher Patentschutz für Computerprogramme in Europa, ÖBl 2003, 59, mag zur Darstellung wenig beitragen, da die Ausführungen allzu einseitig und oberflächlich dargestellt sind.

⁸³⁹ *Jaeger/Metzger*, Open Source Software 114.

⁸⁴⁰ Siehe bspw *Esslinger/Betten*, Patentschutz im Internet, CR 2000, 18 (21).

⁸⁴¹ *Jaeger/Metzger*, Open Source Software 113.

⁸⁴² *Wiebe*, MR 2004, 200.

Da erscheint ein weiterer Vorschlag von *Lutterbeck*⁸⁴³ sinnvoll, der ein Quelltextprivileg zur Diskussion stellt: „*Der Umgang mit dem Quelltext von Computerprogrammen muss patentrechtlich privilegiert werden. Das Herstellen, Anbieten, in Verkehr bringen, Besitzen oder Einführen des Quelltextes eines Computerprogramms in seiner jeweiligen Ausdrucksform muss vom Patentschutz ausgenommen werden.*“

Die Diskussion um die patentrechtliche Behandlung von Software wird weiter geführt werden, auch wenn ihr momentan die Brisanz fehlt. Die Open Source Gemeinde nutzt diese Zeit, um enger zusammenzurücken und sich neue Argumente für zukünftige Diskussion zu überlegen⁸⁴⁴.

⁸⁴³ <http://ig.cs.tu-berlin.de/oldstatic/bl/061a/Lutterbeck-BT-6-2001.pdf>.

⁸⁴⁴ Siehe bspw Lobbyinggruppen, wie: <http://www.ffii.org/> oder <http://www.patentfrei.de/>.

10. ENDERGEBNIS

“May the source be with you.”

-Anonym

Software war von Anfang an Open Source. Durch die zunehmende Popularität des Computers und dem damit verbundenen wirtschaftlichen Erfolg war es für die entwickelnden Firmen in den 1970er Jahren notwendig geworden, den Quellcode der Software nicht mehr freizugeben, um die zentrale Kontrolle über die eigene Software zu behalten und sich davor zu schützen, dass Konkurrenten den Quellcode in ihren Programmen verwendeten. Die Renaissance des frei verfügbaren Quellcodes wurde durch die Haltung von *Richard Stallman* in beispielloser Weise eingeleitet, der bis heute der „Mastermind“ der Freien Software Bewegung geblieben ist. Der Abspaltung der Open Source Gemeinde, die nicht bereit war *Stallmann* zu folgen, hat inzwischen dazu geführt, dass ein Dialog mit den Entwicklern von proprietärer Software gesucht wird. Dieser Dialog ist mittlerweile soweit fortgeschritten, dass Microsoft seine englischsprachige „Anti-Linux-Webseite“ entfernt⁸⁴⁵ und darüber hinaus erklärt hat, dass Open Source ein Entwicklungs- und Lizenzmodell sei, welches man im eigenen Unternehmen nutze⁸⁴⁶. So gibt es inzwischen Partnerschaften von Microsoft mit Novell und Linspire. Open Source ist damit endgültig als ernst zu nehmendes Lizenzmodell anerkannt worden. Bisher kostete es viel Energie, Ressentiments und Gerüchte gegenüber dem jeweils anderen Lager aufzubauen und zu verbreiten. Diese Energie kann jetzt dazu genutzt werden, Weiterentwicklungen an den Projekten vorzunehmen und den wirtschaftlichen Wettbewerb über die Produkte und weniger über Kampagnen zu führen.

⁸⁴⁵ Die deutschsprachige Version war Ende August 2007 noch unter <http://www.microsoft.com/germany/diefakten/default.mspix> abrufbar.

⁸⁴⁶ <http://www.heise.de/open/news/meldung/94838>.

Es hat sich herausgestellt, dass der Begriff "Open Source" nicht ausschließlich mit Software verbunden ist. Die Open Source Bewegung ist mehr als nur ein Modell der Software Entwicklung. Schon beim Blick auf die Geschichte wird deutlich, dass sich neben der Entwicklung von Software neue Standards und Geschäftsmodelle etablieren konnten. Die Bezeichnung der teilnehmenden Akteure als „Community“ deutet darüber hinaus auf den sozialen Aspekt hin. Im Gegensatz zu einem Unternehmen, kann jeder an der Entwicklung von Open Source teilnehmen.

Die Etablierung von Open Source als seriöses Softwaremodell geschah nicht nur durch den großen Zuspruch der Softwareentwickler, sondern auch durch dessen wirtschaftliche Anerkennung. „*Das Pinguin-Imperium hat längst den Mittelstand erreicht*“⁸⁴⁷ titelt die FAZ im Jahr 2006 und dokumentiert damit, dass die Phase, in der Open Source Software nur von denjenigen benutzt werden konnte, die fähig waren, die Software eigenständig anpassen zu können, überwunden ist. IT-Projektleiter und CIOs treffen eine Entscheidung über den Einsatz von Open Source Software heutzutage leichter, da sie mit den Stärken und Schwächen der einzelnen Produkte besser vertraut sind und sich nicht nur aufgrund eines „Hypes“ eine Meinung bilden müssen. Durch das zunehmende Know-How über Open Source Software konnten die Einsatzstrategien wesentlich effizienter gestaltet werden. Ebenso konnte der Reifegrad erhöht und die Wirtschaftlichkeit durch die Reduzierung der Kosten verbessert werden.

⁸⁴⁷ FAZ vom 21.3.2006.

Nach einer Untersuchung von „Forrester Research⁸⁴⁸“ aus dem Jahr 2004 bewerten IT-Manager vor allem folgende Kriterien von Open Source Software positiv:

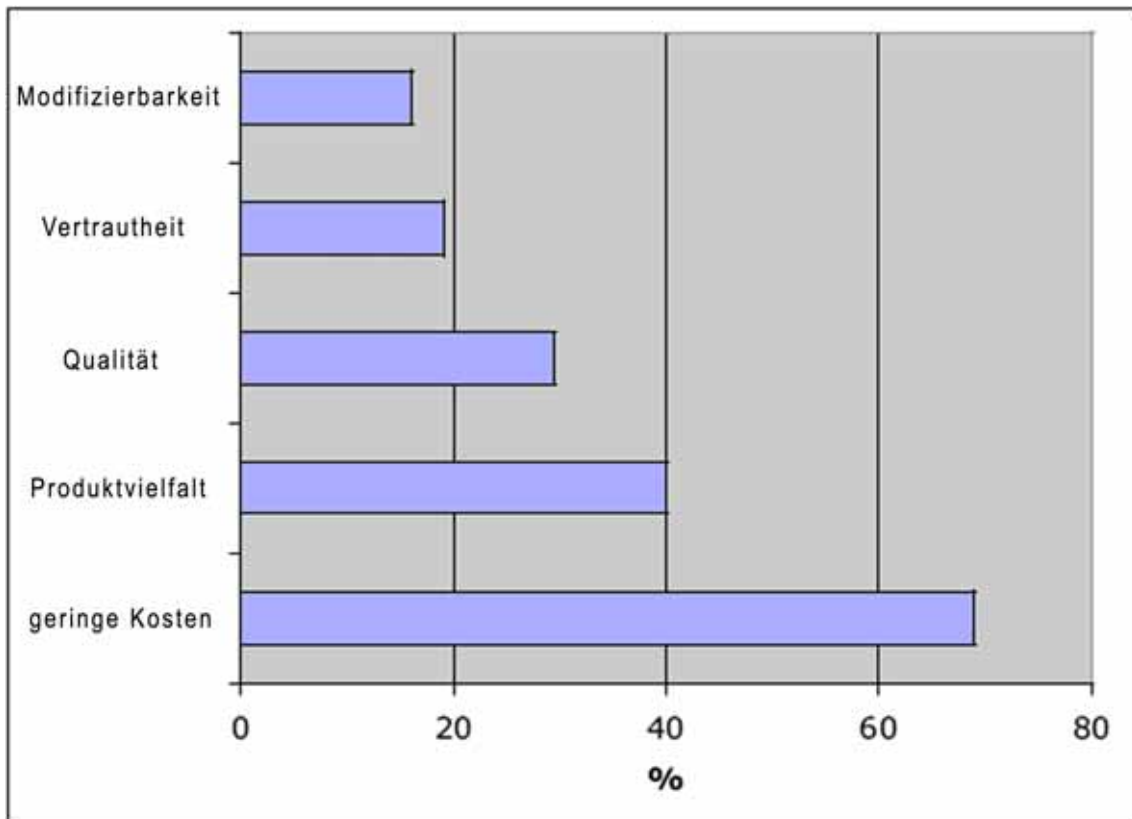


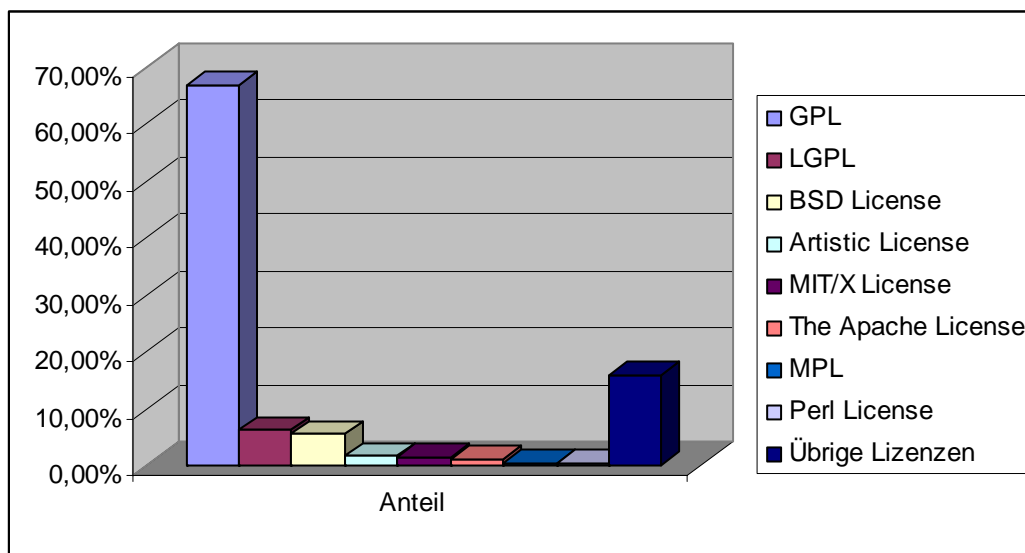
Abb 16⁸⁴⁹

Die Grafik zeigt, dass neben der Möglichkeit die Software eigenständig zu modifizieren, vor allem die geringen Kosten, die mit der Nutzung von Open Source Software verbunden sind, ein Entscheidungskriterium zu Gunsten von Open Source darstellen. Die Vertrautheit mit Open Source Software nimmt stetig zu, ebenso die Qualität der Produkte. Die Benutzerfreundlichkeit von Open Source Software hat mittlerweile sogar Großstädte wie München, Paris oder Wien dazu bewogen, auf Open Source Software umzusteigen.

⁸⁴⁸ Die Ergebnisse der Umfrage wurden von *Välimäki* grafisch umgesetzt. Siehe *Välimäki*, *The Rise of Open Source Licensing* (2005) 38.

⁸⁴⁹ Nach *Välimäki*, *The Rise of Open Source Licensing* 38.

Rechtlichen Schutz für Open Source Software garantieren – neben den Bestimmungen des UrhG – die bei jeder Verbreitung der Software mitzuliefernden Open Source Lizenzen. Diese sind inzwischen in so großer Anzahl vorhanden, dass die OSI ein Zertifizierungsverfahren⁸⁵⁰ für Lizenzen einführen will, um weiterhin eine Übersicht gewährleisten zu können. Ein Vorteil, der sich aus der Vielzahl an Lizenzen ergibt, ist die Auswahlmöglichkeit für Lizenzgeber und -nehmer. Diese können ihre Anforderungen an den Lizenzinhalt aus den bislang 58 von der OSI anerkannten Lizenzen wählen. Die wichtigste Unterscheidung der verschiedenen Schutzmerkmale besteht in der Trennung von Lizenzen mit Copyleft-Effekt und solchen ohne Copyleft-Effekt. Während Lizenzen – wie die GPL 2 bzw 3 – eine freie Verfügbarkeit des Quellcodes für das Ursprungswerk und dessen „derivative Works“ für alle Zeiten garantieren, bieten Open Source Lizenzen ohne Copyleft-Effekt die Möglichkeit, weiterentwickelte Programme auch proprietär lizenzieren zu können. Die nachfolgende Übersicht zeigt die herausragende Popularität der GPL:

Abb 17⁸⁵¹

⁸⁵⁰ http://www.opensource.org/docs/certification_mark.html#approval.

⁸⁵¹ Datenquelle: <http://freshmeat.net/stats/#license> (Stand: 25.07.2007).

Daher ist wichtig, ob eine Open Source Lizenz mit der GPL kompatibel ist. Für Entwickler besteht durch eine mögliche Inkompatibilität die Gefahr, dass ihr Werk entweder nicht zusammen mit GPL lizenzierte Software verwendet werden kann oder es für andere uninteressant ist, ein solches Projekt weiterzuentwickeln.

Die Open Source Lizenzen weisen bei der Überprüfung der Verträglichkeit mit dem österreichischen Recht kaum Problemfelder auf. Lediglich die Haftungs- und Gewährleistungsausschlüsse finden bei verschiedenen Vertragsvarianten keine Anwendung. Die Vertragsautonomie und dispositive Festlegung des Vertragsinhaltes sind mit den Open Source Lizenzen überwiegend als konform zu bewerten. Das Maß der Rechtssicherheit für Lizenzgeber und -nehmer bei Nutzung einer Open Source Lizenz ist dagegen schwerer zu beurteilen. Die Wirksamkeit einer Open Source Lizenz der GPL wurde bisher mehrmalig von deutschen Gerichten bestätigt, sodass die Rechtsverbindlichkeit des Lizenztextes weiterhin Auslegungssache der Lizenzpartner und der Lehre ist. Da jedoch unter Lizenzpartnern der Gang vor Gericht bislang vermieden wurde, ist anzunehmen, dass die Lizenzen weitläufig verstanden und anerkannt werden. Hinsichtlich der Haftungs- und Gewährleistungsausschlüsse wird angemerkt, dass Haftung und Gewährleistung meist gesondert im Rahmen eines Dienstleistungsvertrags – wie bspw. einem Service- oder Supportvertrag – vereinbart werden und oftmals den finanziell umfangreichsten Teil des Rechtsgeschäfts ausmachen. Rechtlich betrachtet muss betont werden, dass nicht jeder, der Open Source Software nutzt, zum Lizenznehmer wird. Dies geschieht erst durch Abschluss des Lizenzvertrags, wenn der Nutzer Rechte zur Anpassung oder Weiterentwicklung benötigt, die über die gesetzlich in § 40d UrhG gewährten hinausgehen. Für einen reinen Nutzer macht daher die Vereinbarung eines gesonderten Servicevertrags Sinn, da er ein Interesse an der ordnungsgemäßen Lauffähigkeit der Software haben wird.

Das Markenrecht hat durch die dezentrale Organisation der Open Source Gemeinschaft bislang nur eine untergeordnete Rolle im Umgang mit Open Source Software gespielt. Durch den immer größer werdenden wirtschaftlichen Erfolg wird diesem Bereich immer mehr Aufmerksamkeit geschenkt. Die missbräuchlichen Versuche, die Marke „Linux“ anzumelden, haben der Open Source Gemeinschaft die Auswirkungen der Nichtbeachtung des Markenrechts aufgezeigt und für eine Sensibilisierung gesorgt. Oft unterschätzen kleine „Spaßprojekte“ die Möglichkeit eines markenrechtlichen Konflikts, wenn sie bspw. aus Verdruss über ein proprietäres Programm eine Open Source Alternative entwickeln, dessen Bezeichnung eine Abwandlung des proprietären Markennamens darstellt. Sollte dieses Projekt über ihren Freundeskreis hinaus weltweit bekannt werden, kann dies unangenehme Konsequenzen nach sich ziehen. Während die untaugliche Eintragungsfähigkeit aufgrund eines Registrierungshindernisses noch im Anfangsstadium eines Open Source Projekts durch Änderung des Marken- bzw. Projektnamens geheilt werden kann, ohne großen wirtschaftlichen oder finanziellen Schaden zu verursachen, ist eine – eventuell unabsichtlich – begangene Markenverletzung für einen Kleinunternehmer existenzbedrohend. Open Source Entwickler sollten sich daher von Beginn an mit einem möglichen Namen für ihr Projekt auseinandersetzen und diesen bei Bedarf markenrechtlich eintragen lassen. Die Eintragung als Gemeinschaftsmarke macht meist Sinn, da Software international genutzt werden kann und so vorzeitig unnötige Barrieren bei einem Markteintritt beseitigt werden können. Der Art 142a EG hilft hier Konflikte zu lösen, wobei in der Praxis nationale Markeninhaber einer konkurrierenden Marke häufig schon am Open Source Projekt beteiligt sind und daher als Partner und nicht als Gegner zu sehen sind. Interessant zu beobachten wird sein, in welchem Ausmaß die Gattungsbezeichnung „Open Source“ rechtlichen Schutz erhält, um sie vor missbräuchlichen Verwendungen zu schützen. Dieser Schutz ist äußerst wichtig, um einer „Verwässerung“ des Begriffs vorzubeugen und die mit Open Source verbundenen Rechte und Pflichten bewahren zu können.

