



DPIIT IPR CHAIR
&
N.C.BANERJEE CENTRE FOR IPR STUDIES
NALSAR UNIVERSITY OF LAW, HYDERABAD



FOSS

[Free & Open Source Software]

Advanced Diploma In Cyber Law
NALSAR University of Law, Hyderabad
Contact Class; April 2023.

ANINDYA SIRCAR

Software

- **Commercial Software** requires payment before the user can download and use the software. Commercial software typically does not allow source code to be modified, and usually restrict access to the source code altogether.
- **Shareware** is free to use, but limits the amount of time the program's features are available to the user.
- **Freeware** is software that is allowed to be shared for no cost, but not allowed to be modified.
- **Free Software** – Free Software Foundation defines Free Software as software that “the users have the freedom to run, copy, distribute, study, change and improve”.
- **Creative Commons** is non-profit organization that facilitates sharing creativity and knowledge by providing free legal tools. They offer free and easy to use copyright licenses that offer flexible ways to share creative work on the conditions that authors are going to choose.
- **Public Domain Software** is the software that is not a subject of copyright. This type of software is freely available through public domain, and can be used without any restrictions.

Software

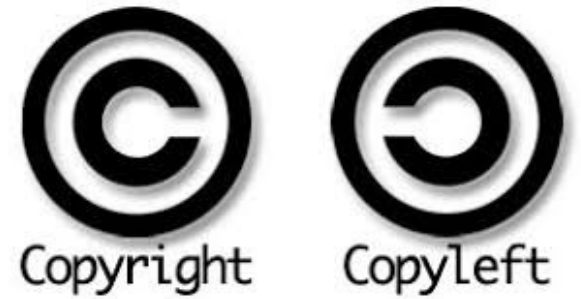
- **Open Source** —. Software has to comply with specific criteria in order to be considered open source. According to Open Source Initiatives, open source software
 - Has to be distributed freely
 - Source code has to be included, and distribution in compiled form must also be allowed
 - Software must allow modifications and derived works, and if used, must be distributed on the same terms
 - Integrity of the author's code should not be compromised. If the code is modified it should only be distributed further using a different name.
 - The license can't discriminate against any individual, or group of individuals
 - The license cannot restrict the field that the software will be used in, nor can it restrict the software from being used for commercial purposes.
 - The rights that are attached to the software should apply to all who received the software without any additional license from the party the software was received from.
 - License must not be specific to a product
 - The license cannot restrict the use of any other software using the same license
 - License must be technology neutral meaning that no technology can dictate the style or interface of the software.
- This type of software always gets confused with free software, and freeware

Copyright



- Copyright protects the expression
- All software is automatically covered by copyright (as long as work is copyrightable)
 - `for (int i = 0; i < FOO; i++)`
 - Not copyrightable
 - `perl nle`
`'print for m/(\S+)\s+the\s+(\S+)/g' paragraphs`
 - maybe copyrightable
 - `main(v,c)char**c;{for(v[c++]="Hello, World!\n");(!c)[*c]&&(v||`
`c&&`
`execvp(*c,*c,c[!c]+!c,!c));**c=!c)write(!*c,*c,!**c);}`
 - Copyrightable
- *“As to copyright, a single sentence is generally accepted to be too short to qualify for copyright protection. Still, at the same time, a haiku is likely to be protected even though it's easy to write single sentences that are longer than haikus. ”Anindya Sircar, Anindya@sircar.org October 14, 2015”*
- In case of software one can't say the LOC that qualifies for copyright protection Copyright holder has the following rights to his work (covered by the international copyright laws)
 - Exclusive right to make copies
 - Exclusive right to prepare derivative works
 - Exclusive right to distribute copies of the original or derivative works
- In case of literature, music, movies etc. you have an exclusive right to display the work publicly
- Others don't have these rights and they are not allowed to perform these actions without copyright holders permission
- There can be many copyright holders for a single work.

Copyleft



- Copyleft is a general method for making a program free software and requiring all modified and extended versions of the program to be free software as well.
- The simplest way to make a program free is to put it in the public domain, uncopyrighted.
- Copyleft, distinguished from copyright, is the practice of offering people the right to freely distribute copies and modified versions of a work with the stipulation that the same rights be preserved in derivative works created later. Copyleft software licenses are considered *protective* or *reciprocal*, as contrasted with permissive free-software licenses.
- In contrast, under copyleft, an author must give every person who receives a copy of the work permission to reproduce, adapt, or distribute it, with the accompanying requirement that any resulting copies or adaptations are also bound by the same licensing terms.
- Copyleft licenses for software require that information necessary for reproducing and modifying the work must be made available to recipients of the binaries. The source code files will usually contain a copy of the license terms and acknowledge the authors.
- The strength of the copyleft governing a work is an expression of the extent that the copyleft provisions can be efficiently imposed on all kinds of derived works.

Patents

- Patents protect the inventions
- Patents must be obtained separately
 - Long and costly process
- Patent holder rights
 - Right to exclude others from making products embodying your patented invention
 - Right to exclude others from using products embodying your patented invention
 - Right to exclude others from selling or offering for sale products embodying your patented invention
 - Right to exclude others from importing products embodying your patented invention



Trademarks



- Purpose is to differentiate from other products
- Can be owned, sold and licensed
- OS licenses don't license trademarks
- For example if you want to use Linux trademark in your product the license must be obtained from Linux Mark Institute

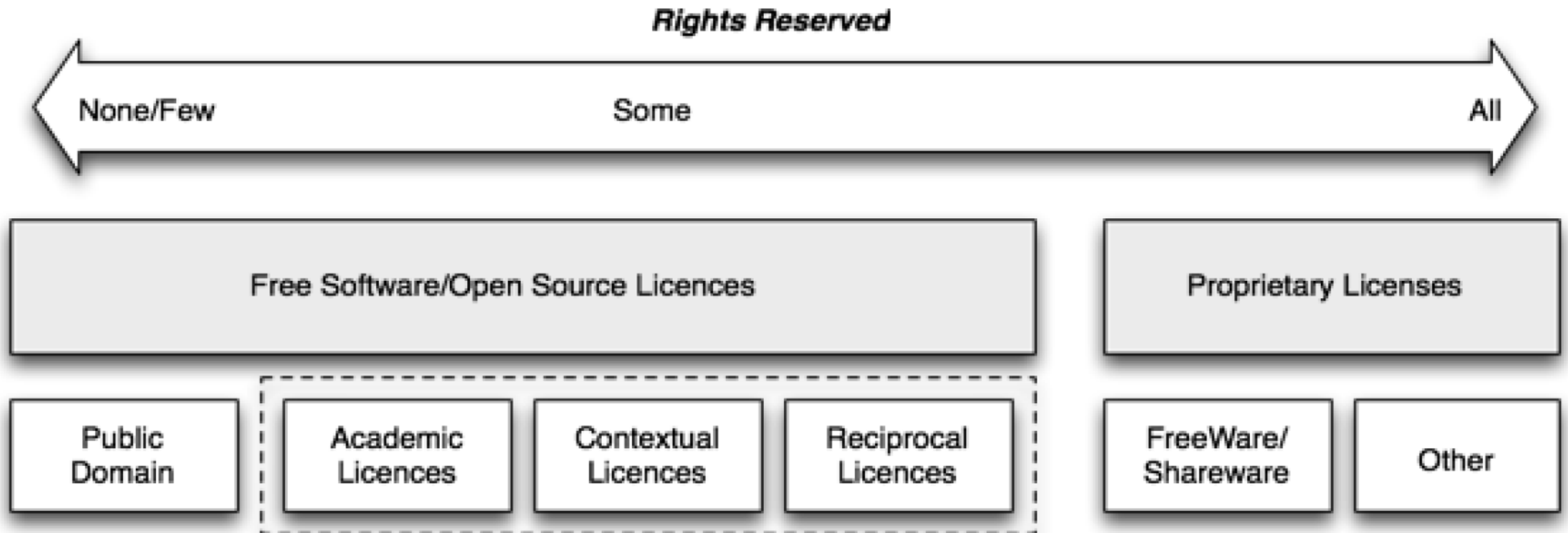
Software Licenses

- License is simply a permit to do something that is not legal otherwise.
- Software license describes copyright and patent holders permission to use their intellectual Property.
- Open Source Initiative approves OS licenses. Approved licenses can be found from:

[<http://www.opensource.org/licenses/>](http://www.opensource.org/licenses/)



Software Licenses



Software Licenses & IP



- Software is not sold - but is instead licensed.
- A software license is a legal instrument that governs the use or redistribution of software. These contracts transfer IP rights from the owner (licensor) to another party who wants to use them (licensee). They can be exclusive – allowing only one licensee – or nonexclusive – granting rights to multiple users.
- The copyright holders retain “ownership” of their software, but they grant to licensees the right to use the software subject to certain restraints.
- The kinds of restraints placed on users distinguish proprietary software from open-source software.
- Patents are particularly insidious because they prevent open-source software programmers from doing what they want.
- Patents can hold standards hostage.
- Free license: (from Apache) *“each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license”*
- Patent retaliation: (again from Apache) *“If You institute patent litigation against any entity (...), then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed”*
- Bear in mind that innovation is not something that either does or does not happen - it happens either more or less.
- Saying that the industry was innovative and productive in the past does not prove that the rate of innovation and investment cannot be improved today, or even that the rate could not have been improved had patents existed in the past.
- There is a definite “need” for patent protection in the software industry
- Patent thickets are not inevitable - Software development is a sequential and cumulative process.
- The alternative to patents may be worse.

