Open Source Software and Licenses

Peter C Rigby

Acknowledgement

- This slides are based on Daniel German’s presentation and are used with permission.

Warning

- Dr. German and I are not a lawyers.
- These are my interpretations, and therefore are not legal advice.
- Most terms and discussion applies to the United States, unless specifically stated
- Law changes rapidly, some of this info might be outdated

OSS and Reuse

- OSS is widely reused:
  - Source is modified to create new products;
  - Products are reused as modules: libraries, executables, etc.
- You might want to release your own product as open source

What is open source software?

- A software product released under an open source license.
- What is an open source license?
  - It is approved by the Open Source Initiative (OSI)
  - OSI defines the 10 characteristics than every OSS license should satisfy

Open Source License Definition [2007]

- 1. Free redistribution
- 2. Source code
- 3. Derived (derivative) works
- 4. Integrity of the author's source code
- 5. No discrimination against persons or groups
- 6. No discrimination against fields of endeavours
- 7. Redistribution of the license
- 8. License must not be specific to a product
- 9. License must not restrict other software
- 10. License must be technology-neutral

OSS Licenses

- It guarantees 4 rights (also known as freedoms):
    - 0 The freedom to run a program, for any purpose;
    - 1 the freedom to study how a program works, and adapt it to your needs;
    - 2 the freedom to redistribute copies of a program so you can help your neighbor
    - 3 the freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

Free VS Open Source

- OSS is considered a business-friendly description
- FS is considered a philosophical stance
- FS is a subset of OS
- Free has bad connotation: “free as in beer”
- Open source is over-used and misinterpreted: “source code available”

Popular Licenses

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

Types of Open Source Licenses [Ros04]

- Academic or Permissive: they allow the licensee a wide range of rights, including creating proprietary derivative works: BSD, MIT, Apache
- Reciprocal: they give you rights, but in return you should release any derivative works under the same license: GPL
- Links to software are permissible, but changes to core software are reciprocal (LGPL, MPL)

Reciprocal or Copyleft

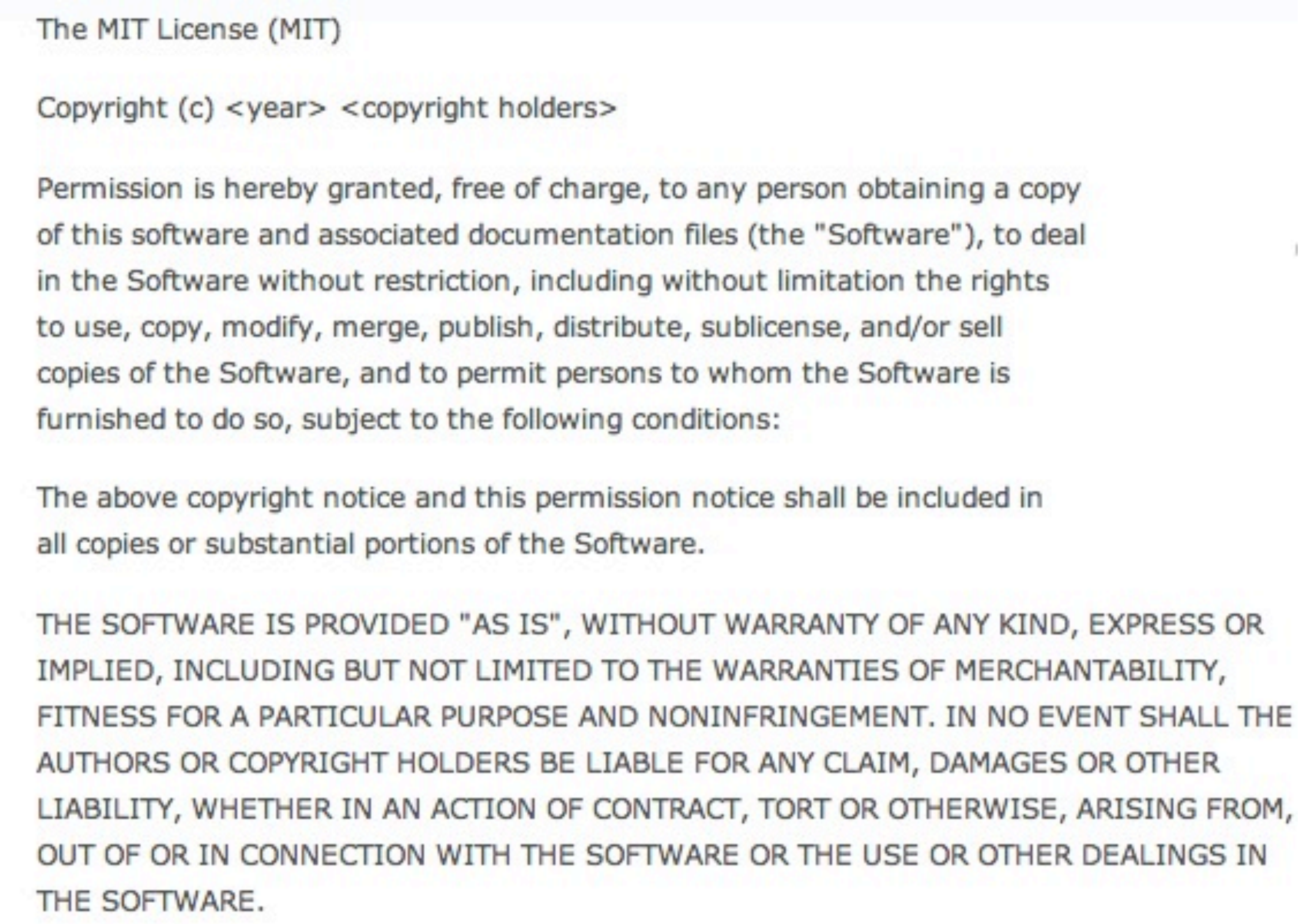
- “Copyleft (a play on the word copyright) is the practice of using copyright law to offer the right to distribute copies and modified versions of a work and requiring that the same rights be preserved in modified versions of the work.”, wikipedia

