Open Source & Licensing

Rogério de Lemos

- After: Budi Arief, Anna Jordanous, Eerke Boiten, Tom Garner, David Chadwick, Kris Welsh, Gwen van der Velden

Consortium

C3ISP
Collaborative and Confidential Information Sharing and Analysis for Cyber Protection

Consiglio Nazionale delle Ricerche

cea

BT

Hewlett Packard Enterprise

University of Kent

SAP

CERT nazionale Italia

CATAPULT Digital

3D Repo

CHINO.IO

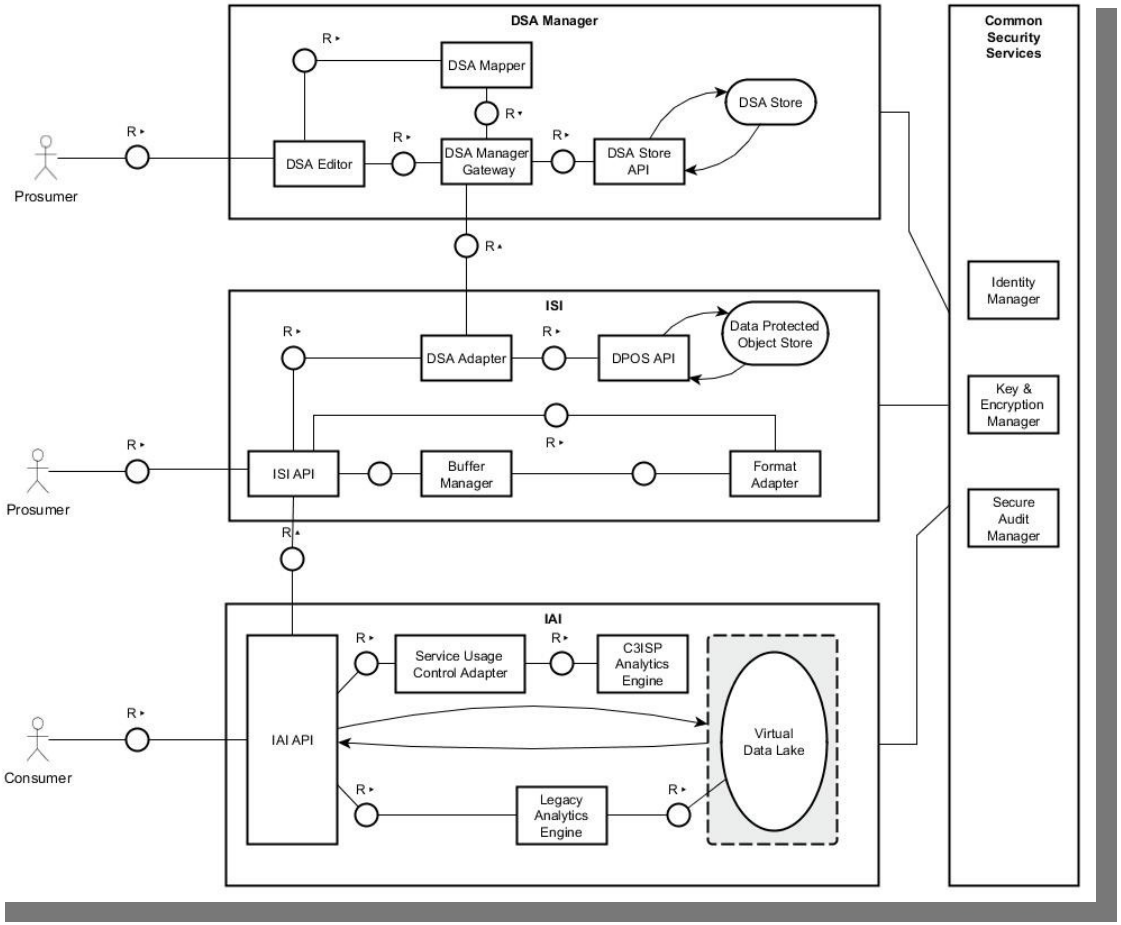
GRIDPOCKET
PERSONAL SMARTGRID SOLUTIONS

Overall Architecture

C3ISP
Collaborative and Confidential Information Sharing and Analysis for Cyber Protection

