

Respecting Open Source Licenses !

What is a Derivative Work?

A review from the viewpoint of a benevolent open source developer

Karsten Reincke,

Open Source CompLianCe and Advisory in Deutsche Telekom

Erleben, was verbindet.



ROSL!

Free and open source software and its **core idea**:

Be free and **feel free to reuse** the software!

Free Software Definition*

- 0: The freedom to **run** the program, for any purpose.
- 1: The freedom to **study** how the program works, and change it ...
- 2: The freedom to **redistribute copies** ...
- 3: The freedom to **distribute copies of your modified versions** ...



* cf: <http://www.gnu.org/philosophy/free-sw.en.html>

ROSL!

Free and open source software and its **core idea**:

Be free and **feel free to reuse** the software!

Open Source Definition*

Free Redistribution: The license shall not restrict any party from selling or giving away the software ...

Source Code: The program must include source code ...

Derived Works: The license must **allow modifications and derived works** ...



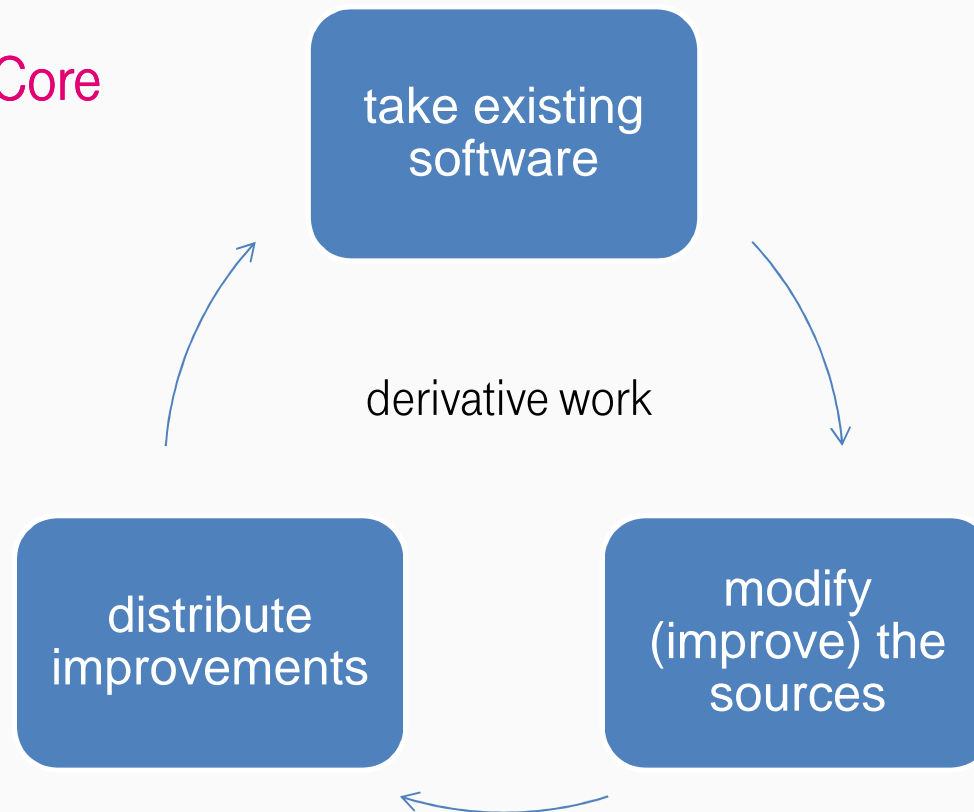
* cf: <http://opensource.org/osd>

R^{OS}L!

Free and open source software and its **core idea**:

Be free and **feel free to reuse** the software!

Open Source Core



* cf: <http://opensource.org/osd>

ROSL!

Free and open source software and its **core idea**:

Be free and **feel free to reuse** the software!

Open Source Licenses

For preserving the protecting power of the open source software, the open source license bind conditions to the distribution of **derivative works**!

“ Preamble: [...] To protect your rights, we need to make restrictions that forbid anyone to deny you these rights [...] “*

* cf.: GPLv2 0, <http://opensource.org/licenses/GPL-2.0>

R^{OS}L!

