



# Lizenzfragen bei Open-Source- Learning-Management- Systemen

Handlungsleitfaden erstellt von Dr. Till Kreutzer, iRights.Law,  
im Auftrag des Stud.IP e.V. und des ILIAS open source e-Learning e.V.  
für die Initiative Open Source LMS

## Vorbemerkung

Die beiden in Deutschland weit verbreiteten Open-Source-Learning-Management-Systeme (OS-LMS) ILIAS und Stud.IP werden unter der GNU General Public License (GPL Version 2 und 3) lizenziert. Nach dieser Lizenz für Freie und Open Source Software (FOSS) können die Systeme ohne individuelle Erlaubnis für jeden Zweck und von jedem genutzt werden. Sie sind zudem kostenfrei verfügbar. Die GPL soll den Nutzern Freiheiten gewähren und gleichzeitig die Interessen der Rechteinhaber am Programmcode schützen. Sie enthält daher Rechte (die den Nutzern eingeräumt werden) und Pflichten (die die Nutzer bei der Verwendung der OS-LMS einhalten müssen). Neben Rechten und Pflichten für Endnutzer und Distributoren enthält die Lizenz Regeln für Bearbeiter, also v. a. Contributors, die Beiträge zum Code beisteuern.

Über die Rechte und Pflichten der GPL herrscht oft Unklarheit. Die Lizenz ist komplex und in Grenzfällen schwer zu interpretieren. Auch kursieren in Mailinglisten und Fachforen allerhand Mythen und Halbwahrheiten über die Grundfunktionen der Lizenz und ihrer Ziele. Die nachstehenden Ausführungen sind zweigeteilt: Im ersten Teil werden einige ausgewählte grundlegende Aspekte erläutert, deren Verständnis für die konkreten Fragen, die im zweiten Teil beantwortet werden, erforderlich ist. Die FAQ-Liste im zweiten Teil wurde von den Vereinen ILIAS open source e-Learning e.V. und Stud.IP e.V. erstellt. Wir haben die Fragen soweit möglich unverändert übernommen. Sie wurden allenfalls leicht abgewandelt oder präzisiert, sofern dies aus fachlicher Sicht notwendig erschien.

Wir verwenden in diesem Leitfaden immer dort, wo nicht ausdrücklich Männer, Frauen oder Menschen anderen Geschlechts angesprochen werden, das generische Maskulinum, das keine Festlegung auf das natürliche Geschlecht beinhaltet und niemanden ausschließt.

# Teil 1: Grundlegende Aspekte

## 1. Allgemein zum Copyleft der GPL

Als „Copyleft“ bezeichnet man Lizenzregeln, mit denen erreicht werden soll, dass die Nutzungsfreiheit einer Open Source Software in all ihren Entwicklungsstadien erhalten bleibt. Sie schreiben daher vor, dass die jeweilige Software in all ihren Fassungen nur unter der Ausgangslizenz verbreitet und geteilt werden darf. Sie sagen sinngemäß: „Sie dürfen diese Software nutzen und verändern. Wenn Sie jedoch eine veränderte Fassung („Bearbeitung“, „*modification*“) in Umlauf bringen, verbreiten oder teilen, müssen Sie Ihre Version ebenfalls unter der Lizenz zur Nutzung freigeben, die für den bearbeiteten Code gilt (mit anderen Worten: die GPL).“ Diese Verpflichtung impliziert, dass Entwickler von Bearbeitungen ihre Versionen nicht unter Regeln verbreiten dürfen, die gegenüber der Ausgangslizenz Einschränkungen der Rechte oder Verschärfungen der Pflichten vorsehen.

Wie weit gehend und in welchen Fällen das Copyleft Nachnutzer an die Ausgangslizenz bindet, hängt von seiner konkreten Ausgestaltung ab (also dem Wortlaut der jeweiligen Lizenzklausel). Es gibt Lizenzen mit eingeschränktem Copyleft-Effekt, die nur in manchen Konstellationen und bei bestimmten Code-Kombinationen greifen. Bei der Mozilla Public Licence (MPL) beispielsweise greift die Lizenzbindung nicht, wenn Modifikationen in eigenständigen Dateien gespeichert vertrieben werden. Wird also der Code einer Modifikation oder eines Patches o. ä. nicht mit dem Ausgangscode vermischt, ist der Bearbeiter nicht an dessen Lizenz gebunden und muss bei der Verbreitung seines Beitrags nicht die MPL verwenden. Die GPL ist diesbezüglich strenger und weniger präzise. Sie besagt im Wesentlichen, dass jede Art von *modification* nur unter der Ausgangslizenz (GPL) verbreitet werden darf. Präzisierungen bzgl. der insofern allentscheidenden Frage, was eine *modification* ist, enthält die Lizenz aber nur sehr spärlich. Vor allem in Grenzfällen besteht daher erheblicher Interpretationsspielraum. Dieses Problem hat rechtspraktisch eine große Bedeutung: Verstößt ein Nachnutzer gegen das Copyleft, weil er die Regel „falsch“ interpretiert, verliert er seine Nutzungsrechte am GPL-Code. Jeder Vertrieb von Bearbeitungen, der nicht unter der GPL erfolgt, ist eine Urheberrechtsverletzung, die Schadensersatzansprüche und andere Sanktionen nach sich ziehen kann.

Nach einem weit verbreiteten Irrtum besagt die Copyleft-Regelung, dass jegliche Anpassung, Modifikation, Bearbeitung des Programms *veröffentlicht werden muss*. Der Auslöser für dieses Missverständnis ist wohl die Idee, dass die Nutznießer des freien Codes ihre Beiträge ebenfalls wieder an die Gemeinschaft zurückgeben und freigeben müssen. Eine Veröffentlichungspflicht sehen jedoch auch die strengsten Copyleft-Lizenzen nicht vor. Copyleft bedeutet nicht, dass der Bearbeiter veröffentlichen muss. Es bedeutet vielmehr, dass wenn der Bearbeiter seine Fassung/seine Modifikation teilt oder veröffentlicht, er an die Ausgangslizenz gebunden ist. Es handelt sich also nicht um eine Pflicht zur Veröffentlichung, sondern um eine Pflicht zur Vergabe einer bestimmten Lizenz, wenn veröffentlicht oder geteilt wird.