- 3 physical servers
- 10 VMs
- 22 micro-services
- 3 OSS packages
- 120+ Open API REST services
- ~300,000 lines of code (Java)

All deployed by a Continuous Delivery engine



Today's lecture: Open Source

- What is Open Source?
- Open Source: Good, bad and the ugly…
- How do we make our code open source?
  - What's allowed – and what's not
  - Licences

Open Source = making software freely available

- “The term “open source” frequently refers to a software development process that relies on the contributions of geographically dispersed developers via the Internet. ”

The Many Meanings of Open Source, Gacek and Arief 2004 – see Moodle

Open Source = making software freely available

- Is a software distribution *open source* ?
- **Open Source Definition (OSD)**: 3 *main* criteria for determining whether software is open source
  - The ability to distribute the software freely
  - The source code's availability
  - The right to create derived works through modification

(http://www.opensource.org/docs/definition.html)

  - Note: The OSD contains 7 more criteria, dealing with licensing issues / the "no discrimination"* caveat

(* anyone may use this software, for any field of endeavour)

The Many Meanings of Open Source, Gacek and Arief 2004 – see Moodle

What does ‘free’ mean?

Yay! Free Beer!

- The Open Source crowd frequently talk about pieces of software being “Free as in Speech” and/or “Free as in Beer”

“Gecko and a beer, Kona, Hawaii” - By SteveD.

Free as in speech

- Means that you can modify and tweak the software yourself, and that there are no licensing clauses or copyright risks to run foul of when doing so (libre)
  - All software considered Open Source is Free as in Speech

"More Press Scrum" - By noodlepie

Free as in speech (contd.)

[https://www.gnu.org/philosophy/free-sw.html]

More precisely, free software means users of a program have the four essential freedoms:

- **Freedom 0** - freedom to run the program as you wish, for any purpose
- **Freedom 1** - freedom to study how the program works, and adapt it to your needs
  - Access to the source code is a precondition for this
- **Freedom 2** - freedom to redistribute copies so you can help your neighbour
- **Freedom 3** - freedom to improve the program, and release your improvements to the public, so that the whole community benefits
  - Access to the source code is a precondition for this

Free as in beer

- Means that the software is provided free of charge (gratis)
  - There's *plenty* of software that's provided free of charge, that isn't Open Source.

"Adnams" - By touring_fishman

Free as in beer (contd.)

- Not comparable to open source
- Free of charge
  - But….

Some history…

- 1983: Richard Stallman Founds GNU
- 1985: Richard Stallman Founds the FSF
- 1991: Linus Torvalds releases first version of Linux
- 1992: First Version of BSD not needing an AT&T license to run is released
- 1995: First Version of mySQL released
- 1996: First Version of Apache released
- 1998: Open Source Initiative (OSI) Founded

Today's lecture: Open Source

- What is Open Source?
- **Open Source: Good, bad and the ugly…**
- How do we make our code open source?
  - What's allowed – and what's not
  - Licences

Advantages of Open Source

- The Open Source movement is important for several reasons
  - Means we can still work with obsolete file formats, or obsolete hardware, if sufficiently motivated
  - Means developers can work together on pieces of software that wouldn't be profitable for an organisation to underpin
  - Provides a political figurehead when campaigning against developments harmful to computer users' freedoms
    - E.g. Trusted Computing or ("Treacherous Computing")
      - to enforce DRM Software Patents

Downsides of Open Source

- Just because a program is open source does not guarantee security
  - "People have to actually review the code"

    [Dan Wheeler, open source security writer]
- Many open source projects have lack of support to users
- There is no-one to hold accountable in the event that there is an issue
- Many open source tools are not enterprise-ready without some significant adaptation
  - Some open source providers and third parties sell commercial ready-to-use binaries with support packages to overcome the above limitations (and improve their bottom lines)

What the (security) experts say

- Bruce Schneier
  - "demand open source code for anything related to security"
- Vincent Rijmen, one of the developers of the Advanced Encryption Standard (AES) believes that Linux is superior to proprietary OSs, because security vulnerabilities are easier to spot and fix
  - "Not only because more people can look at it, but, more importantly, because the model forces people to write more clear code, and to adhere to standards. This in turn facilitates security review."
- Elias Levy, Bugtraq moderator
  - "Open Source Software certainly does have the potential to be more secure than its closed source counterpart. But make no mistake, simply being open source is no guarantee of security."

