

The GPLv3 and Compatibility Issues

European Open source Lawyers Event

Paris, Capitale du Libre
24 September 2008

Philippe LAURENT

Researcher at the CRID
(Research Centre on IT and Law)
University of Namur – Belgium

Lawyer at the Brussels Bar – MVVP
(Marx, Van Ranst, Vermeersch & Partners)



“Compatibility” : semantic issue

■ **Compatibility definitions**

■ **Common sense**

- capable of living together harmoniously
- that can function or be used together without change or alteration
- [comput.] designating or of
 - a) computer components, software, etc. that can be used with a specified computer or computer system: often in hyphenated compounds
 - b) computers or computer systems that can use the same components, software, etc.

○ **Idea of RECIPROCITY**

- no predominance, no alteration of the nature of the compatible elements

■ **“Compatibility” of FLOSS licences**

- **No “Official”/”legal” definition** of what compatibility is
- **General idea :**

“2 software distributed under 2 different licences can be merged”

- **BUT legal effects may be different from one case to another**

- In a broad sense “Compatibility” of a FLOSS license could be described as :
 - “the characteristic of two (or more) licences according to which the codes distributed under these licences may be put together in order to create a bigger distributable software”
 - + : * this definition takes into account many of the combination possibilities
 - ⇒ “BSD and GPL are compatible”
 - ⇒ “BSD and Apache are compatible”
 - ⇒ “LGPL and Mozilla *may, in some cases, be compatible*”
 - : * this definition does not take into account the results of the combination
 - * It creates a false idea of reciprocity, which could lead to legal mistakes
 - * licence compatibility = combinability of code (no added value)

- In a narrow sense, “Compatibility” of a FLOSS license is commonly understood as

“the characteristic of a licence according to which the code distributed under this licence may be integrated in a bigger software *that will be distributed under another licence*”

Ex.: cfr. the use of the terms “GPL-Compatible” on the FSF website

Adopted definition

⇒ **COMPATIBILITY of FLOSS Licences = usually a ONE WAY ROAD**
(and should therefore not be considered as a reciprocal relationship)

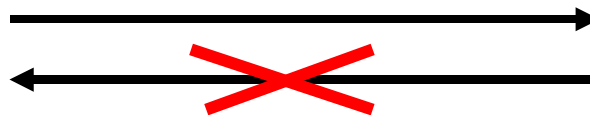
BSD is “GPL-compatible”

BSD code **can** be added in a software distributed under GPL

BUT

GPL is NOT “BSD-compatible”

GPL code **cannot** be added in a software distributed under BSD



- + : * This way of using the term “compatibility” conveys more information : it describes the result of the merging of the code.

BSD code + GPL code = GPL code

- * It does not imply reciprocity

- : * this use of the term “compatibility” calls for more rigour

⇒ One must be aware that the information conveyed by the sentence “is XYZ-compatible” is not complete, and that there could be different possible situations.

- **BSD is MIT compatible** / **MIT is BSD compatible**
- **BSD is GPL compatible** / **GPL is not BSD compatible**
- **GPL is not Mozilla compatible** / **Mozilla is not GPL compatible**