Free and open source software and its **core idea**:

Be free and **feel free to reuse** the software!

Open Source Licenses

Do they really do that?

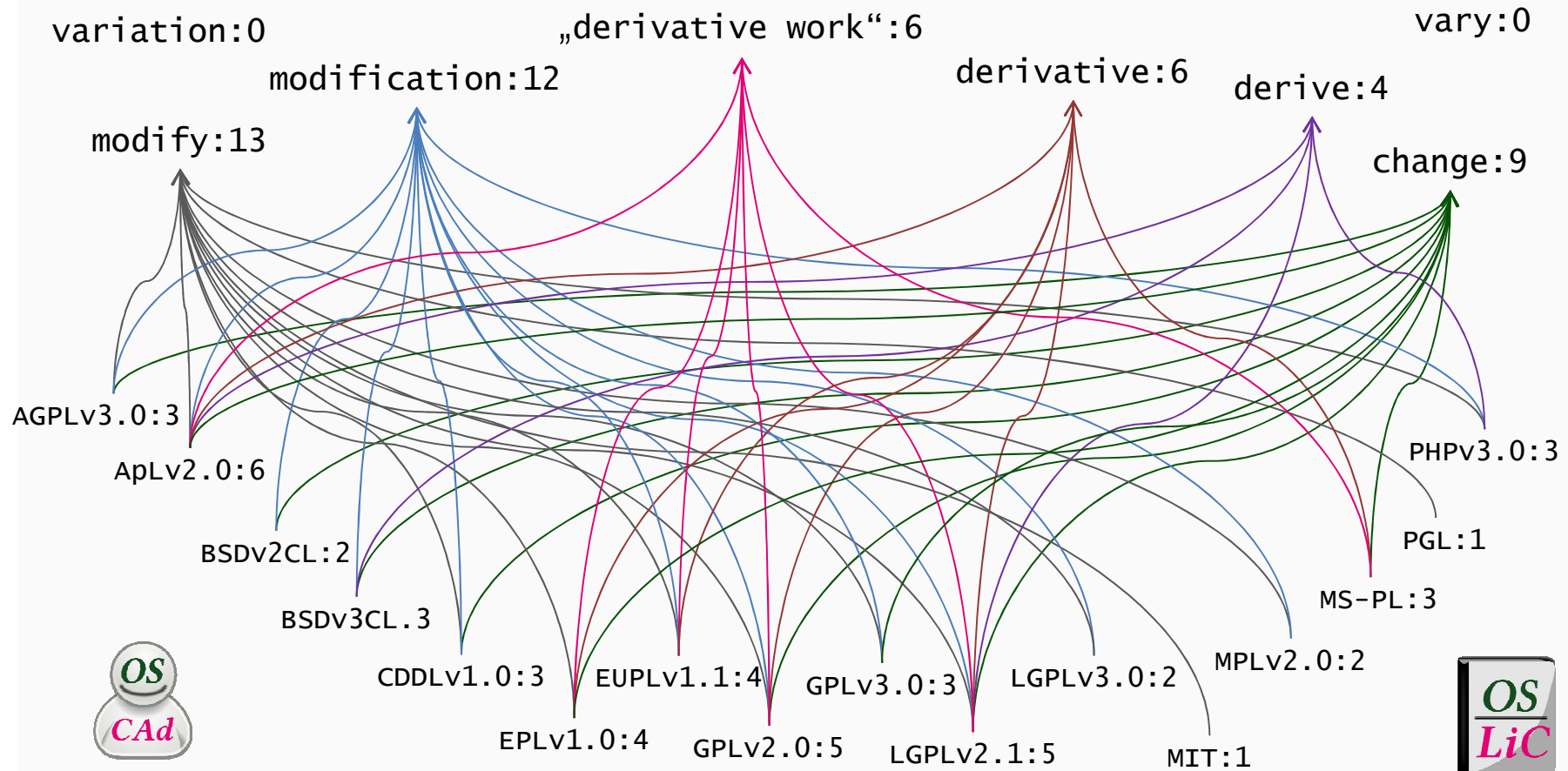
Do they indeed talk about **derivative works**?

