

Mineure OSS @ EPITA

Session 4 (draft)

Dernières sessions

- Unix et sa "philosophie"
- Emergence du libre et de Linux
- Définitions du logiciel libre et de l'open source
- Textes fondateurs
- Economie du logiciel

Cette session (et la suivante?)

- (Économie du logiciel - suite)
- Droit du logiciel
- Droit du logiciel libre

Sources

- Slides de Stefano Zacchiroli (dérivés de ceux de Roberto Di Cosmo)
- Slides de Dirk Riehle
- Slides de François Pellegrini

Prélude: révolution numérique et économie de l'immatériel

La révolution numérique

Une longue histoire, qui commence à peine

Révolutions de la connaissance

- Révolution : « Changement brusque, d'ordre économique, moral, culturel, qui se produit dans une société »
- Historiquement, la révolution numérique fait suite à deux révolutions précédentes dans le champ de la connaissance :
 - Révolution de l'écriture
 - Révolution de l'imprimerie

Révolution de l'écriture

- Issue de la fixation de l'information sur un support mobile
- Crédit des premières cités-États et des premiers grands empires
 - Invention de l'administration

Révolution de l'imprimerie (1)

- Issue de la mécanisation de la copie de l'information sur un support
- Du temps où les livres étaient copiés à la main :
 - La connaissance circulait peu
 - Quelques rares lettrés
 - De bibliothèque en bibliothèque, souvent au sein des monastères
 - Contrôle fort sur le contenu

Révolution de l'imprimerie (2)

- Conséquences de la mécanisation de la copie :
 - Faible coût d'accès à la connaissance
 - Diffusion massive d'idées nouvelles
 - Parfois contraires à l'ordre établi
- Ferment de réformes
 - Réforme religieuse
 - Imprimeurs en Suisse et aux Pays-Bas
 - Siècle des Lumières
 - Révolution industrielle
 - Révolutions républicaines (entre 1750-1790)
 - Printemps des peuples

Révolution de l'imprimerie (3)

- Pyramidalisation de la diffusion de l'information
 - Quelques-uns décident de ce que la multitude consomme de façon passive
 - Grands média : imprimerie, radio, télévision
- Construction des États-Nations modernes
 - Normalisation par la langue
 - Uniformisation par la copie d'un « parler de référence »
 - Normalisation par la technique
 - Claviers de machines à écrire, écartement des rails, etc.

Révolution industrielle (1)

- Son moteur et objet est la machine
- Avant les machines, il existait des outils
 - Permettaient de bouger la matière
 - Mus par la force animale
- Les machines consomment de l'énergie fossile pour augmenter notre capacité d'action sur la matière
 - Charbon et acier → premières machines
 - Premières machines → plus de charbon et d'acier
 - Réaction en chaîne

Révolution industrielle (2)

- Deux siècles après, nous vivons dans un monde dans lequel la majorité de nos actions physiques sont déléguées à des machines :
 - Nous déplacer (avions, automobiles, ascenseurs, etc.), laver notre linge, notre vaisselle, etc.

Révolution numérique (1)

- Issue de l'extraction de l'information de son support physique
 - « Numériser », c'est transformer en nombres
 - En « tas de 0 et de 1 »
- Utilisation d'outils pour extraire de l'information du monde physique et la coder sous forme numérique
- Support toujours nécessaire, mais pas unique

Révolution numérique (2)

- Rendue possible grâce à deux innovations majeures :
 - Le logiciel
 - L'Internet

Révolution numérique (3)

- Le logiciel est le premier outil mécanisé qui soit une extension de l'esprit humain et non de son corps
 - Révolution dans la façon de produire et traiter la connaissance

Révolution numérique (4)

- Le logiciel joue, pour la révolution numérique, le même rôle que la machine a joué pour la révolution industrielle
 - Il en est le moteur et l'objet
- Même effet de réaction en chaîne :
 - Production de logiciels → échange de plus d'idées
 - Échange de plus d'idées → production de plus de logiciels
- Délégation de nos processus intellectuels aux logiciels