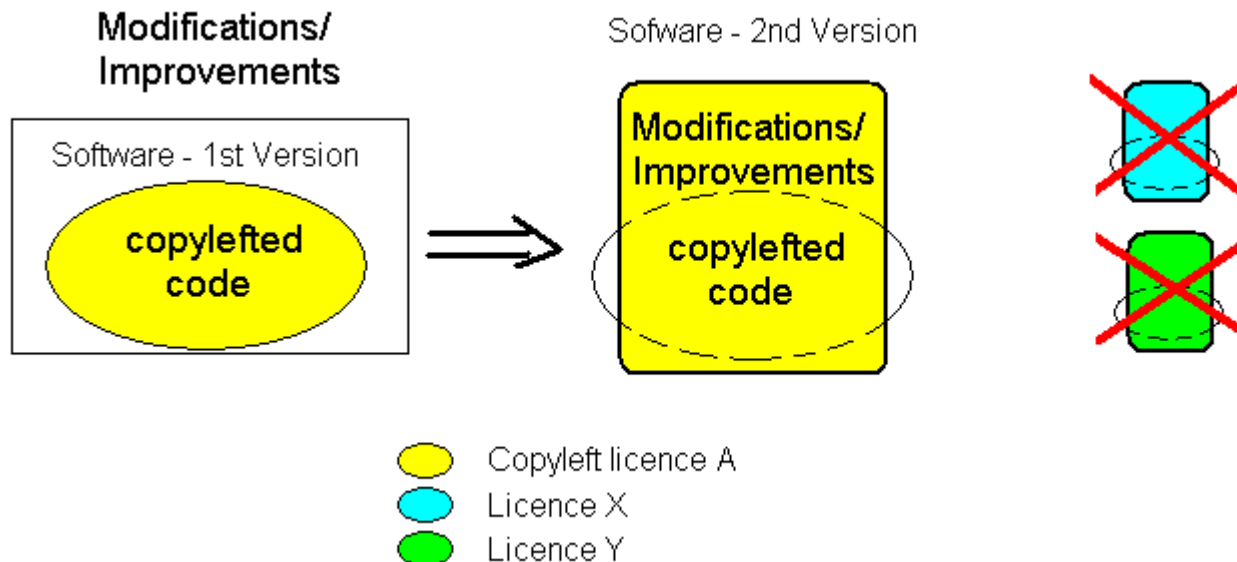
Incompatibility

- **Incompatibility** is due to contradictory obligations provided in the different licences under which two codes to be merged are distributed.
- It can be due to clauses of a multitude of kinds...
 - ⇒ Basically : anything that puts the licensee in a position where he could not fulfill all his/her obligations under one of the two licences.
Ex.: compatibility problems between Apache v2.0 and GPLv2 : due to indemnification and patent termination clauses.
- ... but in the great majority of cases, it is due to the copyleft effect of some copyleft licences.

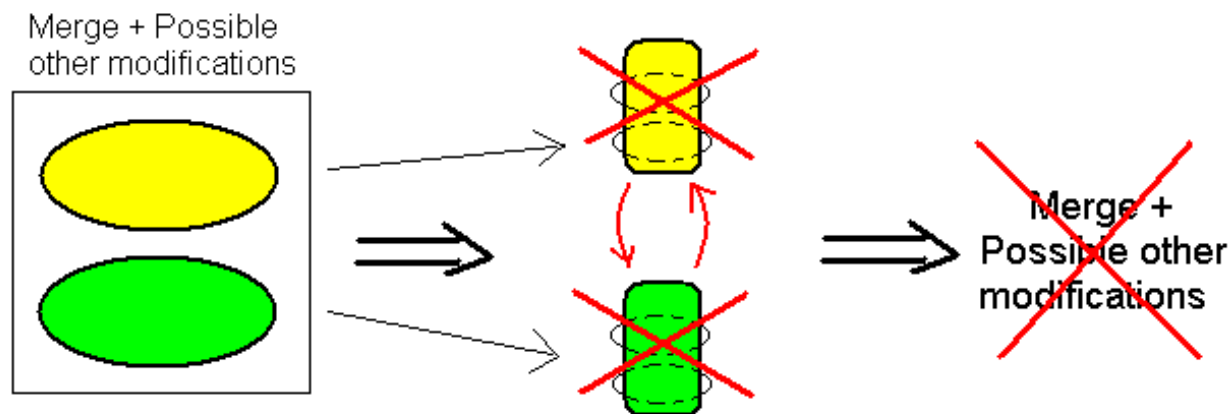
Copyleft is the main source of compatibility problems

- This copyleft effect is reached by introducing a **copyleft clause** in the FLOSS licence, which, in general, reads more or less as follows:

“ You are free to modify or merge the software with another one, but if you redistribute the modified or merged version of the software, this redistribution must be done under the same licence”



- Merging some code with copyleft licensed code usually means that the **copyleft licence is predominant**
 - ⇒ The result of the merger must be licensed under this copyleft licence
 - ⇒ The “compatible licence” steps aside for the copyleft licence
- **Copyleft Incompatibility** is due to the fact that each one of the copyleft licences, under which the codes **to be merged** are licensed, oblige the licensee to redistribute the result under itself :