Which other circumscriptions do they use?

R_{OS}L!

The leading question **for open source users:**

What do they mean when talking about **derivatives**?

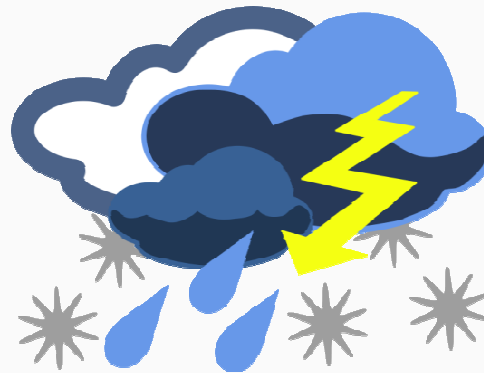


ROSL!

The leading question **for open source users:**

What do they mean when talking about **derivatives**?

What is a **derivative work**?



The question must
be answered for
acting **compliantly**.

An answer concerns
the spirit of **open**
source software.

Open source licenses do not **answer this question**

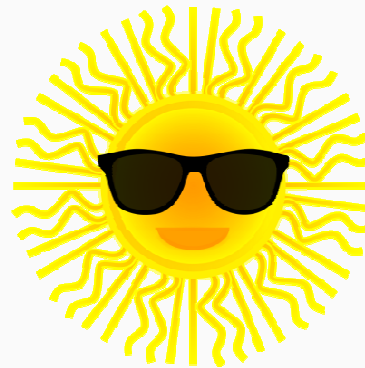
(consistently / unequivocally).

ROSL!

The leading question **for open source users:**

What do they mean when talking about **derivatives**?

What is a **derivative work**?



Open source licenses
are (often) **written by**
developers.

Open source licenses
are (mostly) **written for**
developers.

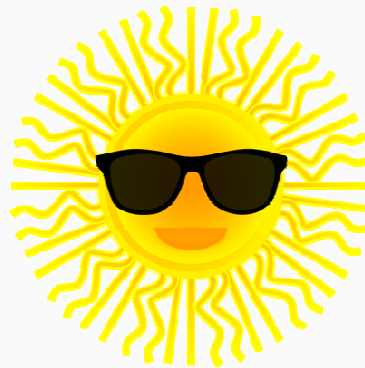
Let us **ask a** benevolent
software developer!

R^{OS}L!

The leading question **for open source users:**

What do they mean when talking about **derivatives**?

What is a **derivative work**?



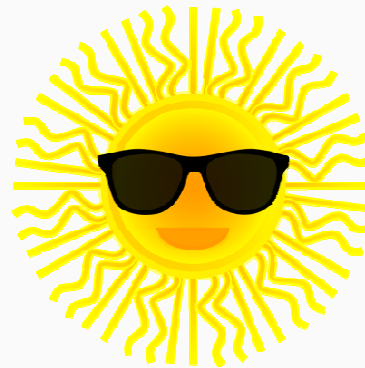
Let us **find** a **consensus**!

R^{OS}L!

The leading question **for open source users:**

What do they mean when talking about **derivatives**?