Other types

- Non-Reusable licenses: Licenses that can only be used by its creator: Apple Public Source License, Nokia Open Source License, PHP License
- Superseded licenses: Older versions of licenses that have been replaced by new versions and are no longer in use: Apache Software License.
- Retired: They have been “retired” and are no longer in use: Intel Open Source License

Academic Licenses

- The MIT and BSD are two of the earliest OSS licenses:
  - Very simple and short
  - They allow you to do almost anything with the source code, including creating proprietary derivative works



The MIT License (MIT)

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

MIT License

BSD vs MIT

- 4 clause BSD
  - “advertising clauses”
- 3 clause BSD
  - Cannot **endorse** product with original authors
- 2 clause BSD = MIT

Apache License version 2.0 (ALv2)

- Approved by the Apache Foundation in Jan. 2004.
- It explicitly grants to the licensee:
  - A non-exclusive copyright license
  - A non-exclusive patent license
- Explicitly states conditions for redistribution: mainly
  - Include license and keep any notices
  - State if you modified any files
- Any new contributions to the product are to be licensed in the same way
- No grant of trademark license
- Disclaimer of warranty and limitation of liability
- Allows a third party to offer warranty and liability on the product

GPL

- A milestone in copyright law, it was written by Stallman
- Based on the Emacs General Public License
- Version 1.0 was released in 1989
- Version 2.0 was released in 1991 (the discussion in these slides applies to this version)
- Version 3.0 was released in July 29, 2007

GPL Bargain

- You can (re)use this software on the condition that any derivative works you create from it and distribute, are also licensed under the GPL.
- This makes the GPL a reciprocal license

How to use the GPL

- The copyright owner has to include a notice in the software that links it to the GPL
- The FSF recommends using "Version 3 or any later version"
- Many projects are still using Version 2 or "Version 2 or any later version" (some even version 1 –perl)
  - This is both a risk and an advantage

GPL applies to programs and derivatives

- The GPL uses always the term program, not software
- In section 0 the GPL defines "work based on the Program" as either "a Program" or a "derivative work under copyright law"
- But it also includes "a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language"
    - where does it become a collective work and not a derivative work?
    - that is one of the main ambiguities of the GPL
- Is linking a collective or a derivative work?
- The GPL seems to imply that linking is a derivative work!

