Licensing Research Software Made Easy: Introducing the License Checker Tool

Dorothea Iglezakis [1,2,*], Iqra Imran [1,2]

[1] Research Data Competence Center, University of Stuttgart, Stuttgart, Germany;
[2] University Library, University of Stuttgart, Stuttgart, Germany;
[*] Corresponding author: fokus@izus.uni-stuttgart.de

Developing research software involves decisions about software licenses. These decisions can get complicated, especially if the code to be licensed builds on already licensed components. The challenge is to select a license that both aligns with the project's goals and remains compatible with predefined licenses.

The *License Checker* helps to find an appropriate license by providing an overview of common open-source software licenses, by analysing existing code for license information and by providing information about the compatibility of these licenses. It builds on a license ontology and offers a webbased interface. Target groups are research software engineers that have a basic understanding about software licenses in general but no deeper knowledge about license compatibility. It can be used to check the compatibility of an arbitrary set of licenses, but does not handle licenses for other research artefacts than software. It offers functionality to extract licenses from existing code and analyses dependency files, but so far only for Python and Javascript. The License Checker is also not handling complex licensing situations and does not distinguish between linking against a dependency and integrating a dependency into the code for the compatibility check.

Keywords: Research Software, Open-Source Licenses, Software Licenses, License Ontology

Published in: Vincent Heuveline, Philipp Kling, Florian Heuschkel, Sophie G. Habinger, and Cora F. Krömer (Hrsg.): E-Science-Tage 2025. Research Data Management: Challenges in a Changing World. Heidelberg: heiBOOKS, 2025. DOI: https://doi.org/10.11588/heibooks.1652.c23939 (CC BY-SA 4.0).

1 Introduction

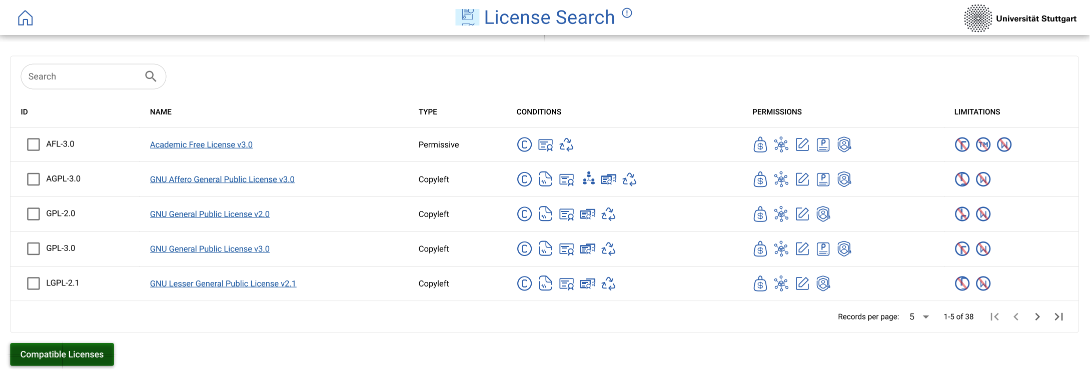
A license serves as a clear way to state what is allowed with a piece of software by defining permissions, conditions and restrictions associated with code usage. However, it is also a legal document, and few research software engineers feel confident interpreting legal texts (e.g., Almeida et al. 2017). Moreover, developers frequently integrate external libraries or build on code with predefined licenses to enhance the functionality of their code. Then, the challenge is to find a license that aligns with the project's goals and is compatible with the predefined licenses at the same time.

2 The License Checker

The *License Checker*[1] is a webbased tool, that helps researchers to choose an appropriate license for their research software.

2.1 License Search

The tool offers an overview over the most established open-source licenses to compare and choose the licenses that fit to the intended usage of the software. An overview page (see Figure 1) displays the license type, conditions, permissions, and limitations of each software license and allows to identify compatible licenses to any chosen set of licenses.