Two main types of Compatibility issues

When drafting a FLOSS licence “XYZ”, 2 main questions arise :

1. Will there be “XYZ-compatible” licences? [**upstream compatibility**]

⇒ This depends a lot (but not exclusively) on how the upstream licence is drafted

2. Will XYZ be compatible with other licences?

Will XYZ be “ABC-compatible”? [**downstream compatibility**]

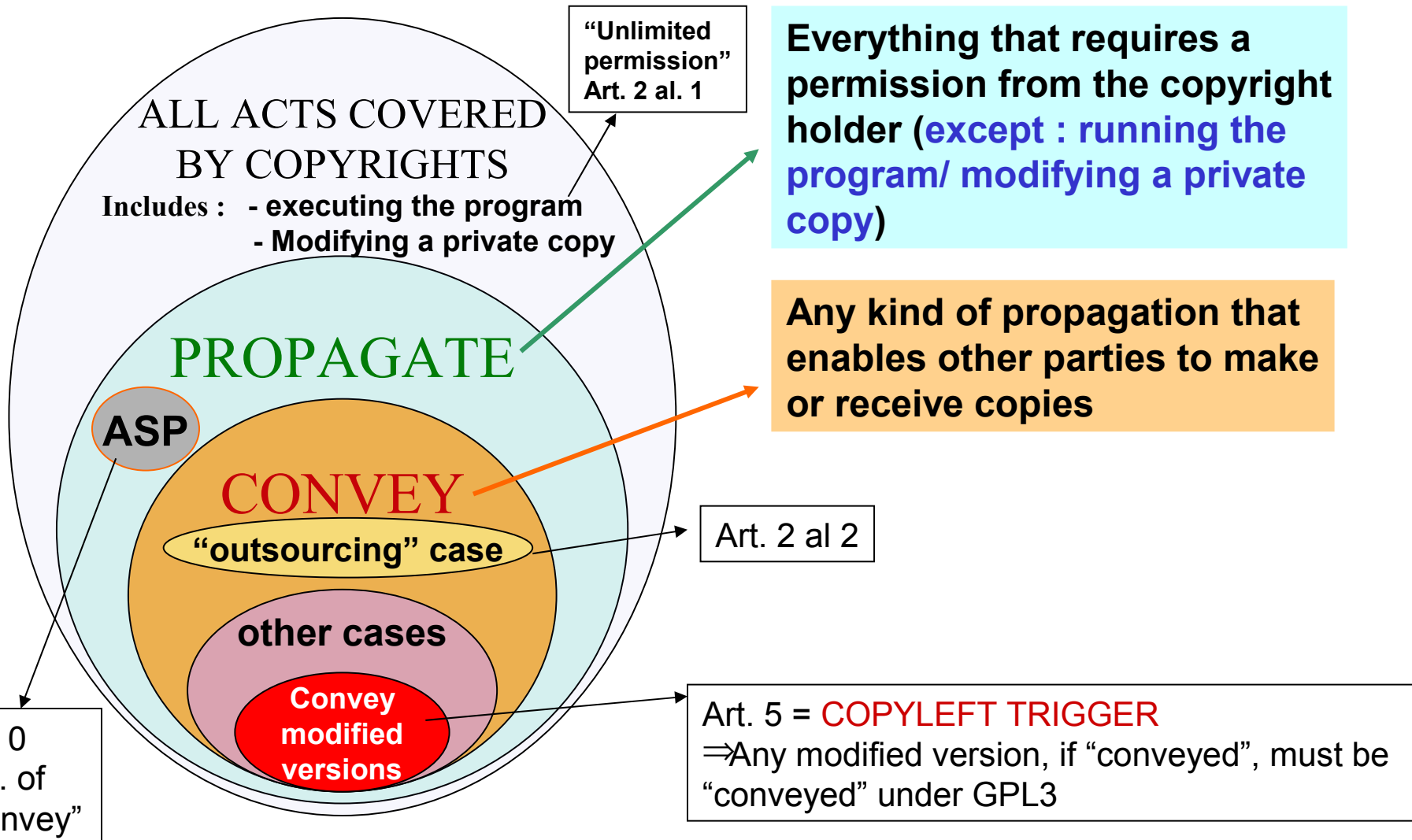
⇒ This depends mainly on how the XYZ licence is drafted