## 2. Wirksamwerden des Copylefts aus der GPL

Copyleft und andere wichtige Vertriebspflichten (z. B. zur Bereitstellung des Quellcodes) werden also erst und nur dann wirksam, wenn das Programm geteilt, verbreitet, veröffentlicht wird. In der GPLv2 werden die Weitergabevorgänge pauschal als *Distribution* bezeichnet. Die Copyleft-Klausel findet sich in Ziff. 2b). Sie lautet:

*“You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.”*

In der GPLv3 wurden die Begriffe geändert. Handlungen der Weitergabe werden als *propagate* bzw. *convey* bezeichnet (s. die Definitionen in Ziff. 0). Nachstehend soll zur Vereinfachung pauschalisierend der Begriff *Distribution* zur Bezeichnung von Weitergabevorgängen verwendet werden, die die Vertriebspflichten in Open-Source-Lizenzen auslösen. Soweit Unterschiede zwischen GPLv2 und v3 bestehen, wird hierauf hingewiesen.

Generell wird mit dem Begriff der Distribution eine Überlassung von Programmcode an einen Dritten (eine Person oder ein Unternehmen) umschrieben. Diese Überlassung kann darin liegen, dass die Software auf einem Server bereitgestellt wird, von dem sie von Dritten heruntergeladen werden kann oder durch individuelle Überlassungsvorgänge, z. B. durch Vertrieb von Datenträgern oder Übersendung per E-Mail.

Keine Distribution liegt dagegen vor, wenn eine Software nur intern, also innerhalb eines Unternehmens oder einer Behörde verwendet wird.

Zudem sieht eine weit verbreitete Ansicht in der Online-Bereitstellung von Software als Service, also etwa bei Webservices, SaaS, Cloud Computing oder ASP, keine Distribution. Hiernach müssen also die Lizenzpflichten, v. a. die Offenlegungspflicht des Sourcecodes, bei diesen Nutzungsformen nicht eingehalten werden. In dieser Frage ist zwischen GPLv2 und GPLv3 zu unterscheiden. Ob die Nutzung als Service dazu führt, dass der Sourcecode der Software und ihrer *derivative works* offengelegt werden müssen, ist in Bezug auf die GPLv2 umstritten. Nach der GPLv3 ist dies dagegen eindeutig zu verneinen<sup>1</sup>.

Die wohl herrschende Rechtsauffassung sieht dies bei der GPLv2 genauso<sup>2</sup>. In der Bereitstellung einer Software als Dienst liege keine Distribution, da hierbei keine Kopien verteilt oder zugänglich gemacht werden. Vielmehr kann der Nutzer lediglich mit der Software interagieren und dessen Funktionen nutzen. Jedenfalls nach US-Rechtsverständnis liege hierin keine Distribution, so dass die Lizenzpflichten nicht eingehalten werden müssten. Die herrschende Auffassung geht von einem *ASP-loophole* aus, wie es auch in der GPLv3 enthalten ist (dort aber bewusst vorgesehen wurde).

Gegner dieser Auffassung argumentieren, dass zumindest nach deutschem Recht auch das Online-Bereithalten (die „öffentliche Zugänglichmachung“) lizenzpflichtig sei und dass die Einräumung der Nutzungsrechte in der GPL untrennbar an die Einhaltung (und damit die Wirksamkeit) der Vertriebspflichten geknüpft sei<sup>3</sup>.

Angesichts dieser konträren Auffassungen – die beide mit guten Argumenten vertreten werden können – ist eine eindeutige Antwort in Bezug auf Software, die unter der GPLv2 steht, nicht möglich. Nachstehend wird mit der herrschenden Auffassung in der „Open Source-Gemeinde“ davon ausgegangen, dass die Bereitstellung einer Software als Service ebenso wenig eine Distribution im Sinne der GPLv2 darstellt, wie eine *conveyance* nach der GPLv3. Mit anderen Worten wird hier davon ausgegangen, dass das Copyleft und die Vertriebspflichten beider GPL-Versionen nicht ausgelöst werden, wenn eine GPL-Software Dritten lediglich zur Nutzung als Service zugänglich gemacht wird.

---

1 Die GPLv2 stammt aus einer Zeit, in der die Nutzung von Software als Service technisch noch unbekannt war. Sie äußert sich dementsprechend nicht zu dieser Frage, was zu Auslegungsschwierigkeiten führt. Der Lizenztext der GPLv3 ist diesbezüglich dagegen eindeutig. Dass die Lizenzpflichten nicht wirksam werden, wenn die Software lediglich als Dienst angeboten wird, steht ausdrücklich in der Definition des Begriffes „convey“. Dieser wird in der GPLv3 anstelle von „distribute“ verwendet. Er wird in Ziff. 0 wie folgt definiert: „To “convey” a work means any kind of propagation that enables other parties to make or receive copies. **Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.**“ Als Alternative für Entwickler, die wollen, dass die Lizenzpflichten für ihre Software auch im Falle des reinen Server-Einsatzes wirksam werden, wurde die Affero GNU GPL (AGPL) geschaffen. Hierbei handelt es sich um ein Derivat der GPL mit einer zusätzlichen Klausel (Ziff. 13), die bei einer bloßen Bereitstellung der Software über ein Netzwerk dazu verpflichtet, den Sourcecode offenzulegen und verfügbar zu machen.

2 So v. a. die Auffassung der Free Software Foundation in Bezug auf die Auslegung der GPLv2 (die sich am US-amerikanischen Rechtsverständnis orientiert), siehe <http://www.gnu.org/licenses/gpl-faq.html#UnreleasedMods>.

3 Ballhausen: OpenSaaS: Using Free and Open Source Software as Software-as-a-Service, IFOSS Law review 6 2014, S. 61 (siehe <https://www.jolts.world/index.php/jolts/article/view/103>).

### 3. Copyleft beim gemeinsamen Vertrieb verbundener GPL-Software und anderen Komponenten

Die Copyleft-Verpflichtung gilt für den Vertrieb aller abgeleiteten Werke („*derivatives*“). Was ein Derivat ist, ist eine äußerst schwierige und im Detail umstrittene Frage. Besonders problematisch ist die Auslegung des GPL-Copylefts in Bezug auf die Distribution von Code-Kombinationen und von mittelbar abhängigen Programmen.

Grundsätzlich gilt, dass direkte Veränderungen von GPL-Code nur unter der GPL vertrieben werden dürfen. Hiermit gemeint sind Patches, Updates und andere Code-Modifikationen jeglicher Art. Anders ausgedrückt: Modifizierter GPL-Code und Programme, in denen GPL-Code verwendet wird, dürfen nur unter der GPL vertrieben werden. Das gilt beispielsweise immer auch dann, wenn GPL-Dateien und anderer Code zu einem einheitlichen Executable verbunden werden und dies verbreitet wird.

