Copyright, Software Licenses and Python

Python User Group Berlin, 08.09.2022, Online

Tobias Schlauch <Tobias.Schlauch@DLR.de>
Institute for Software Technology
German Aerospace Center (DLR)
`http://www.dlr.de/sc`

Knowledge for Tomorrow

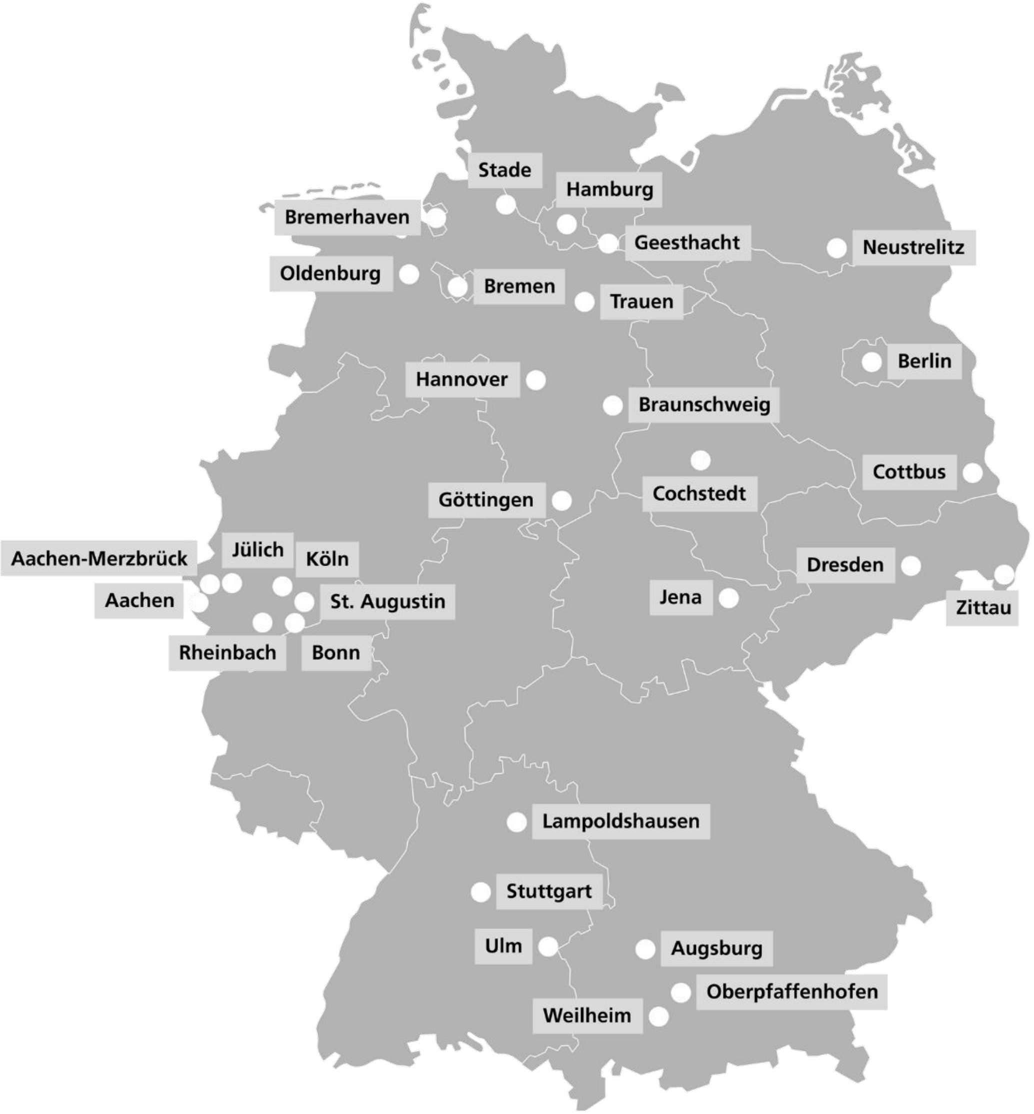
Research Software Development at DLR

Some context

- About 10.000 employees, ~20% of DLR employees involved in software development
- Variety of fields, maturity, and technologies: https://doi.org/10.1145/3387940.3392244

Brief history of DLR`s Software Engineering Initiative

- Activities started in 2005 as part of DLR`s quality assurance program
- Since 2017 focus moved more and more on research software development aspects
- Work is driven by the DLR institute for Software Technology and funded by DLR IT