=> **GPLv3 provides in some extent for a better upstream compatibility**

=> **GPLv3 deals with some specific downstream compatibility issues**

- **GPLv3 is a copyleft licence :**
 - ⇒ Any modified version, if “conveyed”, must be “conveyed” under GPL3
 - **CONVEY** = 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.**
 - **PROPAGATE** a work = to do anything with it that, without permission, would make you directly or secondarily **liable for infringement under applicable copyright law**, **except executing it on a computer or modifying a private copy**. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

GPLv3 : Copyleft trigger



GPLv3 : compatibility related clauses

▪ Art. 5 b) & c) :

“ ... notice stating *it is released under this License and *any conditions added under section 7...”

“... This License will therefore apply, along with any applicable section 7 additional terms”

▪ Section 7 (+section 10) :

- | | |
|--|--------------|
| ▪ Additional permissions (AP) | OK |
| ▪ Additional non-permissive terms (ANPT) | Limited list |
| ▪ Further restrictions (FR) | NO |

Additional Permissions (art. 7 §1 & §2)

- **Art. 7 § 1 & § 2: Additional Permissions**
= additional **exceptions** to one or more of the GPLv3 **conditions**.
 - ⇒ Additional permissions **may only be added** to GPLv3 by the author of additional original code => **These additional permissions will only apply to this author's material**
 - *“You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission”.*
 - ⇒ Additional permissions **may be removed** by any “conveyor”

Additional Permissions (art. 7 §1 & §2)

- **if applicable to the entire Program** => = “shall be treated as though they were included in *this Licence*”
⇒ ? Do they therefore loose their “Status of - Removable - additional permission”?

Art. 0 : “**This License**” refers to version 3 of the GNU GPL

Art. 7 : **Additional permissions** that are applicable to the entire Program shall be treated as though they were included in *this License*

Art. 7 : “**Additional Permissions**” are terms that supplement the terms of *this licence*

⇒ Self-reference problem

**GPLv3
+ AP**

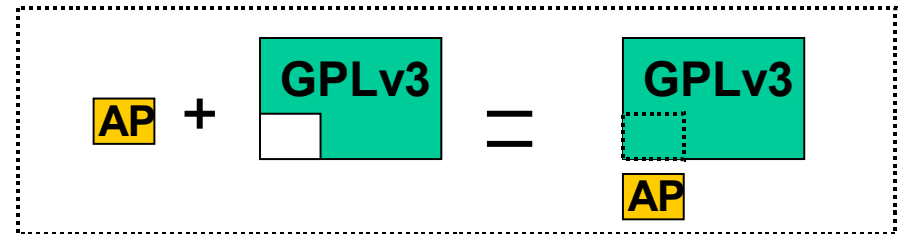
- ⇒ **BUT Art.7, §2** : “*When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it*”.

=> **GENERAL clause ... applicable to all AP cases (I suppose)**

Additional Permissions

(art. 7 §1 & §2)

- if applies only to part of the Program
 - that part may be used separately under those permissions
 - but the entire Program remains governed by GPLv3 without regard to the additional permissions.



- First conclusions :

- ⇒ In any case everything can be conveyed under a “strict” GPLv3
- ⇒ All these modified versions of the GPLv3 are “at least” GPLv3 compatible.



⇒ CATCH ALL

*“Do you want to use a more permissive licence?
 => Add some additional permissions to GPLv3 instead
 ... it will be surely GPLv3-compatible !”*

Non-permissive additional terms (Art. 7 §3 et seq. & art. 10)

Art. 7§3:

For **material you add to a covered work**, you may
(if authorized by the copyright holders of that material)

UPSTREAM
COMPATIBILITY

supplement the terms of this License with terms:

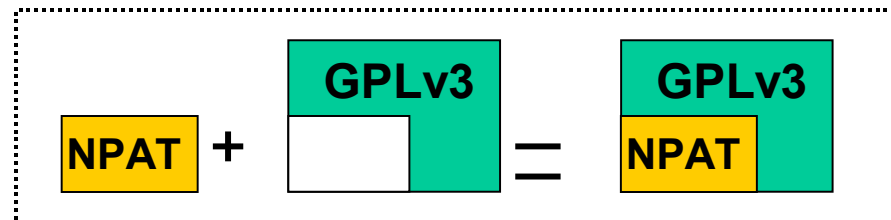
- **Disclaiming warranty** or **limiting liability** differently from the terms of sections 15 and 16 of this License; or
- Requiring preservation of specified reasonable **legal notices** or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- **Prohibiting misrepresentation** of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- Limiting the use for **publicity purposes** of names of licensors or authors of the material; or
- **Declining to grant rights under trademark law** for use of some trade names, trademarks, or service marks; or
- Requiring **indemnification** of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

SUCH CLAUSES ARE PRESENT IN APACHEv2.0 FOR EXAMPLE

Non-permissive additional terms (Art. 7 §3 *et seq.* & art. 10)

■ These authorized NPAT

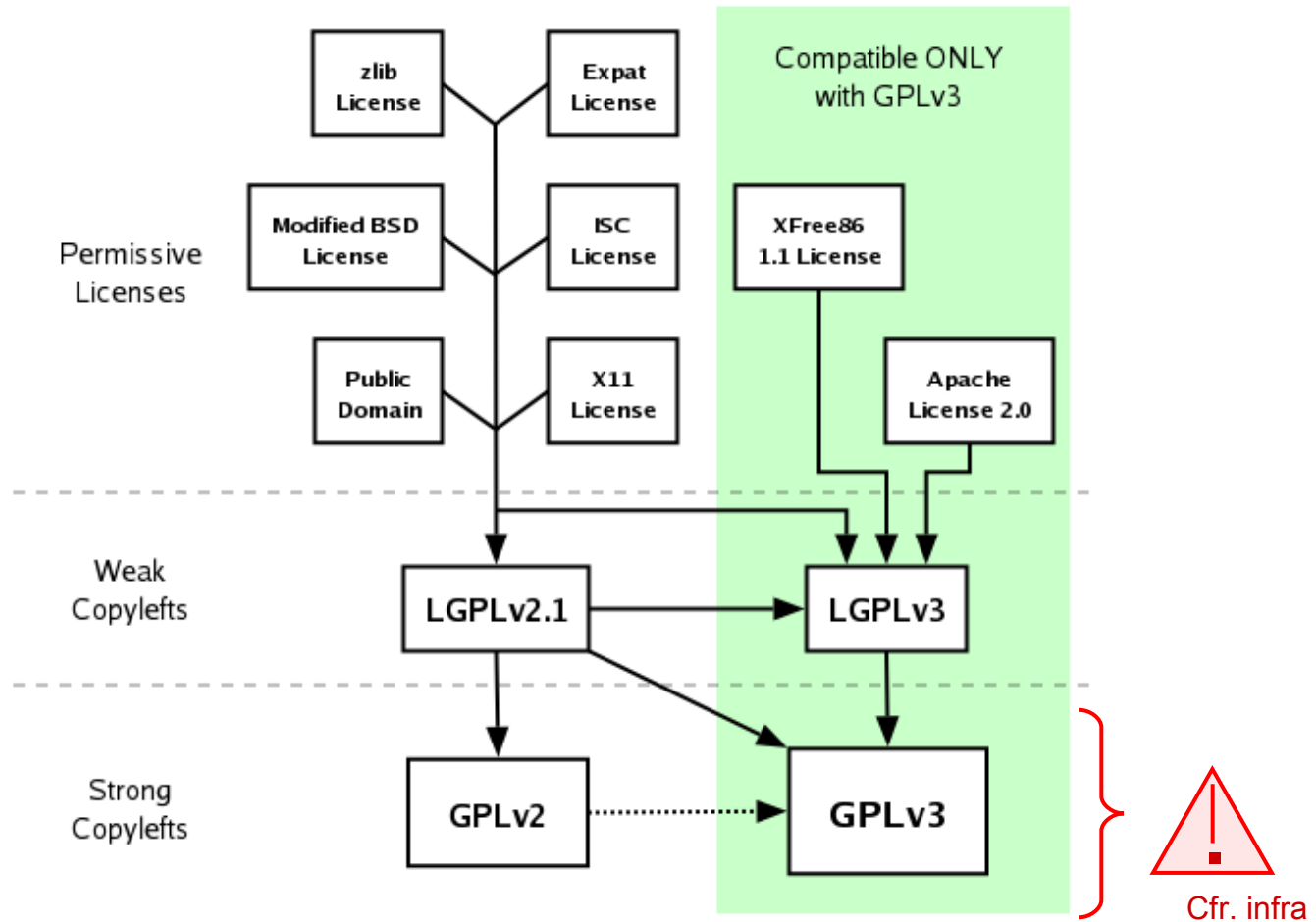
- apply only to material added by licensee to a covered work
- not removable