What the experts say (2)

- Whitfield Diffie, co-inventor of public-key cryptography
  - “It's simply unrealistic to depend on secrecy for security in computer software. You may be able to keep the exact workings of the program out of general circulation, but can you prevent the code from being reverse-engineered by serious opponents? Probably not.”

What the experts say (3)

- Eric S. Raymond author of “The Cathedral and the Bazaar”
  - “Closed source leads not to true security but to a false sense of security. You don't know what's in there, you can't verify it, you can't check the assumptions or honesty of the people who wrote it. Closed source means only the bad guys get to find the security holes. With open source, good guys get to look for them too.”

What the experts say (4)

- Michael H. Warfield, senior researcher with Internet Security Systems Inc., UNIX systems engineer and consultant
  - “In an open source project, to make a mistake and have it known to the entire development community and your friends is mortifying in the extreme. That last moment before hitting the Enter key -- to commit a change or send a patch out into the cold cruel world of your peers -- is the longest moment imaginable.”

Today's lecture: Open Source

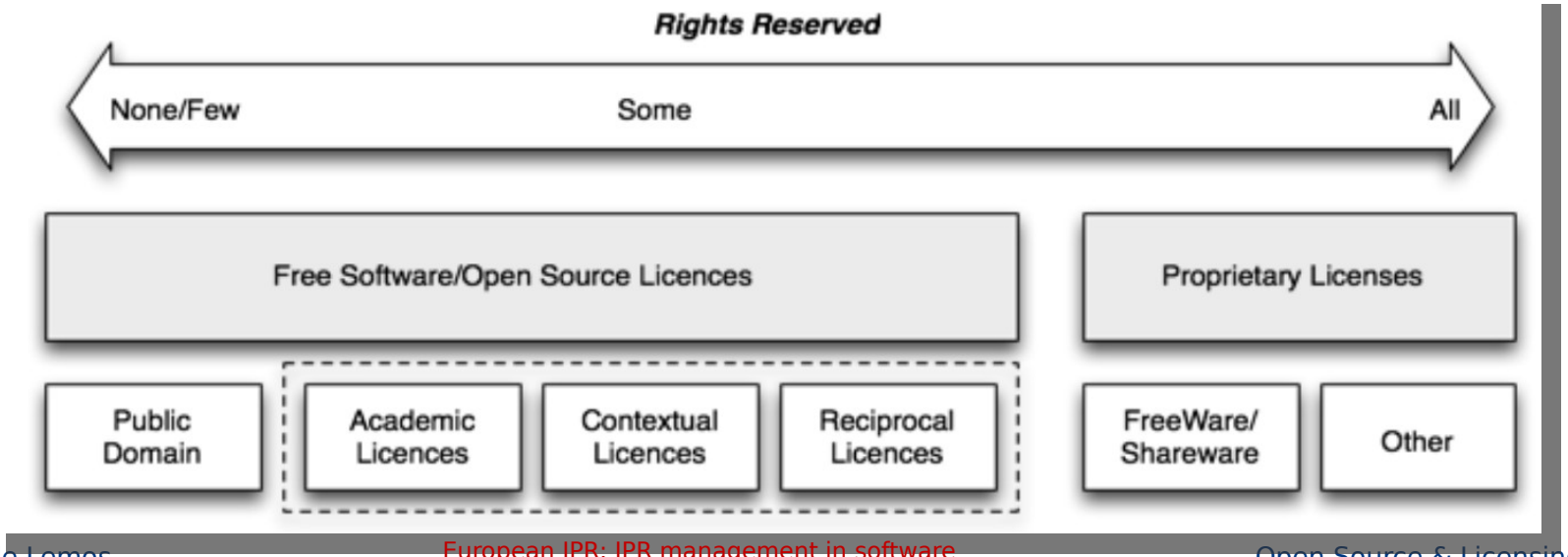
- What is Open Source?
- Open Source: Good, bad and the ugly…
- **How do we make our code open source?**
  - What's allowed – and what's not
  - Licences

