Grenzen und Lizenzen: Passende Lizenzen für Softwareprojekte finden

Das Kleingedruckte

Lizenzen sagen uns, wie wir Softwarepakete verwenden und die Software legal in eigenen Projekten nutzen dürfen. Erfreulicherweise gibt es mittlerweile Werkzeuge, mit denen sich das automatisch überprüfen lässt. Frank Hofmann

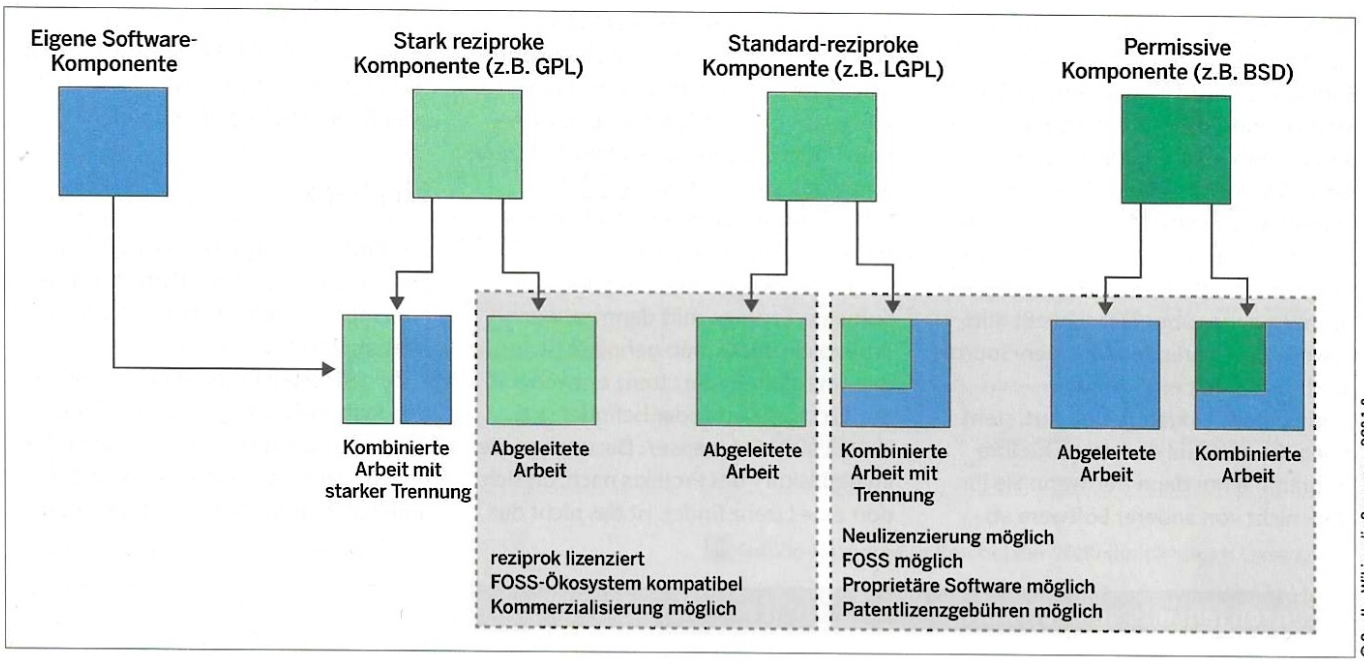
Haben Sie Ihre Programmkomponente aus einem GPL-lizenzierten Softwarepaket abgeleitet, werden Sie Ihre Software unter derselben GPL-Lizenz veröffentlichen müssen. Kombiniert das Programm dagegen LGPL-lizenzierten Code mit Ihrem eigenen Werk, dürfen Sie das Ergebnis unter einer neuen, gegebenenfalls auch proprietären Lizenz verbreiten. Dieselben Möglichkeiten hätten Sie auch dann, wenn Sie Ihre Software aus einem BSD-lizenzierten Paket abgeleitet haben.

Obgleich GPL, LGPL und BSD allesamt zu den Lizenzen für freie Software zählen, gestatten sie sehr unterschiedliche Nutzungen. Um die prinzipiellen Unterschiede differenzierter herauszuarbeiten, unterteilt man Softwarelizenzen in eine Reihe von Kategorien.