Abgesehen von diesen eindeutigen Fällen ist die Beurteilung schwierig und vom Einzelfall abhängig. Dies gilt v. a. für weniger unmittelbare Verbindungen von Software-Modulen. Beispielsweise ist fraglich, ob und unter welchen Umständen eine mit einer GPL-Applikation interagierende Software-Bibliothek – also an sich zwei eigenständige Programme – unter die GPL gestellt werden muss, wenn beide gemeinsam vertrieben werden.

Hier gilt als abstrakte Faustregel: Gemeinsam mit GPL-Code vertriebener Programmcode wird nur dann vom Copyleft erfasst, wenn er aus *inhaltlicher-funktionaler Sicht ein abgeleitetes Programm* ist und in *formaler Sicht nicht isoliert* vertrieben wird, sondern mit dem GPL-Code zu *einem Ganzen* „vermengt“ wird. Anders ausgedrückt: Der hinzugefügte Code bzw. das hinzugefügte Modul darf nicht mit GPL-Code vermischt überlassen werden (sondern muss z. B. in eigenen Dateien oder Containern gespeichert sein) und er darf nicht auf dem GPL-Code basieren (hiervon abgeleitet sein). Entscheidend sind damit einerseits der technische Bezug der Komponenten zueinander und andererseits, wie sie im Rahmen des Vertriebs miteinander verbunden werden.

Von folgender Faustformel geht die herrschende Rechtsliteratur aus<sup>4</sup>:

1. Programme oder Softwarebestandteile, die nicht voneinander abgeleitet sind, können unter unterschiedlichen Lizenzen verbreitet werden, auch gemeinsam auf einem Datenträger;
2. Programme oder Softwarebestandteile, die nicht voneinander abgeleitet sind, müssen aber dann insgesamt unter der GPL verbreitet werden, wenn sie (formal) ein »Ganzes« bilden;
3. Programme oder Softwarebestandteile, die voneinander abgeleitet sind, dürfen stets nur unter der GPL gemeinsam verbreitet werden.

Schwierigkeiten bereitet dabei insbesondere das Merkmal der inhaltlich-funktionalen Abhängigkeit. Dieser Aspekt wird zwar in der GPL angesprochen – und ist daher entscheidungsrelevant – wird aber in keiner Weise präzisiert. So ist beispielsweise bis heute die hoch umstrittene Frage offen, ob Kernel-Module für Linux nur unter der GPL vertrieben werden dürfen.

Als abstrakte Faustregel gilt: Sind zwei Programme als funktionale, voneinander abhängige Einheiten anzusehen, spricht dies dafür, dass sie voneinander abgeleitet sind<sup>5</sup>. Handelt es sich dagegen um multifunktionale Standardkomponenten (wie es bei (Java-)Bibliotheken häufig der Fall ist), ist indiziert, dass sie nicht voneinander abgeleitet sind. Eine konkrete Einschätzung kann aber letztlich nur im Einzelfall unter genauer Betrachtung aller relevanter Umstände getroffen werden.

---

4 S. ifrOSS (Hrsg.): Die GPL kommentiert und erklärt, Ziff. 2, Rz. 28 (<https://www.ifross.org/Druckfassung/Ziffer%202.pdf>) sowie Jaeger/Metzger, Open Source Software, 5. Auflage 2020, Rn. 58.

5 Hierbei kann sich die Folgefrage stellen, welche Programm von welchem abgeleitet wurde (schließlich ist es auch möglich, dass das GPL-Programm vom anderen Programm abgeleitet wurde).

## 4. Fallgruppen Copyleft

Basierend auf diesen Grundlagen haben Fachjuristen (Rechtsprechung gibt es kaum!) einige typische Konstellationen Fallgruppen zugeordnet.

Fallgruppe	Copyleft?
1. Vertrieb von GPL- und anderen Software-Modulen in einem Executable	⊕ Da jedenfalls keine formale Eigenständigkeit der Module
2. Integration von GPL-Code-Fragmenten in anderen Sourcecode	⊕ Da jedenfalls keine formale Eigenständigkeit des Codes
3. Patches, Forkes, Updates etc. einer GPL-Software	⊕ Typische Bearbeitung, das Ergebnis ist ein <i>derivative work</i> im eigentlichen Sinn
4. Plugins	⊕ ⊖ Einzelfallfrage, die nach den o. g. Kriterien geprüft werden muss (Handelt es sich um ein „eigenständiges“ Programm, etwa weil es generische Funktionen ausführt und/oder mit verschiedenen Anwendungen funktioniert? Werden Plugin und Anwendung formal separiert vertrieben? Enthält das Plugin Code der GPL-Anwendung <sup>6</sup> )
5. Anbindung von Anwendungen über Systemaufrufe (Pipes, Sockets, Queues)	⊖ Führt für sich genommen nicht dazu, dass die Anwendung als <i>derivative</i> des Betriebssystems anzusehen ist. Achtung aber bei nicht-separiertem Vertrieb (s. 1.).
6. Anbindung über Schnittstellen	⊖ Führt für sich genommen nicht dazu, dass die Anwendung als <i>derivative</i> des Betriebssystems anzusehen ist. Achtung aber bei nicht-separiertem Vertrieb (s. 1.).
7. Programmbibliotheken	⊕ ⊖ Hängt von der Art der Einbindung ab. Wird eine GPL-Bibliothek statisch verlinkt, entsteht beim Kompilieren ein einzelnes Executable. Dies darf nur unter der GPL vertrieben werden, weil zumindest keine formale Trennung des Codes gegeben ist. Bei dynamischer Verlinkung werden <i>library</i> und Applikation erst beim Nutzer kompiliert. Wenn die Bibliothek auch technisch eigenständig ist, ist ein Vertrieb unter unterschiedlichen Lizenzen zulässig. Letzteres ist umstritten (anders z. B.: Die FSF zur GPL) <sup>7</sup> .
8. Nutzung von Tools, Output von GPL-Applikationen (z. B. Daten)	Der Output, die Arbeitsergebnisse, von Programmen, die unter der GPL stehen, wird vom Copyleft in aller Regel nicht erfasst <sup>8</sup> . Das gilt genauso für Daten, die bei der Programm Benutzung erzeugt werden, wie für Texte, die mit einer Textverarbeitung geschrieben werden. Auch mit einem GPL-Compiler kompilierter Code ist kein „abgeleitetes Werk“. Eine seltene Ausnahme kann hier vorliegen, wenn ein Compiler eigenen Code in die Kompilate einfügt. Solche (wie z. B. Bison) stehen jedoch in der Regel unter einer speziellen Lizenz, die den Copyleft-Effekt aufgrund des in die Output-Files übertragenen Codes ausschließt <sup>9</sup> .