Das Vergaberecht soll seinen Vorgaben gemäß Chancengleichheit bei der Ausschreibung garantieren. Open Source Software ist für den öffentlichen Sektor immer interessanter geworden. Die nachhaltigen Auswirkungen auf Umstellung müssen von den Auftraggebern im Vorfeld gründlich analysiert werden. Die leicht erkennbaren Vorteile der Open Source Software – wie bspw. Offenheit der Schnittstellen, Möglichkeit der eigenen Anpassung und günstiger Support durch die Community – können durch unüberlegte Migrationen leicht als Nachteile gewertet werden, wenn eine Anpassung faktisch nicht stattfindet und die Nutzer die Software nur unzureichend einsetzen können. Durch die dezentrale Organisation fehlte es bislang an kompetenten Ansprechpartnern, welche eine sichere Migration überwachen und gewährleisten konnten. Der Umfang der Ausschreibungen hat in diesem Bereich dafür gesorgt, dass Distributoren und KMUs, die Open Source Software kommerziell vertreiben, sich den Herausforderungen einer vergaberechtlichen Ausschreibung angepasst haben und inzwischen kompetente Unterstützung und Lösungen anbieten können. Der Wettbewerb mit proprietären Anbietern hat bei diesen mittlerweile dazu geführt, ihre Lizenzpolitik zu überdenken⁸⁵². Die Entstehung von gemischten Umgebungen hat zu einem verbreiteten „best of both worlds“-Ansatz geführt, bei der neben Open Source auch proprietäre Software genutzt wird und daher über offene und proprietäre Schnittstellen verfügt⁸⁵³. Die Verlässlichkeit einer solchen Lösung ist inzwischen weitestgehend anerkannt⁸⁵⁴ und sollte bei der Entscheidungsfindung in Bezug auf die Nachhaltigkeit mit einbezogen werden. Letztendlich bleibt festzustellen, dass das Vergaberecht Open Source Software weder bevorzugt noch benachteiligt. Die nächsten „Vergabeschlachten“ werden vorrangig zwischen Microsofts Betriebssystem „Vista“ und einer Linuxdistribution geschlagen werden und eventuelle Defizite des Vergaberechts in struktureller wie inhaltlicher Sicht aufzeigen.

⁸⁵² http://www.computerwoche.de/knowledge_center/linux/557468/; oder <http://www.ipp.mpg.de/~iik/linux.html>.

⁸⁵³ Siehe zur Problematik der offenen und proprietären Schnittstellen: *Varian/Shapiro*, Einführung von Linux im öffentlichen Sektor in *Opensourcejahrbuch 2007*, 213, <http://www.opensourcejahrbuch.de/download/jb2007>.

⁸⁵⁴ Siehe bspw.: <http://www.heise.de/open/artikel/81753>

Über das Patentrecht wurde bis Juli 2005 innerhalb der Open Source Gemeinschaft am heftigsten diskutiert. Durch die Ablehnung der Patentrichtlinie ist dieser Themenbereich wieder in den Hintergrund gerückt. Gerade die Abgrenzungsschwierigkeiten zwischen Software als solcher und Software als patentierbarer Bestandteil von technischen Neuerungen werden in nächster Zukunft nicht gelöst werden. Erschwerend kommt hinzu, dass die Systematik der Patentanmeldungen eine genaue Überprüfung nicht vorsieht – diese würde zudem einen immensen Personalaufwand bzw hohen Wissenstand erfordern – und daher Patentanmeldungen auf Software vorgenommen werden können. Die Open Source Lobbyisten befinden sich derzeit in Lauerstellung, um erneute Versuche, Software als solche in Europa patentschutzfähig zu machen, abzuwehren. Die GPL 3 behandelt die patentrechtliche Problematik explizit, indem sie versucht, den in der Praxis gängigen Kreuzlizenzierungen durch den Verzicht auf die Geltendmachung von Patentansprüchen entgegenzuwirken. Die Wirksamkeit für die Praxis wird sich hier erst zeigen müssen. Fakt bleibt jedoch, dass Open Source Lizenzgeber ihre „Karten auf den Tisch legen“ indem sie den Quellcode offen bereitstellen und so stets der Gefahr ausgesetzt sind, dass Dritte darin eine Patentverletzung erkennen. Proprietäre Lizenzgeber bieten nicht die gleichen Recherchemöglichkeiten, weshalb sie einem ungleich geringeren Risiko ausgesetzt sind, wegen einer Patentrechtsverletzung geklagt zu werden. Diesbezüglich kann auch die GPL 3 keine Abhilfe schaffen, da diese nur jene Lizenzgeber bzw Lizenznehmer vor rechtlicher Inanspruchnahme schützt, die sich über bestehende Patente bewusst sind. Begehen diese unbewusst eine Patentrechtsverletzung, greift dieser Schutz nicht. Die Aufgabe, eine befriedigende Lösung für alle Beteiligten zu schaffen, liegt damit wieder beim Gesetzgeber. Die Verlagerung der Diskussion von einer rechtsdogmatischen zu einer rechtspolitischen wird diese Aufgabe nicht vereinfachen, weshalb in naher Zukunft nicht damit zu rechnen ist, Rechtssicherheit im Softwarebereich hinsichtlich des Patentschutzes zu haben.

Der IT-Bereich hat von Open Source profitiert und vielen Menschen einen Anreiz gegeben, sich mit Problemen im Softwarebereich auseinanderzusetzen. Nicht wenige haben „*Bugs*“ zum Anlass genommen, sich Programmierkenntnisse anzueignen. Die selbständige Problemlösung konnte aber nur durch Software ermöglicht werden, dessen Quellcode frei zugänglich war. Deshalb bleibt abschließend anzumerken, dass es interessant zu beobachten sein wird, ob die „Philosophie“, die mit Open Source verbunden ist, in dieser Weise fortbestehen kann und welche weiteren Auswirkungen die fortschreitende Verbreitung von Open Source Software und den dazugehörigen Lizenzen – nicht nur aus rechtlicher Sicht – haben werden.

11. LITERATURVERZEICHNIS

KOMMENTARE

- Czernich/Heiss* (Hrsg) EVÜ - Das Europäische Schuldvertragsübereinkommen: Übereinkommen über das auf vertragliche Schuldverhältnisse anzuwendende Recht: Kommentar (Wien 1999).
- Dillenz* Praxiskommentar zum österreichischen Urheberrecht und Verwertungsgesellschaftenrecht (Wien 1999).
- Fasching* (Hrsg) Kommentar zu den Zivilprozessgesetzen Bd II Teilbd 1² (Wien 2002).
- Grünzweig*, Österreichisches, europäisches und internationales Markenrecht (Wien 2006).
- Rummel* (Hrsg) Kommentar zum Allgemeinen bürgerlichen Gesetzbuch Band I³ (Wien 2000).
- Kommentar zum Allgemeinen bürgerlichen Gesetzbuch Band II³ (Wien 2004).
- Schönherr/Thaler* Entscheidungen zum Markenrecht (Wien 1985).
- Schwimann* (Hrsg) Praxiskommentar ABGB VI² (Wien 2006).
- Soergel* (Hrsg) Kommentar zum Bürgerlichen Gesetzbuch Bd X¹² (Stuttgart 1996).