Außerhalb der Open-Source-Welt sind proprietäre Softwarelizenzen die häufigsten. Diese folgen nur selten einem Standard. Neben den weit verbreiteten kommerziellen Lizenzen zählen Shareware und Freeware in diese Kategorie.

Kategorisches

FOSS-Softwarelizenzen werden von der Free Software Foundation (FSF) und der Open Source Initiative (OSI) definiert. Dabei kann man im Wesentlichen zwischen Copyleft-, freizügigen sowie gemeinfreien Lizenzen unterscheiden.



1 Diese Übersicht zur Kompatibilität abgeleiteter Softwarelizenzen verdeutlicht die kombinatorischen Möglichkeiten

Copyleft-Lizenzen verpflichten Lizenznehmer, jegliche Bearbeitung der Software (sogenannte Derivate) unter die Lizenz des ursprünglichen Werks zu stellen. Dieser deshalb auch als reziprok bezeichnete Lizenztyp soll Nutzungseinschränkungen der Software verhindern. Die bekannteste Copyleft-Lizenz ist ohne Zweifel die GNU General Public License (GPL). Dabei gilt das Copyleft der GPL als sehr stark, jenes der Mozilla Public License dagegen als recht schwach.

Da die Lizenzgeber nicht selbst an ihr eigenes Copyleft gebunden sind, können sie neue Versionen ihrer Software durchaus unter einer proprietären Lizenz veröffentlichen oder dasselbe Dritten erlauben. Man spricht in diesem Fall von einer Mehrfachlizenzierung.

Durch Copyleft-Lizenzen können bei der Verbreitung zusammen mit Software unter anderen freien Lizenzen allerdings schnell Inkompatibilitäten entstehen. Die 3-Clause-BSD-Lizenz ist zum Beispiel mit der GPL inkompatibel. Bei der European Union Public License (EUPL) handelt es sich dagegen um eine mit den meisten anderen offenen reziproken Lizenzen kompatible und interoperable reziproke Lizenz: Die kompatiblen Lizenzverpflichtungen haben Vorrang, wenn sie mit den sich aus der EUPL ergebenden Verpflichtungen in Konflikt geraten.

Freizügige (permissive) Open-Source-Lizenzen erlauben eine breitere Wiederverwendung als die Copyleft-Lizenzen. Ableitungen und Kopien des Quellcodes darf man unter Bedingungen verbreiten, die grundlegend andere Eigenschaften haben als die Originallizenz. Die bekanntesten Beispiele für solche Lizenzen sind die MIT- und die BSD-Lizenz.

Schließlich gibt es noch gemeinfreie oder neuhochdeutsch Public-Domain-Lizenzen: Bei ihnen gehen die Urheberrechte an die Allgemeinheit über. Zur Kennzeichnung der Gemeinfreiheit von Software wurde die UnLicense aus der Taufe gehoben.

Jenseits von Software

Manche Open-Source-Softwarelizenzen lassen sich auch für Werke verwenden, bei denen es sich nicht um Software handelt. Häufig stellen sie tatsächlich die beste Wahl dar, insbesondere, wenn man die betreffenden Werke als Quelltext bearbeitet und sie versioniert.

Listing 1: Prüfung mit Liccheck

```
$ liccheck -s liccheck.ini -r requirements.txt
gathering licenses...
3 packages and dependencies.
check unknown packages...
3 packages.
  cffi (1.15.1): ['MIT']
    dependency:
      cffi << cryptography
  cryptography (41.0.3): ['Apache Software', 'BSD']
    dependency:
      cryptography
  pycparser (2.21): ['BSD']
    dependency:
      pycparser << cffi << cryptography
```

Hingegen sind die populären Creative-Commons-Lizenzen explizit nicht für Software gedacht, sondern für Daten, Medien und Dokumentation. Für Daten und Datenbanken hat die Open Knowledge Foundation die Open-Data-Commons-Lizenzen formuliert. Die Tabelle Lizenzen jenseits von Software stellt die Varianten von Nicht-Software-Lizenzen gegenüber. Das schließt mittlerweile auch Varianten für Open-Source-Hardware mit ein.