6 Zu weiteren Einzelheiten siehe die Fragen in Teil 2, Nr. 3.

7 <https://www.gnu.org/licenses/gpl-faq.html#GPLStaticVsDynamic>

8 <https://www.gnu.org/licenses/gpl-faq.html#IfInterpreterIsGPL>

9 S. im Einzelnen ifrOSS (Hrsg.): Die GPL kommentiert und erklärt, Ziff. 2, Rz. 48 (<https://www.ifross.org/Druckfassung/Ziffer%202.pdf>).

## 5. GPL-Pflichten und Verträge zwischen Anbietern und Kunden

Die Pflichten aus der GPL – sofern sie denn ausgelöst werden – können durch Verträge mit Dritten (z. B. Kundenverträge) nicht ausgeschlossen oder eingeschränkt werden<sup>10</sup>. Dies verbietet sie ausdrücklich.

Siehe z. B. Ziff. 6 Satz 2 GPLv2:

*“You may not impose any further restrictions on the recipients’ exercise of the rights granted herein.”*

Ein Beispiel: Ein Entwickler will ein Open-Source-LMS anbieten, das er für bestimmte Nutzungszwecke modifiziert hat. Das LMS steht unter der GPLv2. Der Entwickler hat an dem Ausgangsprogramm nicht sämtliche Rechte. Der Kunde erhält die Rechte zur Nutzung des Systems – abgesehen von dessen hinzugefügtem Programmcode – daher nicht vom Entwickler, sondern von den Rechteinhabern am Ausgangsprogramm. Sie werden durch die und unter den Bedingungen der GPL eingeräumt. Die GPL verpflichtet den Entwickler (als Distributor), all seinen Kunden den Quellcode an seiner modifizierten Version zu überlassen und diese ebenfalls unter der GPL zu lizenzieren. Ein Vertrag mit einem Kunden, in dem dieser beispielsweise darauf verzichten soll, dass er die modifizierte Version nach der GPL verwenden kann oder ohne Überlassung des Quellcodes würde zu einer Einschränkung der Nutzungsbefugnisse und -freiheiten führen, die die GPL gewährt. Die Vorgaben der Lizenz würden verletzt. Nach Ziff. 4 der GPLv2 verliert der Entwickler hierdurch seine Nutzungsrechte und kann – von den Rechteinhabern am Open-Source-LMS – rechtlich in Anspruch genommen werden.

Ein anderes Ergebnis würde gleichermaßen dem Grundgedanken von Copyleft-Lizenzen widersprechen wie dem Willen der Rechteinhaber. Wäre dies möglich, könnte sich der Bearbeiter ohne Weiteres nach Gutdünken über die Lizenzpflichten hinwegsetzen, die sie zur Bedingung für die Nachnutzung aufgestellt haben. Das Copyleft-Prinzip und Open-Source-Lizenzen im Allgemeinen wären damit rechtlich wirkungslos.

## 6. Lizenzkompatibilität

Lizenzkompatibilität besteht, wenn bei der Kombination mehrerer Softwarekomponenten zu einem einheitlichen Ganzen sämtliche Lizenzpflichten eingehalten werden können<sup>11</sup>. Zwei Lizenzen sind kompatibel, wenn sie keine Pflichten enthalten, die die Einhaltung der Pflichten aus der jeweils anderen Lizenz unmöglich machen würde. Vor allem Copyleft-Lizenzen sind meist zueinander inkompatibel. Sogar die GPLv2 ist nicht mit der GPLv3 kompatibel. GPLv2- und GPLv3-Code können also nicht gemeinsam vertrieben werden, wenn hierdurch das Copyleft ausgelöst würde<sup>12</sup>. Beide Lizenzen sagen sinngemäß: „Wenn Du die Software mit anderer Software kombinierst, darfst Du die Kombination nur unter dieser Lizenz vertreiben.“ Da sich beide Lizenzen unterscheiden, verstößt man unweigerlich gegen eine der Lizenzen, wenn man die andere einhält. Stelle ich die Kombination des Codes unter die GPLv2 kann sie nicht gleichzeitig unter der GPLv3 stehen. Ich verstoße so also gegen die GPLv3.

---

<sup>10</sup> Dies gilt natürlich nur im Verhältnis zu Software, für die der Anbieter nicht selbst sämtliche Rechte besitzt. Der Rechteinhaber selbst kann mit seinem Computerprogramm „machen, was er will“. Er kann z. B. Dual-Licensing-Modelle verwenden, in denen die Software einerseits unter einer Open-Source-Lizenz angeboten wird und andererseits unter proprietären Lizenzen.

<sup>11</sup> S. die FAQ der FSF hierzu: <https://www.gnu.org/licenses/gpl-faq.html#WhatDoesCompatMean>

<sup>12</sup> Eine Ausnahme besteht dann, wenn die Lizenz eine „any later version“-Klausel enthält.  
Siehe hierzu: <https://www.gnu.org/licenses/gpl-faq.html#VersionThreeOrLater>

Die Free Software Foundation (FSF) bietet auf ihren Webseiten viele Informationen zur Kompatibilität der GNU-Lizenzen untereinander<sup>13</sup> als auch zu anderen Lizenzen<sup>14</sup>.

Als Faustregel gilt:

1. Copyleft-Lizenzen sind in der Regel zueinander inkompatibel. Manche Copyleft-Lizenz enthalten jedoch Öffnungsklauseln, gerade um Kompatibilität zu den GNU-Lizenzen zu ermöglichen (siehe z. B. Ziff. 3.2 der Eclipse Public License 2.0 – EPL 2.0).
2. Permissive Lizenzen ohne Copyleft (wie BSD, MIT usw.) sind oft auch mit strengen Copyleft-Lizenzen kompatibel.
3. Permissive Lizenzen ohne Copyleft sind untereinander immer kompatibel. Sie enthalten keine Einschränkungen hinsichtlich der Lizenzierung von Code-Kombinationen.

