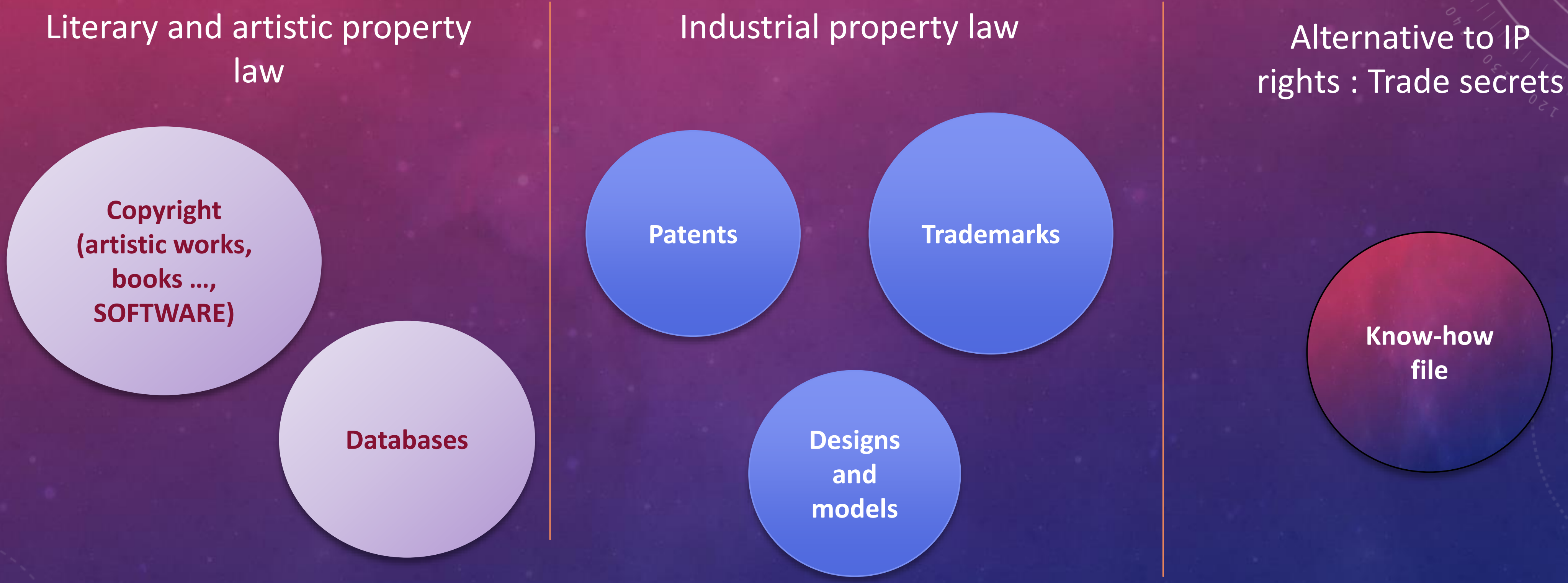
Université de Lille

Software & Data : some elements of Intellectual Property

Sylvain Karpf
sylvain.karpf@univ-lille.fr

What is Intellectual Property law ?

- Defines the legal regime(s) applicable to intangible creations (inventions, artworks, software, etc.)



Literary and artistic property law

Copyright (artistic works, books …, SOFTWARE)

Databases

Industrial property law

Patents

Trademarks

Designs and models

Alternative to IP rights : Trade secrets

Know-how file

Links with other intellectual property rights

- Software cannot be protected as such by patent (in Europe)
- On the other hand, a technical invention can be implemented in software (device, process, etc.)
- The name of a software can be the subject of a trademark registration
- Software often uses/integrates data or databases, which are themselves subject to specific legal provisions.
    - Database law
    - Copyright for certain data (images, photos, etc.)

Copyrights, what type of rights ?

Moral rights :

- belong to the author;
- they are perpetual and inalienable (the author cannot be deprived of them)
- In terms of software, these rights are essentially reduced to the right to authorship

Property rights :

- Correspond to exploitation rights (modification, distribution, etc.);
- Have a limited but nevertheless long lifespan (70 years after first broadcast or after the death of the last author, depending on the case)
- For software: When the author is an employee and created the software as part of his duties, the property rights belong to his employer (law of May 10, 1994)
- **Since the order of 15/12/2021: when the author is a rewarded intern, the property rights belong to the establishment which hosts the intern**

Software: what is protectable under copyright

- Source code and object code
- Detailed specifications and design materials
- Software architecture
- Technical documentation

What is NOT protectable under sofware copyright:

- Algorithms
- Functional specifications
- User guide

Materialization in source code

Source file headers

software xxx version 1.0

Authors : yyy, zzz

© 2018-2020 Université de Lille & CNRS

how to prove that you are the rights holder?