Wie bereits eingangs erläutert, steht Ihnen die Auswahl der Lizenz für Ihre Programme nur dann frei, wenn Sie Ihr Werk nicht von anderer Software abgeleitet oder es mit anderer Software kombiniert haben. Die Lizenzkompatibilität für abgeleitete oder kombinierte Werke aus eigener Software und externem Code, der einer Open-Source-Lizenz unterliegt, stellt Abbildung 1 dar.

Lizenzen abgeleiteter oder kombinierter Werke finden Sie auf verschiedenen Wegen, entweder von Hand oder per Software. Ersteres trifft dann zu, wenn das Projekt auf Github gehostet ist. In diesem Fall steht die Lizenz entweder in der `license`-Datei oder befindet sich im Verzeichnis `license/`. Dazu sehen Sie im Repository des Projekts nach, ob sich dort eine Lizenz findet. Ist das nicht der Fall, gilt der nächste Blick den Hinweisen in der Datei `README`. Per Software lassen sich Lizenzen mit Tools wie Liccheck oder über REUSE identifizieren.

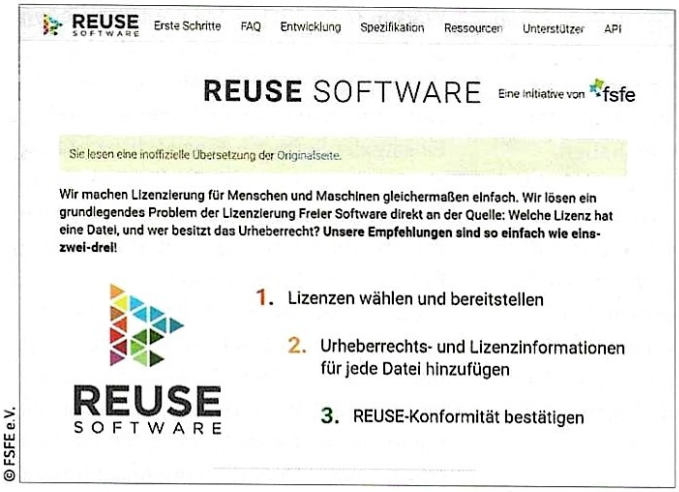
Liccheck

Das Prüfwerkzeug Liccheck ist für diejenigen interessant, die Python in ihren Projekten verwenden, entweder direkt oder als Paketabhängigkeit.

Zu den festen Bestandteilen eines jeden Python-Pakets gehört eine Datei mit dem Namen `requirements.txt`. Darin legt der Programmierer fest, welche anderen Python-Packages das Python-

