

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# Licensing in a software project

Rémi Boulle mail@remiboulle.fr

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

## Main goal

Acquire and strengthen your expertise on free software licenses

- History and context
- Different categories of licenses
- Diffusivity and compliance rules
- Auditing your project
- Free software economics

Rémi Boulle  
mail@remiboulle.fr

## Context

TP1

TP3 et TP4

## Context

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

- GNU project in 1984 by Richard Stallman (RMS)
- GPLv1 : 25th of february 1989
- la technique est un moyen pour atteindre un but social
- 1991 : Minix kernel (*just a hobby, won't be big and professional like gnu* by Linus Torvalds (LT))
- 1995 : Red Hat (Nasdaq in 1999), Apache license
- 1998 : release of Netscape source-code (fight back IE. Free the lizard, mozilla)
- 1998 : "Free software" versus "Open Source Software" (OSI). Rebranding the free software movement to emphasize the business potential ?



**Linus Torvalds**

GNU GENERAL PUBLIC LICENSE - Version 3,  
29 June 2007 - Copyright © 2007 Free  
Software Foundation, Inc. <<http://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. As 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.

```
boot __init__();
page_address_init();
pr_notice("1.1. linux_banner");
setup_arch(command_line);
mm_init_callbacks(&init_mm);
setup_command_line(command_line);
setup_nr_cpu_ids();
setup_per_cpu_areas();
boot_cpu_state_init();
cpu_features_boot_cpu();
build_all_zonelists(NULL, NULL);
page_init();
pr_notice("kernel command line: '%s'",
        parse_early_param());
after_forked = parse_args("Booting kernel",
        static_command_line, &start_param,
        &stop_param, &start_panic,
        &__end, &unknown_bootoption);
if (!IS_ERR(&start_panic))
        parse_args("Setting init args", after_forked,
        NULL, &init_arg);
just_label_init();
setup_io_huff();
pidmax_init();
vfs_caching_init_early();
soft_main_table();
trap_init();
mm_init();
sched_init();
preempt_disable();
if (WARN(!irqs_disabled(),
        "Interrupts were disabled 'early' but "
        "local irq disabled()"));
ide_init_cache(&__init());
trace_init(); /* enables 'tracking' init */
early_irq_init(&__init_irq); task_init();
cpu_init_data(); init_timers();
hrtimers_init(); softirq_init();
clockchasing_init(); time_init();
sched_clock_postinit();
printk_init(); perf_event_init();
profile_init(); call_function_init();
WARN(!irqs_disabled(), "Interrupts were "
        "early boot_irqs_disabled = false; local "
        "mmu cache_initLate()");
hrtimer_init_freeable();
async_synchronize_full();
free_initmem();
mark_readonly();
current_state = SYSTEM_RUNNING;
```

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

## International

Free Software Fondation : <https://www.fsf.org/>

Open Source Initiative : <https://opensource.org/>

Linux Fondation : <https://www.linuxfoundation.org/>

Debian, python, Ubuntu, KDE... communities



## Context

## International

Free Software Fondation : <https://www.fsf.org/>

Open Source Initiative : <https://opensource.org/>

Linux Fondation : <https://www.linuxfoundation.org/>

Debian, python, Ubuntu, KDE... communities

## Europe

France : CNLL : <http://cnll.fr/>, April :

`https://april.org/`, Framasoft :

<https://framasoftware.org/>

We are always the ideologue of someone... Outdated debate ?

Rémi Boulle  
mail@remiboulle.fr

Context

**Vocabulary**

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# Vocabulary

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

- **proprietary software** but all software have authors.
- **Copyleft** : *we give rights*, all modified and extended versions of the program must be free as well, under the same terms. You cannot add restrictions to deny other people freedoms.
- **Copyright** : legal term describing rights given to creators for their works. Under the Berne Convention, everything written is automatically copyrighted from whenever it is put in fixed form
- Copyleft is not opposed to Copyright but a clever hack

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

**Copyright**

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# Copyright

With the creation authors obtain intellectual property rights.