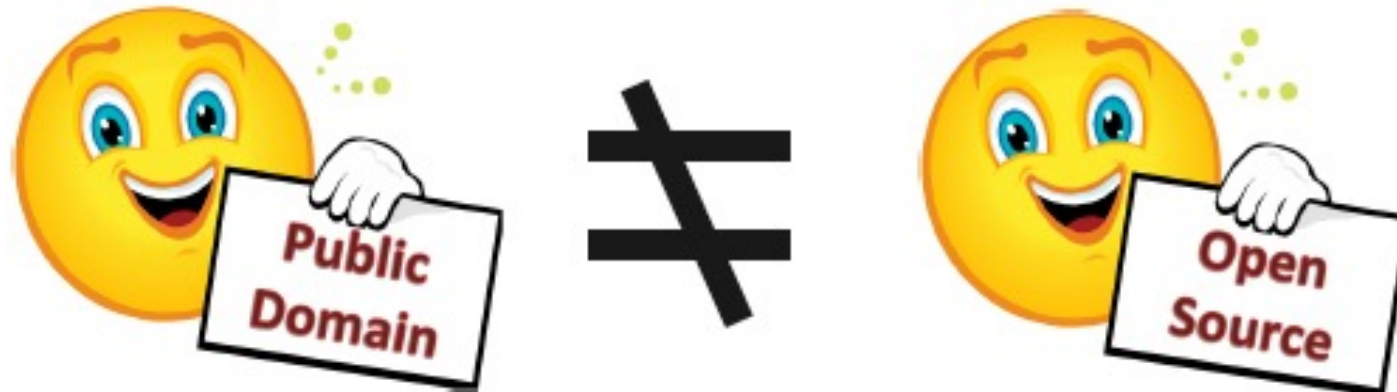
Licenses – Closed & Open

CLOSED & 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

PUBLIC DOMAIN

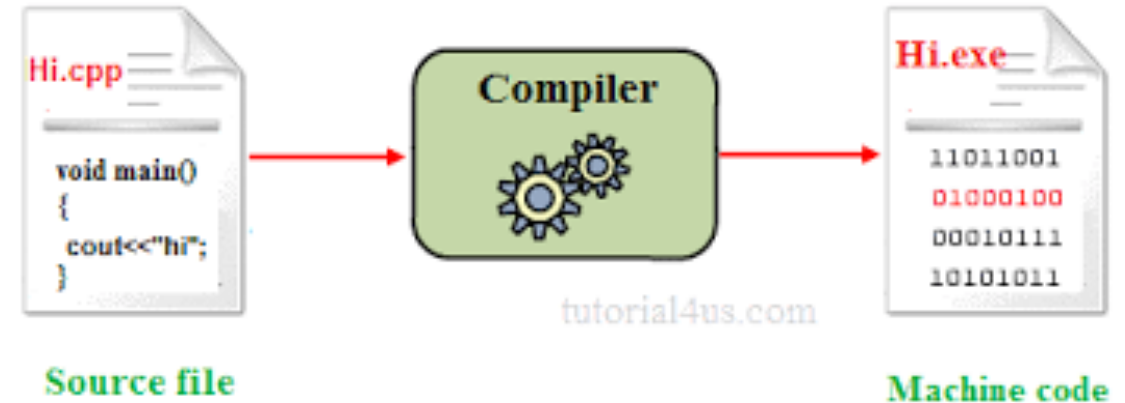
- When something is public domain, it means it is completely open.
- Anybody can use it. No restrictions!
- “Works in the public domain are those whose intellectual property rights have expired, have been forfeited, or are inapplicable.”



Source & Object Code

Two important concepts:

- Source code
 - The human-readable and editable software.
 - Is served as an input to the compiler of the language in which it is written.
- Object code
 - The machine executable file having instructions for the machine in the form of binary digits, generated by the compiler.
 - The compiled, binary software. In some cases, there's not really a distinction (PHP), or it is blurred (Java can be decompiled)



Restrictive vs Permissive

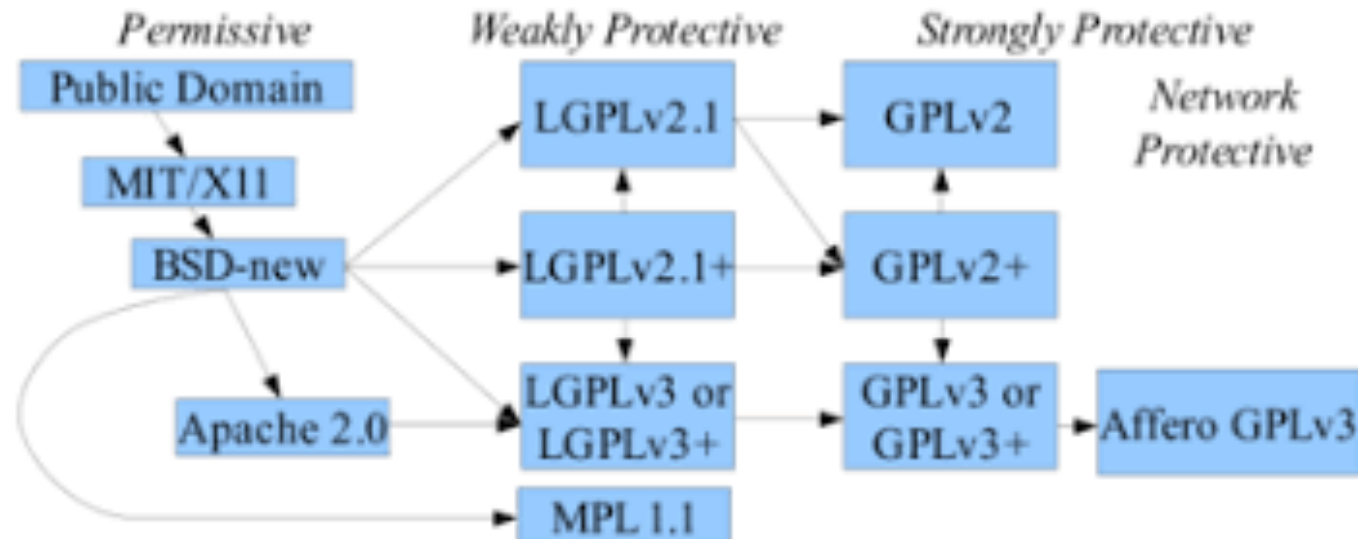
➤ Restrictive:

- 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.
- Changed software must be made available under similar terms with respect to. the original .

■ Permissive:

- One **MAY** distribute the source code when you distribute the object code, but one **MUST** typically mention the authors/organisations where the open source components came from.
- Modifications/enhancements may remain proprietary.

■ “Distributive” is also an important concept.



Open Source Licenses

- Open Source licenses guarantee following rights to the user:
 - Licensees are free to use Open Source software for any purpose whatsoever
 - Licensees are free to make copies of Open Source software and to distribute them without payment of royalties to a licensor
 - Licensees are free to create derivative works of Open Source software and to distribute them without payment of royalties to a licensor
 - Licensees are free to access and use the source code of Open Source software
 - Licensees are free to combine Open Source and other software
- User rights can be expressed explicitly or implicitly in OS licenses
- By contrast commercial software licenses grant users a limited right to use the program
- Open Source license types
 - Academic licenses (for example BSD and MIT)
 - Reciprocal (for example LGPL and MPL) licenses
 - Extreme cases are viral licenses (for example GPL)
 - Content licenses (for example AFL, Creative Commons)
- More than 80 % of the OS software is licensed under GPL or LGPL



Open Source Licenses - Warranties

...Doesn't Exist (*unless someone really wants to give one*)

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(Part of BSD license)

Open Source Licenses - Criteria

Freedom to run the program

Allow the licensee to run the program, for any purpose, therefore with no restrictions.

Freedom to study and change

Allow the licensee to study how the program works and modify it according to expectations or in order to answer the needs that the software is supposed to meet, with the precondition that access to the source code is given.

Freedom to redistribute

Allow the licensee to redistribute unmodified copies, without restrictions.

