



Open source licenses and IPR protection

INF5750 - Lecture 11

Intro 1



- What is IPR?
- What are IPR control points?
- Patents
- Copyright
- Licenses
- Other control points
- Why is IPR important?

Intro 2



- Open Innovation
- Open source hardware ++
- Selling your company - due diligence

An earlier and more detailed open source presentation by Prodromos Tsiavos can be found [here](#). This presentation has parts from that presentation.

Disclaimer



Copyright, patents, IPR and licensing is very difficult, and is really food for lawyers. If you're doing this for real, it's good to involve lawyers to understand all the details.

Open Source licensing is not always very clear about the limits and possibilities. Any disputes would have to be settled in or outside court.

But it's good for everyone to have a basic understanding of what the issues are...

IPR? What???



IPR = Intellectual property rights

The rights to **'intangible assets'**.

For a company, we speak of **'tangible'** and **'intangible'** assets.

Tangible = can be touched

Tangible assets = property, machines etc...

Intangible assets = knowledge etc.



Intangible assets



- A company has many **intangible assets**
 - Designs, plans, documents, source code...
 - Know-how, processes, hierarchy, structure...
 - The intangible assets are often inside people...
 - Distribution networks, agreements...
 - Brand, customer loyalty...
 - Patents, copyright, design patents (formalized)
- How do you protect something non-physical?

IPR Control Points



- Control point = a way to control IPR leakage
 - Copyright
 - Trademarks
 - Patenting
 - Keep things secret
 - Branding
 - Community and distribution
 - ... more?

Copyright



*“A **copyright** gives the creator of an original work **exclusive rights** to it, usually for a limited time. Copyright may apply to a wide range of creative, intellectual, or artistic forms, or “works”.^{[14][15]} Copyright does not cover ideas and information themselves, only the form or manner in which they are expressed.^[16]”* Wikipedia

Creative works are not protected automatically **internationally**, but there are international treaties that provide some protection.

Copyright 2



- If you write something, you own the rights to that content
- But someone else could write software that does the same thing, so not a very strong protection
- Typically an individual copyright is passed to an employer
- In OS projects, many copyright holders...
- If you get someone's source code, you may not understand it and it may be difficult to replicate it.
- Regardless of copyright, the understanding might be

Copyright 3 - why?



- If something costs a lot to make the first time, but is very cheap to replicate!
- Should have an incentive for people to create works
- But how about those who wish to build upon it?
- Sometimes copyright is about becoming famous, not about becoming rich (recognize author)
- Life of author + x years (but sometimes longer)

Moral rights



“Moral rights are rights of creators of copyrighted works generally recognized in civil law jurisdictions and, to a lesser extent, in some common law jurisdictions. They include the right of attribution, the right to have a work published anonymously or pseudonymously, and the right to the integrity of the work.^[1] The preserving of the integrity of the work bars the work from alteration, distortion, or mutilation.” Wikipedia

Patents



*“A **patent** grants an inventor **exclusive rights** to make, use, sell, and import an **invention** for a limited period of time, in exchange for the public disclosure of the invention. An invention is a solution to a specific technological problem, which may be a product or a process.” Wikipedia*

- A legalized monopoly of an invention.
- Has a timeout period and you must apply per country.
- Dubious status of ‘software patents’
- Lots of patents are held by patent-trolls

Patent example



1. A method in a mobile telephone network having recording and positioning capabilities, wherein content recorded on a mobile device and subsequently submitted to a content platform is managed on a content platform and stored on a content database, positions of the mobile telephones are provided by means of a positioning centre, characterized in establishing a link in the content platform between a content recorded and submitted to the content platform by one of the mobile telephones and the position of the mobile telephone at the time the content is being submitted.

...

WO 2004008790 A1

What is a license?



Permission by the owner of property to take some act that the owner has the ability to control

What is a sublicense?