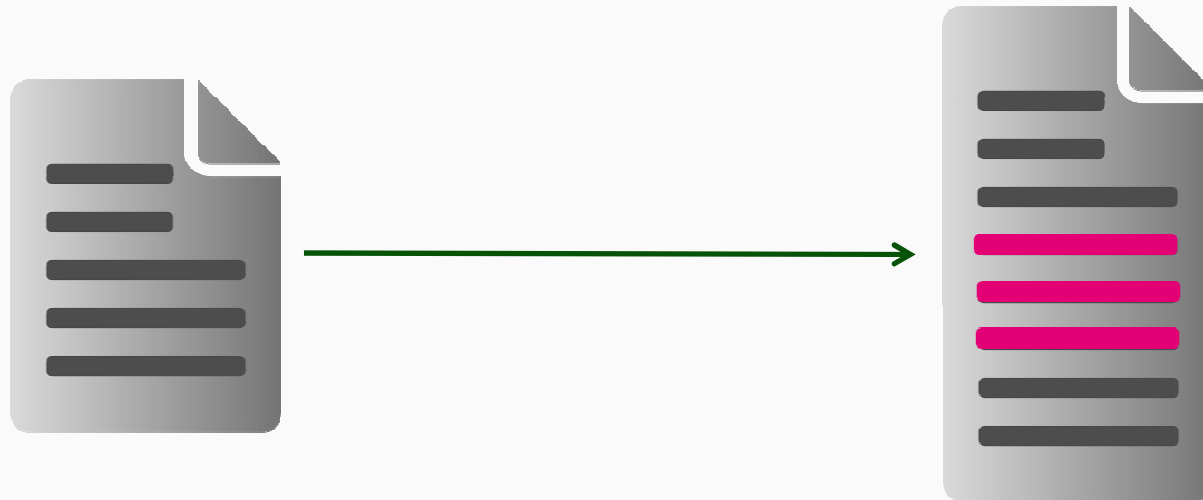
... being to greedy is not a **problem!**



Let us find a consensus,
from **simple to complex** cases

R^OS^L!

The **enlarging case**:
For a broad consensus!



Enlarging an existing source file by our own text **evokes a derivative work**, because we want to reuse the existing code for simplifying our life.

R^OS^L!

The **reducing case**:
For a broad consensus!



Reducing a source file **evokes a derivative work**:
– because we want prepare the given file for a better reuse.

R_OS_L!

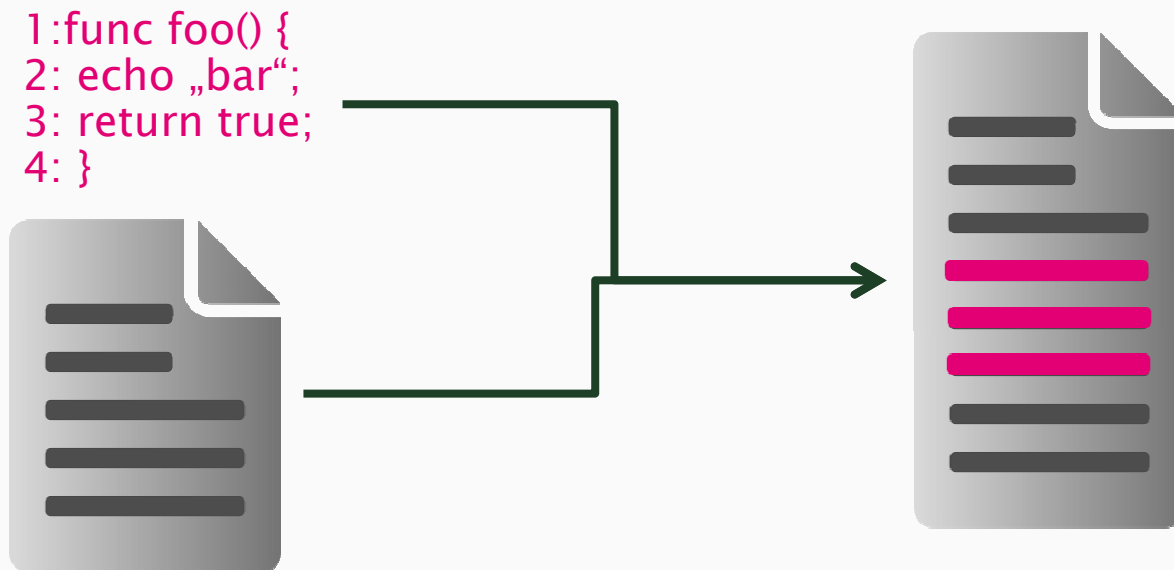
The **replacing case**:
For a broad consensus!



Replacing something in a source file **evokes a derivative work**: – follows from the reducing and the enlarging case

R^OSL!