Freedom to distribute

Allow the licensee to distribute modified versions (known as derivative or larger works).



Closed Source

Vs

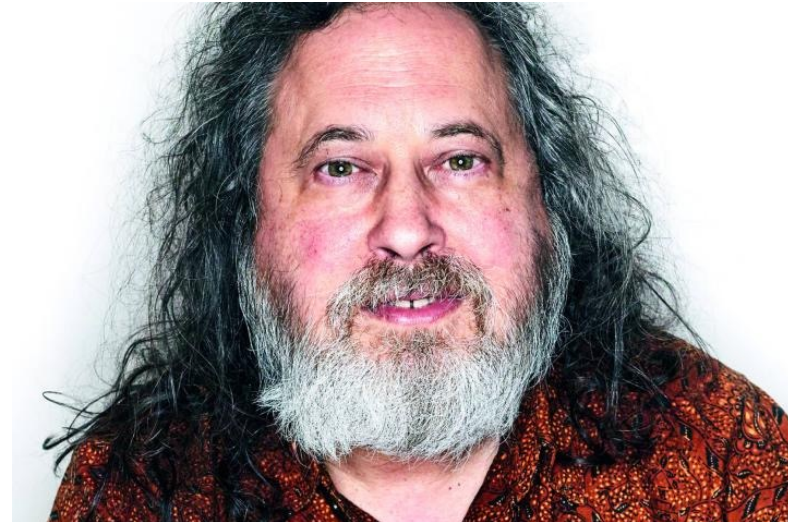
Open Source



Free Software Foundation (FSF)



- In the early 1970s, programmers in academic settings shared programs and swapped sections of source code. This could be viewed as a precursor to today's open source movement.
 - In 1984, Richard Stallman founded the Free Software Foundation (FSF).
 - The idea began in 1980, when Stallman was refused access to the source code for software used to access a printer he used at the MIT Artificial Intelligence Lab. His indignation over the incident eventually led to his morals-based view of software sharing and his founding of the FSF.
-
- Stallman ardently rejects the label “open source” software, preferring instead “free software.”
 - As Stallman expresses it, *“My work on free software is motivated by an idealistic goal: spreading freedom and cooperation. I want to encourage free software to spread, replacing proprietary software that forbids cooperation, and thus make our society better”*.
 - One of the best-known open-source programs is Linux.
 - Developed by Linus Torvalds starting in 1991, Linux is an operating system patterned after Unix, an operating system developed by AT&T in 1969 and still used extensively today, particularly on server computers



Academic Licenses

- Origins of these licenses is at universities
- Most common academic licenses: BSD, MIT, Apache
- Idea is to give the software to the users and let them use it any way they want
- Permission to relicense the software.
- Derivative works can be closed source (commercial) software



MIT License



- MIT license <<http://www.opensource.org/licenses/mitlicense.php>> permits the use of copyright holders intellectual property on two conditions:
 - Copyright notice and permission notice must not be removed
 - Copyright holder gives absolutely no warranty for the program
- MIT license is often found from the beginning of each source code file.
- 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 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.

BSD License



- BSD license <http://www.opensource.org/licenses/bsdlicense.php> is very close to the MIT license
- Some additional restrictions:
 - “Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.”
 - Distribution in binary form must include copyright notice and the license itself.

Reciprocal Licenses

- Derivative works must be distributed under the same license as the original work
- In GNU world, a term Copyleft is often used:
<<http://www.gnu.org/copyleft/copyleft.html>>
- Viral licenses require that all software that use the original work must be licensed under the same license
- LGPL and MPL are reciprocal but not viral licenses
- GPL is a viral license



This condition has been called “*viral*” because it typically means that once code is licensed under the GPL, any other code that comes in contact with it falls under the GPL as well.

According to the actual text of the license: “*You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License*”.

GPL License

- GPL is the most important OS license
- GPL v. 2 was very unclear in many ways
 - Lots of technical “babble” that was based on old software techniques
 - License was contradictory with itself
- GPL v. 3 was created to clarify and to enhance the old license
 - Better compatibility with other licenses
 - More difficult for engineers to understand but much better for lawyers
- Copyleft in GPL 3
- Clause 5c) *“You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it”.*
- Whenever someone conveys software covered by GPLv3 that they've written or modified, they must provide every recipient with any patent licenses necessary to exercise the rights that the GPL gives them. In addition to that, if any licensee tries to use a patent suit to stop another user from exercising those rights, their license will be terminated.
- What this means for users and developers is that they'll be able to work with GPL v.3 covered software without worrying that a desperate contributor will try to sue them for patent infringement later.
- All derivative works of GPL must be GPL
 - *What constitutes combining two parts into one program? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).*
 - *If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program.*
 - *By contrast, pipes, sockets and command line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program.*

LGPL License

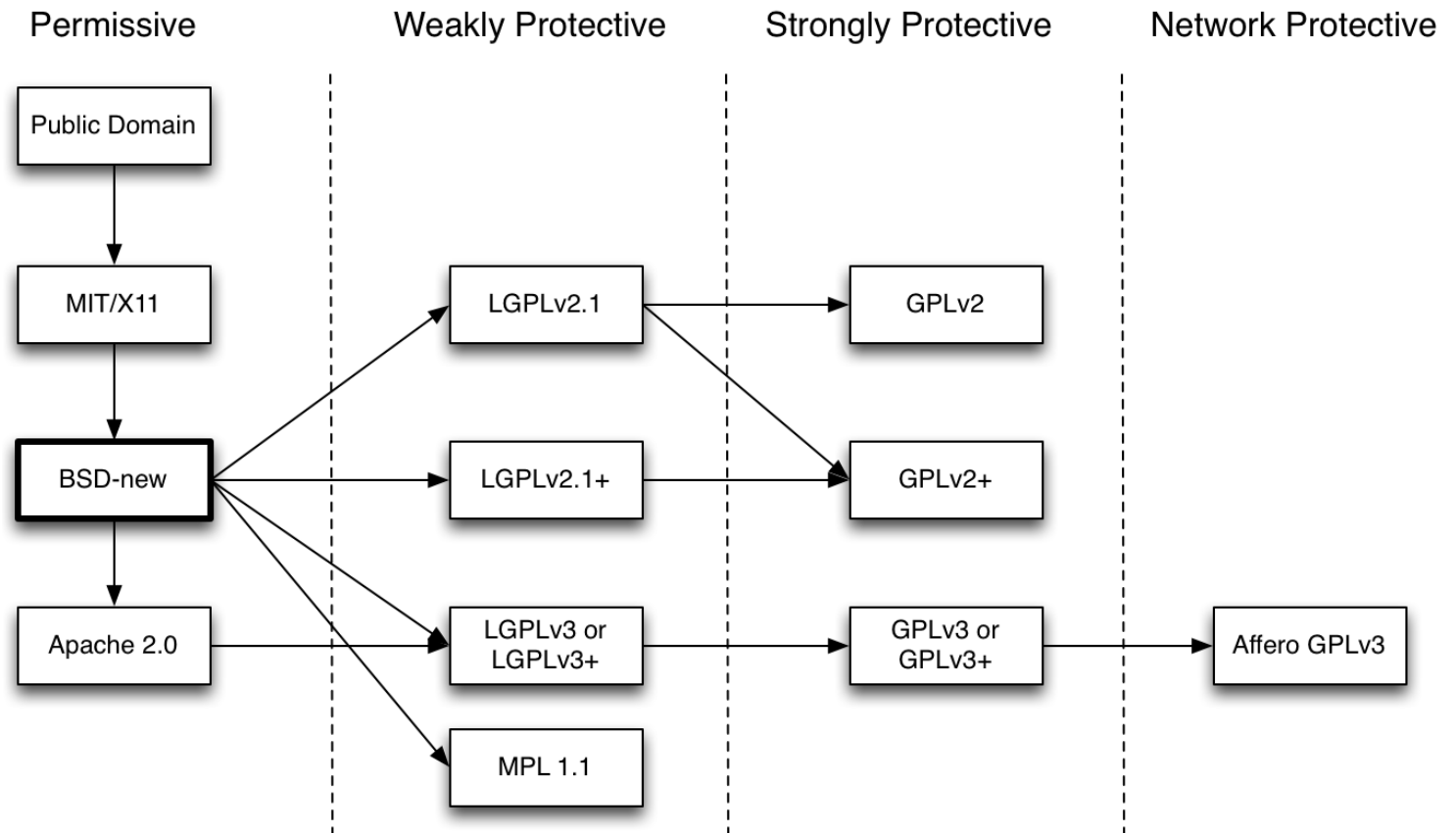
- LGPL, Lesser General Public License (formerly Library General Public License) is mostly a copy from GPL
- Not a strong copyleft license, because it permits linking with non free modules
- *“You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications”.*