Bedenke: Lizenzen verändern sich. Eine Version kann mit einer anderen Lizenz kompatibel, eine andere Version inkompatibel sein. Es genügt also z. B. nicht zu schauen, ob die *Apache Public License* (APL) mit „der GPL“ kompatibel ist. Zu untersuchen sind die jeweils konkret involvierten Lizenzversionen, also in diesem Beispiel etwa, ob die APL 2.0 mit der GPLv3 kompatibel ist.

---

<sup>13</sup> <https://www.gnu.org/licenses/gpl-faq.html#AllCompatibility>

<sup>14</sup> <https://www.gnu.org/licenses/license-list.html.en#Introduction>

## Teil 2: FAQ zu GPL und Open-Source-LMS

### 1. Allgemeine Fragen zur GPL

**a) Müssen Änderungen an einer GPL-lizenzierten Software dem ursprünglichen Anbieter zur Verfügung gestellt werden?**

Nein. Die GPL enthält keine Pflicht, nach der Änderungen wieder veröffentlicht oder anderen zur Verfügung gestellt werden müssen. Sie besagt lediglich, dass *wenn* Änderungen der GPL-Software veröffentlicht oder verbreitet werden, dies unter der GPL zu erfolgen hat.

**b) Müssen Änderungen an einer GPL-lizenzierten Software der allgemeinen Öffentlichkeit zur Verfügung gestellt werden?**

Nein, s. Antwort zu a)

**c) Welche Änderungen an GPL-lizenzierter Software werden vom Copyleft erfasst?**

Wird ein GPL-Programm geändert, z. B. umgeschrieben oder weiterentwickelt, greift die Copyleft-Verpflichtung. Dies gilt für jede Art der Umarbeitung oder Bearbeitung. Die geänderte Fassung ist somit ein *derivative work* und muss, wenn sie wieder in Umlauf gebracht werden soll, unter die GPL gestellt werden. Darüber hinaus können auch Code-Kombinationen das Copyleft auslösen (siehe hierzu unten Teil 1, 3.)

**d) Unter welchen Umständen muss ich meine Software unter die GPL stellen, wenn ich Code aus GPL-lizenzierter Software einbinde?**

Wird GPL-Code in ein anderes Programm eingebunden, greift das Copyleft nur dann, wenn hierdurch ein *derivative work* (abgeleitetes Werk) entsteht. Dies wird bei der Einbindung von Code-Fragmenten in aller Regel der Fall sein. Dies gilt jedenfalls, soweit der GPL-Code in den anderen Code integriert wird und damit nicht mehr formal getrennt ist. Etwas anderes kann bei der Einbindung als Bibliothek oder über eine Schnittstelle gelten (s. Teil 1, 4.).

**e) Unter welchen Umständen muss ich meine Software unter die GPL stellen, wenn ich sie mit einer GPL-lizenzierten Software verbinde?**

Eine Verbindung zweier eigenständiger Programme löst an sich den Copyleft-Effekt nicht aus<sup>15</sup>. Ob er wirksam wird, hängt aber von der technischen Art und Weise der Verbindung ab. Werden beide Programme auf eine Weise verbunden (und gemeinsam vertrieben), dass aus den zwei *separate works* ein *derivative work* entsteht, darf das entstehende Ganze nur unter der GPL vertrieben werden. Bei dieser Beurteilung sind formale und inhaltliche Faktoren zu berücksichtigen. Um das Copyleft zu vermeiden, müssen beide Programme formal getrennt gespeichert sein, damit eindeutig ersichtlich ist, welcher Code der GPL untersteht (und das Werk von Dritten darstellt) und welcher nicht. Hierfür kann man beispielsweise sorgen, indem die eigenständigen Programme/Module beim Vertrieb in unterschiedlichen Dateien überlassen werden. Zudem müssen beide Programme inhaltlich voneinander unabhängig sein. Die eigene Software darf kein vom GPL-Programm abgeleitetes Werk sein. Ob dies der Fall ist, muss im Einzelfall beurteilt werden. Die Rechtsliteratur hat Fallgruppen gebildet, die Anhaltspunkte für die Beurteilung bieten (s. Teil 1, 4.).

---

<sup>15</sup> S. Ziff. 2 letzter Absatz der GPLv2: „In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.”

**f) Inwiefern hat die Anbindung einer GPL-lizenzierten Software über eine Schnittstelle an eine andere Software Einfluss darauf, ob ich die andere Software unter die GPL stellen muss?**

Die Anbindung einer GPL-Software an ein anderes Programm führt für sich genommen nicht dazu, dass ein *derivative work* entsteht, das nur unter der GPL vertrieben werden darf (s. hierzu Teil 1, 4.).

**g) Welche Kriterien können zur Bewertung dienen, ob eine Software, die auf irgendeine Weise an eine GPL-lizenzierte Software angebunden ist, ebenfalls unter die GPL gestellt werden muss?**

Inhaltliche und formale Kriterien (s. hierzu die Ausführungen in Teil 1, 3.-4.).

## **2. Fragen zum Hosting für Kunden**

### **a) Grundkonstellation**

Ein Anbieter hostet ein Open-Source-LMS für einen Kunden, das unter der GPL steht. Der Anbieter installiert die Software auf einem Server, der über das Internet erreichbar ist, aber nicht vom Kunden bereitgestellt wird. Der Kunde erhält den Zugriff auf die Oberfläche, nicht jedoch auf den Server selbst. In der Installation sind Patches, Plugins und Skins enthalten.