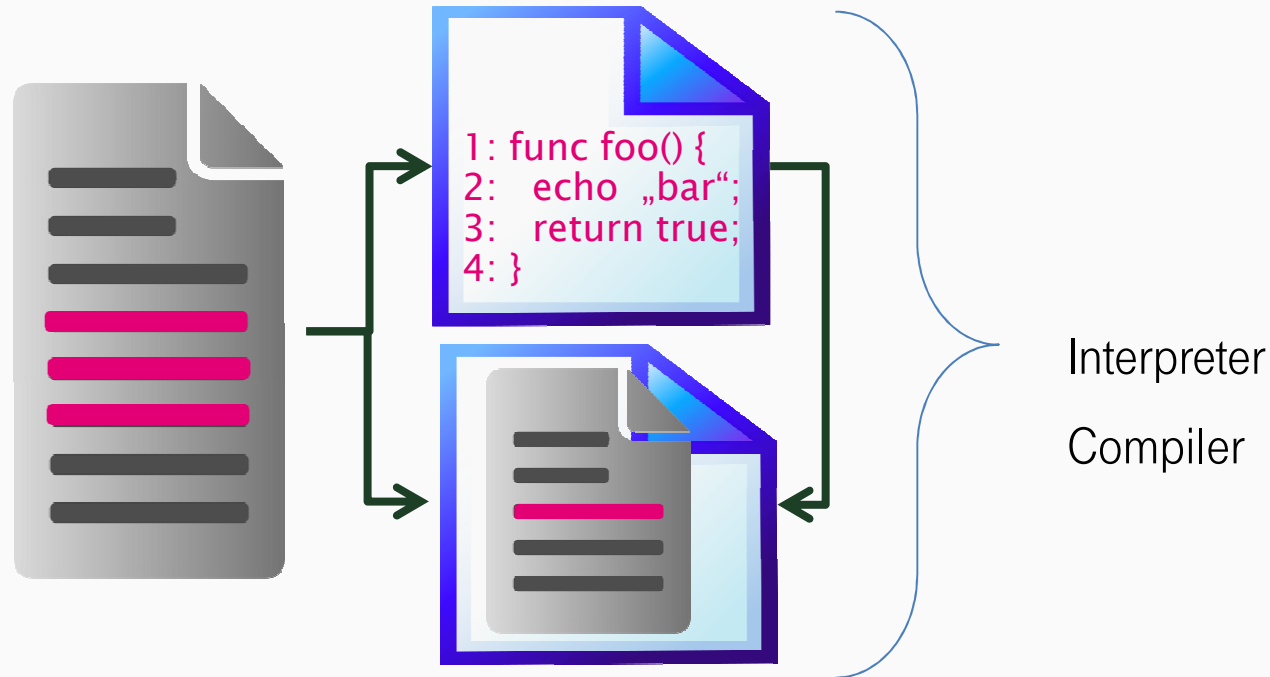
The **snippet case**:
For a broad consensus!



Integrating a foreign **snippet** into an existing source **evokes a derivative work**, twice: we are simplifying our life as by reusing both, the foreign snippet and the original file

R_{OSL}!

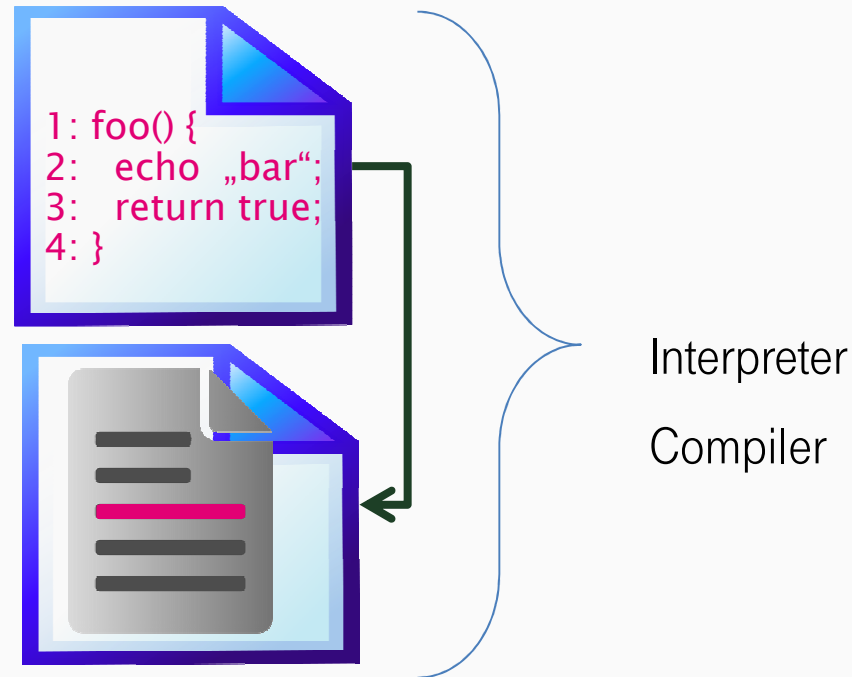
The **refactoring case**:
Hopefully for a broad consensus!