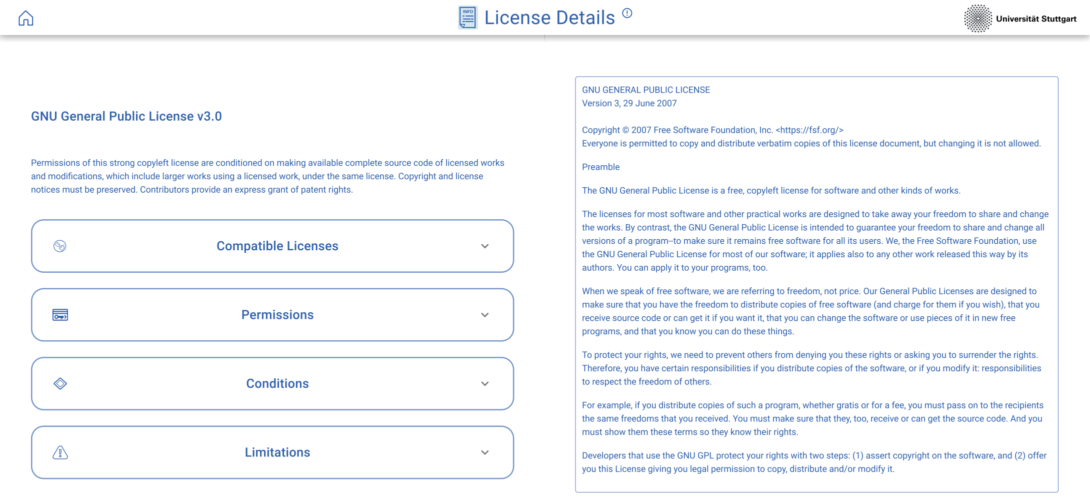


ID	NAME	TYPE	CONDITIONS	PERMISSIONS	LIMITATIONS
□ AFL-3.0	Academic Free License v3.0	Permissive			
□ AGPL-3.0	GNU Affero General Public License v3.0	Copyleft			
□ GPL-2.0	GNU General Public License v2.0	Copyleft			
□ GPL-3.0	GNU General Public License v3.0	Copyleft			
□ LGPL-2.1	GNU Lesser General Public License v2.1	Copyleft			

Figure 1: License Overview.

A detail page provides for each license additionally compatible licenses and the license text (see Figure 2).

[1] https://checklicense.uni-stuttgart.de; *Visited on March 28, 2025.*



GNU General Public License v3.0

Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <https://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

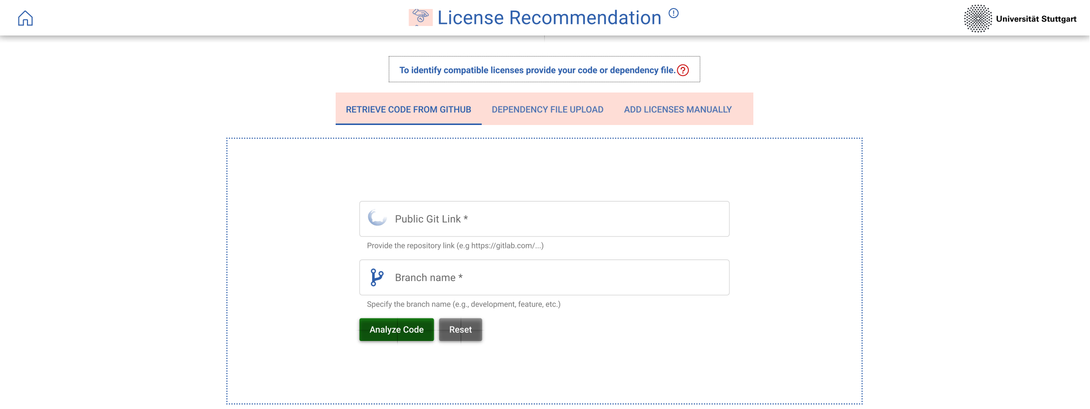
For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

Figure 2: License Details.

2.2 License Recommendation

The *License Checker* also assists users in finding compatible licenses, allowing them to seamlessly incorporate external pieces of software codes without risking licensing conflicts. Existing code can be uploaded to the tool (see Figure 3) to be scanned for available license files with the help of fossology, an open source license compliance software system and toolkit[2].



To identify compatible licenses provide your code or dependency file.

Provide the repository link (e.g https://gitlab.com/...)

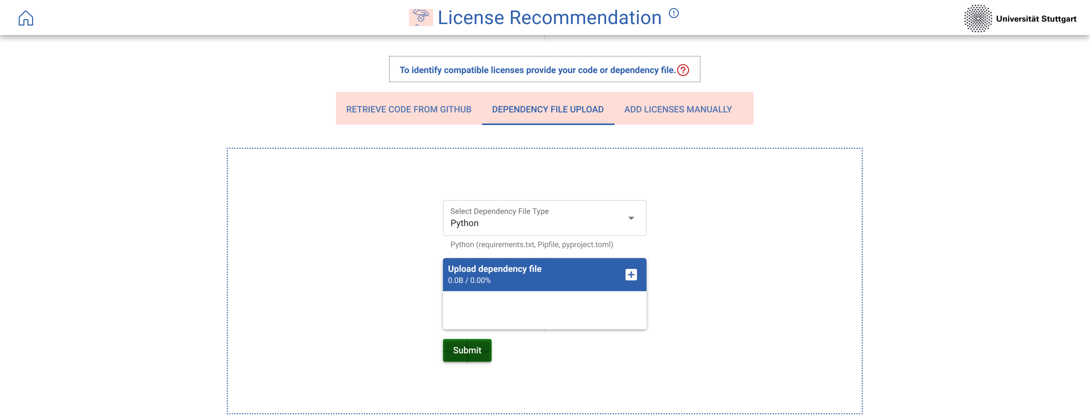
Specify the branch name (e.g., development, feature, etc.)

