

Lizenz zum Coden

Was Open-Source-Lizenzen voneinander unterscheidet

Um Software quelloffen zu lizenzieren, gibt es Hunderte von Möglichkeiten. Aber was sind eigentlich die wesentlichen Unterschiede zwischen den Lizenzen und warum gibt es überhaupt so viele?

Von Sylvester Tremmel

Die Europäische Union, IBM, Microsoft, Nokia, Sun – wer als Organisation etwas auf sich hält, hat offenbar schon aus Prinzip eine eigene Open-Source-Lizenz entworfen. Solche Eitelkeiten kommen zwar vor, doch die gängigen Open-Source-Lizenzen sind aus guten Gründen entstanden. Prinzipiell sollen solche Lizenzen die Zusammenarbeit von Softwareentwicklern erleichtern. Die daraus resultierenden, teilweise globalen Entwicklergemeinschaften sind eine der großen Stärken quelloffener Software gegenüber Closed-Source-Entwicklungen.

Allerdings gibt es Hunderte von Open-Source-Lizenzen, die man vor allem bei Code-Integration und Software-Vertrieb im Blick behalten muss. Schwierig wird es, wenn unterschiedlich lizenzierte Code in einem Produkt kombiniert werden soll: Oft ist das zwar möglich, aber die Details sind nicht trivial – und manche Lizenzen haben schlicht widersprüchliche Anforderungen, sodass sie nicht zusammenkommen können.

Für die hohe Zahl verschiedener Lizenzen gibt es verschiedene Gründe: Zunächst spiegeln sich darin die unterschiedlichen Vorstellungen „Open Source“ und „freie“ Software wider (siehe Artikel auf

S. 62). Im Folgenden verwenden wir „quelloffen“ als – nicht ganz korrekten – Oberbegriff. Hinzu kommt, dass es von vielen Lizenzen verschiedene Versionen gibt, die unter anderem Unklarheiten ihrer Vorgänger beheben, unerwünschte Schlupflöcher schließen oder Entwicklungen der Softwaretechnik Rechnung tragen.

Einige Lizenzen sind viel gebräuchlicher als der große Rest, der aus relativ selten verwendeten Exoten besteht. Häufig trifft man etwa auf Versionen der GNU General Public License (GPL) oder auf eine moderne Abwandlung der BSD-Lizenz. Ebenfalls weit verbreitet sind Apache- und MIT-Lizenzen, die Mozilla Public License (MPL) oder Varianten der GPL wie die LGPL (GNU Lesser General Public License).

Was ist erlaubt?

Ganz grob lassen sich Lizenzen danach unterteilen, ob sie „permissive“ sind oder „Copyleft“ forcieren. Lizenzen mit Copyleft erzwingen Folgendes: Wenn Programmcode verändert oder in andere Programme integriert wird, dann steht auch der so entstandene Quellcode unter derselben Lizenz. Das Projekt bleibt dadurch quelloffen und Verbesserungen und Anpassungen können wieder der Open-Source-Bewegung zugute kommen. Damit das funktioniert, erzwingen Copyleft-Lizenzen neben ihrer eigenen Weiterentwicklung noch etwas anderes: So lizenzierte Software darf nur weitergegeben werden, wenn auch ihr Quellcode zugänglich gemacht wird – zumindest den Empfängern der Software.

Die bekannteste Copyleft-Lizenz ist die GPL, mit der beispielsweise der Linux-Kernel lizenziert wird. Nimmt ein Unternehmen Verbesserungen an dessen Quell-

code vor, so darf es auch diesen veränderten Kernel und seinen Code nur unter den Bedingungen der GPL weitergeben. Zum Beispiel darf der Kernel nur vertrieben werden, wenn Käufern auch sein Quellcode zugänglich gemacht wird – sodass die Käufer selber weitere Verbesserungen vornehmen können.

Copyleft und Copyright

Innerhalb der Copyleft-Lizenzen gibt es feinere Unterteilungen, die tendenziell als starkes und schwaches Copyleft bezeichnet werden. Schwache Lizenzen wie die LGPL oder MPL erlauben etwa, quelloffene Softwarebibliotheken in einem proprietären Programm zu nutzen. Starke Copyleft-Lizenzen wie die GPL werten das schon als unzulässige Kombination. Ein ebenfalls starkes und dazu noch besonders weitreichendes Copyleft erzwingt die Affero GPL (AGPL). Selbst wenn Nutzer lediglich über ein Netzwerk mit AGPL-Software interagieren, also „Software as a Service“ nutzen, muss der Quellcode dieser Software an sie weitergegeben werden.

Copyleft hat übrigens nichts mit der Abwesenheit von Copyright zu tun. Im Gegenteil, Copyleft-Lizenzen funktionieren nur durch das Urheberrecht. Sie tragen ihren Namen, weil sie den Zweck des Copyrights umkehren, nicht weil sie dem Copyright an sich entgegenstehen. Der tatsächliche Verzicht auf Copyright läuft stattdessen unter dem Begriff „Public Domain“. Ein Schritt, der in Jurisdiktionen wie der deutschen streng genommen nicht möglich ist.

Die andere große Kategorie von Lizenzen sind solche, die eben kein Copyleft erzwingen. Neben dem englischen „permissive“ bezeichnet man sie auch mit der Übersetzung „freizügig“. Solche Lizenzen



erlauben, Abwandlungen und Kombinationen von Programmen unter beliebigen – auch proprietären – Lizenzen zu veröffentlichen. Sie erzwingen auch nicht, dass Software zusammen mit ihrem Quellcode ausgeliefert wird. Ein solches Gebot wäre unter einer freizügigen Lizenz ohnehin wirkungslos: Das Programm ließe sich einfach etwas verändern, um dann mit einer anderen Lizenz, die keine Quellcode-Weitergabe erfordert, ausgeliefert zu werden.