- Economic right : right to exploit the work
  - permit or prohibit the fixation and reproduction of his or her work
  - translation, arrangement, adaptation and alteration of the work
  - distribution of the original or copies
  - Related rights
- Moral right
  - right to protect their personal connection with the work

Copyright is not opposed to copyright

Free and proprietary software *copyright*.

EU directive

Only the expression of a computer program is protected and that ideas and principles which underlie any element of a program, including those which underlie its interfaces, are not protected by copyright.

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

**Proprietary**

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# Proprietary

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

General principle : limitations against use, distribution and modification. Used by end-users under predefined conditions.

## Definition : proprietary software

Software who limits one or more of the 4 freedoms of free software.



General principle : limitations against use, distribution and modification. Used by end-users under predefined conditions.

## Definition : proprietary software

Software who limits one or more of the 4 freedoms of free software.

Michel Rocard, 2002, patent battle : 648 no, 14 yes, 18 abs

*Creation, freedom, innovation were on the side of free software. The pursuit of profit, and above all the rent, the desire to restrict competition, and to restrain the external innovations, were on the side of big industry.*

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

*If you can not enter through the door, go in through the window !*

Who is in charge of governing which free software project ?  
Just asking...

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# Free Software

## General principle

*Free as in freedom, not free beer.* Source code is free.

Their main difference : modalities of redistribution

- 1 redistribution but code must stay free : user's should receive a copy of the code under the same terms
- 2 redistribution but code must stay free : user's are not guaranteed to receive a copy

## Definition : free software

A program is free software if the **program's users** have the four essential freedoms.



Rémi Boulle  
mail@remibouille.fr

## Free Software

TP1

TP2

- study without any obstacles (technicals, rights...)
- includes the freedom to use your changed version in place of the originalfreedom d'étude
- Not obliged to give source code with binaries (by pragmatsim)
- Source code should be made available for free or at a low cost
- Source code available for at least 3 years (GPLv3)

## Principe

Again, it is the user's purpose that matters, not the developer's one

Rémi Boulle  
mail@remibouille.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

- absolute freedom. Only restrictions when distribution.
- Preserve copyright ownership (Berne convention)
- Identify contributions

## Principe

Always, it is the user's purpose that matters

# freedom 3 : freedom to redistribute

Anyone, anywhere (export control, trade sanctions...)

- Conditions of redistribution : main difference between all free software licenses :
  - **Strong Copyleft**
  - **Weak Copyleft**
  - **Copyfree**
- Always identify contributors (Berne convention)
- Preserve original authors' reputation

## Principle

User's purpose again and mostly necessary condition for the three others freedoms.

## When does license apply ?

## Redistribution triggers !

Exception : APGL



Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# TP1



Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# TP2

# Free Software and corporations ?

Rémi Boulle  
mail@remibouille.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

- Is there a particular governance with free software in the corporations you know or you work at ?
- Do you use free licensed librairies for your company projects ?

We will share all replies onto the given pad.

Rémi Boulle  
mail@remibouille.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

**Classification**

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# Classification

## Main purpose

Prevent to extract value from free software without giving back own contributions.

### Definition (Pellegrini, Canevet)

**Copyleft** free software license where every contribution using a code under this licensed is subjected to the same terms.

Also know as ***strong copleft***.

Less relevant terms : "viral licenses", "contaminants" (terms to avoid ideology inside...).

Examples : GNU GPL, CeCILL(A)....

## Strategically **expansionnary**.

## Main purpose

Prevent any privatization of the work without applying to derivative works.

## Definition (Pellegrini, Canevet)

**Copyleft** free software license not preventing the use of works placed under their terms with others works under other under licenses (including proprietary ones)

Also known as ***weak copyleft***, sustainable licenses. Mostly for libs.

## Exemple : GNU LGPL

## Strategically **defensives**.

## Main purpose

## Absolute freedom !

## Definition (Pellegrini, Canevet)

Free software license without copyleft which authorize distribution of software without its source code

Also known as **copyfree**

A less rigorous term : permissive licences

Examples : BSD (BSD 4 clauses, BSD 3, BSD 2), MIT, Apache, CeCILL-B...