Figure 3: Upload Code.

By uploading a dependency file, the licenses of additional dependencies can be added to the list of licenses that should be taken into account (see Figure 4).

Based on a license ontology, the *License Checker* then takes this list of used licenses as input and provides a list of licenses as output that contains only licenses compatible with each of the input licenses (see Figure 5).

[2] https://www.fossology.org/; *Visited on March 29, 2025.*

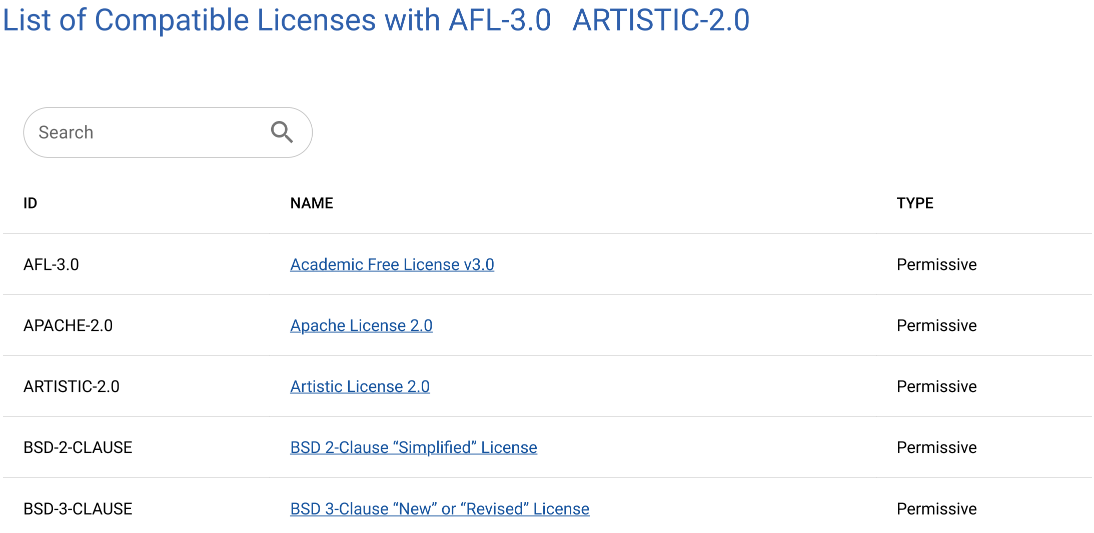


To identify compatible licenses provide your code or dependency file.

Select Dependency File Type: Python

Python (requirements.txt, Pipfile, pyproject.toml)

Figure 4: Upload Dependency File.



List of Compatible Licenses with AFL-3.0 ARTISTIC-2.0

ID	NAME	TYPE
AFL-3.0	Academic Free License v3.0	Permissive
APACHE-2.0	Apache License 2.0	Permissive
ARTISTIC-2.0	Artistic License 2.0	Permissive
BSD-2-CLAUSE	BSD 2-Clause “Simplified” License	Permissive
BSD-3-CLAUSE	BSD 3-Clause “New” or “Revised” License	Permissive

Figure 5: Compatibility Check Result.

2.3 The License Ontology

The license ontology models common open source licenses and license types together with their compatibility. Figure 6 visualizes the main classes and properties. The ontology builds on the SPDX license list[3] and the appendix of choosealicense.com[4].

Each software license has a type and a set of permissions, conditions and limitations. Compatibility between licenses is defined between license types (e.g, Permissive isForwardCompatibleWith Copyleft) for the general rules and between licenses for specific exceptions (e.g., EUPL-1.1 isExplicitlyCompatibleWith BSL-1.0).