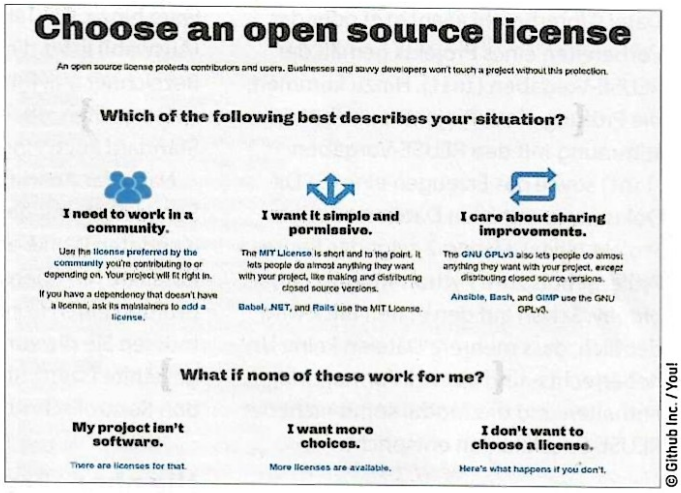
Lizenzen jenseits von Software

Lizenz	Anmerkung
Daten, Medien, Content	
Creative Commons	Offene Lizenzen für Nicht-Software-Material, von Datensätzen bis zu Videos. Die CC-Lizenzen sind jedoch nicht für Software empfohlen.
Open Data Commons	Lizenzen für Daten(-banken) der Open Knowledge Foundation.
Datenlizenz Deutschland – Namensnennung – Version 2.0, Datenlizenz Deutschland – Zero – Version 2.0	GovData hat die Datenlizenz Deutschland in zwei Varianten vorgelegt.
Community Data License Agreements	Datenlizenzen der Linux Foundation.
Free Art License	Lizenz für künstlerische Werke.
Dokumentation	
GNU Free Documentation License (FDL)	Copyleft-Lizenz für Dokumentationen, die für alle GNU-Handbücher verwendet werden soll. Ihre Anwendbarkeit beschränkt sich auf textbasierte Werke wie Bücher.
FreeBSD Documentation License	Freizügige Dokumentationslizenz mit Copyleft, die sich mit der GNU FDL vereinen lässt.
Open Publication License 1.0	Freie Dokumentationslizenz mit Copyleft, sofern keine der Lizenzoptionen aus Abschnitt VI der Lizenz wahrgenommen werden. Diese Lizenz lässt sich grundsätzlich nicht mit der GNU FDL vereinen.
Schriftarten	
SIL Open Font License 1.1	Lizenz für Schriftarten, die man in anderen Werken frei verwenden darf.
GNU General Public License 3 (GPLv3)	Die GPLv3 lässt sich auch für Schriftarten verwenden, sie darf jedoch nur mit der Schriftausnahme in Dokumente eingebunden werden.
LaTeX EC Fonts	Freie Schriftarten (European Computer Modern, Text Companion), die üblicherweise mit LaTeX verwendet werden.
Arphic Public License	Freie Lizenz mit Copyleft.
IPA Font License	Freie Lizenz mit Copyleft, deren abgeleitete Werte jedoch nicht den Namen des Originals verwenden oder enthalten dürfen.
Hardware (basierend auf den CERN-Open-Hardware-Lizenzen)	
CERN-OHL-P-2.0	freizügig (permissiv)
CERN-OHL-W-2.0	schwach reziprok
CERN-OHL-S-2.0	stark reziprok



© FSFE e.V.

2 **Die Homepage des FSFE-Projekts REUSE.**



© Github Inc. / You!

3 **Online-Ratgeber helfen bei der Wahl der richtigen Lizenz.**

Paket benötigt, um zu funktionieren. Liccheck überprüft die Lizenzen der in dieser Datei aufgezählten Pakete und zeigt alle entdeckten Lizenzkonflikte an.

Zuvor legen Sie eine Strategie fest, anhand derer Liccheck seine Prüfung vornimmt: Welche Lizenzen soll es als Verstoß werten, welche nicht? Je genauer diese Vorbereitung erfolgt, desto exakter fällt das Ergebnis der Prüfung aus. Listing 1 zeigt den Aufruf und die Analyse zum Python-Modul *cryptography*.

REUSE

Das Projekt REUSE entstand unter der Federführung der Free Software Foundation Europe und basiert auf zwei Säulen: maschinell lesbaren Copyright-Dateien aus dem Debian-Projekt sowie SPDX. SPDX steht für Software Package Data Exchange und definiert eine standardisierte Methode für den Austausch von Copyright- und Lizenzinformationen zwischen Projekten und Menschen.

REUSE bietet sowohl die Anleitung als auch das Werkzeug Reuse, sodass Ihr Projekt damit die richtigen Lizenzangaben erhält. Die Projektwebseite 2 stellt REUSE anderen Lösungen gegenüber, darunter Scancode, ClearlyDefined, Fossology und OpenChain.

Das Kommandozeilenwerkzeug Reuse basiert auf Python. Die Installation via Pip ist vollständig dokumentiert. Seit Debian GNU/Linux 12 „Bookworm" ist die Software zudem als gleichnamiges DEB-Package in den Paketquellen verfügbar.

Reuse bietet eine ganze Reihe nützlicher Funktionen. Die beschränken sich keineswegs auf das Hinzufügen von Copyright- und Lizenzinformationen im Kopf einer

Listing 2: Reuse-Prüfergebnis

```
$ cd cryptography
$ reuse lint

# FEHLENDE URHEBERRECHTS- UND LIZENZINFORMATIONEN
Die folgenden Dateien haben keine Urheberrechts- und
Lizenzinformationen:
* .gitattributes
* .github/ISSUE_TEMPLATE/openssl-release.md
[...]
* vectors/cryptography_vectors/x509/wosign-bc-invalid.pem
* vectors/pyproject.toml
Die folgenden Dateien haben keine Lizenzinformationen:
* docs/_ext/linkcode_res.py
* src/cryptography/__about__.py

# ZUSAMMENFASSUNG
* Falsche Lizenzen: 0
* Veraltete Lizenzen: 0
* Lizenzen ohne Dateiendung: 0
* Fehlende Lizenzen: 0
* Unbenutzte Lizenzen: 0
* Verwendete Lizenzen: 0
* Read errors: 0
* files with copyright information: 2 / 2806
* files with license information: 0 / 2806

Allerdings ist Ihr Projekt nicht konform mit Version 3.0 der
REUSE-Spezifikation :-( $
```