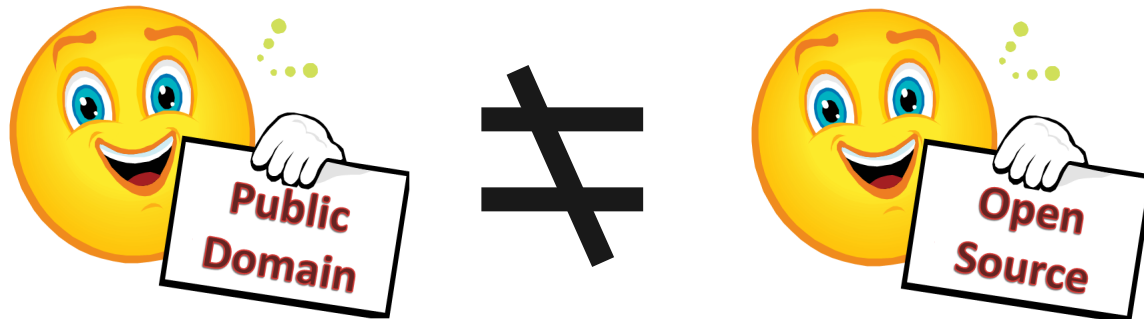
- A license granted by a licensee
- The right to grant a sublicense must be expressly stated in a license and will NOT be implied

Public domain



When something is public domain, it means it is completely open. Anybody can use it. No restrictions!

*“Works in the **public domain** are those whose intellectual property rights have expired, have been forfeited, or are inapplicable.” Wikipedia*



Closed vs open license



- Both are **based on ownership of intellectual property**
- Both **grant certain rights and retain others**
- Both are **governed by the same laws**
- Both may include provisions which may be incompatible with the obligations of other licenses
- License obligations can be **incompatible**, but the issue is whether the obligations are triggered
- As opposed to **public domain**, which is completely open

Source and object code



Two important concepts:

- Source code - The human-readable and editable software
- The compiled, binary software.

In some cases, there's not really a distinction (PHP), or it is blurred (Java can be de-compiled)

Permissive vs Restrictive



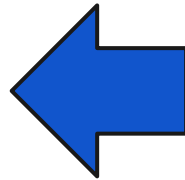
- Restrictive free software licenses want to keep the software free, with same rights for future users. When distributing object code, you **MUST** distribute source code with same license.
- Permissive: you **MAY** distribute the source code when you distribute the object code, but you **MUST** typically mention the authors/organisations where the open source components came from.
- “Distribute” is also an important concept.

Copyleft - very important!



“Copyleft is a general method for making a program (or other work) free, and requiring all modified and extended versions of the program to be free as well.” (GNU Website)

Your existing code-base.
Commercial, secret
software or open source
software.



OSS can be viral



Your software



Weak
Copyleft

Your software



Strong
Copyleft

Open source licenses



Attribution Licenses – compliance is easy

- BSD, MIT, Apache (Example: Apple/MS use BSD IP stack)

Weak Copyleft licenses – more challenging

- Mozilla, EPL, CDDL
- LGPL (LGPLv2 differs from LGPLv3)

Strong Copyleft licenses: most challenging

- GPL (GPLv2 differs from GPLv3) - Linux Kernel etc.
- AGPL (for application service providers / Cloud)

Copyleft - once more



So, if you include software with a permissive license (**no copyleft**), **you do not have the re-distribute** the changed code

If you include software with **weak copyleft**, you have to **redistribute that module's source**

If you include software with **strong copyleft**, you must distribute the **entire software source**