**I. Muss der Anbieter im Sinne der GPL, abgesehen von weiteren vertraglichen Regelungen, dem Kunden den Quellcode der Installation des Open-Source-LMS übergeben? Umfasst dies auch den Code der Patches, Plugins und Skins?**

*In der Bereitstellung (hosting) eines Servers liegt nach der hier vertretenen – nicht unumstrittenen – Auffassung weder eine Distribution gem. GPLv2 noch eine conveyance nach der GPLv3 der Server-Software (Letzteres ist eindeutig, s. hierzu Teil 1, 2.). Entsprechend gelten die Lizenzpflichten der GPL (und auch der meisten anderen Open-Source-Lizenzen) hier nicht. Der Auftraggeber muss seinem Kunden keinen Quellcode übergeben und auch die anderen Vertriebspflichten nicht einhalten. Anders wäre dies, wenn das Open-Source-LMS unter einer speziellen SaaS-Lizenz (wie der AGPL oder der SSPL) stünde. Solche enthalten besondere Klauseln, nach denen die Lizenzpflichten u. U. auch einzuhalten sind, wenn die Software lediglich als Dienst angeboten, Dritten aber nicht als Kopie überlassen wird.*

**II. Kann die Übergabe des Quellcodes eines Plugins oder Patches vertraglich ausgeschlossen werden? Wenn ja, gilt dies auch, wenn das Plugin im Auftrag des Kunden entwickelt wurde?**

*Nach der hier vertretenen Auffassung muss dem Kunden ohnehin kein Quellcode überlassen werden. Wäre dies aber der Fall, etwa, weil das System nicht nur als Service bereitgestellt, sondern auch als Download überlassen wird, hinge die Pflicht zur Quellcodeüberlassung zunächst davon ab, ob es sich bei dem Plugin/Patch um ein abgeleitetes Werk handelt. Wäre dies der Fall, würde das Copyleft greifen. Versuche, das Copyleft vertraglich zu umgehen, würden zum Erlöschen der Nutzungslizenz aus der GPL führen und gegen das Urheberrecht verstoßen (s. hierzu Teil 1, 5.)*

**III. Kann die Übergabe des Quellcodes eines Plugins oder Patches vertraglich für einen bestimmten Zeitraum ausgeschlossen werden?**

*Siehe Antwort zu a) II.*

#### **IV. Kann dem Kunden die Veröffentlichung/Verbreitung von Quellcode, insbesondere Plugins, untersagt werden?**

*Handelt es sich um ein Plugin, das Dritte als Open Source Software veröffentlicht haben (also nicht um eine eigene Software), würden hierdurch die durch dessen Open-Source-Lizenzen gewährten Freiheiten vertraglich ausgehebelt. Dies mag bei permissiven Lizenzen ohne Copyleft unter Umständen möglich sein. Bei strengen Copyleft-Lizenzen, v. a. der GPL, läge hierin jedoch ein Lizenzverstoß. Hierdurch würden die Nutzungsrechte erlöschen und der Anbieter könnte wegen Urheberrechtsverletzung belangt werden (s. hierzu auch Antwort zu II. und Teil 1, 5.).*

#### **V. Muss auf explizite Anfrage, z.B. eines Nutzers der Plattform, der Quellcode veröffentlicht werden? Gilt dies dann für alle Patches, Plugins und Skins?**

*Siehe Antwort auf Frage a) I.*

#### **VI. Kann der Quellcode der Patches, Plugins oder Skins unter einer anderen Lizenz als der GPL an den Kunden weitergegeben werden?**

*Wenn das Copyleft für diese Komponenten greift, ist das nicht gestattet (s. Antwort zu II.)*

### **b) Fallvariante**

In einem ähnlichen Szenario gehört der Hosting-Server dem Kunden, wird aber ansonsten analog zum vorherigen Szenario verwaltet.

*Auch in der Variante ist die zentrale Frage, ob der Anbieter des Open-Source-LMS dem Kunden die Quellcodes für das System, Patches, Plugins oder Skins überlassen muss. Dies wäre der Fall, wenn deren OS-Lizenzen entsprechende Pflichten enthielten (diese also beispielsweise unter der GPL stünden oder abgeleitete Werke darstellen) und die Installation auf dem Server des Kunden und/oder der Betrieb des Dienstes für den Kunden diese Lizenzpflichten auslösen würde.*

*Nach den Ausführungen zum Ausgangsfall (s. o. unter Frage a) I.) liegt in der Bereitstellung einer Software als Dienst (ASP, SaaS) u. E. weder nach der GPLv2 noch der GPLv3 eine Distribution. Entsprechend löst das Hosting für einen Kunden für sich genommen weder eine Pflicht zur Offenlegung von Sourcecode noch andere Vertriebspflichten aus.*

*Diese Bewertung würde sich u. U. ändern, wenn der Kunde die Software nicht nur als Dienst nutzen könnte, sondern sie ihm durch deren Installation auf seinem Server **überlassen** würde. Hierin liegt generell – außer in Sonderfällen – eine Distribution und der Quellcode für das Open-Source-LMS sowie abgeleitete Werke müssen ebenfalls überlassen werden.*

*Dies könnte wiederum anders zu beurteilen sein, wenn dem Kunden keinerlei Zugriff auf die Installation (also die Kopie der Software) ermöglicht wird. Hat nur der Anbieter Zugriff auf die Installation und kann nur er den Server administrieren, die Software löschen etc. würde der Kunde nicht anders stehen als wenn ihm die Software lediglich als Dienst vom Anbieter (von dessen Server) zur Verfügung gestellt würde. Er wäre, mit anderen Worten, auch hier darauf beschränkt, die Funktionen der Software als Dienst zu nutzen.*

*In diesem Szenario könnte man argumentieren, dass dem Kunden faktisch keine Kopie der Software verschafft wird, da sie im ausschließlichen Herrschaftsbereich des Anbieters verbleibt. Der Umstand, dass die Hardware dem Kunden gehört, mag zwar grundsätzlich relevant sein. Auch in diesem Fall könnte jedoch durch vertragliche Vereinbarungen sowie die dementsprechend ausgestalteten faktischen Zugriffsmöglichkeiten auf die Infrastruktur (Ausgestaltung der Admin-Rechte etc.) sichergestellt werden, dass dem Kunden kein Zugriff auf die Kopie der Software möglich ist. Ob dies der Fall ist, hängt von der konkreten Ausgestaltung des Anbieter-Kunden-Verhältnisses ab. Von Bedeutung wird hierbei z. B. sein, ob die Software nach Ende der Vereinbarung und der Rückgabe des Servers gelöscht wird.*

- I. Muss der Anbieter im Sinne der GPL, abgesehen von weiteren vertraglichen Regelungen, dem Kunden den Quellcode der Installation des Open-Source-LMS übergeben? Umfasst dies auch alle Patches, Plugins und Skins?**

*Das hängt entscheidend von den Vereinbarungen mit dem Kunden ab, siehe einleitende Betrachtung.*

- II. Kann die Übergabe des Quellcodes eines Plugins oder Patches vertraglich ausgeschlossen werden? Wenn ja, gilt dies auch, wenn das Plugin im Auftrag des Kunden entwickelt wurde?**

*Das hängt entscheidend von den Vereinbarungen mit dem Kunden ab, siehe einleitende Betrachtung. Siehe im Übrigen die Antwort auf Frage a) II. oben.*