MONOGRAPHIEN, SAMMELBÄNDE UND LEHRBÜCHER

- Blatz* Das Recht des Urhebers an literarischen Arbeitsergebnissen (Köln 1967).
- Ciresa* Urheberrecht aktuell: Wegweiser durch das österreichische Urheber- und Leistungsschutzrecht für die Praxis (Wien 1997).
- DiBona/Ockman/Stone* Open Sources: Voices from the Open Source Revolution (Sebastopol 1999).
- Dinnes* Softwareentwicklung im Vertragsverhältnis (Frankfurt 2002)
- Dittrich* Österreichisches und Internationales Urheberrecht⁴ (Wien 2004).
- Fallenböck/Galla/Stockinger* Urheberrecht in der digitalen Wirtschaft (Wien 2005).
- Fasching* Lehrbuch des österreichischen Zivilprozeßrechts² (Wien 1990).
- Gulbins/Obermayr* Unix System V.4. Begriffe, Konzepte, Kommandos, Schnittstellen (Berlin 1995).
- Grassmuck* Freie Software - Zwischen Privat- und Gemeineigentum² (Bonn 2004).
- Harbrecht* Die Genossenschaft als Rechtsform für junge Unternehmen (Wien 2001).
- Heid/Preslmayr* (Hrsg) Handbuch Vergaberecht² (Wien 2005).
- Jaeger/Metzger* Open Source Software, Rechtliche Rahmenbedingungen der Freien Software (München 2002).
- Jahnel/Schramm/Staudegger* (Hrsg) Informatikrecht² (Wien 2002).
- Klein* Alles Zufall – Die Kraft die unser Leben bestimmt (Hamburg 2004).
- Kucsko* Geistiges Eigentum (Wien 2003).
- Marly* Softwareüberlassungsverträge³ (München 2000).

- Omsels/Schertz* (Hrsg) Festschrift für Paul W. Hertin zum 60. Geburtstag (München 2000).
- Plöckinger* Der Erschöpfungsgrundsatz im Marken- und Urheberrecht (Wien/Graz 2002).
- Posch* Bürgerliches Recht VII Internationales Privatrecht³ (Wien 2002).
- Rehbinder* Urheberrecht, ein Studienbuch¹³ (München 2004).
- Renner* Rechtsschutz von Computerprogrammen (Wien 1998).
- Scheuch/Holzmüller* Die wirtschaftliche Bedeutung des Urheberrechts in Österreich - Die Wertschöpfung der Copyright Industries (Wien 1989).
- Sieckmann* Bravehack, Technische, wirtschaftliche und gesellschaftliche Aspekte von freier Software und Open Source; ihr Wesen, ihre Geschichte, ihre Organisationen und Projekte (Bonn 2000).
- Spindler* Rechtsfragen der Open Source (Köln 2004).
- Surowiecki* The Wisdom of Crowds – Why the Many Are Smarter Than the Few (London 2004).
- Stumpf/Groß* Der Lizenzvertrag⁸ (Frankfurt 2005).
- Ulrich/Körner* Der internationale Softwarevertrag (Heidelberg 1995).
- v.Lewinski/Walter* (Hrsg) Europäisches Urheberrecht: Kommentar ; insbesondere Software-, Vermiet- und Verleih-, Satelliten- und Kabel-, Schutzdauer-, Datenbank-, Folgerecht-, Informationsgesellschaft-Richtlinie, Produktpiraterie-Verordnung (Wien 2001).
- Wachter/Winter* Computerrecht für die Praxis (Wien 1996).
- Walter* Urheberrechtsgesetz – UrhGNov 2003 (Wien 2003).
- Wartinger* Urheberrecht und Arbeitsverhältnis (Wien 2006).
- Williams* Free as in Freedom. Richard Stallman's Crusade for Free Software (Sebastopol 2002).

AUFSÄTZE UND BEITRÄGE

• Beiträge in Zeitschriften und Sammelbänden

- Andréewitch* Zur Anwendung des Produkthaftungsgesetzes für Softwarefehler, EDVuR 2/1990, 50.
- Risiken bei firmen- und behördeninterner Bearbeitung von Open Source Software, MR 2005, 36.
- Berger* Urheberrechtliche Erschöpfungslehre und digitale Informationstechnologie, GRUR 2002, 198.
- Blocher* Die Rechtsstellung des Software-Anwenders nach österreichischem und deutschem Urheberrecht, EDVuR 1994, 5.
- Urheberrecht im Internet, in *Studiengesellschaft für Wirtschaft und Recht* (Hrsg), Internet und Recht. Rechtsfragen von E-Commerce und E-Government (2002) 223.
- Deike* Open Source Software: IPR-Fragen und Einordnung ins deutsche Rechtssystem, CR 1/2003, 9.
- Dillenz* Urheberrechtsschutz heute, ÖBI 1990, 1.
- Dittrich* Gedanken zur so genannten Zweckübertragungstheorie, RfR 1979, 41.
- Urheberrechtsschutz für Computersoftware? RdW 1983, 39.
- Noch einmal: Gedanken zur so genannten Zweckübertragungstheorie, RfR 1984, 1.
- Ertl* Allgemeine Geschäftsbedingungen der Softwareverträge, EDVuR 1994, 19.
- Gutgläubiger Erwerb von Softwarepiraten – Zugleich ein Beitrag zum Gutgläubenserwerb an Forderungen, MR 1997, 314.

- Fallenböck/Trappitsch* Application Service Providing (ASP) – rechtlich betrachtet, MR 2002 3.
- Gamerith* Sind Rechtsgemeinschaften an Immaterialgüterrechten Gesamthandgemein-schaften? ÖBI 1996, 63.
- Grass* “Wir Urheber”, FAZ vom 18.01.2005, 36.
- Grillberger* Zur Einziehung fremder Forderungen in eigenem Namen, ÖJZ 1978, 141.
- Handig* Urheberrechtliche Erschöpfung von Downloads im World Wide Web, RdW 2003, 2
- Heckmann* IT-Vergabe, Open Source Software und Vergaberecht, CR 6/2004, 401.
- Heidinger* Von Copyright zu „Copyleft“: Wenn Programm freigegeben werden, Die Presse, 22.12.2003
- Holzer* Einheitlicher Patentschutz für Computerprogramme in Europa? ÖBI 2003, 59.
- Holzinger* Bedienerhandbuch – Schlüssel für EDV-Entscheidungen? EDVuR 1994, 54.
- Horns* Anmerkungen zu begrifflichen Fragen des Softwareschutzes, GRUR 2001, 1.
- Horwath* Software – ein Produkt? ecolex 2000, 784.
- Jaeger* Copyright oder Copyleft, Computerwoche Spezial 4/2000, 36.
Version 3 der GPL fertiggestellt, c` t 2007 Heft 15, 38.
- Koch* Urheber und kartellrechtliche Aspekte der Nutzung von Open-Source-Software (II) CR, 2000, 333.
- Knies* Erschöpfung online? GRUR Int 2002, 314.
- Küng* Urheberrechtliche Aspekte freier Software, MR 2004, 21.
- Lutz* WIENUX und OpenOffice – Die freiwillige Migration im Magistrat der Stadt Wien, in *Lutterbeck/Bärwolff/Gehring*, Open Source Jahrbuch 2006 (2006) 73.

- Mellulis* Zur Patentfähigkeit von Programmen für Datenverarbeitungsanlagen, GRUR 1998, 843;
- Metzger* Das FLA - Die treuhänderische Lizenzvereinbarung der FSF Europe, Linux-Magazin 5/2003, 75.
- Müller/Gerlach* Open Source-Software und Vergaberecht, CR 2/2005, 87.
- Omsels* Open Source und das deutsche Vertrags- und Urheberrecht, in *Schertz/Omsels*, Festschrift für *Paul W.Hertin* zum 60. Geburtstag am 15.November 2000 (2000) 141.
- Reinbothe* Die EG-Richtlinie zum Urheberrecht in der Informationsgesellschaft, GRUR Int 2001, 733.
- Reindl* Die Nebenrechtseinräumung im Musikverlagsvertrag, ÖSGRUM 12/1993; *Reimer*, Der Erschöpfungsgrundsatz im Urheberrecht und gewerblichen Rechtsschutz unter Berücksichtigung der Rechtsprechung des Europäischen Gerichtshofes, GRURInt 1972, 221.
- Reischauer* Das neue Gewährleistungsrecht und seine schadensersatzrechtlichen Folgen, JBI 2002, 137.
- Röttiger* Der Urheberrechtsschutz von Computersoftware in Österreich, ÖJZ 1990, 33.
- Schanda* Kritisches zur Aktivlegitimation des Markenlizenznehmers ÖBI 2001, 151.
- Schimanko* Die Klage auf Unterlassung und § 45 ZPO, ÖJZ 2002, 127.
- Spindler* Europäisches Urheberrecht in der Informationsgesellschaft, GRUR 2002, 105 (108)
- Spindler/Wiebe* Open Source Vertrieb, CR 2003, 873.
- Staudegger* Zur Qualifikation von Verträgen, die der Überlassung von Computersoftware dienen, JBI 1998, 604.
- Rechtsfragen beim Erwerb von IT-Systemen in *Jahnel/Schramm/Staudegger*, Informatikrecht² (2002) 87.