Distributing software

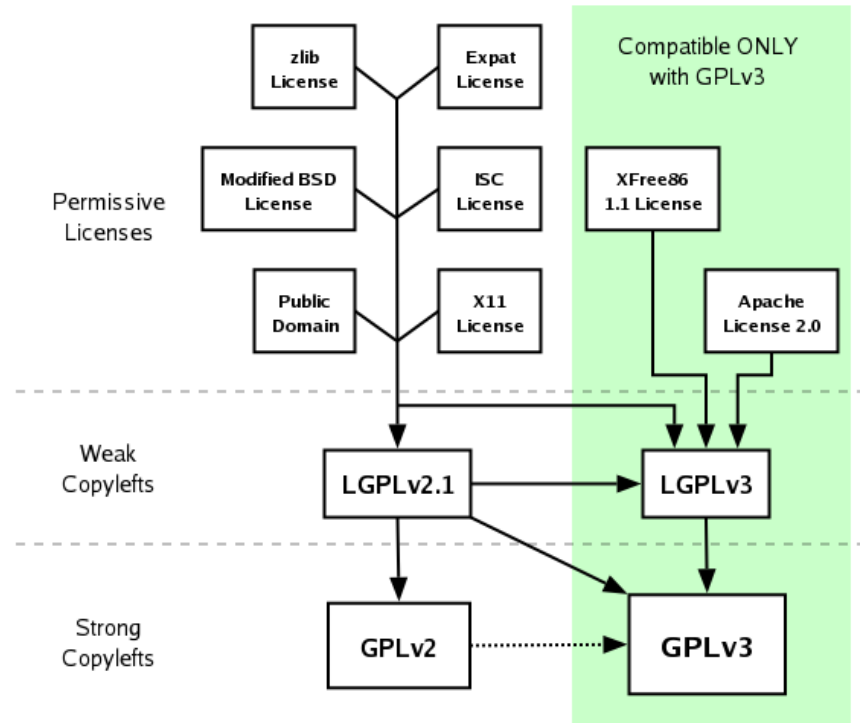


- The [copyleft](#) and requirement to distribute source is typically linked to act of distributing the object code
- Much OSS you can modify and use internally without requirement to distribute
- [AGPL](#) and some other new licenses are different. If you have a **service** based on AGPL, must distribute source.
- If you make your own OSS, you should still claim copyright on it, but distribute it with an OSS license. The copyright gives you the right to offer it as OSS.

License compatibility



Not all licenses are compatible, because they impose different restrictions. For example, you cannot combine EPL and GPL.



Dual licensing



- Sometimes software is licensed under dual licenses.
- The copyright-holder can issue whatever license they want, as long as they hold the copyright to the whole source.
- Common dual licensing: A restrictive license (GPL or similar) that most commercial companies try to avoid, and a commercial license that you must pay for.

Secrecy of your source



- There are many other ways to control your IPR than just keeping your source secret
- Publishing your source may even protect you against patents. If you keep it secret, you are not protected.
- In fact, sometimes opening your source is a great business proposition (more in later lecture)
- Which other control points are there?

Commercial impact of OSS



- If you want to sell your company, the buyer will want to go through a 'Due Diligence'
- Due Diligence means looking into all your financials, customer, your source code etc.
- Buyers often use **automated tools** to go through source code, looking for undesirable OSS licenses
- Remember that if you did use OSS licenses in a way that requires you to distribute software, someone could take you to court to force you. [gpl-violations.org](https://www.gnu.org/licenses/gpl-violations.html)

Using OSS in companies



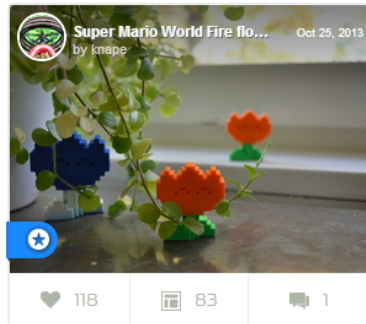
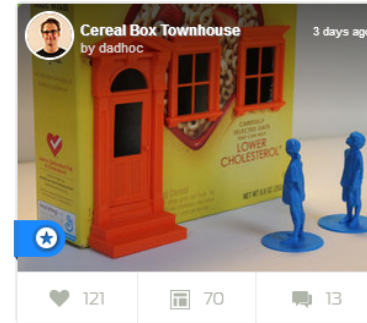
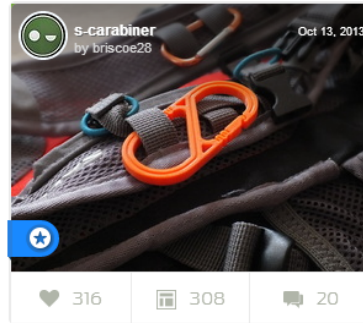
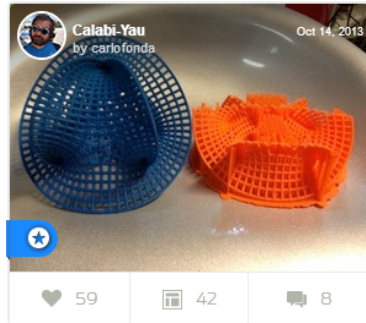
- Using OSS in commercial companies is critical to most companies businesses
- Just make sure you do it right
- Give developers OSS training, so they don't violate the company's strategy with a wrong OSS license
- Maintain lists of all accepted OSS licenses
- Maintain lists of all used OSS modules
- Use automatic scanning tools to detect issues