- All other non-permissive additional terms are considered **“further restrictions”** within the meaning of section 10.
 - ⇒ **Art. 10** : **You may not impose any further restrictions** on the exercise of the rights granted or affirmed under this License.
- If a license document contains a further restriction **but permits relicensing or conveying under this License**, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.
 - ⇒ Licences with a specific GPL-compatibility clause :
Cecill , next v. of EUPL?

**UPSTREAM
COMPATIBILITY**

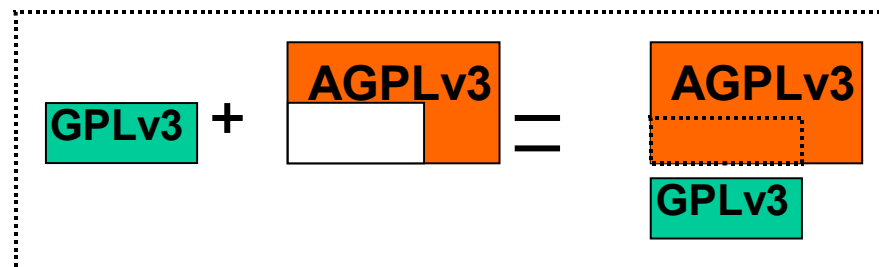
GPLv3 as a CATCH ALL



B. Smith, *A quick guide to GPLv3*, FSF Inc., 2007, p.4
<http://www.fsf.org/licensing/licenses/quick-guide-gplv3.html>

Art. 13 : Use with the GNU Affero General Public License.

- Notwithstanding any other provision of this License, you have permission **to link or combine** any covered work with a work licensed under version 3 of the GNU Affero General Public License **into a single combined work, and to convey the resulting work.**
- The terms of this License will continue to apply to the part which is the covered work, but **the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.**



**DOWN STREAM
COMPATIBILITY**

Art. 14 : Revised Versions of the GPLv3 (idem as Art. 9 of GPLv2)

- Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. **If the Program does not specify a version number** of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

DOWN STREAM
COMPATIBILITY

UPSTREAM
COMPATIBILITY

- **BUT : ... (IMPLICIT) ...** If the program specifies the applicable numbered version of GPL **only : no option**

I want to
copy code
under:

I want to release a project under:

GPL 2/GPL 3 (& LGPL's) Compatibility MATRIX

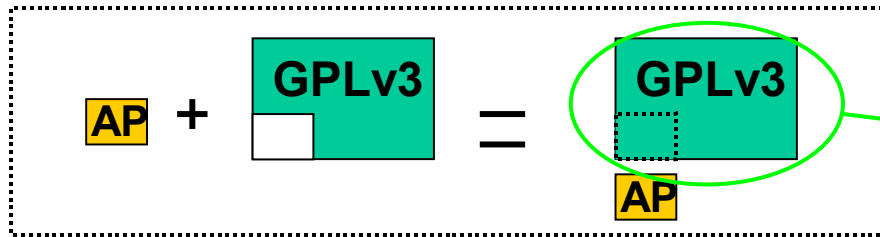
		GPLv2 only	GPLv2 or later	GPLv3 or later	LGPLv2.1 only	LGPLv2.1 or later	LGPLv3 or later
I want to copy code under:	GPLv2 only	OK	OK [2]	NO	OK if you convert to GPLv2 [7]	OK if you convert to GPLv2 [7][2]	NO
	GPLv2 or later	OK [1]	OK	OK	OK if you convert to GPL [7]	OK if you convert to GPL [7]	OK if you convert to GPLv3 [8]
	GPLv3	NO	OK if you upgrade to GPLv3 [3]	OK	OK if you convert to GPLv3 [7]	OK if you convert to GPLv3 [7][3]	OK if you convert to GPLv3 [8]
	LGPLv2.1 only	OK if you convert to GPLv2 [7]	OK if you convert to GPL [7][2]	OK if you convert to GPLv3 [7]	OK	OK [6]	OK if you convert to GPLv3 [7][8]
	LGPLv2.1 or later	OK if you convert to GPLv2 [7][1]	OK if you convert to GPL [7]	OK if you convert to GPLv3 [7]	OK [5]	OK	OK
	LGPLv3	NO	OK if you upgrade and convert to GPLv3 [8][3]	OK if you convert to GPLv3 [8]	OK if you convert to GPLv3 [8]	OK if you upgrade to LGPLv3 [4]	OK
I want to use a library under:	GPLv2 only	OK	OK [2]	NO	OK if you convert to GPLv2 [7]	OK if you convert to GPLv2 [7][2]	NO
	GPLv2 or later	OK [1]	OK	OK	OK if you convert to GPL [7][1]	OK if you convert to GPL [7]	OK if you convert to GPLv3 [8]
	GPLv3	NO	OK if you upgrade to GPLv3 [3]	OK	OK if you convert to GPLv3 [7]	OK if you convert to GPLv3 [7][3]	OK if you convert to GPLv3 [8]
	LGPLv2.1 only	OK	OK	OK	OK	OK	OK
	LGPLv2.1 or later	OK	OK	OK	OK	OK	OK
	LGPLv3	NO	OK	OK	OK	OK	OK

▪ FOOTNOTES

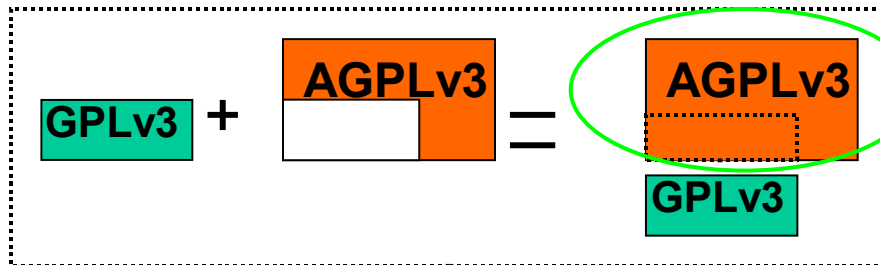
- **1: You must follow the terms of GPLv2 when incorporating the code in this case. You cannot take advantage of terms in later versions of the GPL.**
- **2: If you do this, as long as the project contains the code released under GPLv2 only, you will not be able to upgrade the project's license to GPLv3 or later.**
- **3: If you have the ability to release the project under GPLv2 or any later version, you can choose to release it under GPLv3 or any later version—and once you do that, you'll be able to incorporate the code released under GPLv3.**
- **4: If you have the ability to release the project under LGPLv2.1 or any later version, you can choose to release it under LGPLv3 or any later version—and once you do that, you'll be able to incorporate the code released under LGPLv3.**
- **5: You must follow the terms of LGPLv2.1 when incorporating the code in this case. You cannot take advantage of terms in later versions of the LGPL.**
- **6: If you do this, as long as the project contains the code released under LGPLv2.1 only, you will not be able to upgrade the project's license to LGPLv3 or later.**
- **7: LGPLv2.1 gives you permission to relicense the code under any version of the GPL since GPLv2. If you can switch the LGPLed code in this case to using an appropriate version of the GPL instead (as noted in the table), you can make this combination.**
- **8: LGPLv3 gives you permission to relicense the code under GPLv3. In these cases, you can combine the code if you convert the LGPLed code to GPLv3.**

Last considerations : Compatibility clauses and ...