- III. Kann die Übergabe des Quellcodes eines Plugins oder Patches vertraglich für einen bestimmten Zeitraum ausgeschlossen werden?**

*Das hängt entscheidend von den Vereinbarungen mit dem Kunden ab, siehe einleitende Betrachtung. Siehe im Übrigen die Antwort auf Frage a) III oben.*

- IV. Kann dem Kunden die Veröffentlichung/Verbreitung von Quellcode, insbesondere Plugins, untersagt werden?**

*Das hängt entscheidend von den Vereinbarungen mit dem Kunden ab, siehe einleitende Betrachtung. Siehe im Übrigen die Antwort auf Frage a) IV. oben.*

- V. Muss auf explizite Anfrage, z.B. eines Nutzers der Plattform, der Quellcode veröffentlicht werden? Gilt dies dann für alle Patches, Plugins und Skins?**

*Das hängt entscheidend von den Vereinbarungen mit dem Kunden ab, siehe einleitende Betrachtung. Siehe im Übrigen die Antwort auf Frage a) I. oben.*

- VI. Kann der Quellcode der Patches, Plugins oder Skins unter einer anderen Lizenz als der GPL an den Kunden weitergegeben werden?**

*Das hängt entscheidend von den Vereinbarungen mit dem Kunden ab, siehe einleitende Betrachtung. Siehe im Übrigen die Antwort auf Frage a) IV. oben.*

### 3. Fragen zu Plugins

#### a) Grundfragen

Ein Anbieter erstellt ein Plugin für einen Pluginslot für ein Open-Source-LMS, das unter der GPL steht.

##### I. Darf der Anbieter das Plugin an Kunden verkaufen?

*Ja. Selbst wenn es als abgeleitetes Werk anzusehen wäre und daher aufgrund des Copylefts der GPL unter der GPL lizenziert werden müsste, dürfte es der Anbieter „verkaufen“. Die GPL untersagt es nicht, Software zu verkaufen (also kostenpflichtig zu überlassen), sondern lediglich für die Nutzung der Software Lizenzgebühren zu verlangen. „Free“ oder „Open“ bedeutet nicht „kostenfrei“, sondern „frei zur Nutzung“. Die Free Software Foundation drückt dieses Grundprinzip wie folgt aus:*

*“Thus, “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”. We sometimes call it “libre software,” borrowing the French or Spanish word for “free” as in freedom, to show we do not mean the software is gratis.”<sup>16</sup>*

*Eine andere Frage ist, ob sich das Copyleft der GPL am Open-Source-LMS auf das Plugin erstreckt und es daher nur unter der GPL vertrieben werden darf. Das wäre der Fall, wenn das Plugin ein vom Open-Source-LMS abgeleitetes Programm (derivative work) wäre. Dies hängt von verschiedenen Faktoren ab. Grundsätzlich wird ein Plugin in der Regel ein eigenständiges Programm sein. Der Umstand allein, dass es dazu dient, einem anderen Programm zusätzliche Funktionen hinzuzufügen, ändert hieran per se nichts. Die konkrete Situation bedarf jedoch einer Einzelfallprüfung anhand der in Teil 1, 3. und 4. beschriebenen inhaltlich-funktionalen und formalen Kriterien.*

*Folgende Umstände würden dafür sprechen, dass das Plugin nur unter der GPL vertrieben werden darf:*

- Es wird in einem Executable gemeinsam mit der GPL-Anwendung vertrieben. Wird der Sourcecode gemeinsam kompiliert, darf das entstehende *Binary* insgesamt nur unter der GPL vertrieben werden.
- Es enthält Code aus der GPL-Anwendung (z. B. ein urheberrechtlich geschütztes Headerfile).
- Es ist ausschließlich mit einem bestimmten GPL-Programm nutzbar und verfügt nicht über generische Funktionen. Dieser Faktor ist sehr vage und mit großer Vorsicht zu genießen, weil dies für die meisten Programme – abgesehen von Standard-Bibliotheken – zutreffen wird. Daher kann er auch nur unter Hinzutreten weiterer Anhaltspunkte zur Erkenntnis beitragen.

*Siehe grundsätzlich zu dieser Frage Teil 1, 3.*

##### II. Muss das Plugin öffentlich zur Verfügung gestellt werden?

*Nein. Die GPL verpflichtet nicht dazu, Code zu veröffentlichen, auch nicht bei abgeleiteten Werken (s. Teil 1, 1.).*

##### III. Kann das Plugin unter eine andere Lizenz als GPL gestellt werden?

**Wenn ja, unter welche, bzw. welche Anforderungen bestehen an die andere Lizenz?**

*Zunächst ist im Einzelfall zu prüfen, ob es sich bei dem Plugin um ein derivative work handelt (s. Antwort zu Frage I.). Sollte dies **nicht** der Fall sein, kann das Plugin unter jede beliebige Open-Source-Lizenz gestellt oder auch proprietär vertrieben werden. Greift dagegen das Copyleft der GPL, ist ein Vertrieb nur unter der GPL zulässig (s. Teil 1, 1.).*

---

<sup>16</sup> <https://www.gnu.org/philosophy/free-sw.html.en>.

#### IV. Muss der Quellcode des Plugins öffentlich zur Verfügung gestellt werden?

*Nein. Selbst wenn das Copyleft der GPL greifen würde, müsste kein Quellcode veröffentlicht werden. Gilt das Copyleft, muss jedoch – wenn die Software im Object Code überlassen wird – jedem Empfänger der complete corresponding machine-readable source code überlassen oder zumindest die Überlassung angeboten werden (s. Ziff. 3a GPLv2, 6a GPLv3).*

#### b) Ergänzungsfragen

Ein Kunde kauft das Plugin zum Einsatz in seiner Installation des Open-Source-LMS.

##### I. Muss der Quellcode des Plugins dem Kunden zur Verfügung gestellt werden?

*Wenn das Copyleft der GPL greift: ja (s. vorstehende Frage).*

##### II. Kann die Weitergabe des Plugins vertraglich untersagt werden?

*Wenn das Copyleft der GPL greift: nein (s. Teil 1, 5. und Frage 2. a. IV). Wenn das nicht der Fall ist, ist dies ohne Weiteres möglich.*

##### III. Kann die Weitergabe des Quellcodes des Plugins vertraglich untersagt werden?

*Wenn das Copyleft der GPL greift: nein (vorstehende Frage). Wenn das nicht der Fall ist, ist dies ohne Weiteres möglich.*

### 4. Fragen zu Schnittstellen

#### a) Konstellation 1

Ein Anbieter erstellt ein Programm. Dieses greift über eine im Kern des Open-Source-LMS implementierte Schnittstelle auf dieses zu. Muss das Programm unter der GPL stehen?