Datei (Unterbefehl `annotate`) oder das Vorbereiten eines Projekts gemäß der REUSE-Vorgaben (`init`). Hinzu kommen die Prüfung eines Projekts auf Übereinstimmung mit den REUSE-Vorgaben (`lint`) sowie das Erzeugen eines SPDX-Dokuments mit allen Dateien aus dem Projekt (`spdx`). Listing 2 zeigt das Reuse-Prüfergebnis zum Python-Modul *cryptography*. Schon auf den ersten Blick wird deutlich, dass mehrere Dateien keine Urheberrechts- und Lizenzinformationen enthalten und das Modul somit nicht der REUSE-Spezifikation entspricht.

Lizenz auswählen

Sind die Lizenzen der einzelnen Werke geklärt, steht die Wahl der Lizenz für Ihr eigenes Projekt an. Hilfestellung bietet dabei neben Github auch SPDX. Dazu wählen Sie zuerst die passenden SPDX-Begriffe aus und fügen sie dem Kopf der Lizenzbeschreibung oder Ihrer Software hinzu. Die Tabelle SPDX-Bezeichner (Auswahl) listet die wichtigsten SPDX-Bezeichner auf. Mit Reuse können Sie später prüfen, ob Sie den angepeilten Standard auch eingehalten haben.

Nach der Auswahl einer Lizenz müssen Sie sie noch mit den Lizenzen der verwendeten Werke abgleichen und auf Konflikte hin überprüfen. Wenn die Überprüfung einen Konflikt zutage fördert, müssen Sie die von Ihnen ursprünglich gewählte Lizenz überdenken und dann den Kontrollschritt wiederholen.

Nie ohne Lizenz

Zu jeder Software gehört eine Softwarelizenz einfach dazu. Für freie Software wählen Sie sie aus einer ziemlich großen Palette aus. Eine Übersicht dazu stellt beispielsweise die Open Source Initiative (OSI) zusammen.

Als Autor wählen Sie passende Lizenzbestimmungen aus und fügen sie Ihrem Werk in Textform bei. Mit der Lizenz legen Sie fest, was andere mit Ihrem Werk tun dürfen. Das betrifft sowohl die private Nutzung als auch einen kommerziellen Einsatz der Software. Eine Mehrfachlizenzierung liegt vor, wenn Sie für die beiden Nutzungsarten unterschiedliche Lizenzen angeben. Listing 3 zeigt das für das Python-Modul *cryptography*.

Dabei gibt es einige Fallen, die später für ungewollte Konflikte sorgen, zum Beispiel, wenn Sie die Server Side Public License (SSPL) verwenden. Die OSI hat festgestellt, dass sie nicht mit ihrer Open-Source-Definition übereinstimmt. Sie kommt jedoch bei bedeutenden Softwareprojekten wie DBMS MongoDB, der Suchmaschine Elasticsearch und dem Visualisierungswerkzeug Kibana zum Einsatz.

Die Wahl einer proprietären Softwarelizenz ist legitim, aber es fehlen dafür Standards. Das macht Vergleiche komplizierter. So lizenzierte Programme können kommerzielle Programme im engeren Sinn sein, aber auch Share- oder Freeware. Bei gemeinfreien oder Public-Domain-Lizenzen gehen die Urheberrechte an die Allgemeinheit über. Zur Kennzeichnung der Freigabe weitestmöglicher Nutzungsrechte wurde die Creative-Commons-Zero-Lizenz erstellt.