Révolution numérique (5)

- Les outils numériques dé-pyramidalisent et ré-horizontalisent les échanges
 - Chacun peut échanger avec chacun ou tous
- Bouleverse le modèle économique de la production et de la diffusion des savoirs et des œuvres numériques
- Crée de nouveaux outils et usages
 - Majorité de contenu auto-produit
 - Nous sommes tous des auteurs
 - Création collective de biens communs informationnels

Révolution numérique (6)

- Vastes bouleversements économiques et sociaux induits par l'usage des outils numériques
- Nombreuses questions ouvertes :
 - Gouvernance de l'Internet
 - Identité numérique et anonymat
 - Statut des données à caractère personnel et « droit à l'oubli »
 - ...

Quelques notions d'économie des biens immatériels

Copier n'est pas voler...

Économie des biens immatériels (1)

- L'économie des biens immatériels diffère fondamentalement de l'économie des biens matériels
- On ne peut pas penser la révolution numérique et les changements profonds qu'elle induit si l'on reste prisonnier des schémas de pensée du millénaire précédent

Économie des biens immatériels (2)

- À la différence des biens matériels, les biens immatériels sont non rivaux
 - On n'est pas en rivalité pour les consommer
 - On peut les partager sans s'appauvrir
 - On ne « donne » pas une idée, on la copie !
- La notion de « vol » n'est pas pertinente
 - Pas de « propriétaire »
 - Le terme « propriété intellectuelle » est intellectuellement erroné
 - C'est un oxymore
 - Différentes incriminations : contrefaçon, détournement de finalité, parasitisme, etc.

Économie des biens immatériels (3)

- Le coût de copie (coût marginal) des biens numériques est nul
- Coût marginal : coût de copie d'une unité supplémentaire d'un bien
 - Bien matériel : voiture : prototype → série
 - Bien immatériel : copie à coût marginal nul
 - Même si ordinateur et électricité sont des biens rivaux
 - Un bien numérique (logiciel, livre, musique) peut être distribué gratuitement dès le moment où son développement a été financé

Économie des biens immatériels (4)

- Les effets de réseau sont considérables
 - La valeur d'un produit augmente avec le nombre de personnes qui l'utilisent
 - Ex. : téléphone
- Exemple des réseaux sociaux
 - Rivalité par rapport au temps disponible
 - On tend à aller sur le réseau social où l'on espère avoir le plus d'interactions
 - Économie de l'attention

Économie des biens immatériels (5)

- Très grande volatilité du marché
 - Alors que le logiciel ne s'use pas...
 - Obsolescence très rapide
- Stratégies basées sur la création de « communautés »
 - D'usage, de création, etc.

Droit du logiciel et "Propriété intellectuelle"

Outline

- 1 Intellectual Property concepts
 - What is “Intellectual Property”?
 - Main branches of IP
 - Is “Intellectual Property” Property?
- 2 Copyright on software
 - General notions
 - Copyright in France
 - La notion de licence de logiciel
 - Reverse engineering
- 3 Patents & “software patents”
 - Definitions et histoire
 - Le brevet en Europe
 - Conditions de la brevétabilité
 - Le brevet logiciel en Europe
 - Le brevet en quelques chiffres
 - Le brevet en une analogie

QUIZ

Qui connaît l'expression "IANAL" ?

The importance of licenses

"I Am Not A Lawyer (IANAL) and I never read licenses... why should I care about licenses?"

The importance of licenses

"I Am Not A Lawyer (IANAL) and I never read licenses... why should I care about licenses?"

- Licenses provides terms of use of a work.
- Licenses enable the opportunity to “free” a work.
- “Free licenses” are not just ordinary licenses: they are also a declaration of principles, a social contract.

The importance of licenses

“I Am Not A Lawyer (IANAL) and I never read licenses... why should I care about licenses?”