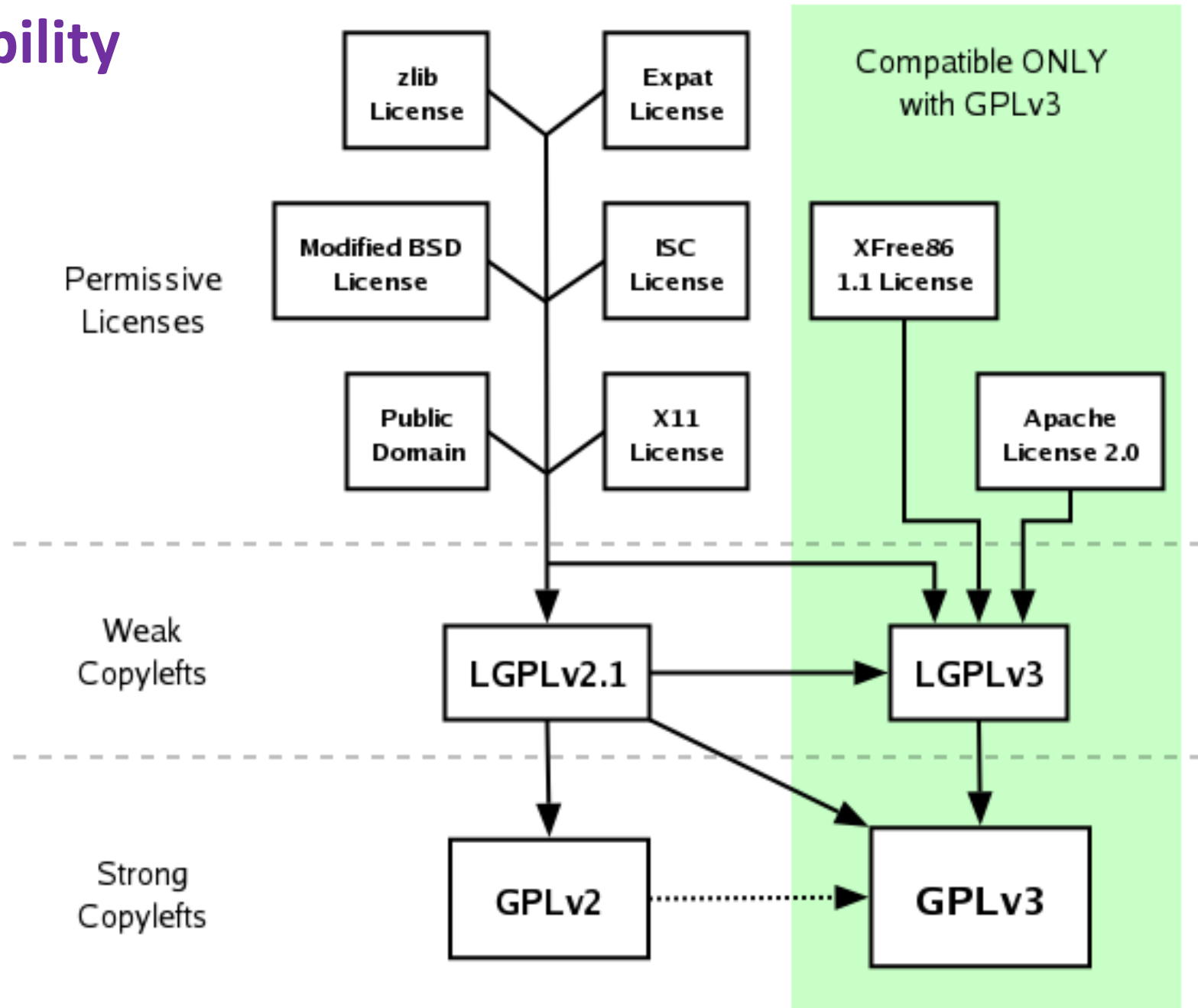
License Compatibility

- Licenses must be compatible in order to create works that contain software that is licensed under different licenses
- List of GPL compatible and incompatible licenses can be found from FSF web pages:
<http://www.fsf.org/licensing/licenses/index_html#SoftwareLicenses>
- Well known GPLv2 incompatible licenses
 - Mozilla Public License (MPL)
 - Xfree 86 1.1 license
 - Original BSD license
 - Apache license