Würden Sie Ihrem Werk dagegen überhaupt keine Lizenz beifügen, müsste jeder Interessent erst bei Ihnen nachfragen, wie Sie sich die Benutzung Ihrer Software vorstellen, also welche Rechte Sie gewähren und welche Pflichten auf einen Benutzer zukommen. Dasselbe gilt umgekehrt, wenn Sie Software oder Bibliotheken benutzen möchten, denen der Urheber keine Lizenz beigefügt hat:

SPDX-Bezeichner (Auswahl)

offizieller Lizenzname	SPDX-Bezeichner
GNU Affero General Public License v3.0 or later	AGPL-3.0-or-later
Apache License 2.0	Apache-2.0
BSD 3-Clause „New" or „Revised" License	BSD-3-Clause
Common Development and Distribution License 1.1	CDDL-1.1
Creative Commons Attribution Share Alike 4.0 International	CC-BY-SA-4.0
Deutsche Freie Software Lizenz	D-FSL-1.0
European Union Public License 1.2	EUPL-1.2
GNU Free Documentation License v1.3 or later	GFDL-1.3-or-later
GNU General Public License v3.0 or later	GPL-3.0-or-later
GNU Lesser General Public License v3.0 or later	LGPL-3.0-or-later
Mozilla Public License 2.0	MPL-2.0
XFree86 License 1.1	XFree86-1.1

Listing 3: Mehrfachlizenz

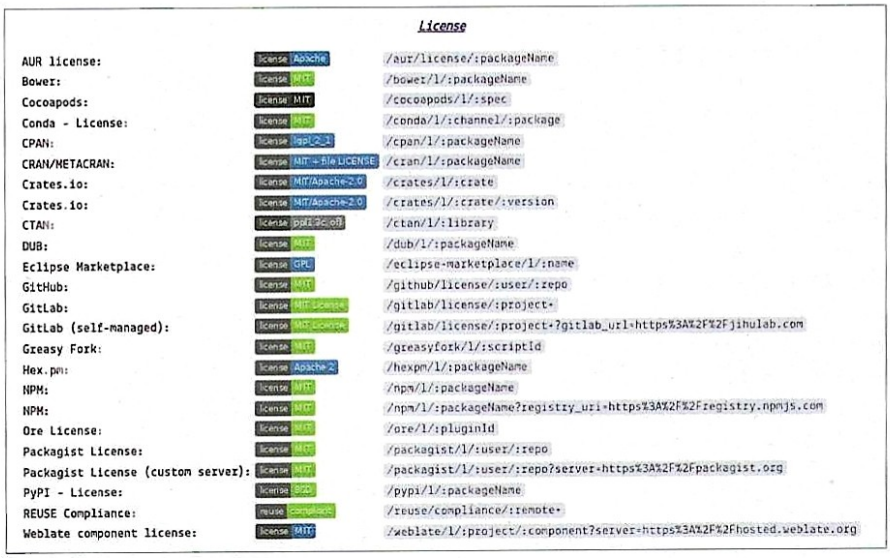
```
This software is made available under the terms of *either* of the
licenses found in LICENSE.APACHE or LICENSE.BSD. Contributions to
cryptography are made under the terms of *both* licenses.

The code used in OpenSSL locking callback and OS random engine is
derived from the same in CPython, and is licensed under the terms
of the PSF License Agreement.
```

Lizenzlogos einbinden

Um es für potenzielle Benutzer leichter zu machen zu erkennen, unter welcher Lizenz Sie ein Werk veröffentlicht haben, statten Sie es mit einem eingängigen Lizenzschild aus, umgangssprachlich auch Badge genannt. Das Projekt Shields.io nimmt Ihnen diesen Schritt ab und bietet sowohl fertige als auch individuell generierte Lizenzlogos an, die Sie dann lediglich noch in der Dokumentation zu Ihrem Projekt ergänzen müssen [4].

In diesem Fall liegt der Aufwand zur Klärung der Situation bei Ihnen als Nutzer.

Dieser Weg der Rückfrage stellt zwar das übliche Verfahren beim Verwenden fremder Werke dar, ist aber umständlich und behindert schnelle Entscheidungen, wie sie heute vielfach stattfinden. Es ist deshalb nicht auszuschließen, dass der Interessent dann entweder Ihr Werk benutzt, ohne vorher Ihre Vorstellung zur Verwendung zu erfragen, oder stattdessen ein anderes Werk auswählt, das klar und eindeutig lizenziert ist, um möglichen Streitigkeiten mit Ihnen aus dem Weg zu gehen. Harald Welte dokumentiert entsprechende Streitfälle seit 2004 mit seinem Projekt GPL Violations.