Refactoring a given work let (a) depend all modified / generated files on the original and (b) those with function calls on the function defining files.

R^{OS}L!

The **module case**:
Hopefully for a broad consensus!

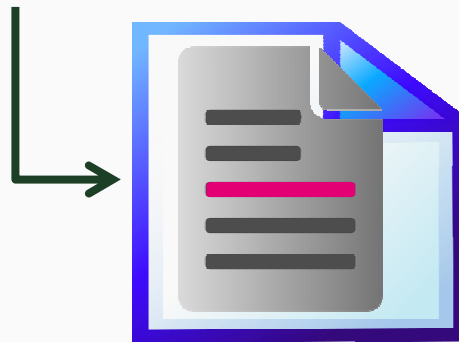


Calling a function - served by a defining module - **let the calling file become a derivative work** of the serving module.

R_{OSL}!

The **library case**:
Hopefully for a broad consensus!

```
1: foo() {  
2:   echo „bar“;  
3:   return true;  
4: }  
1: class Foo {  
2:   private int bar;  
3:   public Foo() {  
4:     [...]  
5:   }  
1: #define FOO „bar“;  
2:  
3: string foo=FOO;
```

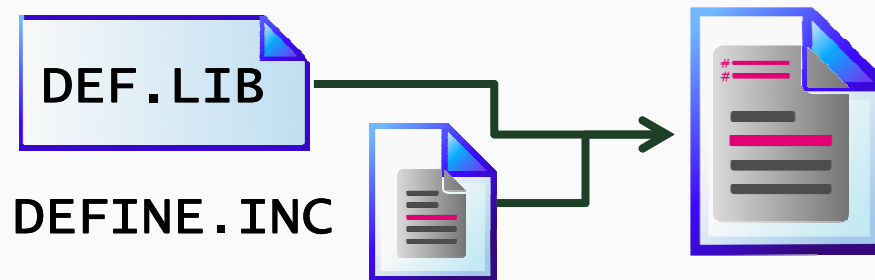
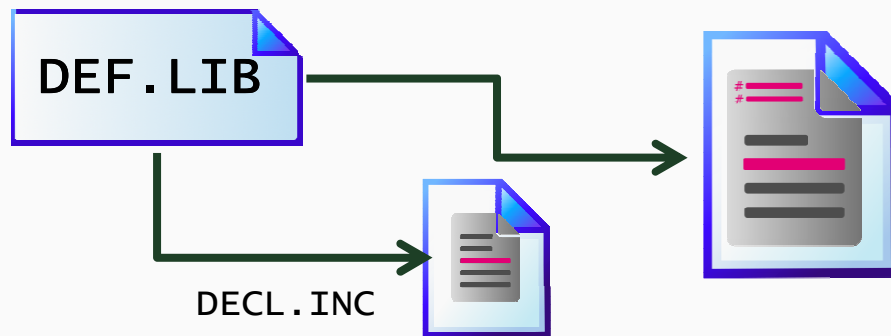


Interpreter
Compiler

Calling elements - served by a defining library - **let the calling file become a derivative work** of the serving library.

R^OSL!

The **import case**:
Hopefully for a broad consensus!