- Unlike patents, software is protected by copyright from its creation, without any particular formality.
- Technical proof (for example: copy of CVS or Github on a certain date) can constitute the beginning of proof but are not considered sufficient.
- There is also legal and certain proof (probatory deposits), carried out by trusted third parties (notaries, organizations carrying out certified deposits, etc.), which currently constitute the best way to assert one's rights.
  - APP (Agence de Protection des Programmes)
  - e-soleau (INPI)
  - …
- Such procedures make it possible to take a date and facilitates the identification and defense of the software (in the event of infringement or if the University is attacked for infringement)

Exploitation of property rights

- **Only property rights can be exploited.**
  - by the software owner himself
  - by a third party, to whom the owner of the software grants a **license**.
- An operating license is a contract that defines the terms and conditions under which the licensee is authorized to operate the software.
- In practice, this results in the existence of a great diversity of licenses in the world of software: opensource licenses, proprietary licenses

Many existing licences

Opensource / free sofware

Licenses for testing and evaluation purposes

license for industrial and commercial purposes

With or without access to source code

License for research purposes

Including the right to modify/adapt the software or not

Free or paid

And many more …

What is Free Software / Opensource sofware ?

A free licence must meet 4 criteria, defined by the Free Software Foundation (FSF, founded in 1985):

- Freedom to use and reproduce the software
- Freedom to study the sofware
- Freedom to modify the software
- Freedom to redistribute the software, modified or not, at least under the original license

The obligation to provide access to the source code is implicit (necessary to study and modify the software)

If a license does not meet one of these criteria, it is not free within the meaning of the FSF

The Open Source Initiative (OSI, founded in 1998) has also defined ten criteria, which are implicitly included in the 4 freedoms of the FSF.

Examples of “fake” free/open source licenses

- Access to source code, right to use, modify, but **no right to redistribute**

- Access to source code, right to use, modify, and right to redistribute **for non-commercial purposes only**

- Access to the source, right to modify, redistribute, **but not the right to use for certain purposes (e.g. military purposes)**

Types of Free/Opensource Licenses

The typology presented below is based on the degrees of permissiveness of a free license in terms of redistribution, depending on the way in which pre-existing code is reused.

Licence type	Owner Philosophy	
Non-permissive license (copyleft) GNU GPL, GNU AGPL, CeCILL	« If you redistribute my code in your software, it is with the same rules for both and by sharing with the community »	« If you redistribute my code in your software or if you use it in SaaS mode, it is with the same rules for both and by sharing with the community »
Permissive license in composition (weak copyleft) GNU LGPL, CeCILL-C, Mozilla Public License, Eclipse Public License	«If you redistribute my code (modified or not) in your software, it must remain accessible and benefit the community ("licensing back"); on the other hand, you choose your rules for the rest of your software »	
Permissive license in composition and derivation (permissive) MIT, BSD, CeCILL-B, Apache	« My code is a gift, do with it what you want! »	

Impact of code reuse

- Reuse of external software components
    - Open Source
    - Non Open Source
        - proprietary code from a partner
        - code that looks Open Source
- For each pre-existing component integrated into an assembly, it is important to identify and qualify the license, to verify its adequacy with the publishing strategy (redistribution/exploitation license)
- If there is a license compatibility problem with a component
    - Ask the owner to change the license
    - Find an equivalent component with a suitable license
    - Redevelop the component

About Data

For example Creative Commons type licenses, inspired by free licenses

CC BY : ok to redistribute intial or modified version, obligation to quote

CC BY-SA : ok to redistribute initial or modified version but under CC BY-SA only, obligation to quote

CC BY-ND : modification/adaptations prohibited for redistribution, obligation to quote

CC BY-NC : ok to redistribute initial or modified version, obligation to quote, no commercial use

CC BY-NC-SA : ok to redistribute initial or modified version but under CC BY-NC-SA only, no commercial use, obligation to quote

CC BY-NC-ND : no modifications, no commercial uses, obligation to quote

Some Business Models using Open Source

Service

- Use of Open Source software, or development and distribution of Open Source software (all licenses possible)
- Associated service activity
  - distribution, packaging
  - Installation
  - Maintenance
  - integration, specific developments
  - etc …

Dual Licensing

- Development of a software
- Distribution of an Open Source version with a non-permissive GPL type license
- Distribution of **the same version** under a proprietary license

Well suited for libraries

Examples : MySQL, Qt

Open Core

- Distribution of the core software under a free license (generally permissive)
  - distribution of a more complete version under proprietary license
  - distribution of specific plug-ins with high added value, under proprietary license