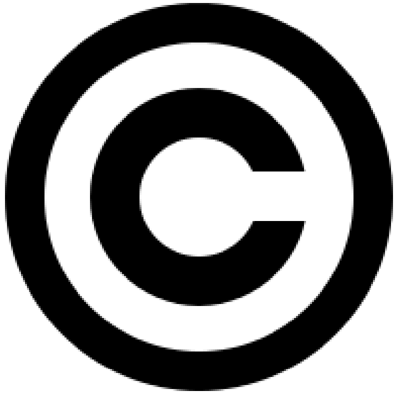


Copyright

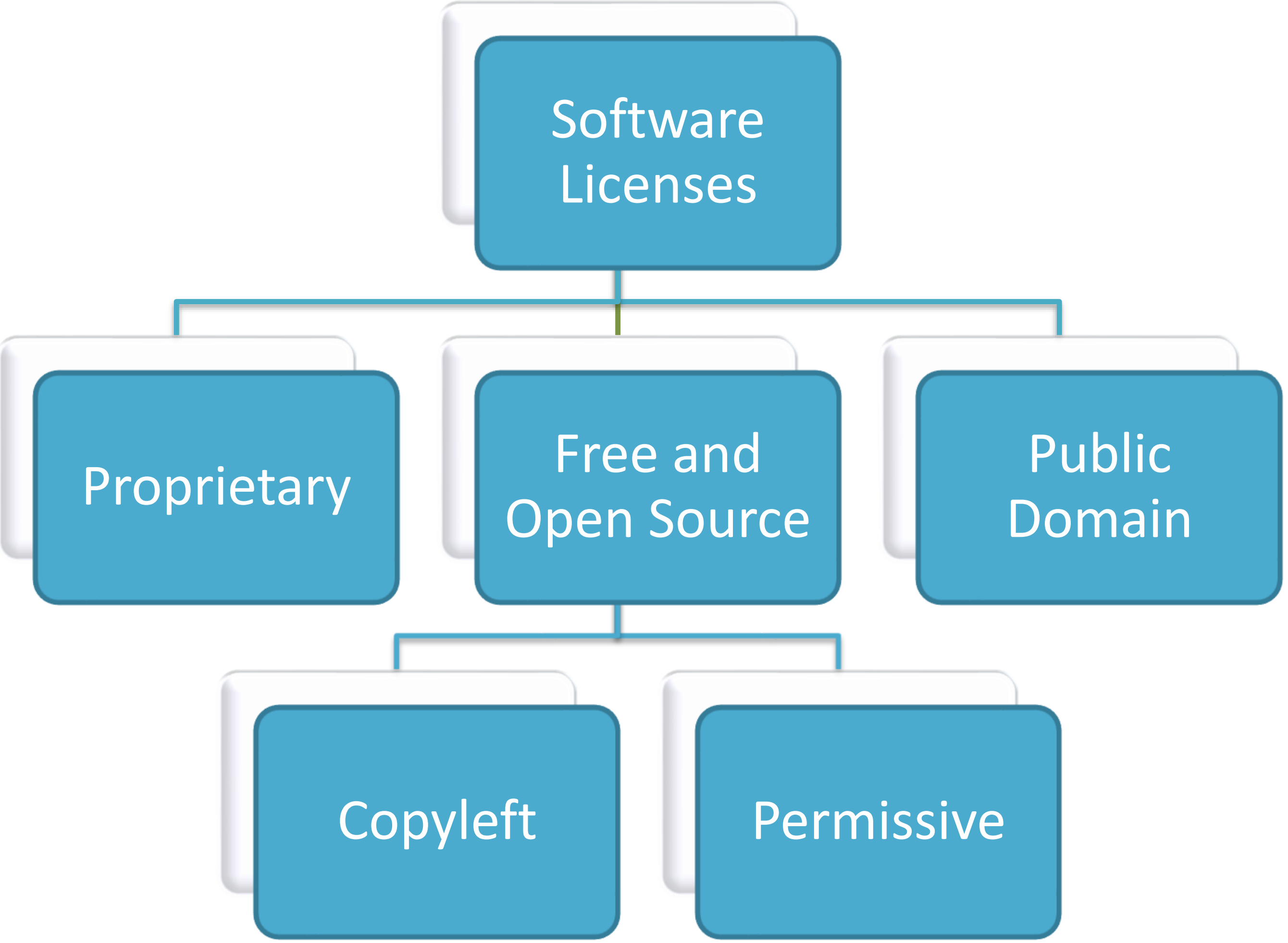
- **Copyright**
  - Software is protected by copyright.
  - Copyright protects the expression of an idea.
  - Copyright grants exclusive rights to the copyright holder.

Copyright

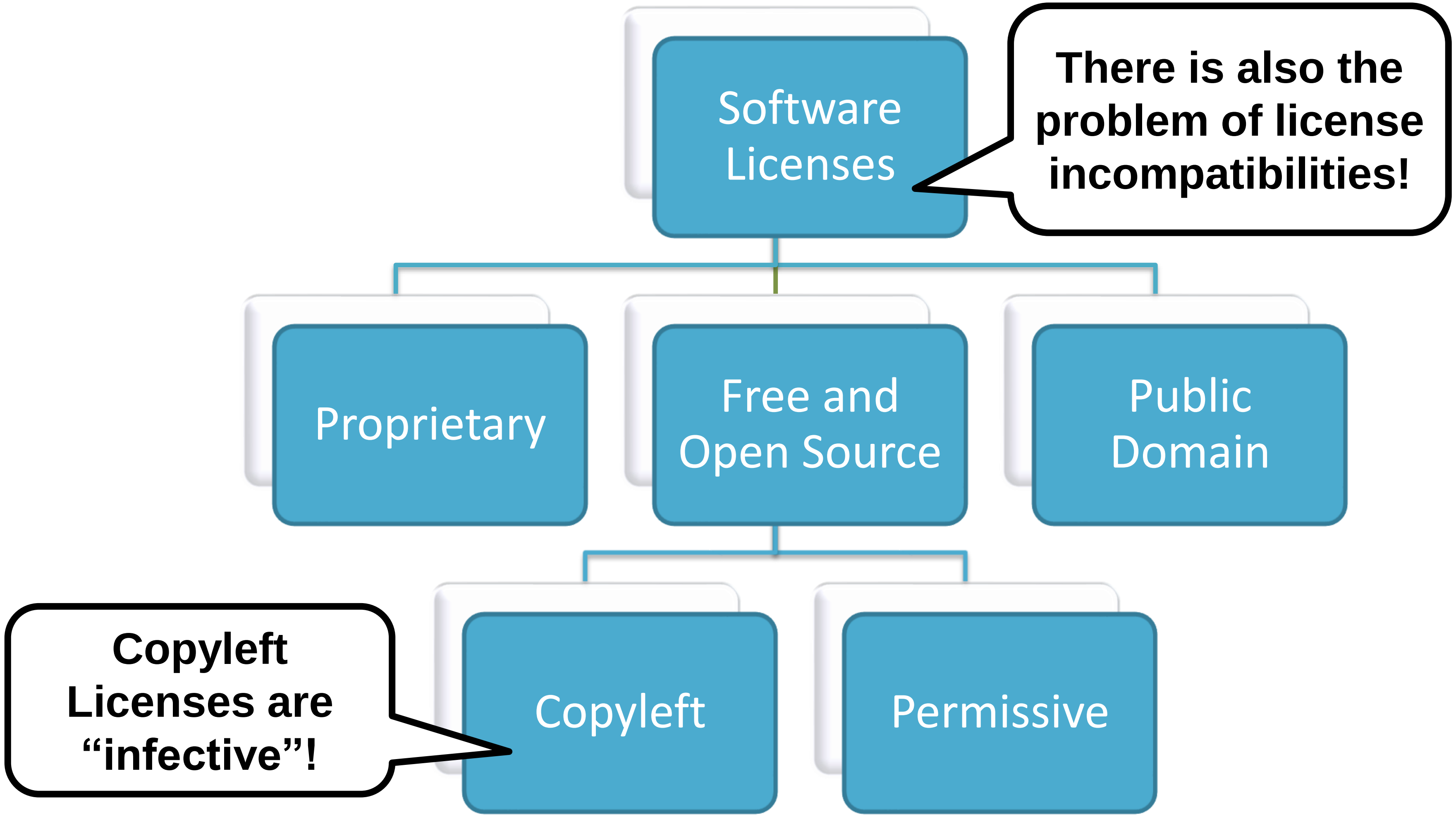
- **Copyright**
  - Software is protected by copyright.
  - Copyright protects the expression of an idea.
  - Copyright grants exclusive rights to the copyright holder.
- **Who is the copyright holder of a software?**
  - All contributors are considered as copyright holders and jointly exercise the rights granted by copyright.
  - A company paying an employed developer obtains most of the exclusive rights.