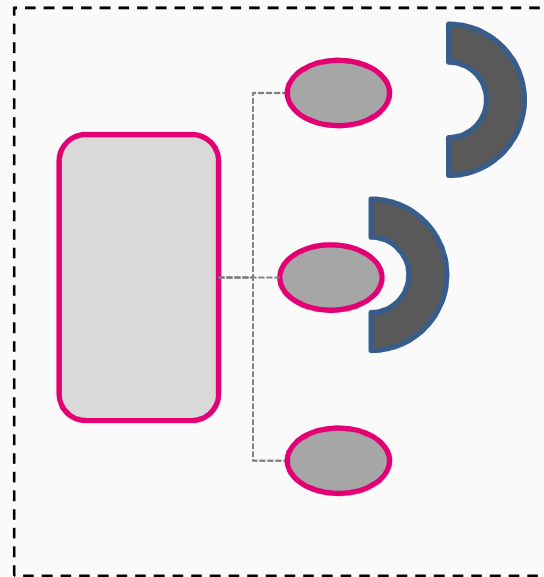
Interpreter
Compiler

Meaningless differentiation: the including file is a derivative work even if it only includes a declaring file

ROSL!

The **plugin case**:
Hopefully for a broad consensus!

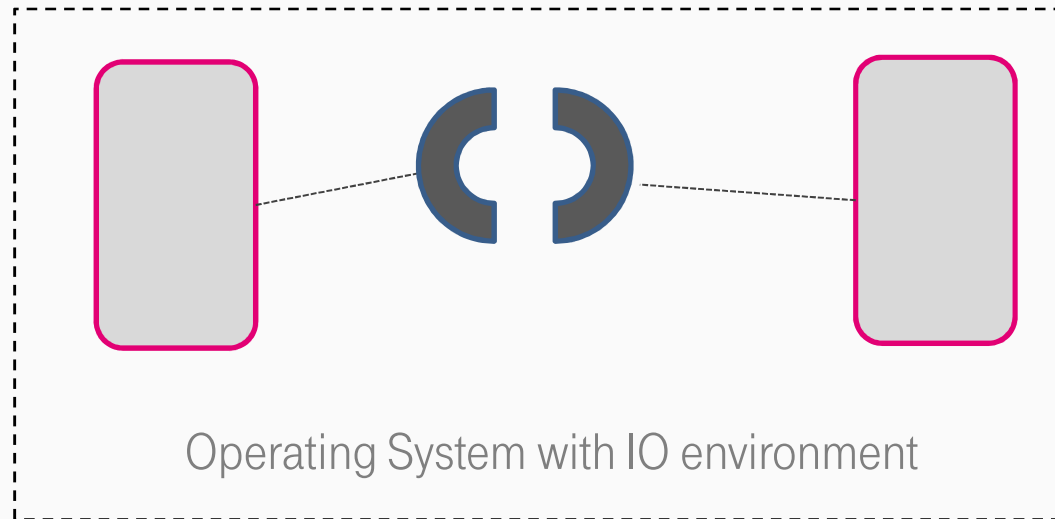
PROPOSAL



Does a **framework** depend on its plugins? No – because it does not crash if a plugin is missed. Does a plugin depend on its framework? No – because it is like a module. Hence: **no derivative work**.

R^OS^L!

The **pipe** case:
Hopefully for a broad consensus!



Does a program using inter process communication depend on its IO-partners? Definitely not. Hence: **no derivative work.**

ROSL!

Yes, some cases evoke a **derivative work** :

A condensed answer as offer for a consensus.

Derivative Work establishing cases

- Enlarging Case
- Reducing Case
- Replacing Case
- Snippet Case
- Refactoring Case
- Module Case
- Library Case
- Import Case