Software Licensing

- IP grants a set of exclusive rights to the owner of intangible assets
  - e.g., inventions, brands or designs
- Exclusive rights allows the owner
  - to exclude others from using its IP assets
  - to grant third parties the rights for exploitation
- **Licensing** is a fundamental means of exploiting IPR
  - to share existing source codes in order to improve quality while lowering investment needs without users having to reinvent the wheel

Software Licensing

- Licensing is a contract by which a **licensor** grants a **licensee** an authorisation to use an identified asset under certain conditions
  - licensors can either be the owners of the IPR or act under a mandate from the actual owners
  - software as a whole or just some components



University of Kent

Software Licensing

- Free software licence specificities are defined by the **Free Software Foundation**
  - relies on philosophical principles such as the notion of freedom
  - editor and owner of some of the most widely used licences such as the GNU General Public Licence (GPL)
- Open Source licence specificities are defined by the **Open Source Initiative**
  - refers to a more business- oriented approach
- Both
  - both share similar objectives
  - well-established international reputation

Making software Open Source by using Licences

Freedom to run

- to use the software for any purpose, with no restrictions

Freedom to study and change

- to change the software to suit your needs

Freedom to redistribute

- to redistribute unmodified copies, without restrictions

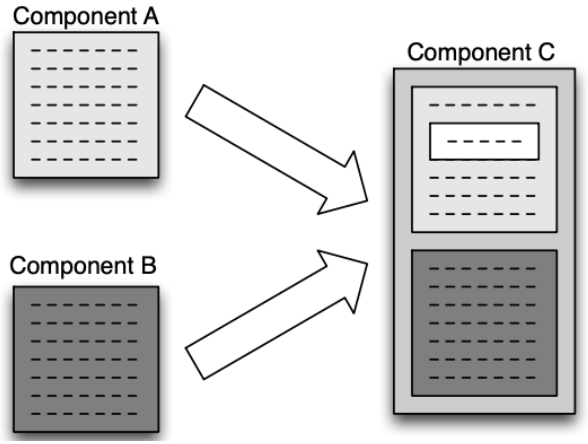
Freedom to distribute

- to distribute modified versions
  - known as derivative or larger works

Adapted from http://www.gnu.org/

Derivative vs Larger Work

- Mergers and modifications of third party code can have a strong impact on licensing options
  - **derivative creations** based on pre-existing components, and will include a code modification
  - **larger works** based on pre-existing components, but they are made up of the use of unmodified original

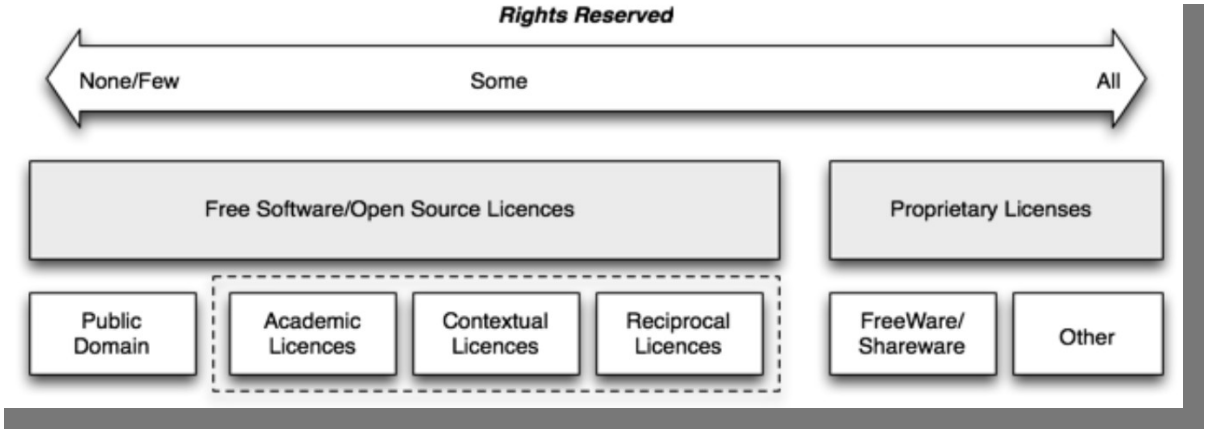


- component C is both a derivative work of component A and a larger work of component B
  - modified version of component A
  - unmodified component B