Software Licenses



Software Licenses



Software Licenses

Proprietary

Free and Open Source

Public Domain

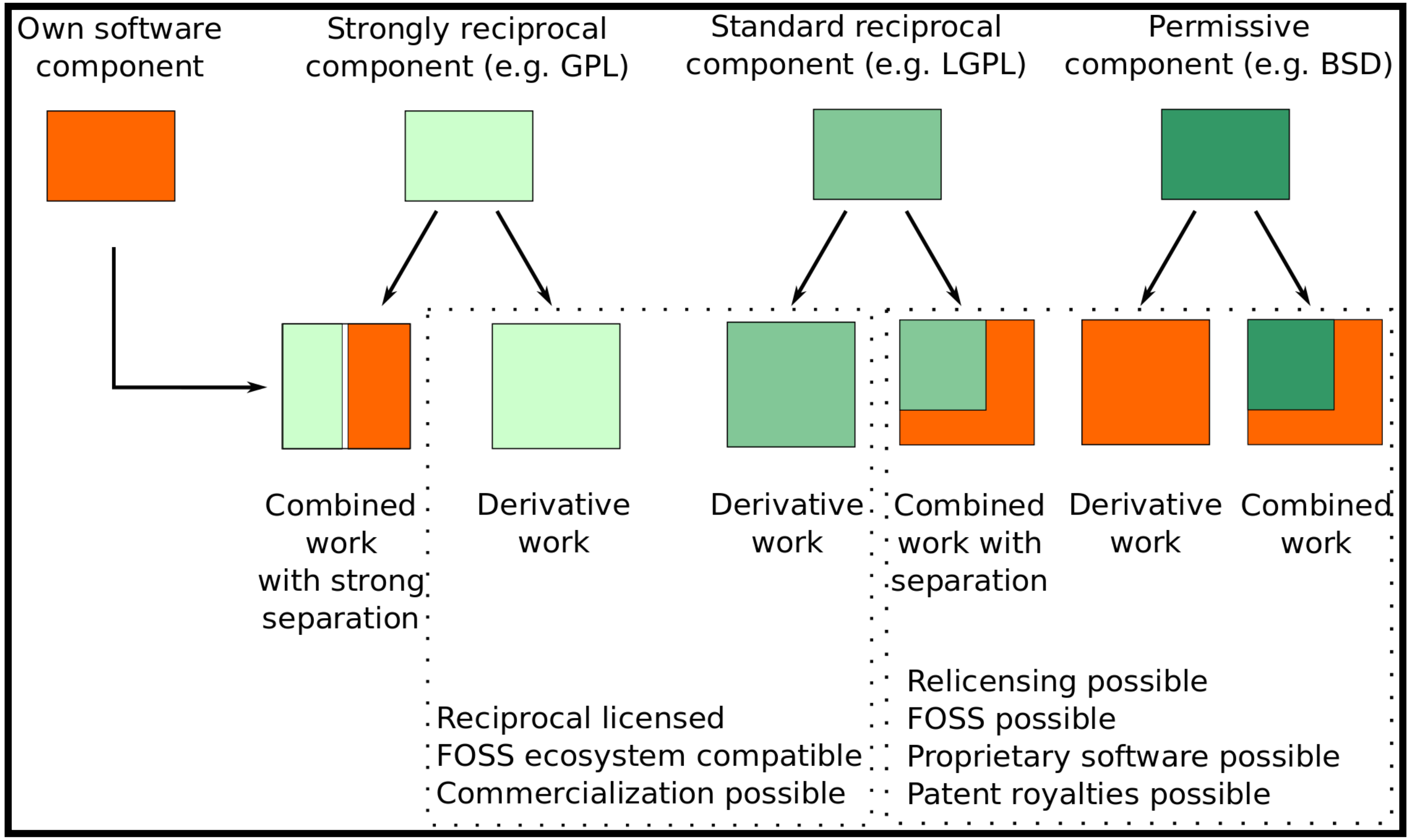
Copyleft

Permissive

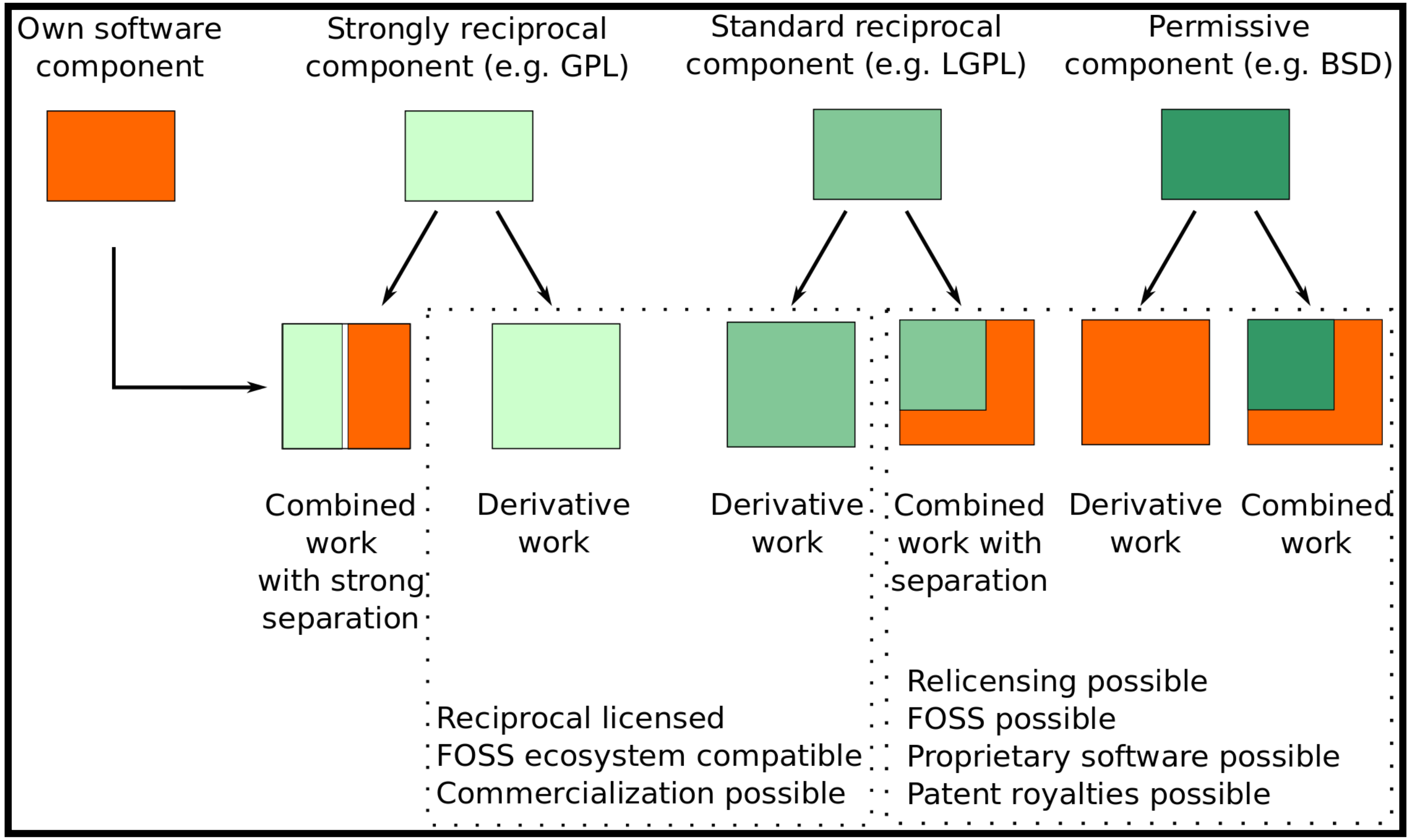
There is also the problem of license incompatibilities!