Grundsätzlich: Nein. Der Umstand, dass ein Programm über eine Schnittstelle mit einem GPL-Programm kommuniziert, sagt nichts darüber aus, ob es sich um ein abgeleitetes Werk handelt. Handelt es sich um eigenständige Programme, die auch technisch-formal eigenständig vertrieben werden, greift das Copyleft nicht (s. Teil 1, 3. und 4.).

#### b) Konstellation 2

In einem anderen Fall wird über eine spezifische, für die Anbindung des Programms geschaffene, Schnittstelle auf das Open-Source-LMS zugegriffen. Die spezifische Schnittstelle wurde als Plugin realisiert. Muss das Programm unter GPL stehen?

Siehe Antwort auf Frage a). Wie die Schnittstelle technisch realisiert wurde, ist für die Beurteilung, ob es sich bei dem Programm um ein *derivative work* des Open-Source-LMS handelt, irrelevant.

#### c) Weitere Fragen zu beiden Konstellationen

##### I. Gilt dies auch für z.B. Apps, die in einem App-Store publiziert werden?

*Auch hier gilt das in der Antwort auf Frage 4. a) Gesagte. Auf welchen Marktplätzen die Applikation angeboten wird, ist für die Frage nach der (Un-)Abhängigkeit derselben vom Open-Source-LMS nicht relevant.*

**II. Kann die Verbreitung eines Programms, das über eine Schnittstelle auf das Open-Source-LMS zugreift, vertraglich unterbunden werden?**

*Sofern das Copyleft nicht greift (s. Antwort auf Frage a), ist das unproblematisch. Handelt es sich bei dem Programm um ein abgeleitetes Werk (-> Copyleft), ist das unzulässig (s. Antwort auf Frage 2. a) IV. und Teil 1, 3. und 4.).*

**III. Ist die Art der Schnittstelle oder das Wesen der übertragenen Daten für diese Fragen relevant?**

*Grundsätzlich: Nein.*

**d) Konstellation 3**

In einem dritten Szenario erstellt ein Anbieter ein Programm, welches nicht über eine Schnittstelle mit dem Open-Source-LMS kommuniziert, aber die durch das Open-Source-LMS erzeugte Datenbank verwendet.

**I. Gilt dies als Schnittstelle? Hat diese Art Zugriff Implikationen für die Lizenz des Programms?**

*Die von einem Open-Source-LMS erzeugten Daten werden vom Copyleft des Systems nicht erfasst. Sie können unter beliebigen Lizenzen veröffentlicht oder als public domain zur Nutzung freigestellt werden. Siehe hierzu Teil 1, 4.). Entsprechend hat das Copyleft des Open-Source-LMS in dieser Konstellation – abstrakt betrachtet – keinen Einfluss auf ein Programm, das lediglich dessen Output-Daten verwendet.*

**II. Wie verhält es sich, wenn das Open-Source-LMS auf Datenbanken anderer Programme zugreift?**

*Ob und unter welchen Bedingungen die Datenbanken oder darin enthaltene Daten verwendet werden können, wird sich meist aus der hierfür geltenden Lizenz ergeben. Diese ist auf etwaige Beschränkungen bei der Nachnutzung zu untersuchen. Davon abgesehen hat die Nutzung von Daten grundsätzlich keine Auswirkungen auf die Lizenzierung des nutzenden Programms.*

## **5. Fragen zu 3rd Party Software**

Ein Anbieter möchte Komponenten anderer Anbieter in seine Erweiterung integrieren. Beispielsweise wird in einem Plugin ein HTML-Editor integriert oder der HTML-Editor soll als Erweiterung im Kern integriert werden.

**a) Welche Lizenz muss die Komponente haben, um in einem unter der GPL stehenden Open-Source-LMS eingebunden werden zu können? Sind folgende Lizenzen zulässig: MIT-Lizenz, MPL, Apache License, ...?**

Zunächst stellt sich hier die Frage, ob die Erweiterung des Anbieters überhaupt Restriktionen aus der Lizenz des Open-Source-LMS unterliegt. Nur in manchen Konstellationen greift beim kombinierten Vertrieb von Software das Copyleft der GPL. Siehe hierzu Teil 1, 3. und die Fragen zu Plugins (3.).

Handelt es sich bei der Erweiterung um ein abgeleitetes Werk, darf sie insgesamt (einschließlich der Fremdkomponenten) nur unter der GPL vertrieben werden. Hier gilt Folgendes: Bilden die Fremdkomponenten mit der Erweiterung eine Einheit, dürfen sie nur integriert werden, wenn sie unter einer mit der GPL kompatiblen Lizenz stehen. Die Kombination muss dann beim Vertrieb unter die GPL gestellt werden (s. zur Lizenzkompatibilität Teil 1, 5.).

Zu beachten ist v. a., dass die Lizenzkompatibilität immer anhand konkreter Lizenzversionen untersucht werden muss. So ist beispielsweise (nach Auffassung der FSF) die Apache Public License in Version 2 (APL 2.0) mit der GPLv3 kompatibel, nicht aber mit der GPLv2. Einzelheiten zur Kompatibilität mit den GNU-Lizenzen finden sich auf den Hilfeseiten der FSF: <https://www.gnu.org/licenses/license-list.html.en#Introduction>.

**b) Wie wirkt sich dies auf die Lizenz des Plugins aus?**

Siehe die Antwort zu a). Wenn der Vertrieb des Plugins unter das Copyleft der GPL fällt, muss es entsprechend lizenziert werden.

**6. Fragen zu 3rd Party Services**

Ein Anbieter greift in einem von ihm erstellten Plugin auf eine Schnittstelle eines weiteren Programms zu (z.B. Google Maps).

**a) Darf vom Anbieter bereitgestellter Code (z. B. zur Implementierung der Schnittstelle) unter die GPL gestellt werden?**

Solange der Anbieter nicht den Code des anderen Programms verwendet, ist das generell unproblematisch. Der Anbieter kann grundsätzlich mit seinem eigenen Code „machen, was er will“.

**b) Darf das Plugin weiterhin unter GPL stehen?**

Siehe die Antwort zu 6. a).