Legal Basics

- Open Source licences rely on copyright to work
- They present terms under which software may be copied
  - If the terms are breached (e.g. there's no permission for copying), then person doing the copying breaches copyright
- Because copyright is usually used to *prevent* sharing, and some Open Source licences *promote* or even *force* it, they're a bit of a hack
  - So we nickname them *copyleft* licences

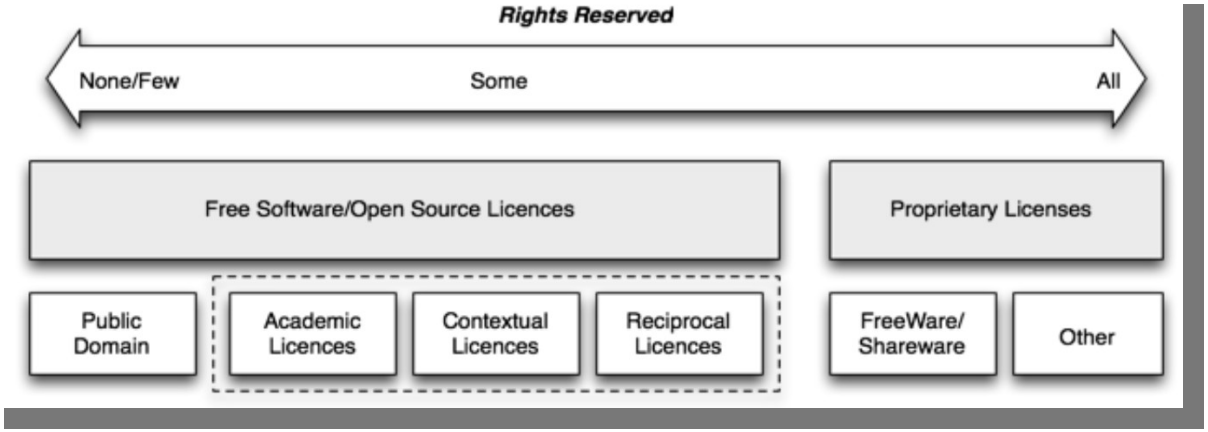
Categories of Free and Open Source Software Licences



Academic Licences (eg, MIT Licences)

- licensees are allowed to run, modify, and distribute derivatives of the code under licence with no restrictions whatsoever
  - licences over derivative works can in turn not only be granted to sub-licensees but also lead to new licensing terms, including proprietary ones
- are extremely open
  - an excellent choice for widespread dissemination of the code under licence where patrimonial restrictions or legal lock-ins on licensees might be at issue

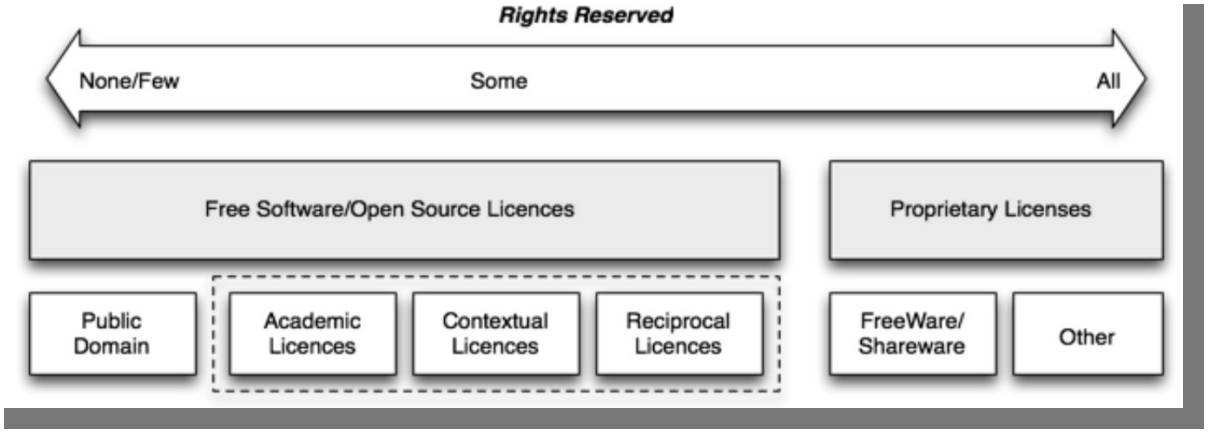
Categories of Free and Open Source Software Licences