Entwickler geben also mehr Kontrolle über ihren Code ab, wenn sie freizügige Lizenzen nutzen. Bekannte Vertreter dieses Lizenztyps sind die BSD- und Apache-Lizenzen. Auch das erwähnte Entlassen von Software in die Public Domain ist eine extreme Form von freizügiger „Lizenzierung“.

Seid fruchtbar und mehret euch

Darüber hinaus gibt es von vielen Lizenzen verschiedene Versionen. Die entstehen in der Regel, weil mit der Zeit gewisse Defizite in bestehenden Lizenzen auffallen oder neue Anwendungsfälle entstehen, die die Lizenzen nicht berücksichtigten. Ein bekanntes Beispiel dafür waren digitale Videorekorder der Firma TiVo. Diese nutzten quelloffene Software unter Version 2 der GPL und machten Kunden den Quellcode zugänglich. Nutzer konnten dadurch – wie von der GPL intendiert – eigene Varianten der TiVo-Software erzeugen. Allerdings weigerte sich die TiVo-Hardware, solche Varianten auszuführen, was sie praktisch nutzlos machte. Das wurde vielfach als Verstoß gegen den Geist der GPL aufgefasst und von Gegnern als Tivoisierung bezeichnet.

Version 3 der GPL enthält deshalb Klauseln, die solches Verhalten unterbinden – neben anderen Änderungen, etwa zum Schutz vor den Auswirkungen von Softwarepatenten. Allerdings machen neue Versionen einer Lizenz alte nicht zwingend obsolet, sondern sind nur eine weitere Alternative. Während alte Versionen von Software mangels Support mehr



Alte Garde: Richard Stallman ist unter anderem Gründer der FSF, ursprünglicher Autor der GPL und Wegbereiter des Copyleft-Konzepts.

oder weniger schnell von der Bildfläche verschwinden, bleiben Lizenzen nutzbar und gültig. Linus Torvalds zum Beispiel, als Hüter des Linux-Kernels recht einflussreich, hält nicht allzu viel von der GPLv3 und bevorzugt Version 2.

Kompatibilitätsprobleme

Das Hauptproblem mit der wachsenden Anzahl von Lizenzen und Lizenzvarianten ist, dass diese nicht unbedingt miteinander kompatibel sind. Zum Beispiel lassen sich reine GPLv2- und reine GPLv3-Programme nicht in einem Softwareprojekt kombinieren: Beide Versionen der Lizenz erfordern jeweils, dass genau sie auch für das Gesamtprojekt zum Einsatz kommen. Eine Kombination würde einen Lizenzwechsel und damit die Zustimmung aller Copyright-Inhaber erfordern, die Code zu zumindest einem der Projekte beigetragen haben – bei alten, großen Projekten fast ein Ding der Unmöglichkeit.

Ironischerweise begegnet man dem Problem unter anderem mit zusätzlichen Lizenzvarianten: Ein Vorteil der GPLv3 gegenüber der GPLv2 ist ihre Kompatibilität mit mehr Drittlizenzen. Und die heute verbreiteten 2- und 3-Klausel-Varianten der BSD-Lizenz sind, im Unter-

schied zur kaum noch anzutreffenden 4-Klausel-Variante, kompatibel mit der GPL (in beiden Versionen).

Alternativ lässt sich Code auch von vornherein unter mehr als einer Lizenz veröffentlichen, was als Mehrfachlizenzierung bekannt ist. Um ein mehrfachlizenziertes Programm mit einem anderen zu kombinieren, genügt es, wenn nur eine der verwendeten Lizenzen mit der Lizenz des anderen Programms kompatibel ist. Eine besonders häufige Variante solcher Mehrfachlizenzierung ist, Code unter die GPLv2 beziehungsweise v3 mit dem Zusatz „oder später“ zu stellen. Die Software ist dann sozusagen automatisch mit nachfolgenden Versionen der GPL mehrfachlizenziert und Kombinationen von Code mit unterschiedlichen Versionen sind kein Problem.

Übersicht verschaffen

Die Free Software Foundation (FSF), die Organisation hinter den GPL-Lizenzen, pflegt Listen, um Übersicht über die Situation zu behalten. Auch die Open Source Initiative (OSI), als zentrale Hüterin des Labels „Open Source“, bietet entsprechende Informationen wie auch verschiedene Linux-Distributionen (alle Links unter ct.de/yvvt).

Die Organisationen geben Einschätzungen ab, welche Lizenzen mit welchen anderen kompatibel sind und unter welcher Lizenz man neue Projekte grundsätzlich stellen sollte. Außerdem sollen solchen Listen neuen Lizenzen vorgreifen, die nur aus Unkenntnis bestehender Lizenzen geschaffen werden.

Letztlich ist die Situation zwar nicht ideal, aber – und das darf man nicht vergessen – deutlich besser als im Closed-Source-Umfeld. Dank Open-Source-Lizenzen können Firmen ohne vorherige Vertragsverhandlungen zusammenarbeiten und Privatpersonen können frei mit von ihnen gekauften Produkten experimentieren – inklusive der Software dieser Produkte. (syt@ct.de) **ct**

Lizenz-Übersichten: ct.de/yvvt



Bunte Tüte: Varianten der GPLv3 haben sogar eigenen Logos, um damit zu werben. Die meisten anderen bekannten Lizenzen müssen ohne eigenes Logo auskommen.