Copyleft Licenses are “infective”!

Combining Modules under Different Software Licenses


Own software component
Strongly reciprocal component (e.g. GPL)
Standard reciprocal component (e.g. LGPL)
Permissive component (e.g. BSD)
Combined work with strong separation
Derivative work
Derivative work
Combined work with separation
Derivative work
Combined work
Reciprocal licensed
FOSS ecosystem compatible
Commercialization possible
Relicensing possible
FOSS possible
Proprietary software possible
Patent royalties possible

Combining Modules under Different Software Licenses


Own software component
Strongly reciprocal component (e.g. GPL)
Standard reciprocal component (e.g. LGPL)
Permissive component (e.g. BSD)
Combined work with strong separation
Derivative work
Derivative work
Combined work with separation
Derivative work
Combined work

Reciprocal licensed
FOSS ecosystem compatible
Commercialization possible

Relicensing possible
FOSS possible
Proprietary software possible
Patent royalties possible

Make sure that every code that you use is covered by a license!

Take care when combining code under different licenses!

Minimal License Checklist

1. **Choose a license**
   - Consider strategical implications
   - Comply with licenses of third-party dependencies
2. **Ask your boss for permission**
3. **Add copyright holder and license information**

- **Assumptions:**
  - You created the software as part of your job.
  - You created the software completely on your own.

Minimal License Checklist

1. **Choose a license**
   - Consider strategical implications
   - Comply with licenses of third-party dependencies
2. **Ask your boss for permission**
3. **Add copyright holder and license information**

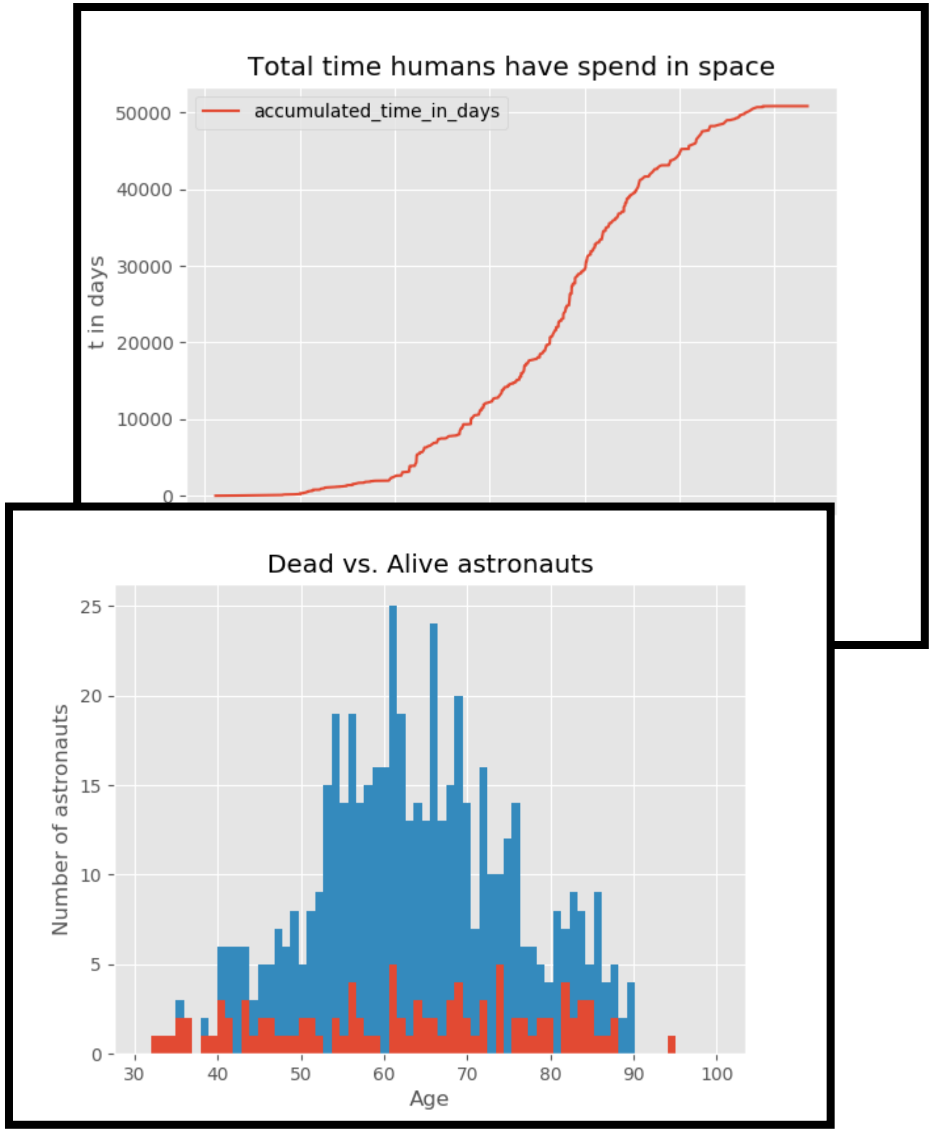
Find out about your organizational processes!

- **Assumptions:**
  - You created the software as part of your job.
  - You created the software completely on your own.

Ask for legal advice if you are unsure!

Example: Astronaut Analysis

- **Astronaut Analysis is a data publication consisting of:**
  - Data set
  - Analyze script written in Python using [pandas]() and [matplotlib]()
  - Result plots
- **Scenario:**
  - I created that work completely on my own as part of my job.
  - I want to publish it alongside with my paper to support reproducibility of my research results.
  - I want to make its usage as simple as possible.
- **Code repository:** [https://gitlab.com/hifis/hifis-workshops/make-your-code-ready-for-publication/astronaut-analysis/-/tree/3-add-docs]()



Example: Astronaut Analysis

Choose a License

- **I follow the recommendation from https://choosealicense.com/ and want to use the MIT License. But do the licenses of my dependencies fit?**

- **Python offers various tools to find out about licenses of your dependencies. I tried out the following:**
  - pylic: https://github.com/ubersan/pylic
  - liccheck: https://github.com/dhatim/python-license-check
  - pip-licenses: https://github.com/raimon49/pip-licenses

Example: Astronaut Analysis

Choose a License

- **I follow the recommendation from https://choosealicense.com/ and want to use the MIT License. But do the licenses of my dependencies fit?**