Reciprocal Licences (eg, GPL v3)

- licensees are free to run, modify, and distribute derivatives based on the code under licence, as long as the distributed derivatives or larger works are themselves distributed under the same licence
- usually called **copyleft** licences
- major advantage
  - secure a common investment as no derivative or larger works can be licensed under another licence

Categories of Free and Open Source Software Licences



Contextual Licences (eg, L-GPL v2.1)

- a contextual technical trigger where reciprocal licensing obligations are likely to impact derivative or larger works
  - larger works are not restricted by the original licence
  - derivative product has to be released under same licence
- compatible with multiple licensing schemes

Proprietary Software Licences

- All licences which are not Free and Open Source licences will be called **proprietary licences**
  - most proprietary licences require a financial contribution
  - exception, are freeware and shareware
    - **freeware** licences imply the product is available free of charge, but generally these licences prohibit any code modification and focus on price
    - **shareware** licences imply the product might be freely used for a limited period of time or with limited functionalities
- Common points in proprietary licences
  - prohibition of software modifications, a strict control of software use conditions, and no access to the source code

Popular Open Source Licences

- GNU General Public License (GPL) (version 2 and version 3)
- GNU Lesser General Public License (LGPL) (also versions 2 and 3)
- BSD (and MIT) License
- Apache License

- http://wiki.civiccommons.org/Choosing_a_License

Copyright

Within business/industry:

- teaching & examination material
- reports, articles, essays, books
- presentations
- questionnaires
- translations
- architectural design
- computer programmes
- music, films, plays,
- paintings, photographs, graphics
- websites, marketing material
- databases

MIT

If you don't care about restrictions or what happens downstream

GPL

If you want to guard against your code ending up in proprietary code

Apache

If you want to encourage commercial adoption of your code

ICE
INNOVATION
CREATIVITY
ENTERPRISE

University of Kent

The GPL

- We're going to start with version 2, because that's still the most popular Open Source licence
- A few terms must be met when you distribute software licenced under GPLv2:
  - Copyright, licensing and disclaimer of warranty information must be included, in the same form you received it
  - Any modified files must state that you changed them (+ date)
  - Distribute the source code along with the binary (…or a written offer, good for at least 3 years, to do so for no more than the cost of the physical distribution of the source)
- There's also a warranty disclaimer in there…

Recursive (“viral”) Licence

- The GPL is a *recursive* licence
- Any software based upon (a derivative work of) GPL-licenced software must itself be GPL licenced
- For a period, big software companies were *terrified* that an errant programmer might accidentally cause their huge product to have to be open sourced
- This hasn’t really happened – and if it did, the big company could always negotiate an alternative licence to the code that has been covered
- …Unless the original developer wants to play hardball

The GPL (Version 3)

- The main restrictions are the same as v2
- Addresses 3 main problems:
  - "Tivoisation" & "Technical Protection Measures"
  - Software Patents
  - Unintended Incompatibilities between the GPL & other Open Source Licenses – preventing crosslinking
- "or later versions" – or not

The LGPL

- The LGPL is designed for use on code libraries
- If my program used a GPL library, that would be a derivative work of the library, and would need to be released under the terms of the GPL
- The LGPL says that new versions of the library must also be LGPL, but that a work that *uses* the library, can be licenced however you want

Other GPL Variants

- Affero GPL (AGPL) is based on the GPL v2 or v3, and adds a single extra clause:
  - If you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network [...] an opportunity to receive the [...] Source of your version [...] from a network server at no charge...
- There's also the Gnu Free Documentation License, which is for docs rather than programs

BSD Licenses

- BSD Licenses are simple:
  - Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
    - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
    - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

    …<LARGE WARRANTY DISCLAIMER IN ALL CAPS>…

BSD Licenses (contd.)

- That was what's known as a "two clause" BSD License, because of the 2 sub-bullets
- There's also a "three clause" variant, adding:
  - 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

MIT License

- MIT license is very similar to the BSD license:
  - 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.