No Dependency establishing cases

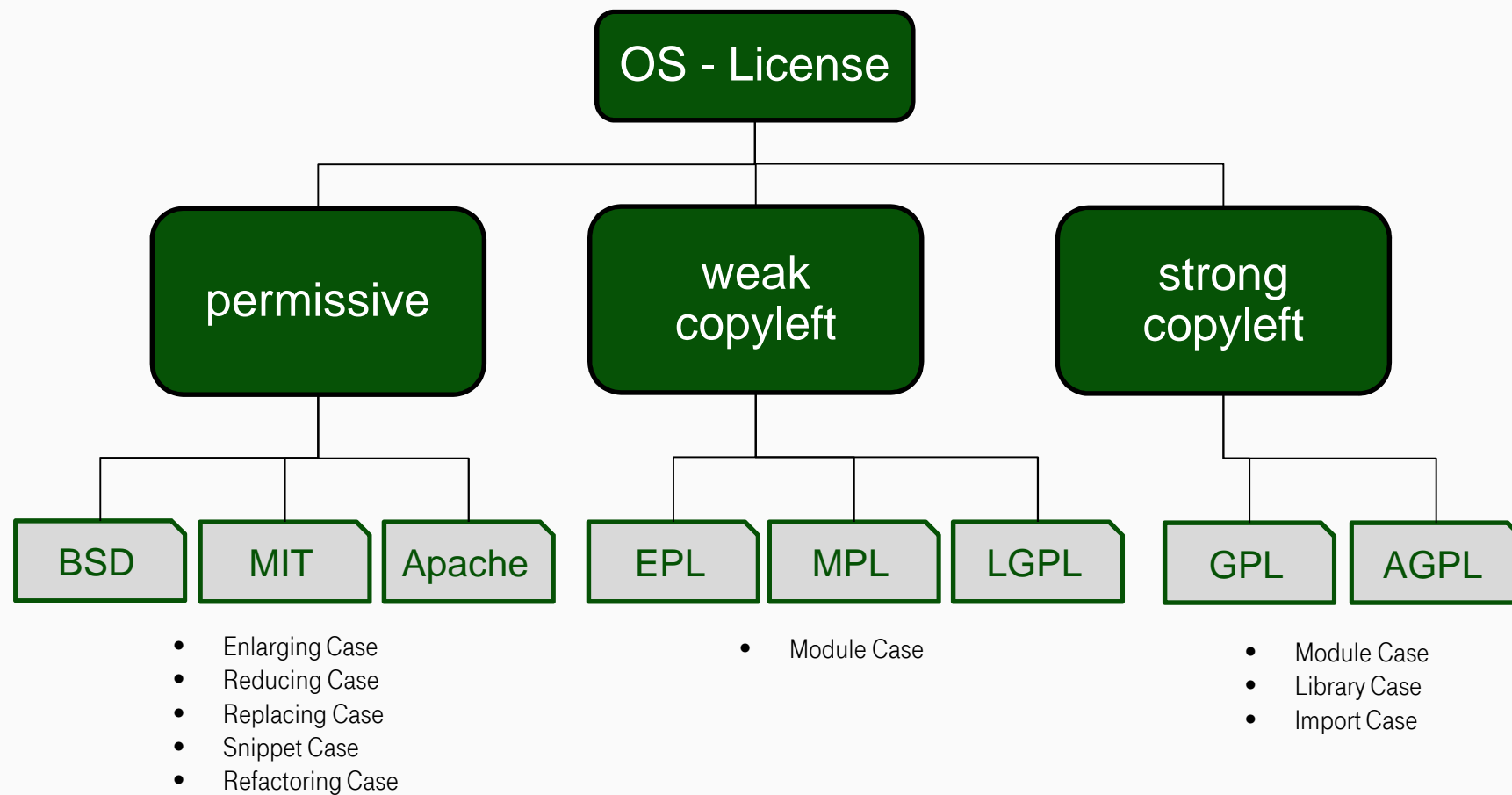
- Plugin Case
- Pipe Case

Purpose ?

R^{OS}L!

License specific consequences

5 sufficient questions!



R^{OS}L!

License specific consequences

2 tools to get the license and use case specific task list



a form of

5

questions

is sufficient



<http://dtag-dbu.github.com/oslic>

<http://dtag-dbu.github.com/oscad>

*<http://oscad.fodina.de/>

Open
Source
Compliance **Ad**visor

Open
Source
License **C**ompendium

ROSL!

Be **invited**, be **welcome**!

We want to collaborate.

Many **thanks** for your time and **attention**!

k.reincke@telekom.de

<http://dtag-dbu.github.com/oscad/>

<http://www.oslic.org/>