Linking

- “These requirements [you may modify...] apply to the modified work as a whole”
- “If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.”
- “But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Linking

- “As there is no record of anyone circumventing the GPL by dynamic linking without backing down when threatened with lawsuits by the copyright holder, the restriction is apparently de facto enforceable even if not currently de jure”
- [Wikipedia]

Lesser GPL

- Proprietary software can be linked with LGPL software without needing to release the source code of the proprietary software
- Changes to LGPL software are reciprocal
  - Libraries, etc
- Eclipse Public License
  - Plugin architecture
  - Core is reciprocal
  - Plugins can be proprietary

The "best" paragraph of the GPL

- "You are not required to accept this License, since you have not signed it.
- However, nothing else grants you permission to modify or distribute the Program or its derivative works.
- These actions are prohibited by law if you do not accept this License.
- Therefore, by modifying or distributing the Program (or any work based on the Program),
- you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

GPL is a copyright license [Ros04]

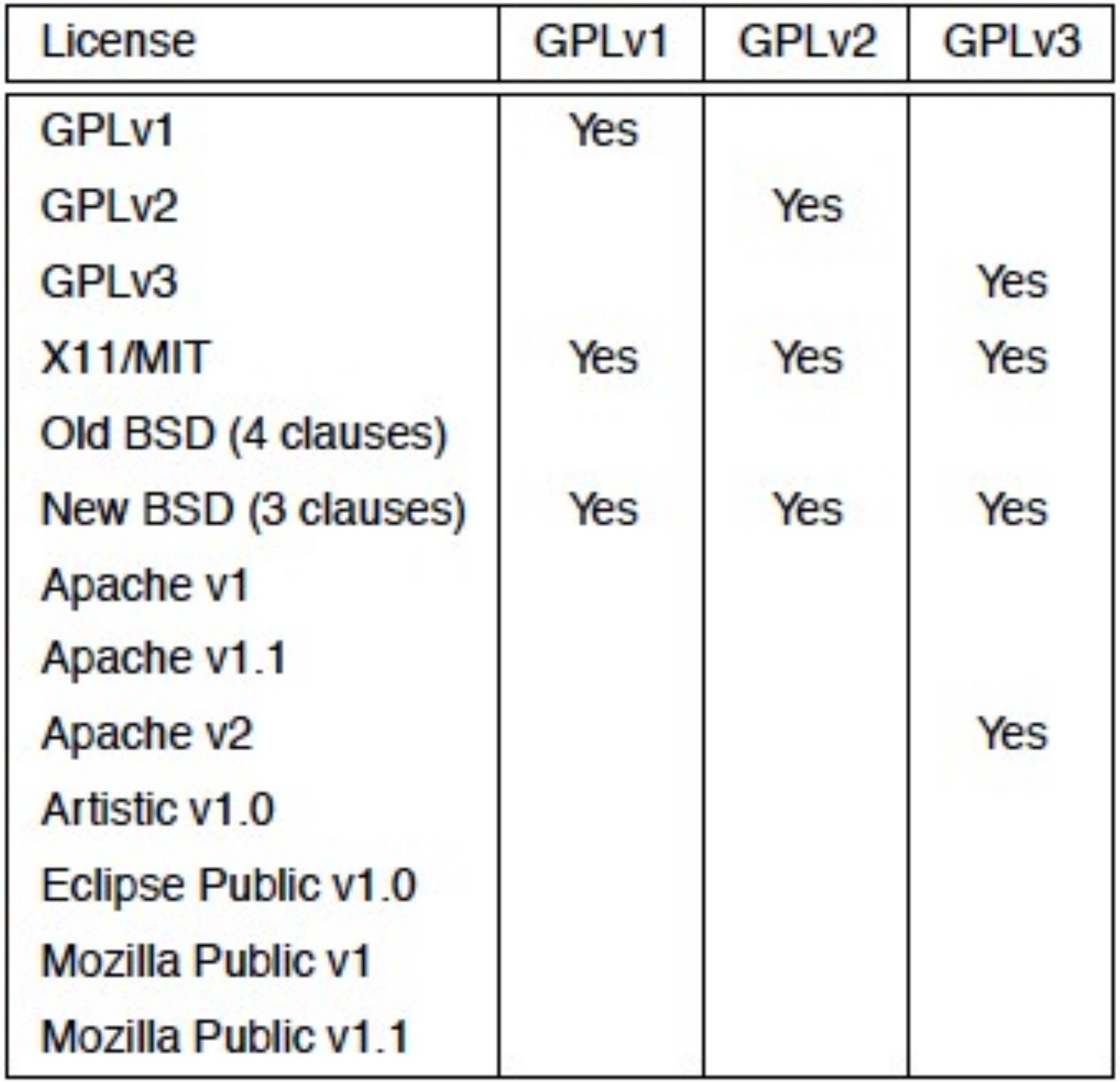
- It is not a contract: it does not ask for any promises in return.
- Disadvantage:
  - Only the copyright owner can sue to enforce it
  - At least in the US, copyright disputes are heard only in federal court (true to Canada too for software)
- “If you want to pursue contract litigation, and obtain contract remedies, you probably don’t want to use the GPL”