## Strategically **proselytes**.



Rémi Boulle  
mail@remibouille.fr

TP1

TP2

## Scope

TP3 et TP4

# Scope



Spread is not at all absolute ! (Linux is GPLv2 only...)

## General purpose of diffusive licenses

- Counterparty
  - Coexistence
- 
- Ascending diffusivity if derivate work based upon original code. Libs and system call (libc is LGPL, Linux is GPLv2 only) ?
  - Descending diffusivity : drivers = derivative work ? (use of wrappers)

When is it a derivative work ? Need to study substitutability (link is generic or specific ?)

## Scope

Scope of those licenses is defined by substitutability...

Three cases :

- work using... : If work could be considered as only using the covered work, no obligations (except statically linked, read license)
- combined work (reverse-ingenierie and not combined version)
- internal modification of the covered work

Preserve the functional scope of the covered work to avoid subversion... (cf MPL, quite easy to subvert)

The scope is defined fonctionnaly

## Scope

The scope of the license is the interface.

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

## Main purpose of evanescent licenses

No obligation to distribute source code with object code.

## Scope

The scope of the license is the file.

Again, read the license (SMOG index, see further)

Rémi Boulle  
mail@remibouille.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

**compatibility**

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

# compatibility







Your code = some code + A + B. Can you distribute it ?

Code A \ Code B	Diffusive	Persistent	Evanescent
Diffusive	<b>Impossible</b> except compatibility exception	<b>Impossible</b> except compatibility exception	<b>Possible</b> distribution under same terms of A
Persistent	<b>Impossible</b> except compatibility exception	<b>Possible.</b> Distribution OK without covering it with A and B licenses if each module is distributed under his own licence	<b>Possible.</b> Distribution OK without A license to cover your code and B and if A is distributed under his license.
Evanescent	<b>Possible.</b> Distribution under terms of B	<b>Possible.</b> Distribution OK without covering it and A with B license if B is distributed under his license	<b>Possible.</b> No diffusivity.

Rémi Boulle  
mail@remibouille.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

**TP3 et TP4**

Good practice  
and auditing

Future ?

Bibliography  
and credits

## TP3 et TP4

# Read and analyse licences. At last !

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

**TP3 et TP4**

Good practice  
and auditing

Future ?

Bibliography  
and credits

*The SMOG grade is a measure of readability that estimates the years of education needed to understand a piece of writing. SMOG is an acronym for Simple Measure of Gobbledygook. According to Alexios Zavras, average SMOG grade is 17. Some over 25...*

By groups of 2/3 students and into the given document (GPLv3) read and comment (1h max) :

- no paraphrasing. Your little brother/sister would have to understand
- explain what are : software patents, DRM, Anti-Circumvention Law...
- explain why this license take this in charge
- try to compare with others free software licenses (MIT, Apache, LGPL, AGPL...)

Each group will have to teach randomly chosen parts of the GPLv3 to the whole group.



# 3 questions

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

**TP3 et TP4**

Good practice  
and auditing

Future ?

Bibliography  
and credits

By groups of 2/3 students :

- Draw by lot 3 questions.
- TEach group will have to teach his answers to the whole groupe in 10 minutes max.

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

**Good practice  
and auditing**

Future ?

Bibliography  
and credits

# Good practice and auditing

- DO NOT remove existing copyrights mentions !
- Manage diffusivity
- Respect formalism of each license
- Think license at the very beginning of the project and all along it !
- You will need to only lawyers... But also skillful profiles (not very common)

If not :

- emergency rewrite of the covered code (worst case scenario)
- delays in project delivery
- license violation (court, image of the company, clients loss of trust...hell)

# Auditing, what for ?

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

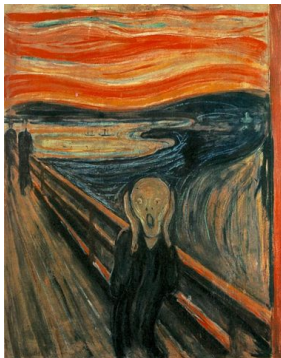
Good practice  
and auditing

Future ?