[3] https://github.com/spdx/license-list-data; *Visited on March 28, 2025.*
[4] https://choosealicense.com/appendix/; *Visited on March 28, 2025.*

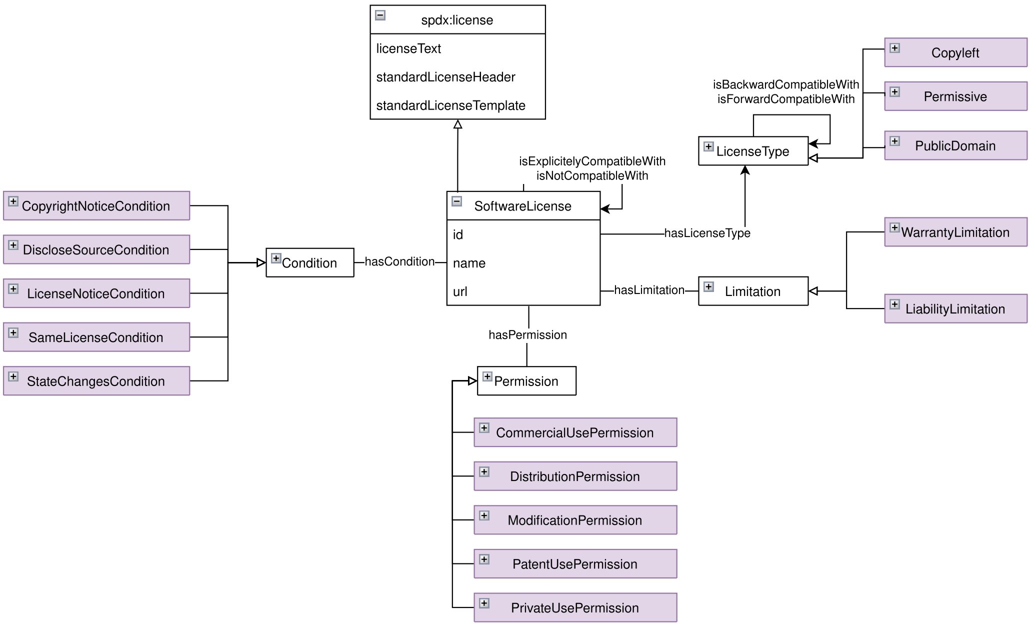


Figure 6: License Ontology.

2.4 Architecture

The license checker consists of several components, depicted in Figure 7. All components are available as docker containers[5] and can be deployed using docker compose. The *License Checker* frontend is a single page application using the frameworks VueJS[6] and Quasar[7]. The license engine uses the fossology toolkit to extract licensing information from existing code and provides the license texts. The *License Checker* backend, a REST API written in Python, uses the license ontology to provide information about software licenses and their compatibility.

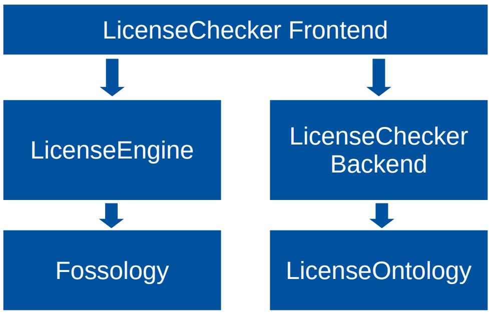


Figure 7: Components of the License Checker.

[5] https://github.com/izus-fokus/LicenseChecker; *Visited on March 28, 2025.*
[6] https://vuejs.org/; *Visited on March 31, 2025.*
[7] https://quasar.dev/; *Visited on March 31, 2025.*

3 Outlook

The license checker is actually in a usable protoype stage and available for testing. It can not replace legal advice and does also not provide legal certainty, but offers a possibility to get aware of possible license conflicts. Further development is planned within the national research data infrastructure NFDI in a newly founded working group "Licensing & Agreement Rule Automation (LARA)" of the NFDI section ELSA for ethical, legal and social aspects[8] by checking the legal soundness of the license ontology and integrating the tool into a framework to support researchers to license not only research software, but also research data.

Acknowledgements

The development of the *License Checker* Tool was funded by the German Research Foundation (DFG) within the ReSUS project with the grant number 425911815.

In addition to the authors, other people were involved in the development of the license checker components. Evgenia Yangulova and Saskia Holland conducted most of the research on the license ontology. Markus Hirsch worked on the license ontology and the licence checker backend, while Michael Zimmermann developed the license engine. Faisal Khan implemented main parts of the license checker backend. Usman Ghani Amin worked on the backend and on the frontend of the license checker. Iqra Imran did most of the work in designing and implementing the license checker frontend.

Authorship Contributions

Dorothea Iglezakis drafted and wrote the first version. Iqra Imran commmented and improved.

Conflict of Interest

The authors declare no conflicts of interest

[8] https://www.nfdi.de/section-elsa/; *Visited on March 28, 2025.*

Bibliography

Almeida, Daniel A., Gail C. Murphy, Greg Wilson, and Mike Hoye. 2017. "Do Software Developers Understand Open Source Licenses?" In *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC),* 1–11. https://doi.org/10.1109/ICPC.2017.7.