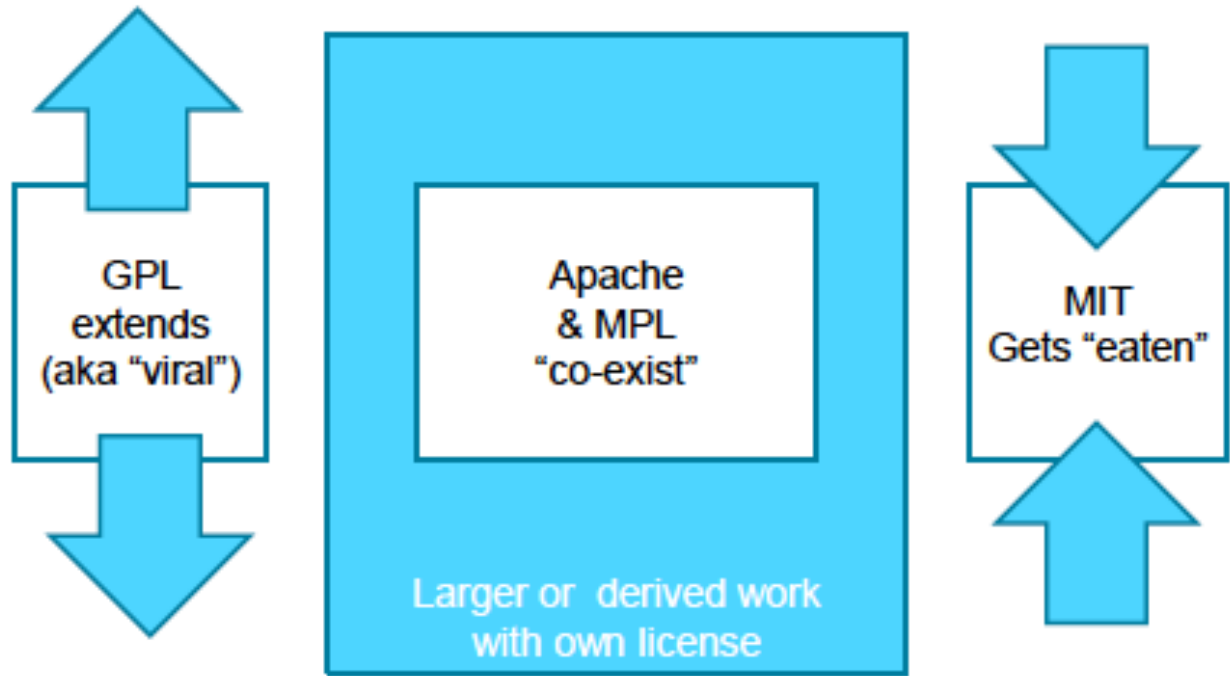
Open Source License Compatibility Chart



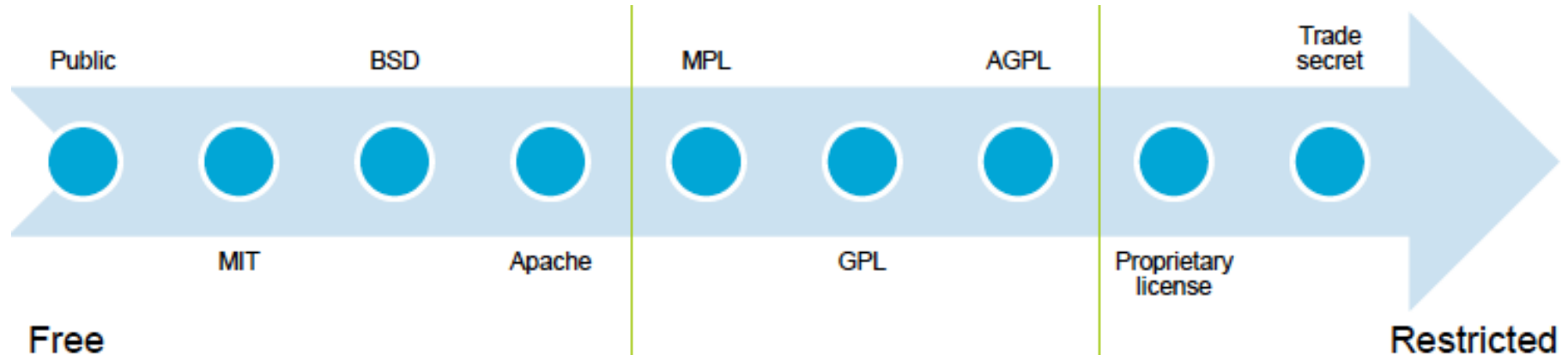
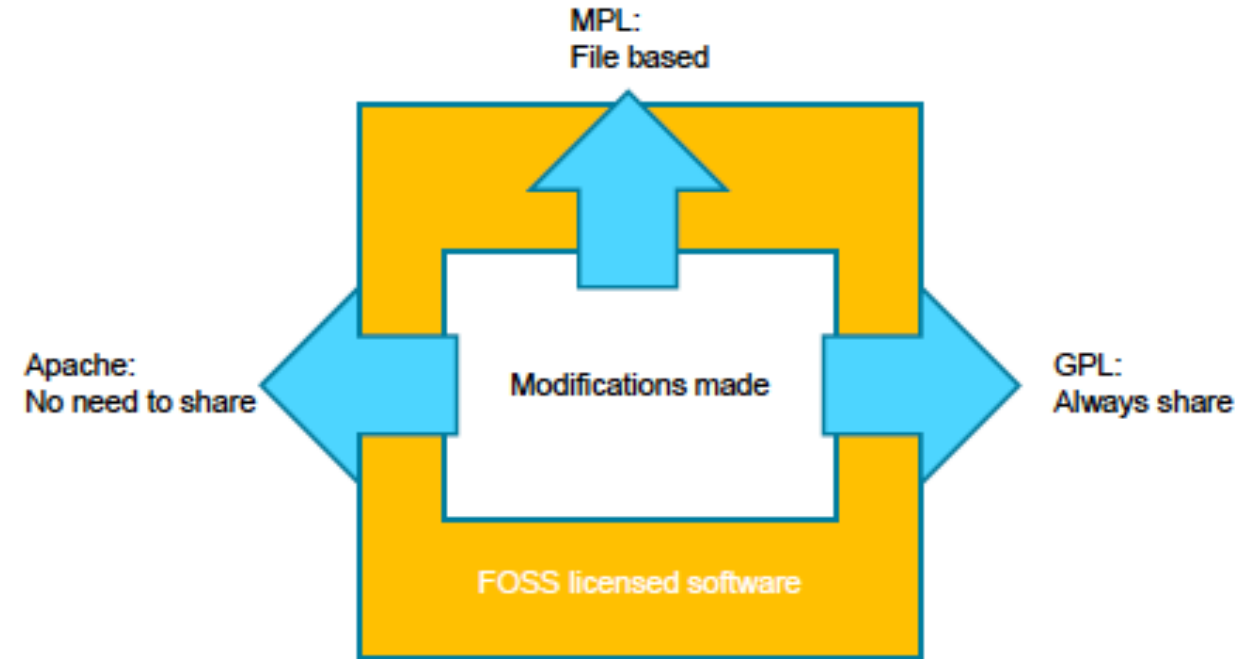
GPL3 – License Compatibility



Derivative Works

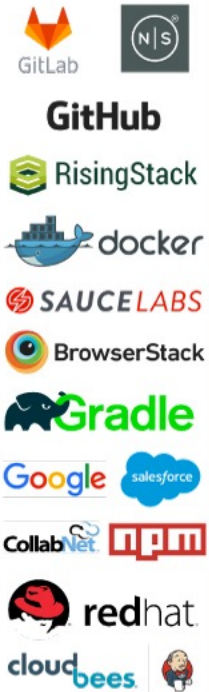


Reciprocity



14

DevOps



11

IT Operations



15

Data & Analytics



Comparison of the Open Source Licences

The bullets mark if the the licence explicitly states the item in question. Implicit items are not marked by this chart

Apache License 2.0

Common Development and Distribution License

GNU General Public License (GPL)

GNU Library General Public License (LGPL)

Microsoft Public License (Ms-PL)

Microsoft Reciprocal License (Ms-RL)

Mozilla Public License 1.1 (MPL)

New BSD License

The MIT License

Must distribute license with binray or source

Cannot use contributors name to endorse

There has to be a notification for changed files

Any change must distributed in source form

Lets you provide warrenty if you want to, normally no

Lets you explicitly charge for providing warrenty or gurantee or transfer of code

All derivative work must be under the same license

Must show License when Run from command line

Non derivative works can have different license

May exclude countries where there is a contradiction with patent in that ocuntry

Must describe any deviation due to regulation

●	●	●	●	●						
●	●	●	●			●				
●		●	●		●	●	●	●	●	
●	●	●	●		●			●	●	
●	●		●					●		
●	●	●	●							●

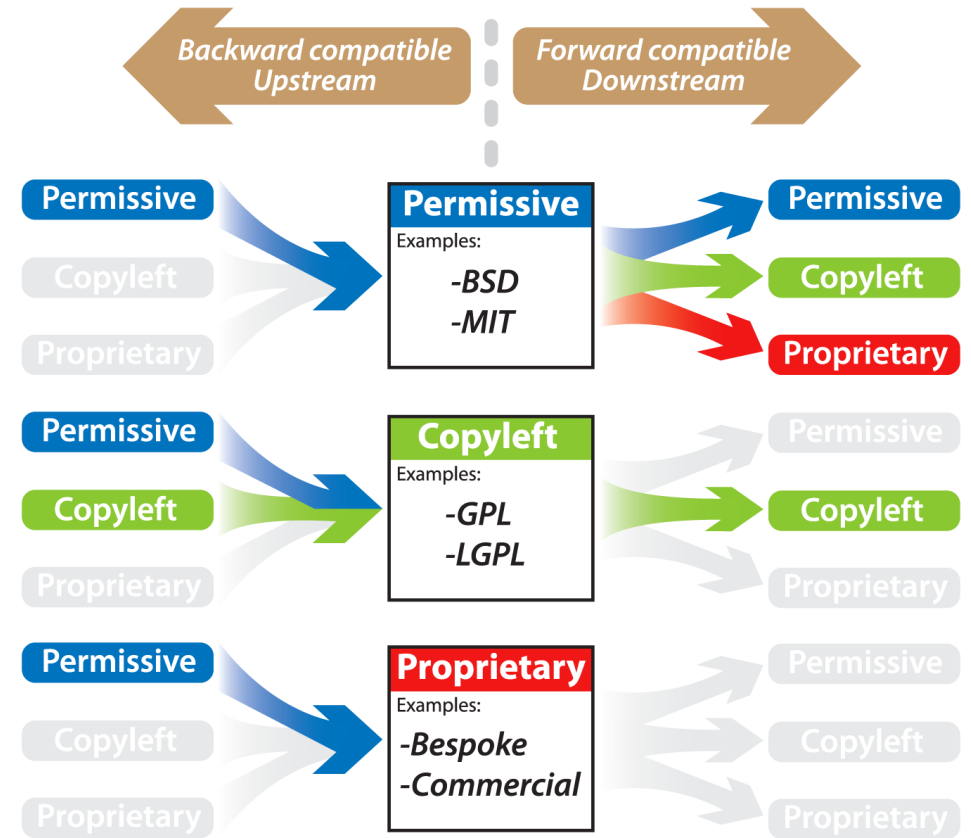
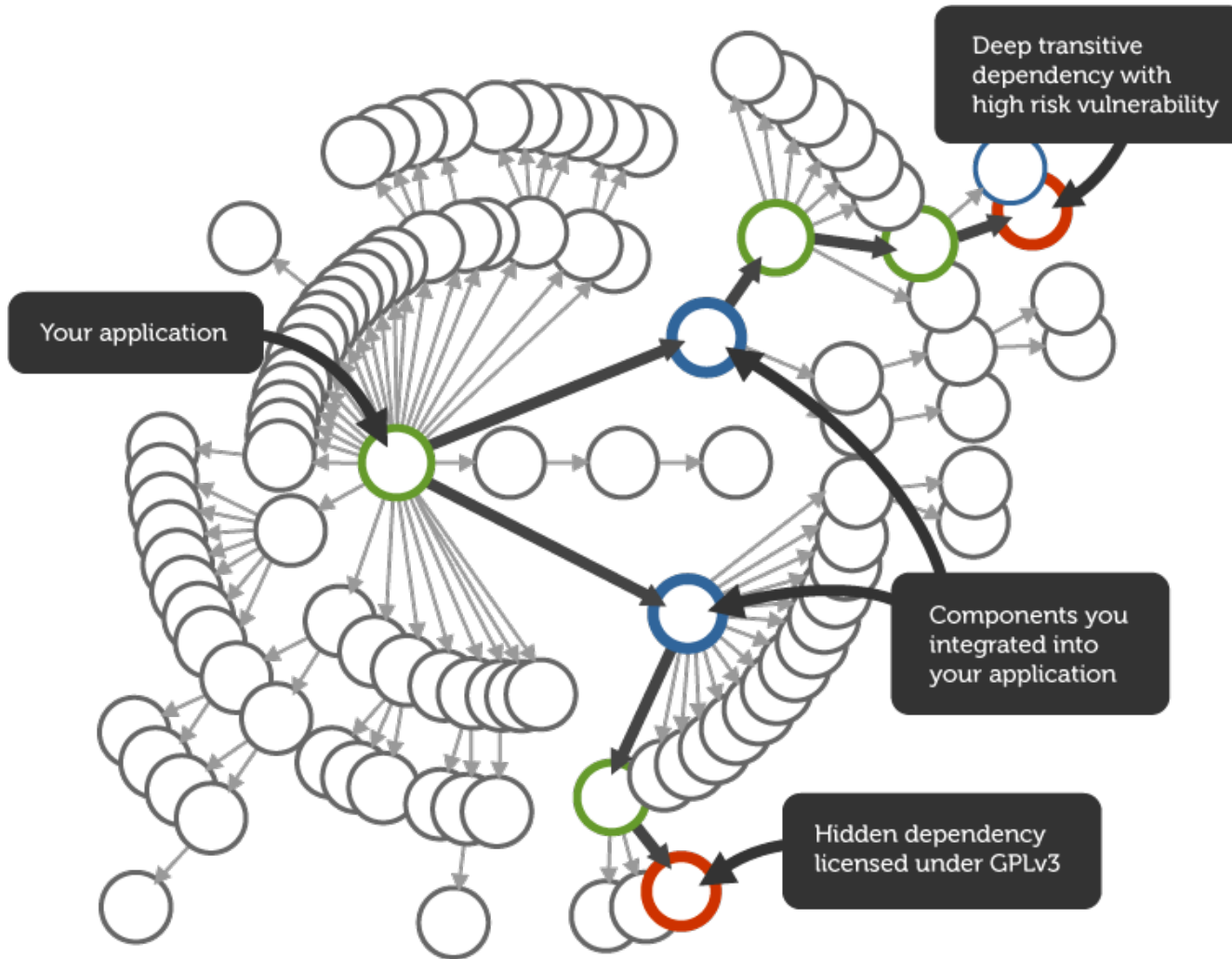
Capabilities (Without Application Licensing Restriction)	GPL (Linux)	Dual-GPL (MySQL)	LGPL/MPL (OpenOffice, Firefox)	Apache/BSD (Apache, FreeBSD)
1) Download	✓	✓	✓	✓
2) Evaluate	✓	✓	✓	✓
3) Deploy	✓	✓	✓	✓
4) Redistribute	✗ ¹	✓ ³	✓	✓
5) Modify	✗ ²	✗ ²	✗ ²	✓ ⁴