Open source hardware++



- Not only software can be open sourced. Any creative artifact, including art, literature and even hardware can be open sourced.
- A very common license for non-Software are the Creative Commons licenses.
- A 'configurable' license.
- <http://creativecommons.org/>

Example: Thingiverse.com



Creative commons



Attribution?



Share alike?



No derivatives?



Commercial use?

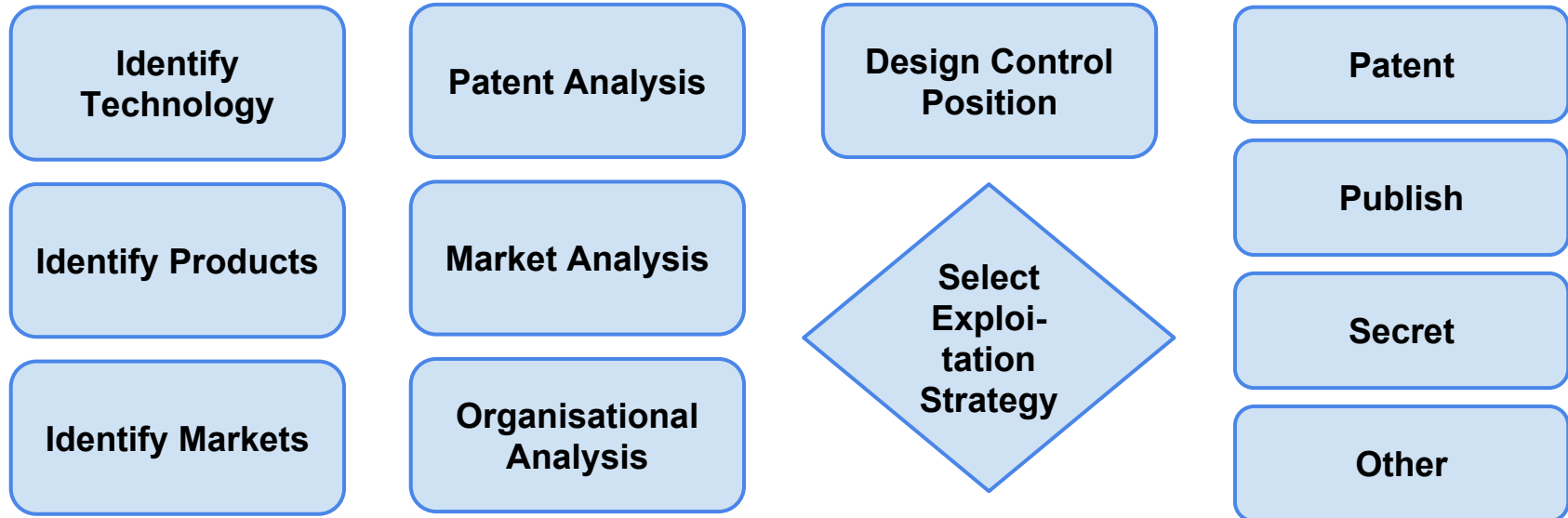


**CONFIGURE
YOUR OWN
License**

IPR Strategy



Identify → **Analyze** → **Select** → **Implement**



Open innovation