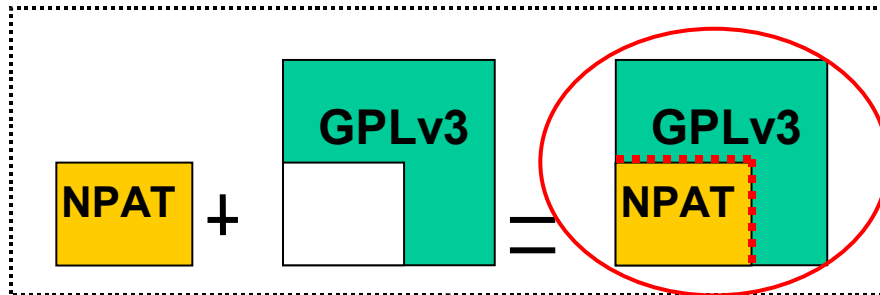
the notion of derivative work (?)



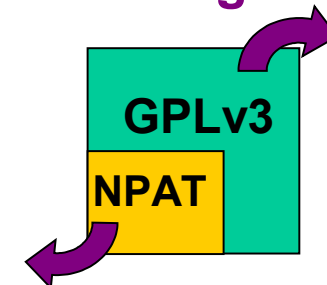
Derivative Works?
OK



???
Artificial
“Separation” ?



& quid with dynamic linking ?

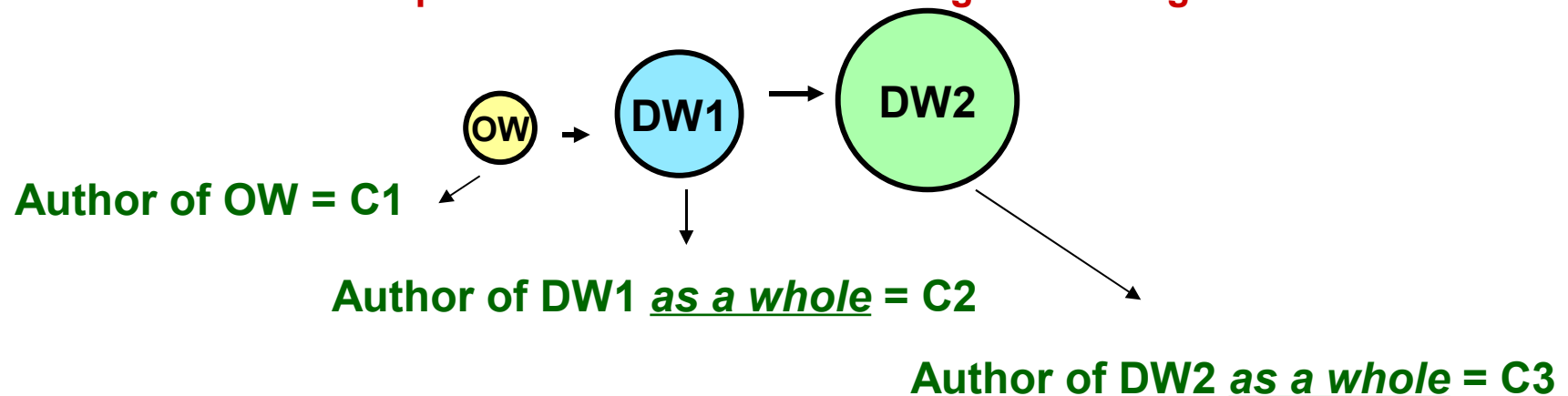


Is this “compatible” with the (EU) Notion of derivative work * ?
 Can an author decide what is part of a derivative Work and what is not? (NB : GPLv3 forbids sub-licencing)

*NB:

Under Belgian & French Law (Doctrine)

- The author of a Derivative Work (DW) based on a Original Work (OW) must have an authorization to do so from the author of the OW = **Licence to create a DW**
 - The author of a DW is **author of this DW as a whole**...
 - ... but he is still bound to the author by his licence to create a DW
- => He still must respect this licence when dealing with his rights on the DW...**



Thank you for your attention !

Philippe LAURENT

Researcher at the CRID

University of Namur



Email :

philippe.laurent@fundp.ac.be

<http://www.crid.be>

Attorney at Law

Lawyer at the Brussels Bar

MVVP

Email :

philippe.laurent@mvvp.be

<http://www.mvvp.be>



These slides are distributed under the Belgian Creative Commons
License : Attribution – Non Commercial – No Modification

<http://creativecommons.org/licenses/by-nc-nd/2.0/be/legalcode.fr>