- 1) Application needs to be licensed under GPL if redistributed with the GPL asset.
- 2) Library code modifications need to be licensed under the same license as the originating asset.
- 3) Usually requires a commercial license from the copyright holder.
- 4) Although much more permissive than an OSI license, some BSD based licenses, such as Apache V2, still have some copyleft materials.

Use of OSS in Corporates

- 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
- 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.
- Secrecy of your source is important:
 - 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
 - Identify necessary control points.

OSS – License Dependencies



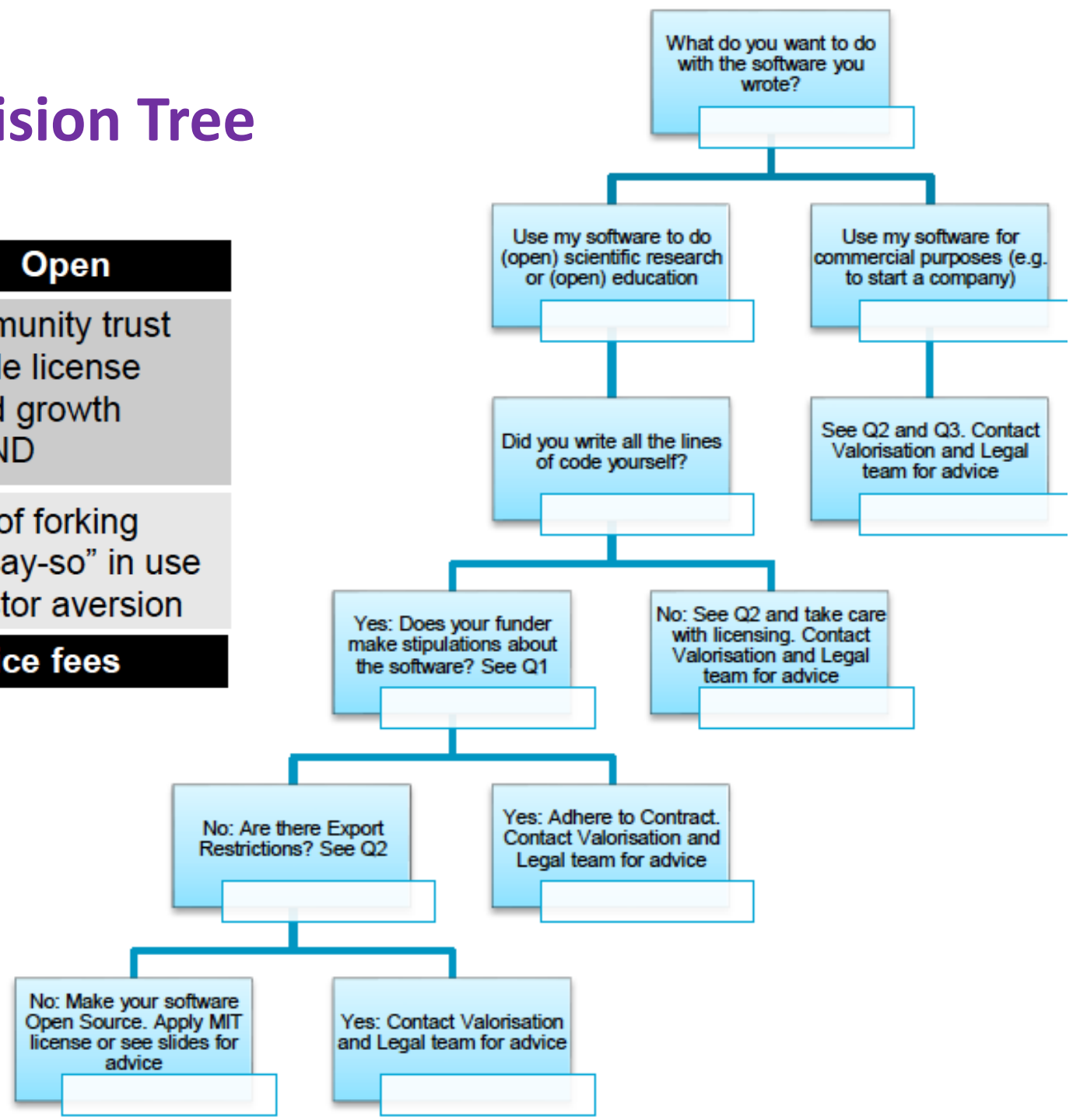
OSS – License Dependencies

Rights abandoned ↑		Zero cost	Distribution allowed	No usage restrictions	Source code available	Source code modifications	Linking with proprietary work	Derivative work can be proprietary	Can be relicensed by anyone	OSS license examples	
										GPL compatible	Not GPL compatible (reason)
OSS	Public domain	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		
	Non-copyleft (permissive)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	BSD mod MIT/X11 Apache v2 AL v2	BSD orig (advertising) AL V1 (patent termination)
	Weak copyleft	Yes	Yes	Yes	Yes	Yes	Yes	No	No	L-GPL	MPL (additional restrictions**) NPL (use of code in Netscape) SISL (minor details) SPL (like MPL) IBM CPL (choice of law) EPL (patent lawsuit language)
	Copyleft (restrictive)	Yes	Yes	Yes	Yes	Yes	No	No	No	G P L	
Rights reserved ↓	Freeware	Yes	Yes	Yes	No *	No	N/A	N/A	No		
	Proprietary	No	No	No	No *	No	N/A	N/A	No		

* Except under special licensing conditions - ** Provision in v1.1 to allow alternative license choice

Decision Tree

Type	Proprietary	Open
Positives	Protection from competitors Control Possibility to differentiate (e.g. student/academic)	Community trust Simple license Rapid growth FRAND
Negatives	Complex licenses No/small community Need to enforce	Risk of forking No “say-so” in use Investor aversion
Revenue	License fees	Service fees