4 Einige Beispiele für die von Shields.io erzeugten Lizenzschilde.

Hilfestellung

Die Auswahl der richtigen Lizenz ist nicht immer leicht. Mehrere Projekte, darunter die Free Software Foundation, Github und Wikipedia, bieten daher online Ratgeber an 3. Darin stellen sie die Ziele der unterschiedlichen freien Lizenzen einandner gegenüber und beschreiben deren Anwendungsfälle. Die FSFE bietet ihre Unterstützung daneben bei der Überprüfung von Lizenzkonformität in Form rechtlicher Beratung durch eine Rechtsfachkraft an.

Diese Möglichkeiten verhelfen Ihnen zum nötigen Durchblick und erleichtern den Weg durch das rechtliche Dickicht, sofern es sich um Einzelprojekte handelt. Bei einer Mehrfachlizenzierung wie im Beispiel aus Listing 3 oder unterschiedlichen Lizenzen für die enthaltenen Dateien/Komponenten steigt zum Beispiel Github aus. Es kann mit einem solchen Fall nicht sauber umgehen, bislang können Sie dort pro Projekt nur eine einzige Lizenz hinterlegen.

Fazit

In dem vorliegenden Beitrag haben wir die verschiedenen Softwarelizenzen beleuchtet, sind auf ihre Nutzung eingegangen und haben auf die Fallstricke hingewiesen, die es bei ihnen zu berücksichtigen gilt. Daneben haben wir Ihnen Methoden und Werkzeuge vorgestellt, mit denen Sie die Lizenz einer von Ihnen verwendeten Software automatisiert erkennen können. In einem Folgeartikel werden wir besprechen, wie Sie einem eigenen Python-Paket und dessen Dokumentation eine oder mehrere Lizenzen hinzufügen. Wir stellen Ihnen dabei die Vorgehensweisen und Werkzeuge vor, die sich dazu im Alltag bewährt haben, und zeigen, wie wir sie in den Entwicklungsprozess integrieren. (*jlu*)

Lizenzdatei auf Github erstellen

Github hilft Ihnen dabei, eine Open-Source-Lizenz für ein Repository zu erstellen. Dazu bedarf es einer Reihe von Schritten. Zunächst begeben Sie sich auf die Hauptseite Ihres Repositorys. Dort wählen Sie *Create new file* und geben anschließend als Dateinamen `LICENSE` oder `LICENSE.md` ein. Anschließend klicken Sie rechts neben dem Feld für den Dateinamen auf die Schaltfläche *Choose a license template*. Es öffnet sich ein Auswahlfeld für die passende Open-Source-Lizenz. Sofern die ausgewählte Lizenz dies erfordert, erfragt der Prozess zusätzliche Angaben. Nach der Angabe einer Commit-Message wie *Add license* schließt ein Klick auf *Commit new file* den Prozess ab.

Befindet sich bereits eine `LICENSE`-Datei im Repository, verwendet Github das Werkzeug Licensee, um sie mit einer kurzen Liste von Open-Source-Lizenzen abzugleichen. Kann Github die Lizenz des Repositorys nicht erkennen, enthält es möglicherweise mehrere Lizenzen oder ist zu komplex. Prüfen Sie dann, ob Sie die Lizenzierung vereinfachen können, beispielsweise indem Sie die Komplexität in die README-Datei auslagern. Umgekehrt bietet Github an, nach Repositories mit bestimmten Lizenzen oder Lizenzfamilien zu suchen. Eine Übersicht über die von Github verwendeten Schlüsselwörter rund um Lizenzen finden Sie unter dem Eintrag *Searching Github by license type*.

Danksagung

Der Autor bedankt sich bei Veit Schiele, Axel Beckert und Gerold Rupprecht für ihre Anregungen während der Erstellung des vorliegenden Artikels.

Weitere Infos und interessante Links

www.lm-online.de/qr/49787

Der Autor

Frank Hofmann arbeitet zumeist von unterwegs als Entwickler, Trainer und Autor. Bevorzugte Arbeitsorte sind Berlin, Genf und Kapstadt. Er gehört zu den Verfassern des Debian-Paketmanagement-Buchs