    <ANOTHER WARRANTY DISCLAIMER IN CAPS>

Apache License

- It is similar to a BSD license, with the notable exception of a “patent retaliation” clause
- Explicitly (rather than implicitly) grants a licence to use any patents required to use the software
- If you sue us for patent infringement, then we revoke your licence to use the patents granted to you by this licence, so we can sue you back…

Heterogeneity/Proliferation

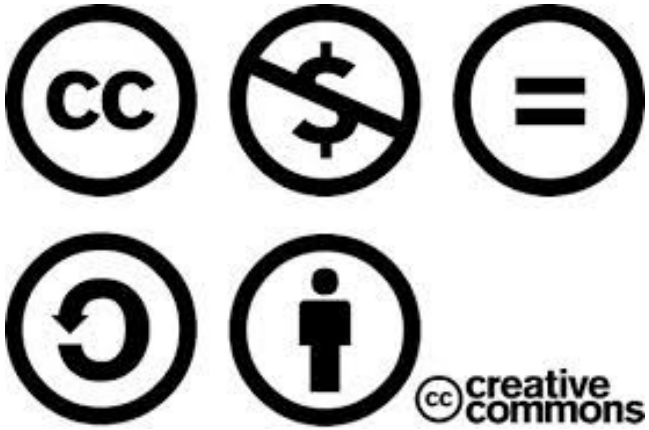
- Not all Open Source Licenses are compatible…
- So if I make a program based off some GPLv2 code, and some Yahoo Public Licensed code, I can’t legally distribute it because I can’t simultaneously adhere to both licences
- This situation doesn’t fit well with the earlier 4 freedoms

Not program source: use Creative Commons

- Intro film:

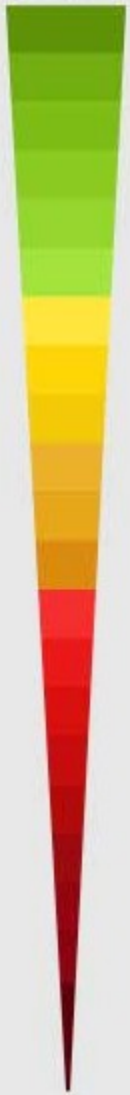
  http://mirrors.creativecommons.org/movingimages/webm/WannaWorkTogether_480p.webm

- Attributes:



- A handy leaflet on CC licenses
  - http://wiki.creativecommons.org/images/0/01/6licenses-folded.pdf

LICENSES



ATTRIBUTION

CC BY

This license lets you distribute, remix, tweak, and build upon the original work, even commercially, as long as you credit the original creation. This is the most accommodating of licenses offered.

ATTRIBUTION-SHAREALIKE

CC BY-SA

This license lets you remix, tweak, and build upon the original work even for commercial purposes, as long as you credit the original work and license your new creations under the identical terms. This license is often compared to "copyleft" free and open source software licenses. All new works based on the work should carry the same license, so any derivatives will also allow commercial use. This is the license used by Wikipedia.

ATTRIBUTION-NODERIVS

CC BY-ND

This license allows for redistribution, commercial and non-commercial, as long as it is passed along unchanged and in whole, with credit to the original work.

ATTRIBUTION-NONCOMMERCIAL

CC BY-NC

This license lets you remix, tweak, and build upon the original work non-commercially. Your new works must be non-commercial and acknowledge the original work, but you don't have to license your derivative works on the same terms.

ATTRIBUTION-NONCOMMERCIAL-SHAREALIKE

CC BY-NC-SA

This license lets you remix, tweak, and build upon the original work non-commercially, as long as you credit the original work and license your new creations under the identical terms.

ATTRIBUTION-NONCOMMERCIAL-NODERIVS

CC BY-NC-ND

This license is the most restrictive of the six main licenses, only allowing you to download the original work and share it with others as long as you credit the original work. You can't change the original work in any way or use it commercially.

Today's lecture: Open Source

- What is Open Source?
- Open Source: Good, bad and the ugly…
- How do we make our code open source?
  - What's allowed – and what's not
  - Licences

Open Source licences are something that anyone writing software should have at least a rudimentary understanding of.

We've gone through the main Open Source licences, so you should be able to answer questions on which to pick when creating code, and what you can and can't do when using it.