Compliance Detection Tools



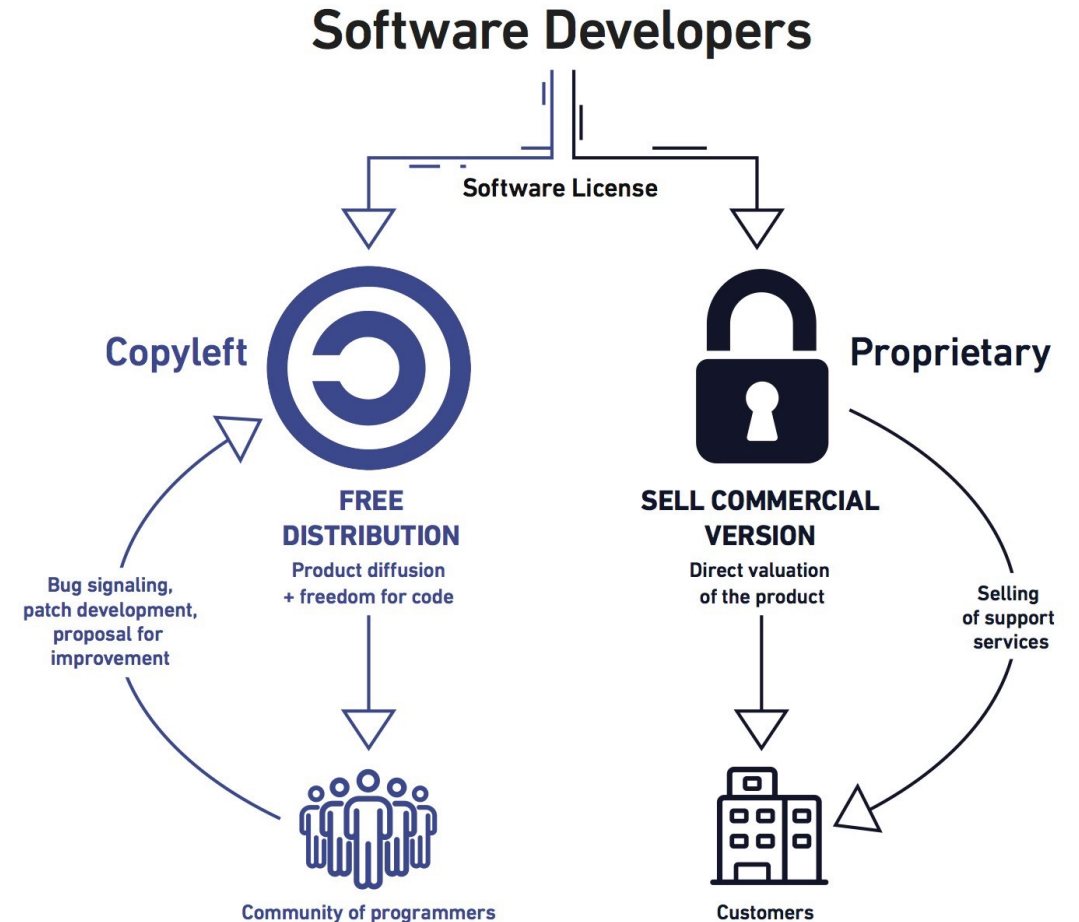
Multifactor open source discovery

- Dependency analysis
- File system scanning
- Snippet matching
- Binary analysis



Dual Licenses

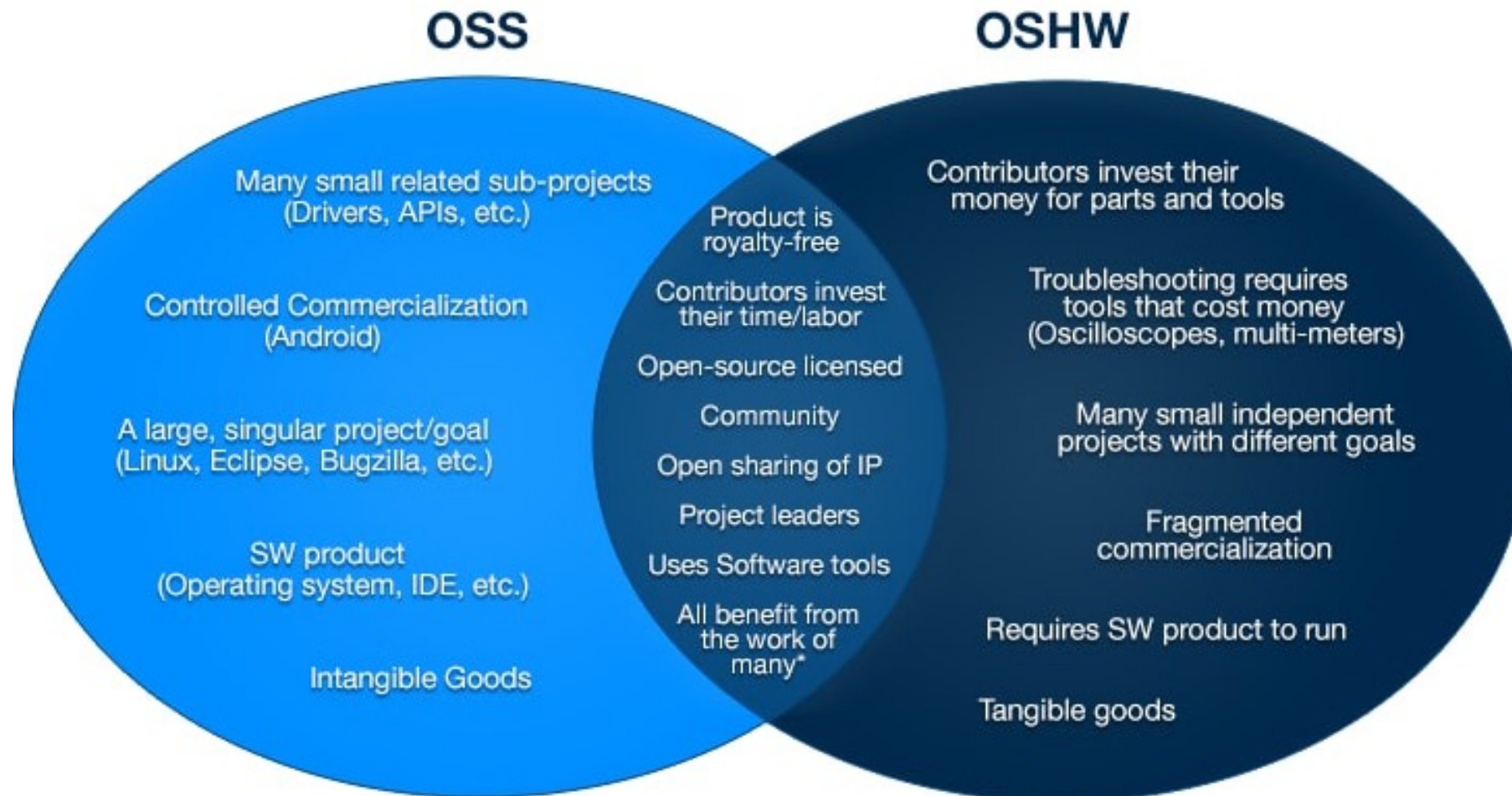
- Sometimes software is licensed under dual licenses.
- The copyright-holder can issue whatever license they want, as long as they hold the copyright to the whole source.
- Common dual licensing: A restrictive license (GPL or similar) that most commercial companies try to avoid, and a commercial license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



Open Source Hardware

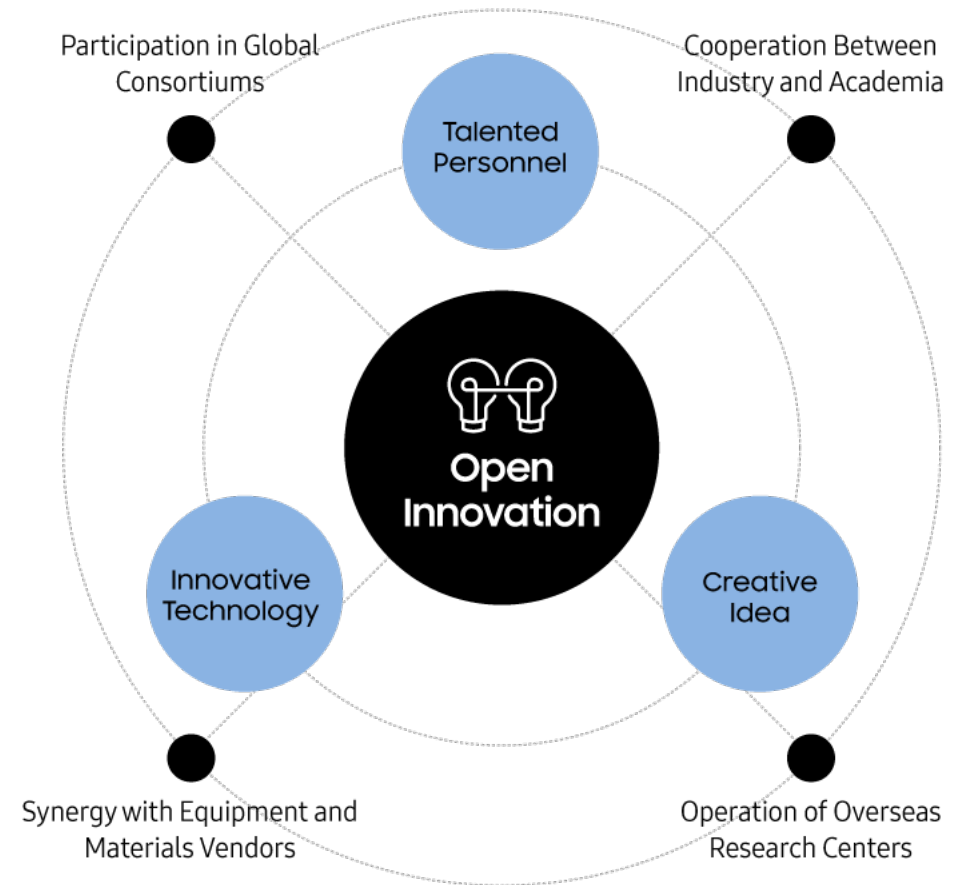


- Not only software can be open sourced – any creative artifact, including art, literature and even hardware can be open sourced.
- A very common license for non-Software are the Creative Commons licenses.
- A 'configurable' license. <<http://creativecommons.org/>>