- Sujecki* Open Source Software im deutschen Vertrags- und Urheberrecht, MR 2005, 40.
- Tangl* Leitfaden für die Einbeziehung elektronischer AGB, ecolex 2001, 896.
- Tauchert* Zur Patentierbarkeit von Programmen für Datenverarbeitungsanlagen, GRUR 1999, 965.
- Thiele* Verwendung von Mitarbeiterfotos auf Firmenwebsites, wbl 2002, 397.
Übertragung von Urheberrechten auf den Arbeitgeber, RdW 2002, 537.
- Walter* Verbreitungsrecht und Öffentlichkeit, MR 1998, 344.
Die vier Säulen des Urheberrechtes, ZfRV 1999, 88.
Entwurf eines Gesetzes zur Verbesserung der vertraglichen Stellung von Urhebern und ausübenden Künstlern in Österreich, GRURInt 2001, 602.
Gutgläubiger Erwerb urheberrechtlicher Nutzungsrechte nach österreichischem Recht? in *Loewenheim* (Hrsg), Urheberrecht im Informationszeitalter: Festschrift für *Wilhelm Nordemann* zum 70. Geburtstag am 8. Januar 2004 (2004) 625.
- Wiebe* Softwarepatente – das Ende von Open Source? MR 2004, 195.
Umsetzung der Geschäftspraktikenrichtlinie und Perspektiven für eine UWG-Reform, JBI 2007, 69.
- Wiebe/Heidinger* GPL 3.0 und EUPL – Aktuelle Entwicklungen im Bereich der Open Source Lizenzen, MR 2006, 258.
- Wiebe/Prändl* Open Source Software, Rechtliche Rahmenbedingungen nach österreichischem Recht, ÖJZ 2004, 628 (632).
- Wolff* Die Schlichtung von EDV – Streitigkeiten EDVuR 1988, 2.
- Zankl* Online-AGB: Erste Entscheidung zum E-Commerce-Gesetz, ecolex 2003, 669.

- **Online-Beiträge**

Bastin/Laurent, Report on Study of the compatibility mechanism of the EUPL,

<http://ec.europa.eu/idabc/servlets/Doc?id=27472>.

Brandon, FYI: Zope Public License 1.1 vague, contradictory, and not DFSG-free,

<http://lists.debian.org/debian-legal/2001/09/msg00014.html>.

False, Markenrecht und Open Source – Sinn oder Unsinn?,

www.ifross.de/Fremdartikel/VortragLinuxTag2003.pdf.

Fottner, EU-Marken im Zeichen der Osterweiterung,

<http://www.ipmunich.de/Grafik/mfs-02-2004.pdf>.

FSF, GPLv3 Third Discussion Draft Rationale,

<http://gplv3.fsf.org/gpl3-dd3-rationale.pdf>.

Jaeger, Copyright oder Copyleft,

http://www.ifross.de/ifross_html/art6.html.

Junger, Überblick über das Urheberrecht,

<http://remus.jura.uni-sb.de/urheberrecht/gw01.html>.

Leonard, The Saint of the Free Software,

http://archive.salon.com/21st/feature/1998/08/cov_31feature2.html.

Metzger, Risiko Softwarepatente,

<http://www.linux-magazin.de/Artikel/ausgabe/2001/07/recht/recht.html>.

Metzger, Nebeneinander von Lizenztypen, Einmal GPL, immer GPL?

<http://www.linux-magazin.de/Artikel/ausgabe/2001/01/recht/recht.html>.

Mühlbauer, Die Resozialisierung des Giganten,

<http://www.heise.de/tp/r4/artikel/8/8778/1.html>.

Perens, Software Patents vs. Free Software,

<http://perens.com/Articles/patents.html>.

Pierson, Softwarepatente – technische und patentrechtliche Grundlagen, JurPC Web-Dok. 184/2004,

<http://www.jurpc.de/aufsatz/20040181.htm>.

Raymond, Software Release Practice HOWTO,
<http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/licensepractice.html>.

Raymond, The Cathedral and the Bazaar,
http://www.firstmonday.org/issues/issue3_3/raymond/index.html.

Raymond, Epilog: Netscape Embraces the Bazaar,
<http://www.catb.org/%7Eesr/writings/cathedral-bazaar/cathedral-bazaar/ar01s13.html>.

Raymond in Leonard, The Richard Stallman Saga,
<http://archive.salon.com/21st/feature/1998/09/11feature2.html>.

Siepmann, Lizenz- und haftungsrechtliche Fragen bei der
kommerziellen Nutzung freier Software,
<http://jura.uni-sb.de/jurpc/aufsatz/19990163.htm>.

Stallman, Voices from the Open Source Revolution,
<http://www.oreilly.com/catalog/opensources/book/toc.html>.

Varian/Shapiro, Einführung von Linux im öffentlichen Sektor
<http://www.opensourcejahrbuch.de/download/jb2007>.

Wheeler, Make Your Open Source Software GPL-Compatible or Else,
<http://www.dwheeler.com/essays/gpl-compatible.html>.

WEBSEITENVERZEICHNIS

http://apache.org/about_apache.html.
http://archive.salon.com/21st/feature/1998/08/cov_31feature2.html.
<http://archive.salon.com/21st/feature/1998/09/11feature2.html>.
<http://blogs.zdnet.com/open-source/?p=1123>.
<http://counter.li.org/>.
<http://de.wikipedia.org/wiki/Freiheit>.
<http://dev.perl.org/perl6/rfc/346.html>.
<http://developer.apple.com/darwin/projects/>.
<http://distrowatch.com/>.
<http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/licensepractice.html>.
http://europa.eu.int/comm/internal_market/en/indprop/com02-92de.pdf#search=computerimplementierte%20Erfindungen'.
<http://freshmeat.net/stats/#license>.
<http://gplv3.fsf.org/>.
<http://gplv3.fsf.org/draft>.
http://groups.google.de/groups?as_umsgid=1991Aug25.205708.9541@klaava.Helsinki.FI.
<http://ig.cs.tu-berlin.de/oldstatic/bl/061a/Lutterbeck-BT-6-2001.pdf>.
<http://ig.cs.tu-berlin.de/oldstatic/bl/061a/Lutterbeck-BT-6-2001.pdf>.
<http://ig.cs.tu-berlin.de/oldstatic/bl/061a/Lutterbeck-BT-6-2001.pdf>.
<http://ig.cs.tu-berlin.de/oldstatic/w2000/ir1/referate1/k-1a/>.
http://info.tuwien.ac.at/ai/tt_pt.htm.
http://judiciary.senate.gov/member_statement.cfm?id=1475&wit_id=2629.
<http://jura.uni-sb.de/jurpc/aufsatz/19990163.htm>.
<http://librosoft.dat.escet.urjc.es/html/downloads/woss-icse-2004.pdf>.
<http://linuxgazette.net/issue13/tmark.html>.
http://www.linuxmark.org/linux_sublicense.php.
<http://lists.debian.org/debian-legal/2001/09/msg00014.html>.
<http://lists.debian.org/debian-legal/2001/09/msg00018.html>.
<http://lists.kde.org/?l=koffice-devel&m=99405898605976&w=2>.
<http://logo-contest.freebsd.org/announce.txt>.
<http://lwn.net/Articles/94763/>.
<http://mail.gnome.org/archives/evolution-list/2001-April/msg00452.html>.
<http://msdn.microsoft.com/embedded/usewinemb/ce/sharedsrccode/eula/default.aspx>.
http://news.com.com/IBM+offers+500+patents+for+open-source+use/2100-7344_3-5524680.html?tag=nl.
http://news.com.com/The+open-source+patent+conundrum/2010-1071_3-5557340.html.