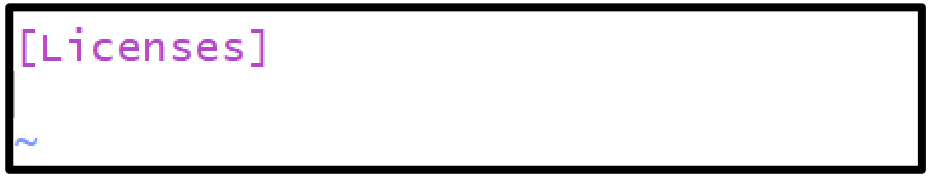
- **Python offers various tools to find out about licenses of your dependencies. I tried out the following:**
  - pylic: https://github.com/ubersan/pylic
  - liccheck: https://github.com/dhatim/python-license-check
  - pip-licenses: https://github.com/raimon49/pip-licenses

- **All tools do a proper job and offer similar features such as:**
  - Using package metadata to find out about licenses
  - Working on the basis of installed Python dependencies
  - Allowing you to check for unwanted licenses

Example: Astronaut Analysis

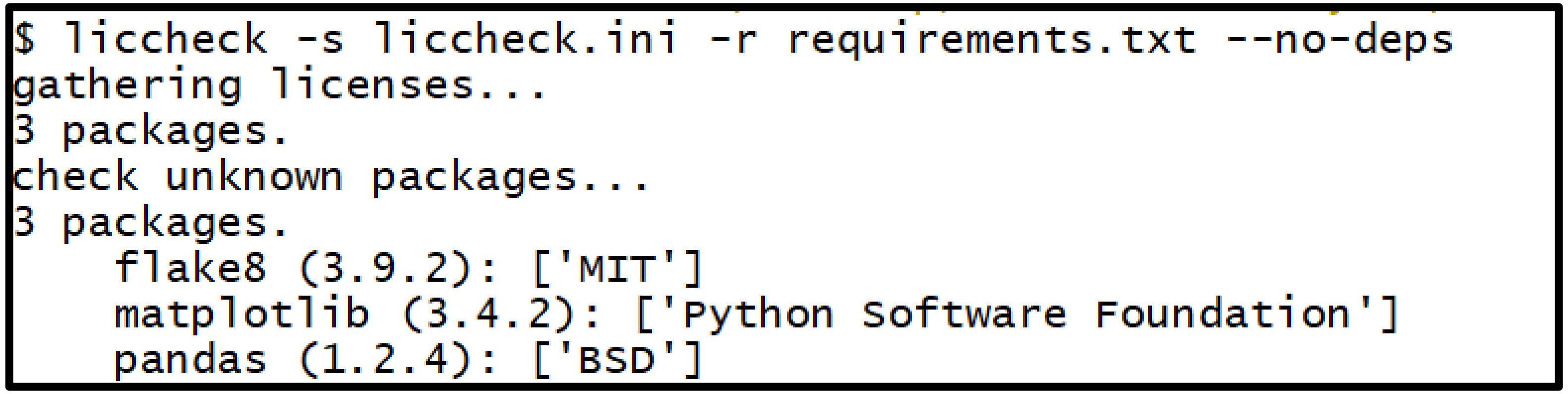
Check the Dependencies with `liccheck`

- Install `liccheck` via: `pip install liccheck`
- Install the dependencies via: `pip install -r requirements.txt`
- Add a minimal `licconfig.ini`:



```
[Licenses]

~
```

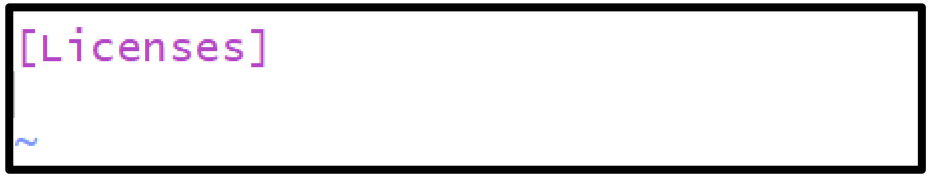
- Check our top-level dependencies:



```
$ liccheck -s liccheck.ini -r requirements.txt --no-deps
gathering licenses...
3 packages.
check unknown packages...
3 packages.
    flake8 (3.9.2): ['MIT']
    matplotlib (3.4.2): ['Python Software Foundation']
    pandas (1.2.4): ['BSD']
```

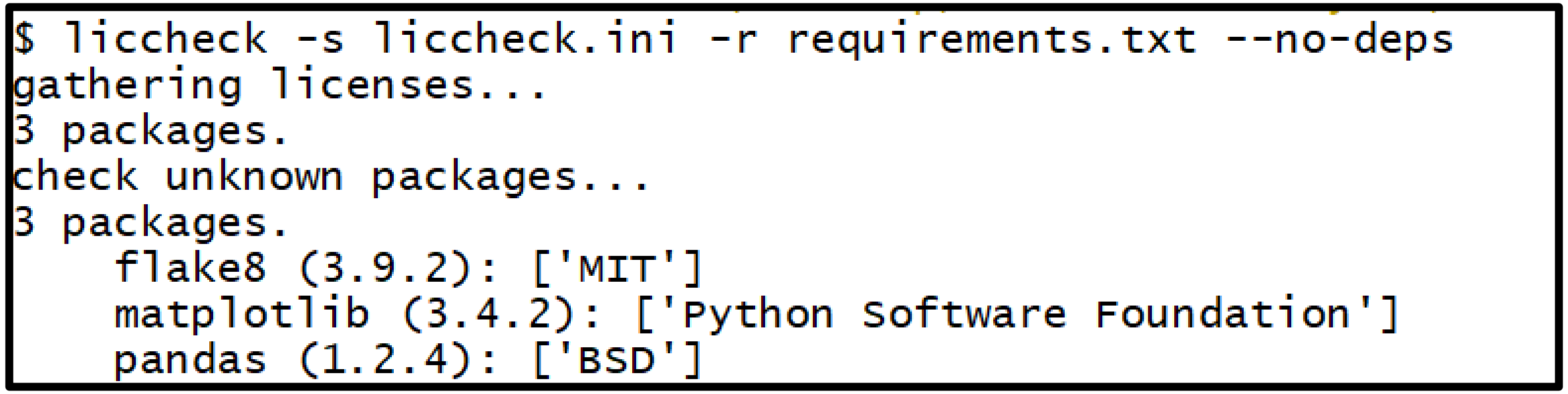
Example: Astronaut Analysis
Check the Dependencies with **`liccheck`**

- Install `liccheck` via: `pip install liccheck`
- Install the dependencies via: `pip install -r requirements.txt`
- Add a minimal `licconfig.ini`:



```
[Licenses]

~
```

- Check our top-level dependencies:



```
$ liccheck -s liccheck.ini -r requirements.txt --no-deps
gathering licenses...
3 packages.
check unknown packages...
3 packages.
    flake8 (3.9.2): ['MIT']
    matplotlib (3.4.2): ['Python Software Foundation']
    pandas (1.2.4): ['BSD']
```

Some pitfalls:

- Non-standard license identifiers
- Find the right dependencies
- Check sub-dependencies

In our example:

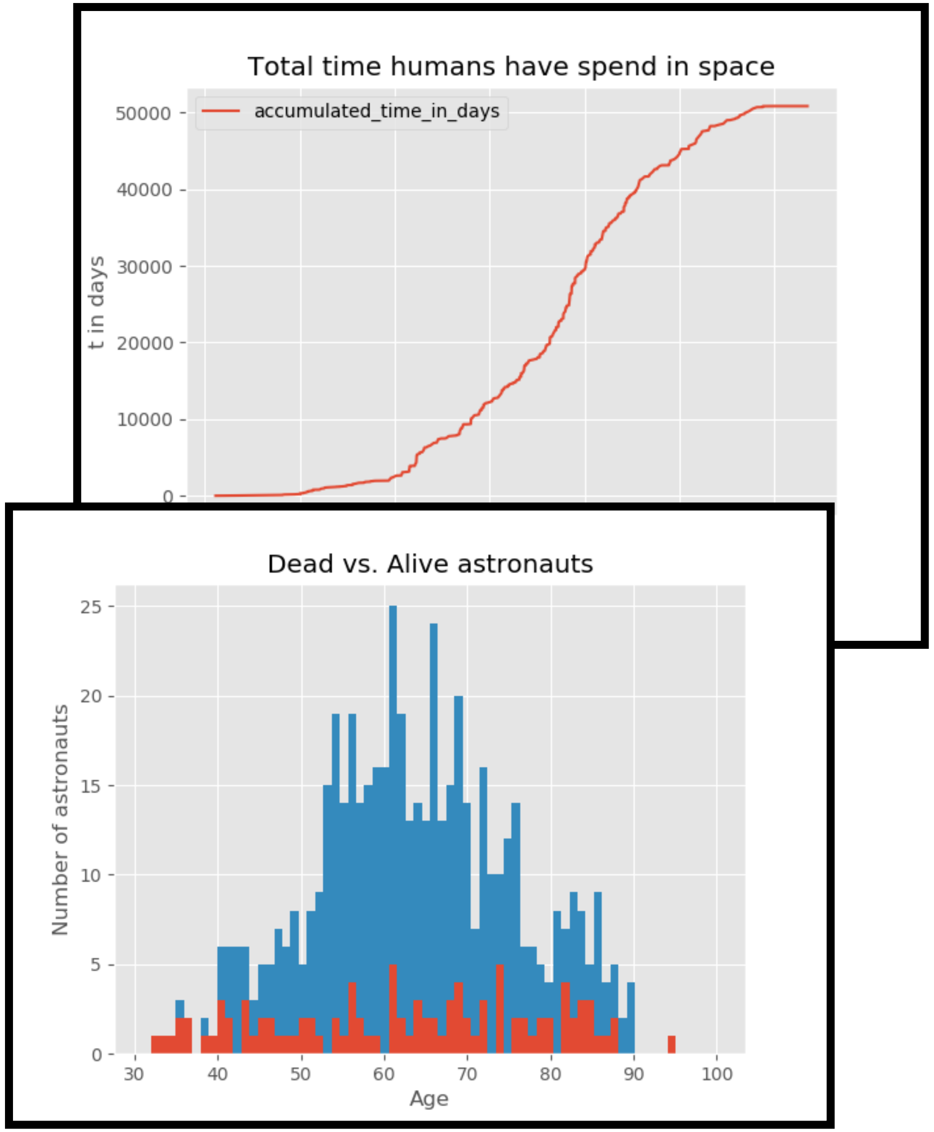
- pandas and matplotlib are the relevant packages
- All involved licenses are compatible with MIT

Example: Astronaut Analysis
Finalize Decisions

- **Although the license decision for our source code is fine, we might not want to use MIT for all content!**

- **Final copyright and license decisions of the example:**
  - Copyright holder: German Aerospace Center
  - Source code: MIT
  - Data set: CC0-1.0
  - Documentation and plots: CC-BY-4.0
  - Insignificant files: CC0-1.0

- **My boss is fine with it. But how to best add this information to our files?**



Example: Astronaut Analysis

How to best add Copyright Holder and Licensing Information?

- **Typical questions:**
  - How and where do I indicate the copyright holders?
  - How and where do I provide license information?
  - Do I really need the long legal texts in all files headers?
  - How to handle different copyright holders and licenses?
  - And much more…

Example: Astronaut Analysis

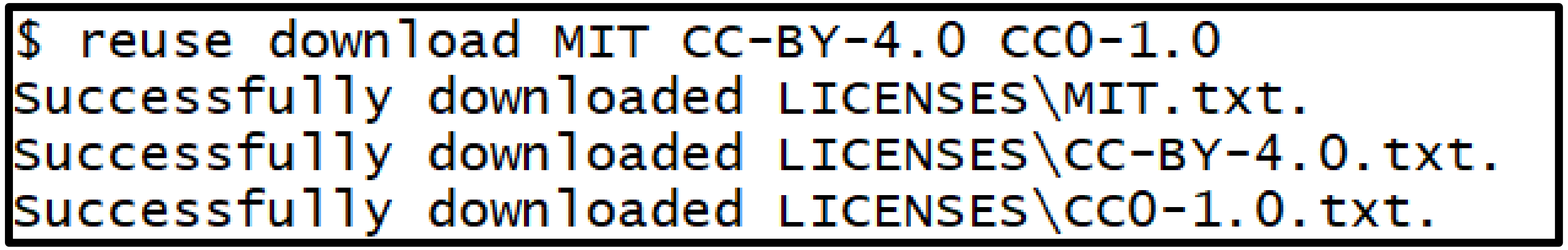
How to best add Copyright Holder and Licensing Information?

- **Typical questions:**
  - How and where do I indicate the copyright holders?
  - How and where do I provide license information?
  - Do I really need the long legal texts in all files headers?
  - How to handle different copyright holders and licenses?
  - And much more…
- **Let us try the REUSE Approach :)**
  - **Goal:** Make it easy to determine license and copyright holders of a file for humans and machines!
  - Heavily builds on SPDX: https://spdx.dev/
  - For more information: Tutorial, FAQ, Specification
  - Provides the `reuse` helper tool written in Python: https://git.fsfe.org/reuse/tool

Example: Astronaut Analysis

REUSE – Step 1: Add License Files

- Add all required license files to the `LICENSES` directory
- You can provide them manually or via `reuse download`:



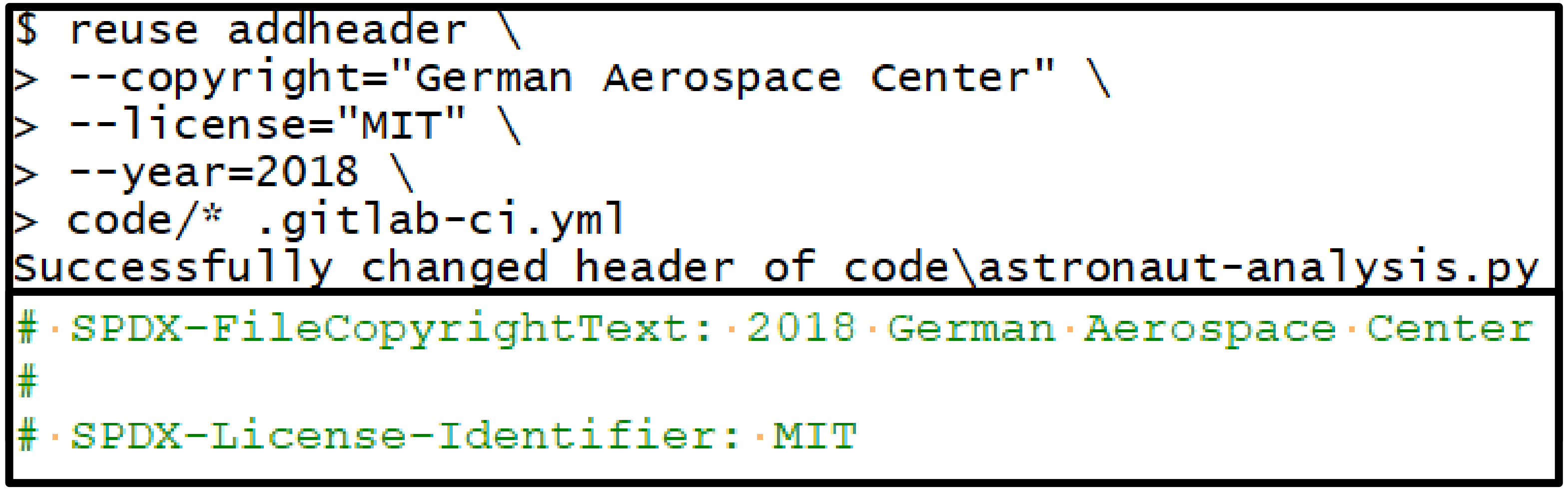
```
$ reuse download MIT CC-BY-4.0 CC0-1.0
Successfully downloaded LICENSES\MIT.txt.
Successfully downloaded LICENSES\CC-BY-4.0.txt.
Successfully downloaded LICENSES\CC0-1.0.txt.
```

- **Notes:**
  - `reuse download` supports all licenses from the SPDX License List Data repository: https://github.com/spdx/license-list-data
  - Other licenses can be added manually to the `LICENSES` directory
  - `reuse download` can download all missing license files

Example: Astronaut Analysis

REUSE – Step 2: Add Copyright and Licensing Information

- Add copyright and licensing information via SPDX tags to all files in the code repository
- You can attach them manually or via `reuse addheader`:



```
$ reuse addheader \
> --copyright="German Aerospace Center" \
> --license="MIT" \
> --year=2018 \
> code/* .gitlab-ci.yml
Successfully changed header of code\astronaut-analysis.py
```

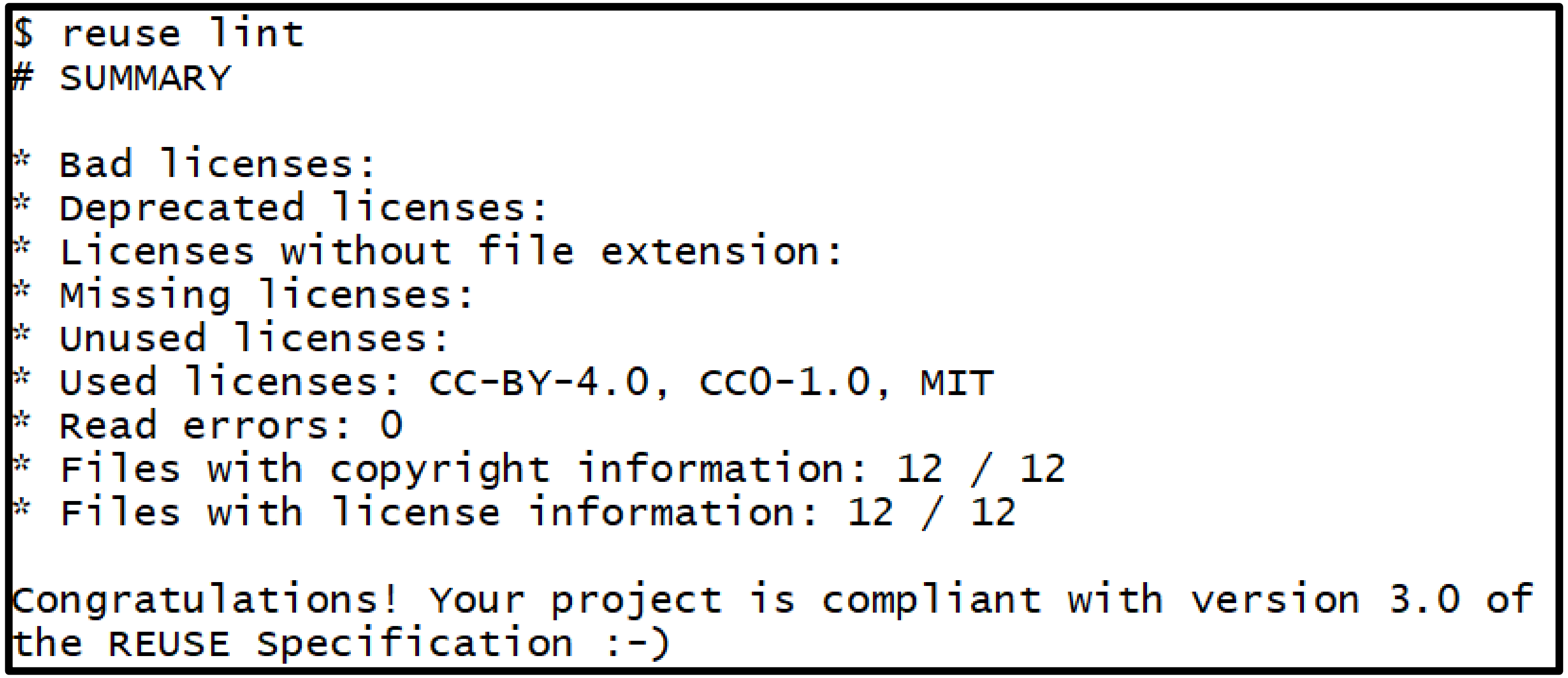
```
# SPDX-FileCopyrightText: 2018 German Aerospace Center
#
# SPDX-License-Identifier: MIT
```

- **Notes:**
  - `reuse addheader` handles comment styles and uncommentable files automatically
  - You can have multiple copyright and license statements in each file
  - License identifiers are standardized (see also: SPDX expressions)

Example: Astronaut Analysis

REUSE – Step 3: Check REUSE Compliance

- You can check whether everything is fine via `reuse lint`:

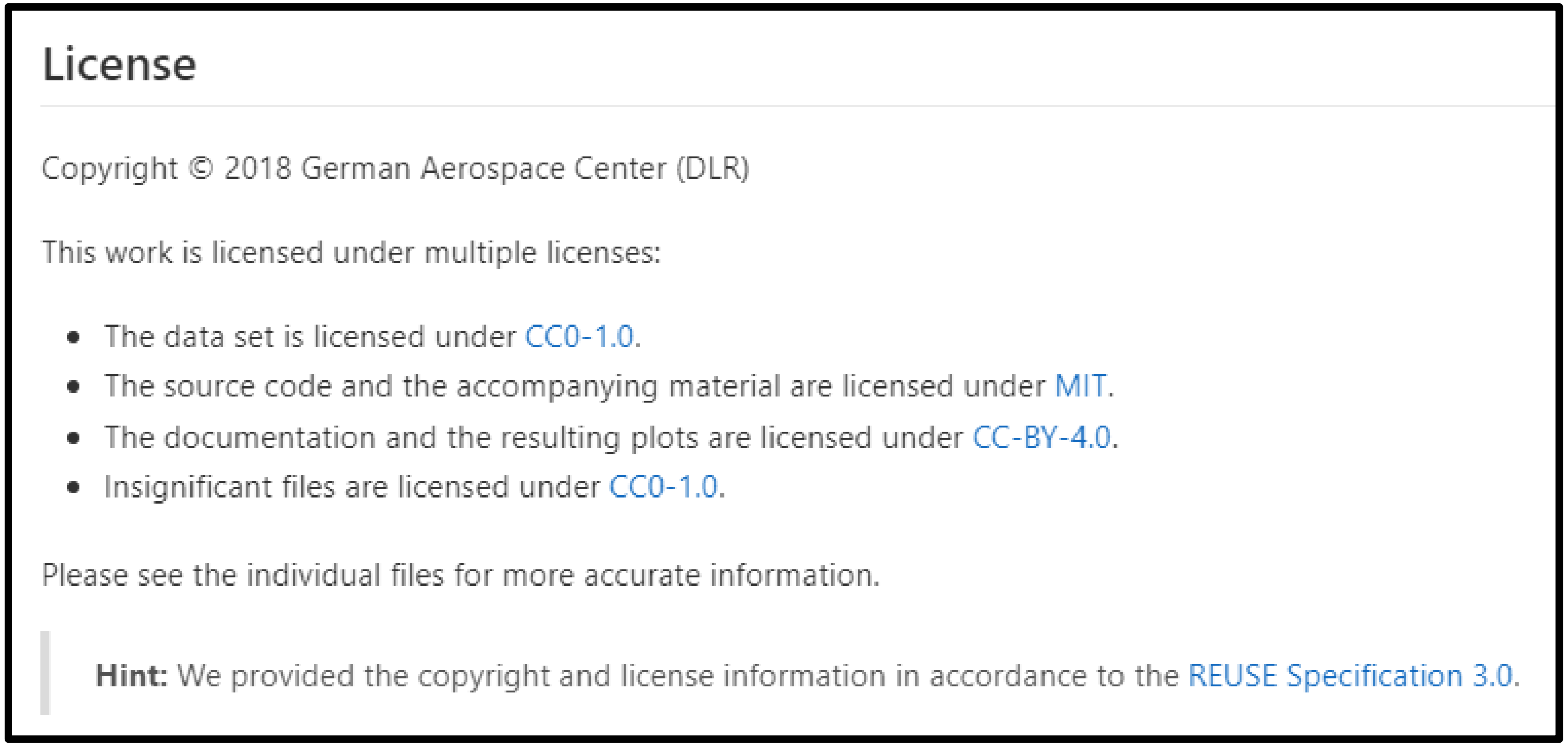


```
$ reuse lint
# SUMMARY

* Bad licenses:
* Deprecated licenses:
* Licenses without file extension:
* Missing licenses:
* Unused licenses:
* Used licenses: CC-BY-4.0, CC0-1.0, MIT
* Read errors: 0
* Files with copyright information: 12 / 12
* Files with license information: 12 / 12

Congratulations! Your project is compliant with version 3.0 of
the REUSE Specification :-)
```

- You can use the linter in pre-commit hooks or in your CI/CD workflow: https://reuse.software/dev/

Example: Astronaut Analysis

Add a License Hint



License

Copyright © 2018 German Aerospace Center (DLR)

This work is licensed under multiple licenses:

- The data set is licensed under CC0-1.0.
- The source code and the accompanying material are licensed under MIT.
- The documentation and the resulting plots are licensed under CC-BY-4.0.
- Insignificant files are licensed under CC0-1.0.

Please see the individual files for more accurate information.

> **Hint:** We provided the copyright and license information in accordance to the REUSE Specification 3.0.

Example: Astronaut Analysis
What about Package Metadata?

- Well, in our example we might not need them and a license hint in our README might be enough! But if you have a Python package, you could use the following package metadata as defined in PEP 621:
  - `authors/maintainers` – for copyright holders
  - `license` – for license information
  - `classifiers` – for license information

Example: Astronaut Analysis
What about Package Metadata?

- Well, in our example we might not need them and a license hint in our README might be enough! But if you have a Python package, you could use the following package metadata as defined in PEP 621:
  - `authors/maintainers` – for copyright holders
  - `license` – for license information
  - `classifiers` – for license information
- Unfortunately, the semantics of the metadata fields are not entirely defined ☹ But hopefully PEP 639 will fix it for license information!
- Some ideas how to deal with the fields when you already have REUSE in place:
  - Define the `license` field using a SPDX expressions (e.g., `MIT AND CC-BY-4.0 AND CC0-1.0`)
  - Add `classifiers` for all licenses if possible
  - Add all license files to your package!
  - Focus on proper contact information in the `authors/maintainers` fields

Summary

- Copyright **protects software** and **grants exclusive rights to the copyright holder(s)**.
- Software licenses offer a **“defined” way to grant rights to others** but also **require you to fulfill certain obligations**.
- **Software publication processes work differently** depending on your organization.
- **In the Python universe:**
  - There are some good tools to find out about licenses of your dependencies!
  - Python package metadata is a bit ambiguous if you look for copyright holder and license information ☹
  - REUSE is a Python-independent approach to properly manage copyright holder and license information for your code and other files ☺

➢You can take a look at the new DLR Open Source Brochure if you interested in more legal background!

Copyright and License Information

- All content is licensed under Attribution 4.0 International (CC BY 4.0) with the following exceptions:
  - DLR logo, slide layout, DLR locations map on slide 2, © German Aerospace Center. All rights reserved.
  - Copyright logo, slides 3/4, public domain, source: https://commons.wikimedia.org/wiki/File:Copyright.svg
  - License compatibility, slides 7/8, image by Mikko Välimäki, public domain, source: https://commons.wikimedia.org/wiki/File:Software-license-compatiblity-graph.svg
  - Philae landing on comet 67 P/Churyumov-Gerasimenko, slide 27: German Aerospace Center, CC BY 3.0

Thank you! Questions?

Want to work with us?
www.dlr.de/jobs

Tobias.Schlauch@dlr.de

www.DLR.de/sc | @TobiasSchlauch