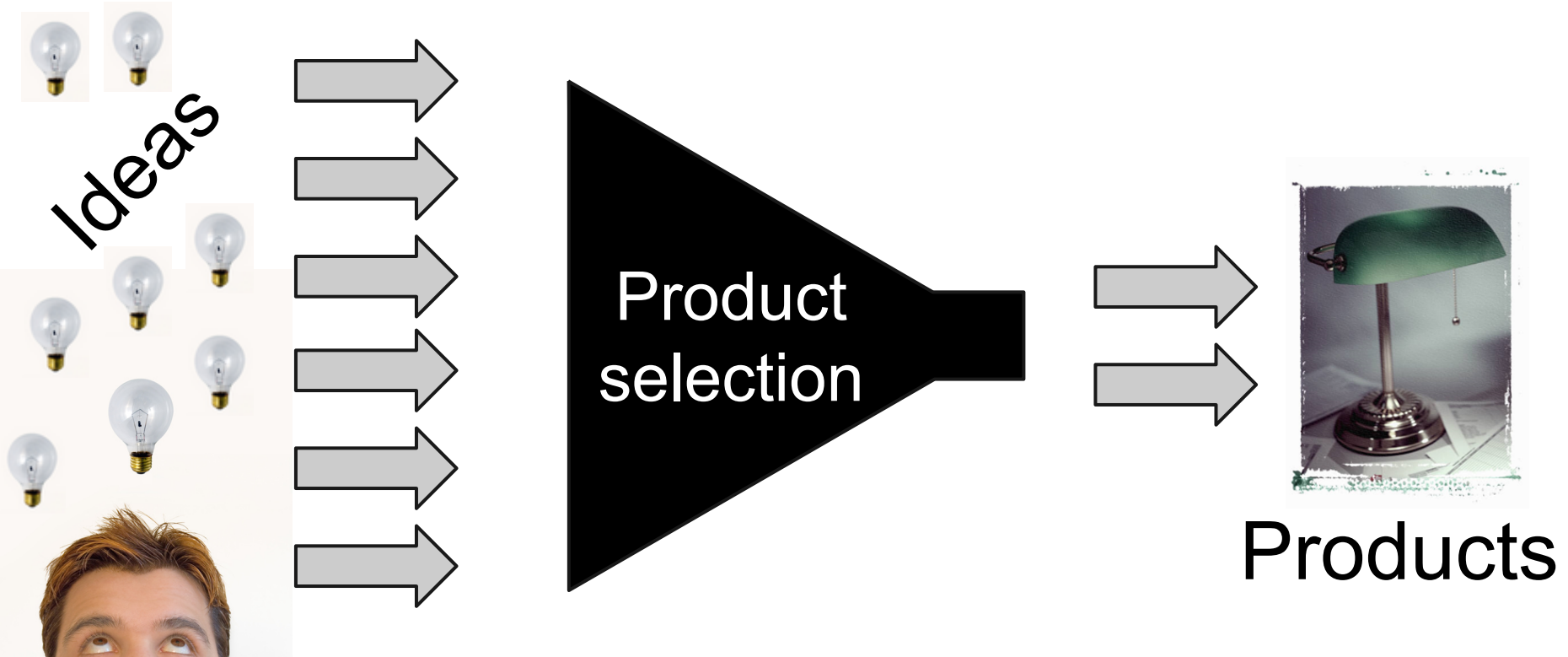


- Companies can no longer rely on own research only
- Companies are increasingly opening up their innovation processes, possibly retaining a level of control
- Example: Split up project and outsource
- Use external experts to solve problems.

Retired/Students

- Create eco-systems, with open interfaces and APIs
- Cathedral (exclusive group of experts) vs Bazaar (developed openly) - *'The individual wizard is where successful bazaar projects generally start' Eric Raymond*

The innovation funnel



The open innovation funnel