http://news.zdnet.com/2100-3513_22-996866.html.
<http://oami.eu.int/de/mark/aspects/reg/reg4094.htm>.
<http://oami.europa.eu/de/office/aspects/communications/05-03.htm>.
<http://opensource.org/docs/policy/licenseproliferation.php>.
<http://perens.com/Articles/APSL.html>.
<http://perens.com/Articles/patents.html>.
<http://perens.sourcelabs.com/articles/2005/11/18/friday-november-18-my-speech-at-the-un-world-summit-on-the-information-society>.
<http://remus.jura.uni-sb.de/urheberrecht/gw01.html>.
<http://samba.sernet.de/lizenz/presse.html>.
<http://samba.sernet.de/old.news.html>.
<http://slashdot.org/article.pl?sid=98/11/22/1029225&tid=117>.
http://software.ericssink.com/Browser_Wars.html.
<http://sourceforge.net/projects/mtstcops/>.
http://sourceforge.net/softwaremap/trove_list.php?form_cat=160.
<http://wp.netscape.com/newsref/pr/newsrelease558.html>.
<http://www.affero.org/oagpl.html>.
<http://www.apache.org>.
<http://www.apache.org/GPL-compatibility.htm>.
<http://www.bittorrent.com>.
<http://www.blinkenlights.com/classiccmp/gateswhine.html>.
<http://www.bva.gv.at/BVA/Rechtsgrundlagen/EURichtlinien/default.htm>.
<http://www.catb.org/%7Eesr/writings/cathedral-bazaar/cathedral-bazaar/ar01s13.html>.
http://www.cbronline.com/article_news.asp?guid=A4BE06FA-EF8F-4D1E-8C62-B3DEF66F893F
<http://www.census.gov/eos/www/napcs/papers/cpcintro.pdf>
<http://www.cheiron.org/misc/license.htm>.
http://www.computerwoche.de/knowledge_center/linux/557468/.
<http://www.croftsoft.com/library/tutorials/gplmpl/>.
<http://www.cs.mun.ca/~ulf/csh/commcomp.html>.
http://www.debian.org/social_contract#guidelines.
<http://www.dpma.de/suche/klass/wd/abkommen.html>.
<http://www.dwheeler.com/essays/gpl-compatible.html>.
<http://www.dzug.org/maillinglisten/zope/archive/2003/2003-03/1047467083020>.
<http://www.eclipse.org/legal/eplfaq.php>.
http://www.ecomm-online.de/fileadmin/template-ecomm/uploads/uin/061103_ecomm_linux.pdf.
<http://www.ffii.org/>.
http://www.firstmonday.org/issues/issue5_3/kuwabara/.
http://www.firstmonday.org/issues/issue3_3/raymond/index.html.

<http://www.foresight.org/FI/Peterson.html>.
http://www.free-soft.org/gpl_history/
<http://www.fsf.org>.
<http://www.fsf.org/blogs/rms/entry-20050130>.
<http://www.fsf.org/campaigns/drm.html>.
<http://www.fsf.org/compatible-license-list.htm>.
<http://www.fsf.org/license-list.html#GPLIncompatibleLicenses>.
<http://www.fsf.org/licensing/essays/categories.html#TOCProprietarySoftware>.
<http://www.fsf.org/licensing/essays/categories.html>.
<http://www.fsf.org/licensing/essays/category.png>.
<http://www.fsf.org/licensing/essays/copyleft.html>.
<http://www.fsf.org/licensing/essays/free-software-for-freedom.html/view?searchterm=freedom>.
<http://www.fsf.org/licensing/essays/free-sw.html>.
<http://www.fsf.org/licensing/licenses/index.html>.
<http://www.fsf.org/licensing/licenses/why-assign.html>.
<http://www.fsf.org/news/gplv3launch>.
<http://www.fsf.org/philosophy/bsd.html>.
<http://www.fsf.org/philosophy/why-not-lgpl.html>
<http://www.fsmlabs.com/openpatentlicense.html>.
<http://www.fsmlabs.com/rtlinuxfree.html>.
<http://www.germany.fsf-europe.org/projects/fla/fla.de.html>.
<http://www.gnu.org/licenses/lgpl.html>.
<http://www.gnu.org/licenses/license-list.html#X11License>.
<http://www.gnu.org/philosophy/apsl.html>.
<http://www.gnu.org/philosophy/categories.de.html#ProprietarySoftware>.
<http://www.gnu.org/philosophy/free-software-for-freedom.html>.
<http://www.gnu.org/philosophy/free-sw.html>.
<http://www.gnu.org/philosophy/historical-apsl.html>.
<http://www.gnu.org/philosophy/license-list.html>.
<http://www.gnu.org/philosophy/rtlinux-patent.html>.
<http://www.golem.de/0302/23880.html>.
<http://www.heise.de/newsticker/meldung/17553>.
<http://www.heise.de/newsticker/meldung/17857>.
<http://www.heise.de/newsticker/meldung/38206>.
<http://www.heise.de/newsticker/meldung/39248-APSL.htm>.
<http://www.heise.de/newsticker/meldung/44891>.
<http://www.heise.de/newsticker/meldung/65576>.
<http://www.heise.de/open/artikel/81753>.
<http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/70291&words=WIPO>.

<http://www.heise.de/open/artikel/81753>.
<http://www.heise.de/open/news/meldung/94838>.
<http://www.heise.de/tp/r4/artikel/6/6475/1.html>.
<http://www.heise.de/tp/r4/artikel/8/8778/1.html>.
<http://www.ifross.de>
http://www.ifross.de/ifross_html/art6.html.
http://www.ifross.de/ifross_html/art9.html.
http://www.ifross.de/ifross_html/lizenzcenter.html.
<http://www.ipp.mpg.de/~iik/linux.html>.
<http://www.isc.tamu.edu/~lewing/linux/>.
<http://www.jurpc.de/aufsatz/20040181.htm>.
<http://www.justiz.nrw.de/IndexSeite/Organisation/projekte/lisa.html>.
<http://www.langenscheidt.de/?fremdwb=lizenz>.
http://www.law.cornell.edu/uscode/html/uscode15/usc_sec_15_00002301----000-.html.
<http://www.lexikon-definition.de/Freiheit.html>
<http://www.li.org/linuxhistory.php>.
<http://www.linux10.org/history/>.
<http://www.linux.com/articles/51613>.
<http://www.linux-community.de/Neues/story?storyid=1651>.
<http://www.linuxinsider.com/story/38172.html>.
<http://www.linuxinsider.com/story/38172.html>.
<http://www.linuxinsider.com/story/38172.html>.
<http://www.linux-magazin.de/Artikel/ausgabe/1999/01/Halloween/halloween.html>
<http://www.linux-magazin.de/Artikel/ausgabe/2001/01/recht/recht.html>.
<http://www.linux-magazin.de/Artikel/ausgabe/2001/07/recht/recht.html>.
<http://www.linux-praxis.de/linux1/geschichte.html>.
<http://www.microsoft.com/resources/sharedsource/default.mspx>.
http://www.microsoft.com/resources/sharedsource/Initiative/speeches/mundie_model.mspx.
<http://www.microsoft.com/resources/sharedsource/Licensing/default.mspx>.
<http://www.mitlinux.de/ldap>.
<http://www.mozilla.org/about/>.
<http://www.mozilla.org/about/staff>).
<http://www.mozilla.org/html/MPL/Faq.html>.
<http://www.mozilla.org/MPL/missing.html>.
<http://www.mozilla.org/MPL/MPL-1.1-annotated.html>.
<http://www.mozilla.org/MPL/relicensing-faq.html>.
<http://www.opensource.org/advocacy/faq.php>.
http://www.opensource.org/docs/certification_mark.html#approval.
<http://www.opensource.org/docs/definition.php>.