Bibliography  
and credits

Selon *Black Duck Software* :

- 30% of companies code source are free software
- 98% of companies don't know that...



Marketing of *fear* but :

Double constraint :

- legally : compatibilities  
(proprietary licences with free ones, free between them...)
- technically : security breaches  
of any external component  
(HeartBleed in OpenSSL, ...)

**Security breaches**

Luckily, it was free software....



# Auditing, how ?

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

**Good practice  
and auditing**

Future ?

Bibliography  
and credits

grep and essentially three solutions...

Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

founded by former Microsoft managers.

- Open Source Application Security
- Open Source Compliance and Management

<https://www.blackducksoftware.com/>



49/58

Rémi Boulle  
mail@remibouille.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

Bibliography  
and credits

*ScanCode is a suite of utilities used to scan a codebase for license, copyright, and other interesting information that can be discovered in files.*

*A typical software project often reuses hundreds of third-party components. License and origin information is often scattered and not easy to find: ScanCode discovers this data for you.*

<https://github.com/nexB/scancode-toolkit>

ffmpeg
compat
doc
libavcodec
libavdevice
libavfilter
tests
x86
af_volume.asm
af_volume_init.c
avf_showcqt.asm
avf_showcqt_init.c
colospacecsp.asm
colospacecsp_init.c
Makefile
vf_blend.asm
vf_blend_init.c
vf_bwdif.asm
vf_bwdif_init.c
vf_eq.c
vf_fspp.asm
vf_fspp_init.c
vf_gradfun.asm
vf_gradfun_init.c
vf_hqdn3d.asm
vf_hqdn3d_init.c
vf_idet.asm
vf_idet_init.c
vf_interlace.asm
vf_interlace_init.c
vf_maskedmerae.asm

## License Summary

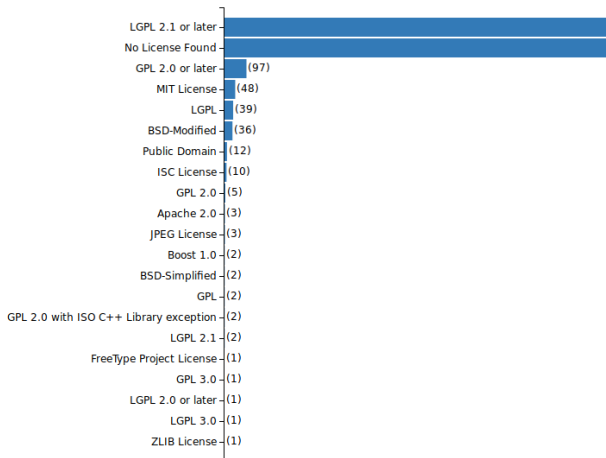
## Copyright Summary

## License &amp; Copyright Details

## File Details

## Packages

Total Files Scanned: 6471



Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

**Future ?**

Bibliography  
and credits

Future ?





Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

**Bibliography  
and credits**

## Bibliography and credits

Bibliography (in french) :

- "Droit des logiciels, logiciels propriétaires, logiciels libres", F.Pellegrini et S.Canevet, PUF 2013.
- "Option libre. Du bon usage des licences libres", B.Jean, Framabook 2012.
- "Histoire et cultures du Libre. Des logiciels partagés aux licences partagés", Collectif, Framabook 2013.

### Websites :

- <http://gplv3.fsf.org/>
- <https://copyleft.org/guide/>
- <http://april.org/en>





Rémi Boulle  
mail@remiboulle.fr

Context

Vocabulary

Copyright

Proprietary

Free Software

TP1

TP2

Classification

Scope

compatibility

TP3 et TP4

Good practice  
and auditing

Future ?

**Bibliography  
and credits**

Document under GFDL1.3+, CC-BY-SA+, LAL+.