Compatibility of Licenses

- OSS licenses might impose contradictory requirements.
- When it is impossible to satisfy the requirements of two different licenses we say these two licenses are incompatible with each other.
- For example:
  - GPL: You cannot impose any further restrictions on this license to redistribute the software.
  - Old-BSD: You must advertise in all your materials the inclusion of the licensed software
- In such cases the two licenses are said to be non-compatible

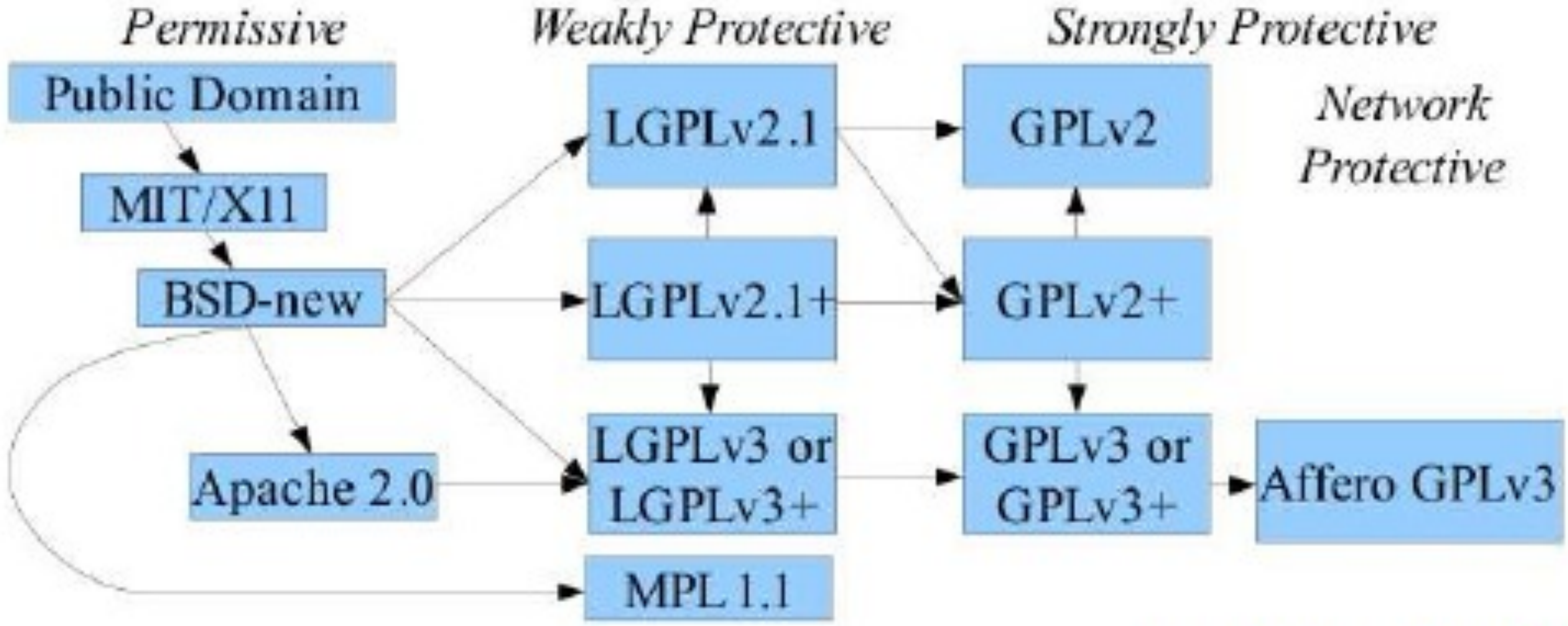
Compatibility of Licenses

- In practical terms, software under one license cannot be "combined" with software with a non-compatible license to create a new product
- The definition of "combined" is a tricky issue.
    - Is the new work a compilation or a derivative work?
    - There exist many ingenious ways to circumvent this.