http://www.opensource.org/docs/definition_plain.php.
<http://www.opensource.org/halloween/>.
<http://www.opensource.org/node/163>.
<http://www.openwatcom.com/index.php/History>.
<http://www.oreilly.com/catalog/opensources/book/toc.html>.
<http://www.patente.bmbf.de/de/pdf/software.pdf>.
<http://www.patentfrei.de/>.
<http://www.popa.org/pdf/misc/epocipo2007.pdf>.
<http://www.pro-linux.de/news/2004/6344.html>.
<http://www.redhat.com/about/corporate/timeline.html>.
<http://www.sensortime.com/Diehr81-de.html>.
<http://www.share.org>.
<http://www.slackware.com/book/index.php?source=x68.html>
<http://www.sleepycat.com/company/licensing.html>.
http://www.softpanorama.org/Copyright/License_classification/social_dynamics_of_BSD_and_GPL.shtml.
http://www.softwarehistory.org/history/d_50s.html.
<http://www.unc.edu/~unclng/public-d.htm>.
http://www.softwarehistory.org/history/d_60s.html.
<http://www.stallman.org/#serious>.
http://www.sybase.com/about_sybase/history.
<http://www.trolltech.com/licenses/gpl-annotated.html>.
<http://www.trolltech.com/products/qt/licensing.html>.
<http://www.tucows.com>.
<http://www.usenet-abc.de/whatis.htm>.
<http://www.usip.com/articles/1st2fil.htm>.
<http://www.w3.org/2005/01/timelines/timeline-2500x998.png>.
<http://www.w3c.org/consortium>.
<http://www.w3.org/DesignIssues/LinkLaw.html>.
<http://www.wien.gv.at/ma14/wienux.html>.
http://www.wipo.int/meetings/2006/scp_of_ge_06/en/.
<http://www.wipo.int/patent/law/en/harmonization.htm>.
http://www.wipo.int/treaties/en/ip/berne/trtdocs_wo001.html#P82_10336.
<http://www.wxwindows.org/newlicen.htm>.
<http://www.x.org>.
http://www.x.org/Downloads_terms.html.
<http://www.x.org/wiki>.
<http://www.zope.com/license/>.
<http://www.zope.com/Marks>.

<http://www.zope.de/ueber-zope/was-ist-zope>.

<http://www.zope.org/Resources/License/>.

<http://www-128.ibm.com/developerworks/library/os-cplfaq.html>.

<http://www2.linuxjournal.com/article/2098>.

<http://www2.linuxjournal.com/article/2559>.

<http://wwwai.wu-wien.ac.at/Publikationen/Janko/paper/main3/node5.html>.

www.ifross.de/Fremdartikel/VortragLinuxTag2003.pdf.

www.linuxmark.org/.

JUDIKATURVERZEICHNIS

OGH:

OGH 02.03.1955,	1 Ob 81/55	= SZ 28/60: Wiederruf/laesio enormis.
OGH 14.02.1958,	3 Ob 577/57	= ÖBI 1958, 71 (73)
OGH 06.12.1961,	5 Ob 399/61	= SZ 34/184.
OGH 08.03.1966,	8 Ob 26/66	= 39/46; Widerruf.
OGH 19.01.1969,	5 Ob 260/68	= SZ 5/42, Widerruf.
OGH 07.02.1969,	7 Ob 99/69	= SZ 42/105 = EvBI 1970/147 = JBI 1970, 198.
OGH 01.09.1970,	4 Ob 336/70	– ZahnärztekammerG I – ÖBI 1970, 146.
OGH 07.03.1978,	4 Ob 317/78	– Stichwörterverzeichnis – ÖBI 1978, 107
OGH 04.10.1979,	7 Ob 727/79	= JBI 1980, 318.
OGH 03.12.1981,	8 Ob 530/81,	= JBI 1983, 255.
OGH 02.03.1982,	4 Ob 427, 428/81	– Blumenstück – ÖBI 1982, 164.
OGH 31.05.1983,	4 Ob 311/83	– Testkäufer – ÖBI 1983, 129.
OGH 14.09.1983,	3 Ob 574/83	= RdW 1984, 43;
OGH 10.07.1984,	4 Ob 337/84	– „Mart Stam-Stuhl I“ – MR 1/92, 21 (<i>Walter</i>) = ÖBI 1985, 24 = GRUR Int 1985, 684.
OGH 09.05.1985,	6 Ob 836/83	= HS 16.609 = RdW 1985, 306 = JBI 1986, 172;
OGH 16.01.1986,	6 Ob 808/83	= JBI 1986, 526 (<i>Pfersmann</i>) = RdW 1986, 174.
OGH 19.05.1987,	4 Ob 323/86	= EDV&Recht 1/86, 26.
OGH 26.01.1989,	8 Ob 620/88	= RdW 1989, 189 = ARD 4086/19/89 = JBI 1989, 383.
OGH 12.09.1989,	4 Ob 76/89	= ÖBI 1990, 136 – Happy Skiing – MR 1989, 210 (<i>Walter</i>).
OGH 06.11.1990,	4 Ob 145/90	„Oberndorfer Gschichtn“ = ÖBI 1991, 1188 = MR 1990, 230 (<i>Walter</i>).
OGH 07.04.1992	4 Ob 36/92	– „Bundesheer-Formblatt“ – MR 1992, 199 (<i>Walter</i>) = ÖBI 1992, 81 = SZ 65/51 = WBI 1992, 340 = EvBI 1993/36 = ÖJZ NRsp 1992/183, 184 = GRUR Int 1993/565.
OGH 16.06.1992	4 Ob 65/92	– Übungsprogramm – EDVuR 1992/2, 133 = JBI 1993, 116 = ÖBI 1992, 281 = SZ 65/89.
OGH 17.09.1992,	7 Ob 579/92	= WBI 1993, 95 = ecolex 1993, 18.
OGH 29.09.1992,	4 Ob 66, 67/92	– MERCEDES – Teyrosowsky – ÖBI 1992, 273 = MR 1992, 252 = ecolex 1993, 99.
OGH 27.10.1992,	5 Ob 124/92	= JBI 1993, 312.

- OGH 08.06.1993, 4 Ob 53/93 = MR 1993, 187 (*Walter*).
- OGH 14.12.1993, 4 Ob 157/93 – TÜV I – ÖBI 1994, 85.
- OGH 11.07.1995, 4 Ob 59/95 – New Yorker – ÖBI 1996, 141 = ecolex 1995, 817
= wbl 1995, 511
- OGH 29.05.1996, 3 Ob 2004/96v HS 27.422 = ecolex 1996, 743 = JBI 1997, 458
(*Staudegger*) = RdW 1997, 73 = ÖBA 1997, 214.
- OGH 17.12.1996, 4 Ob 2363/96w – Head-Kaufvertrag – ÖBI 1997, 256 = MR 1997,
93 (*Walter*) = SZ 69/283 = wbl 1997, 175.
- OGH 22.04.1997, 4 Ob 96/97i – Ramtha – ÖBI 1998, 53 = ecolex 1997, 681
(*Schanda*).
- OGH 14.10.1997, 5 Ob 504/96 = JBI 1998, 577.
- OGH 14.10.1997, 5 Ob 504, 505/96, = ecolex 1998, 127.
- OGH 28.10.1997, 4 Ob 304/97b – „einzigartiges“ EDV-Programm – ÖBI 1999, 57 =
MR 1998, 72 (*Walter*) = wbl 1998, 181.
- OGH 26.05.1998, 4 Ob 63/98p – Rauchfänge – MR 1998, 197 (*Walter*) = JBL 1998,
793 (*Mahr*).
- OGH 24.06.1998, 3 Ob 2427/96z = ecolex 1999, 16.
- OGH 12.08.1998, 4 Ob 193/98f – Wiener Gruppe – MR 1998, 287 (*Walter*) =
GRUR Int 1999, 360.
- OGH 24.11.1998, 4 Ob 266/98s – Tabasco – ecolex 1999, 134 = RdW 1999, 78 = ÖBI
1999, 124.
- OGH 24.11.1998, 4 Ob 292/98i – Mittelschulatlas – ecolex 1999, 167 (409, *Tahedi*) =
MR 1999, 171 (*Walter*).
- OGH 04.02.1999, 4 Ob 305/98a – Red Puma – ÖBI 1999, 191.
- OGH 04.02.1999, 4 Ob 348/98z – Ford-KG – ÖBI 1999, 295.
- OGH 13.04.1999, 4 Ob 17/99z – fremde Waren = ÖBI 1999, 283.
- OGH 02.09.1999, 2 Ob 300/97z = JBI 2000, 249 = ecolex 1999, 821.
- OGH 13.09.1999, 4 Ob 151/99f – Roll up – ÖBI 2000, 133 (*Kucsko*) = MR 1999, 343
(*Walter*).
- OGH 09.11.1999, 4 Ob 282/99w = MR 1999, 346;
- OGH 14.03.2000, 4 Ob 41/00h – Zahnarztprogramm – MR 2000, 312 (*Walter*).
- OGH 21.03.2000, 4 Ob 77/00b = MR 2001, 171 (*Walter*).
- OGH 12.04.2000, 4 Ob 106/00t – Tiroler Tageszeitung – MR 2000, 173 (*Frauenberger*).
- OGH 23.05.2000, 4 Ob 30/00 – Handwerkerpaket WIN 2.3 – ecolex 2000/291 =
MR 2000, 249 = RdW 2000/651 = ÖBI 2001, 141.
- OGH 04.07.2000, 4 Ob 171/00a – Glückwunschkartenmotive – MR 2000, 318
(*Walter*).
- OGH 17.08.2000, 4 Ob 178/00f – BOSS-Brillen II. – ecolex 2000/352 = WBI 2001/34 =
ÖBI 2001/89.