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
- 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).
- “Open innovation” generally does not mean that this traditional role of IPRs becomes less important.
- The role of well-defined, affordable and enforceable IPRs in an economy is to provide incentives to invest in innovation, support the emergence of markets for intellectual assets, and encourage the disclosure and thereby the dissemination of inventions and the diffusion of innovation.



Data Analysis & Platforms



Databases / Data warehousing



In-Memory Computing



ERP BI Solutions

Open Source Business Intelligence Solutions for ERP



Business Intelligence



Data Mining



Big Data search



Multivalue database



Programming



Data aggregation



KeyValue



Document Store



Graph databases



Operational



Social



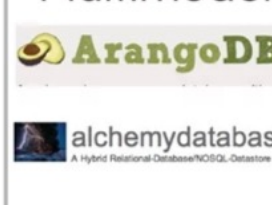
Multidimensional



Object databases



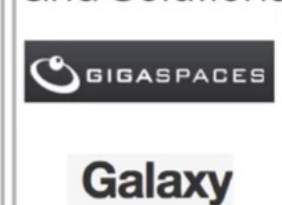
Multimodel



XML Databases



Grid Solutions



thank
you



NALSAR University of Law

Justice City, Shameerpet, Medchal District, Hyderabad - 500101

Annexure

A Copyleft license: You can use, modify and distribute the license under the same or equivalent license, but you should also share the source code.

Berkely Software Distribution license (BSD): You can combine the software with your proprietary software and release it under a proprietary license, but you have to retain the BSD license text and notices. This license may also have one or two more clauses about using the author's name or advertising.

Gnu Public License (GPL): You can use, modify and distribute the software for free or for a fee, but you have to also distribute the source code with it, and if the software is combined with other software, that software will also have to be released as GPL, unless it is not an aggregate (it does not interact with the GPL software)

Lesser Gnu Public License (LGPL): Same as GPL, but you're allowed to link your proprietary software with LGPL software and release it under your terms. But there are some conditions under which you can do so.

Affero Gnu Public License (AGPL): Same as GPL, but it additionally says that your software is considered as having been distributed, if someone else is able to use it (like if you keep it on a server and they can run it from the server).

Massachusetts Institute of Technology license (MIT): You are allowed to use, modify and distribute copies of the software as long as you allow the same rights to the person to whom you distribute the software

Apache License: You are allowed to use, modify and distribute copies of the software, and you can add your own copyright statement to the changes you make. You can also provide additional or different license terms and conditions for use, reproduction or distribution of your modifications.

Mozilla Public License (MPL): You can use, modify, distribute and sell the software, provided you also give the recipient the source code. You can also sub-license the modified work, provided you don't restrict the recipients rights to the source code.

Boost Software License: You can use the software and create derivative works out of it provided you include the license text and disclaimer in all derivative works (no need to include the text if the derivative work is object code).

Expat License: You can use the software for free, use it, modify it, sublicense it and sell it and allow those who receive it to do the same, provided the license text is included in all or a reasonable number of sections of the software.

X11 License: You can use the software for free, use it, modify it, sublicense it and sell it and allow those who receive it to do the same, provided the license text is included in all or a reasonable number of sections of the software. Additionally, you cannot use the name of X Consortium to advertise or promote the sale of the software without their written permission.

Zlib License: You can use the software freely, even commercially. You can modify and redistribute it provided you do not claim to have written the original software. If you're using this software, an acknowledgement in the product documentation would be nice, but not explicitly required. If you alter the source code, it should be explicitly mentioned that it is altered source code. The license text should be present in all unmodified and modified copies of the source code.

Common Development and Distribution License (CDDL): You can use, modify and sublicense the software, even as part of a larger software. You have to make the source code available to those you distribute the software to. Any modifications you make are also covered under this license, and you should mention that you modified it. You aren't allowed to modify any of the other copyright notices, and you can't alter the terms of the license in a way that restricts the rights of those who receive it. You can charge a fee if you offer warranty, support or liability, but none of the other contributors of the software can be held responsible for it.

Common Public Attribution License (CPAL): If you're a developer, you can use, modify, sublicense and use the software as part of a larger work. If you're distributing or contributing to the software, you can use, modify, sublicense and sell the software from the date the contributor who gave you the software first makes commercial use of the software. The software can be distributed only under the CPAL and all copies of the software should contain the license text and if you've modified the software, you should provide documentation of what you modified and include the name of the initial developer. You also have to make the source code available to the recipient. You can distribute the executable under your own license (provided you also comply with CPAL), but the accompanying source code should not restrict the recipient's rights provided by the CPAL. You can combine the CPAL code with other code but have to ensure that the code under CPAL remains covered by CPAL.

Common Public License (CPL): You can use, modify, sublicense, sell the software. There is no guarantee on infringement of intellectual property rights. You can distribute the program in object code form under your own license provided you comply with CPL, not hold any of the contributors liable for any problems, specifically mention that which of the license terms are yours and specify how the recipient can obtain the source code. If you distribute the source code, it has to be under the CPL and you have to supply a copy of the CPL along with it. You cannot alter/remove any copyright notices and if you've contributed to the source code, you have to mention it.

Eclipse Public License (EPL): You can reproduce, distribute, sublicense the software in source code and object code form (provided it complies with the EPL, has a disclaimer of warranties and states that the source code is available and where it is available from). Every contributor to the software should mention which portion of the software they contributed to. You can distribute the software commercially so long as other contributors to the software are not liable for any problems and if you agree to defend the other contributors from losses, damages and costs arising from legal action. Unless explicitly stated, the EPL offers no warranties or liabilities.

License type	Licenses	(1) Collection of royalties is allowed when a program is distributed	(2) The source code is provided when the original program is redistributed	(3) The source code must be provided when a modification is distributed	(4) A modification should be distributed under the same license as the original program	(5) A fee higher than the distribution cost can be collected when a program is distributed without the source code	(6) The program can be sublicensed	(7) A documentation must be provided with a documentation	(8) A fee higher than the distribution cost can be collected when a program is distributed with the source code	Degree of openness
GPL type	GPL	No(+)	Yes(+)	Yes(+)	Yes(+)	No(+)	No(+)	Yes(+)	Yes(-)	1
	LGPL	No(+)	Yes(+)	Yes(+)	Yes(+)	No(+)	No(+)	Yes(+)	Yes(-)	1
Others	MPL	No(+)	Yes(+)	Yes(+)	Yes(+)	No(+) (source code is always redistributed)	Yes(-)	Yes(+)	Yes(-)	2
	QPL	No(+)	Yes(+)	Yes(+)	No(-)	No(+)	No(+)	No(-)	Yes(-)	3
	CPL	No(+)	Yes(+)	Yes(+)	No(-)	No(+)	Yes(-)	No(-)	Yes(-)	4
	Artistic	No(+)	Yes(+)	No(-)	No(-)	No(+) (source code is always Redistributed)	No(+)	Yes(+)	No(+)	5
BSD type	Apache v.2.0	Yes(-)	No(-)	No(-)	No(-)	Yes(-)	No(+)	Yes(+)	Yes(-)	6
	Zlib	Yes(-)	No(-)	No(-)	No(-)	Yes(-)	No(+)	Yes(+)	Yes(-)	6
	Apache v.1.1	Yes(-)	No(-)	No(-)	No(-)	Yes(-)	No(+)	No(-)	Yes(-)	7
	BSD	Yes(-)	No(-)	No(-)	No(-)	Yes(-)	No(+)	No(-)	Yes(-)	7
	MIT	Yes(-)	No(-)	No(-)	No(-)	Yes(-)	Yes(-)	No(-)	Yes(-)	8