License	GPLv1	GPLv2	GPLv3
GPLv1	Yes		
GPLv2		Yes	
GPLv3			Yes
X11/MIT	Yes	Yes	Yes
Old BSD (4 clauses)			
New BSD (3 clauses)	Yes	Yes	Yes
Apache v1			
Apache v1.1			
Apache v2			Yes
Artistic v1.0			
Eclipse Public v1.0			
Mozilla Public v1			
Mozilla Public v1.1			

Compatibility of the GPL



[D. Wheeler, 2007]

Compatibility of Licenses

Compatibility of Licenses

- Many OSS Licenses have been evolving towards compatibility:
  - Artistic Public License v1 is not compatible with the GPLv2 but its v2 is.
- Apache License version 2.0 is not compatible with the GPL v2, but it is with the GPL v3.

Choosing and Open Source License

- Do you want to be a licensee of somebody else's software?
    - To use?
    - To create a derivative work?
- Do you want to be a licensor of your own code?
    - What limitations do you want to impose on others? if any.

As Licensor [Ros04]

- Do you want to avoid the "Free-Rider" problem?
  - Impossible to avoid: anybody can make copies of the software
- But you can avoid proprietary derivative works
  - Use a reciprocal license (MPL, GPL)

Licensor (make money?)

- Make money on what your open source license does not allow
- MySQL AB and Trolltech make money by licensing their software to those who can’t use it under the terms of the GPL (e.g. proprietary derivative works)
- Provide warranties and service (e.g. Cygnus and many GNU/Linux distributions)
- Use your trademark, not the software itself (e.g. RedHat line of GNU/Linux distributions)

Microsoft OSS

- Determine if you want an academic or a reciprocal license
- Microsoft often states it prefers to support projects that use academic licenses:
- " Proposals should evidence a commitment to make all results [...] broadly available by either: (i) [...] the public domain; or (ii) [...] a non-restrictive license that allows modification and redistribution...An example of an acceptable license is the BSD license whereas the widely used GPL and LGPL are not acceptable"

As licensor: Key issues to decide [Ros04]

- Do you agree with the “legal language”: definition of distribution, derivative work, etc.
- Do you care about patent licenses?
- What type of warranties are you willing to give?(provenance? none?)
- Do you want a copyright license or a contract license?

As licensor: Key issues to decide [Ros04]

- Consider your disclaimer of liability (many argue placing software into the public domain is bad because it does not absolve the author from liability)
- Do you agree with the jurisdiction, venue and governing law provided by the license?
- Do you want an attorney’s fees provision?

As Licensor: Contributions

- When you accept contributions a key issue is copyright ownership
- Many projects only accept contributions if their copyright is also transferred
- Is it important to be able to dual-license (OOS and proprietary)?
- Simplifies change of license
- “Foundations” are becoming a common legal owner of the copyright of open source projects (Apache, Mozilla, Free Software, Eclipse)

As Licensee [Ros04]

- Are you planning to create, and potentially distribute derivative works?
  - Patent rights
  - Do you need additional rights?
  - License conditions
  - Liability limitations
  - Warranty limitations
  - Jurisdiction, venue and governing laws of the license
  - Termination provisions

References

- [Fre07] Free Software Foundation. FSF Web Site. http://fsf.org, 2007.
- [Ope07] Open Source Initiative. OSI Web Site. http://opensource.org, 2007.
- [Ros04] Lawrence Rosen. Open Source Licensing: Software Freedom and Intellectual Property Law. Prentice Hall, 2004.
- [Whe07] David Wheeler. Make your open source software gpl-compatible. or else. http://www.dwheeler.com/essays/gpl-compatible.html, 2007.