- OGH, 19.12.2000, 4 Ob 274/00y – jobmonitor.com = ÖBI 2001, 164
OGH 19.12.2000, 4 Ob 256/00a – steuerprofi.at – wbl 2001, 237.
OGH 12.06.2001, 4 Ob 139/01x – täglichalles.at – MR 2001, 245 (*Korn*) = ecolex
2001, 923 (*Schanda*) = wbl 2001, 540.
OGH 10.07.2001, 4 Ob 155/01 – C-Villas – MR 2001, 311 = ÖBI 2003, 252 (254) =
EvBI 2002/7 = RdW 2001/750 = ecolex 2001/352
(*Schanda*) = ZUM-RD 2002, 135 = GRURInt 2002,
452.
OGH 13.11.2001, 4 Ob 249/01y – Wirtschaftskurier – MR 2002, 101.
OGH 29.01.2002, 4 Ob 279/01k – Aufzugsanlagen – MR 2002, 156 (*Walter*).
OGH 13.03.2002, 4 Ob 39/02t – kunstNET – ÖBI-LS 2002/125 = ecolex 2002, 597
(*Schanda*).
OGH 09.04.2002, 4 Ob 77/02f – Geleitwort – MR 2002, 387.
OGH 28.05.2002, 4 Ob 108/02i – Figurstudio – MR 2002, 236 (*Walter*).
OGH 29.08.2002, 8 Oba 130/02d = ARD 5377/1/2003.
OGH 15.10.2002, 4 Ob 209/02t – Brühl – WBI 2003/95 = ÖBI 2003/87 = EvBI 2003/26
OGH 18.02.2003, 4 Ob 10/03d – More II – ÖBI-LS 2003/60.
OGH 29.04.2003, 4 Ob 57/03s – "Die Puppenfee" – MR 2003, 239
RdW 2003/617, 700 (LS) = ecolex 2004, 45
(*Schumacher*).
OGH 17.12.2003, 7 Ob 275/03x = JBI 2004, 449 = ZfRV 2004, 20 = IHR 2004, 148 =
RdW 2004, 252.
OGH 14.01.2004, 7 Ob 259/03v = immolex 2004/93.
OGH 19.10.2004, 4 Ob 182/04z = ecolex 2005/370, 776.
OGH 24.05.2005, 4 Ob 63/05a = MR 2005, 252 (*Walter*).
OGH 12.07.2005, 4 Ob 45/05d – TerraCAD – MR 2005, 379.
OGH 03.08.2005, 9 Ob 81/04h = MR 2005, 480.
OGH 12.07.2006, 4 Ob 38/06a – Shopping City – wbl 2006, 253.
OGH 20. 03. 2007, 17 Ob 3/07a – Immoeast - JusGuide 2007/21/4736.

VwGH:

VwGH 07.09.2006, 2006/16/0054 = MR 2006, 408.

OLG:

OLG Wien 24.05.1976, 1 R 118/76 – MusiCassetten – ÖBI 1976.

OLG Wien 08.08.1986, 3 R 101/86 = EDV&Recht 4/86, 34.

OLG Wien 18.04.1986, 4 R 48/86 – UIP-Repertoire – MR 1986, 18 (*Walter*).

BA:

BA 08.09.1983, B 31/82 – Isolierglas – ÖBI 1984, 114.

BVA:

BVA 07.11.2003, 14N-91/03-29 (A2 Tunnelkette Pack).

Deutschland:

BGH 04.11.1987 – VIII ZR 314/86 = NJW 1988, 406 (409).

LG München I, 19.05.2004, Az 21 O 6123/04.

LG München I, 24.07.2007, Az 7 O 5245/07.

BAG 30.04.1965, 3 AZR 291/63 – Abdampferwertung – GRUR 1966, 88 (*Vollmer*).

EuGH:

EuGH 19.04.1994, Rs C-331/92 (*Gestion Hotelera Internacional*) Slg 1994, I-1329.

EuGH 24.01.1995, Rs C-359/93 (*UNIX*), Slg 1995, I-157.

EuGH 07.12.2000, Rs C-324/98 (*Teleaustria*), Slg. 2000, I-10745.

EuGH 04.10.2001, Rs C-517/99 (*Bravo*), Slg 2001, I-6959 = ÖBI 2002, 105 = *ecolex* 2002, 35 (*Schanda*) = *MarkenR* 2001, 403 = *GRUR Int* 2002, 145.

EuGH 18.10.2001, Rs C-19/00 (*SIAC*), Slg 2001, I-07725.

EuGH 30.05.2002, Rs C-358/00 (*Buchhändler*), Slg 2002, I-4685

EuGH 18.06.2002, Rs C-299/99 (*Philips vs Remington*), Slg. 2002, I-5475 = *GRUR Int.* 2002, 842 = ÖBI 2003, 55 (*Gamerith*) = *GRUR* 2002, 804.

EuGH 17.09.2002, Rs C-513/99 (*Concordia Bus*), Slg 2002, I-07213.

EuGH 23.01.2003, Rs C-57/01 (*Makedoniko*) Slg. 2003, I-1091.

EuG:

EuG 07.02.2002, Rs T-88/00 (*Stabtaschenlampen*), Slg 2002, II-467 = *GRUR Int* 2002, 288 und 531.

EuG 27.02.2002, Rs T-34/00 (*EUROCOOL*), Slg 2002, II-683 = *wbl* 2002, 218.

EuG 20.03.2002, Rs T-355/00 (*TELE AID*), Slg 2002, II-1939 = *wbl* 2002, 271 = *GRUR Int* 2002, 747.

USA:

Diehr, 450 U.S. 175, 209 U.S.P.Q. (BNA) 1 (1981) = *GRUR Int* 1981, 646;

ABBILDUNGSVERZEICHNIS

- Abb 1:** Eigene Darstellung nach:
<http://libresoft.dat.escet.urjc.es/html/downloads/woss-icse-2004.pdf>.
- Abb 2:** Eigene Darstellung nach:
<http://libresoft.dat.escet.urjc.es/html/downloads/woss-icse-2004.pdf>.
- Abb 3:** Eigene Darstellung nach:
<http://libresoft.dat.escet.urjc.es/html/downloads/woss-icse-2004.pdf>.
- Abb 4:** <http://www.fsf.org/licensing/essays/category.png>.
- Abb 5:** <http://de.wikipedia.org/wiki/Bild:Copyleft.svg>.
- Abb 6:** Eigene Darstellung.
- Abb 7:** Eigene Darstellung, nach:
Jaeger/Koglin/Kreutzer/Metzger/Schulz, Die GPL kommentiert und erklärt 41.
- Abb 8:** Eigene Darstellung.
- Abb 9:** Eigene Darstellung.
- Abb 10:** Eigene Darstellung.
- Abb 11:** <http://de.wikipedia.org/wiki/Bild:Linux-Distribution.svg>.
- Abb 12:** Abbildung nicht mehr abrufbar, da *freeBSD* ein neues Logo bekommen hat.
- Abb 13:** <http://www.isc.tamu.edu/~lewing/linux/>.
- Abb 14:** <http://www.isc.tamu.edu/~lewing/linux/>.
- Abb 15:** <http://www.patente.bmbf.de/de/pdf/software.pdf>.
- Abb 16:** Nach *Välämäki*, The Rise of Open Source Licensing 38.
- Abb 17:** Eigene Darstellung, nach: <http://freshmeat.net/stats/#license>.