- Licenses provides terms of use of a work.
- Licenses enable the opportunity to “free” a work.
- “Free licenses” are not just ordinary licenses: they are also a declaration of principles, a social contract.

Software licenses (Free or not) work within the limits imposed by copyright law.

But as to the effects that laws can have on coding-related activities, there is more than copyright:

- Tiny SCO group sued the huge IBM in 2005 put forward a cluster of complaints: trademarks, copyright infringements and theft of trade secrets...
- Software patents lawsuits.

WIPO: the World International Property Organization, UN agency

Definition (Intellectual Property according to WIPO)

Intellectual property (IP) refers to **creations of the mind**, such as inventions; literary and artistic works; designs; and symbols, names and images used in commerce.

IP is protected in law by, *for example*, **patents, copyright and trademarks**, which enable people to earn recognition or financial benefit from what they invent or create. By striking the right balance between the interests of innovators and the wider public interest, the IP system aims to foster an environment in which creativity and innovation can flourish.

<http://www.wipo.int/about-ip/en/>

Types of IP

There is no single “IP law”. Rather there are several bodies of law that, collectively, correspond to what is (improperly) called “IP”.

There is no single “IP law”. Rather there are several bodies of law that, collectively, correspond to what is (improperly) called “IP”.

IP is a broad concept that covers several types of **legally recognized rights**.

- IP rights are rights on **intangible things** (e.g., “ideas”).
- as expressed (e.g., “copyrights”),
- or as embodied in practical implementations (e.g., “patents”)

The set of protected IP rights vary across countries.

The most popular branches of “IP”, found in several countries, are:

- Patents
- Copyrights
- Trademarks
- Trade secrets

Exclusion Rights in Comparison

	Copyright	Patents	Trademarks
Form	Text, other specific expressions of non-trivial complexity	Abstract idea that represents a non-trivial industrial invention	Equivalence set of uniquely identifiable expressions
Purpose	To promote otherwise unprotectable artistic expression by protecting it	To spur innovation by motivating the disclosure of trade secrets	To protect consumers from confusion by providing unique marks
Duration	Will eventually expire, but only after several decades	Will eventually expire, but only after a few decades (shorter than copyright)	Will not expire as long as it is enforced by trademark owner
Granting	Automatic by creation	By application to patent office	By application to trademark authority

- Patents are IP rights on **inventions** or technological developments, i.e., on devices or processes that perform a “useful” function.
- A patent grants the inventor a limited **monopoly** on the **manufacture, use, sale, or import** of the invention.
- The monopoly applies to others, even in case of **independent discovery**.
- In exchange of the grant of a patent, the inventor **fully disclose** the idea to the public as part of the **patent application**.
- For exploiting the invention, interested parties must obtain (and pay for, usually) a **license** from the patent owner.
- A patent actually only grants to the patent holder the **right to exclude** others from practicing the patented invention.

Patents — limitations

- duration: usually 20 years max
- due to their very strong protection, patents are designed to be hard to get (in theory):
 - ▶ usefulness
 - ▶ novelty / prior art
 - ▶ non obviousness
- Patents can be obtained only for “practical applications” of ideas; not for abstract/theoretical ideas.
- Philosophical, mathematical or scientific truths cannot be protected. (Why?)
 - ▶ Christmas Tree Stand Watering System (US Patent)
 - ▶ Einstein’s *discovery* of the relation $E = mc^2$ is unpatentable.
- Unfortunately, the distinction between *creation* (patentable) and *discovery* (unpatentable) is not always clear cut.

- usage limitations on the **expressions** (= “works”) of an idea¹
- designed to protect expressions such as: paintings, writings, architectural drawings, pictures
- copyrights can be used to limit actions like:
 - ▶ produce **copies** of the work and sell them
 - ▶ import/**export** the work
 - ▶ create **derivative** works
 - ▶ **perform** or display the work publicly
 - ▶ **transmit** or display over the media
 - ▶ sell or cede **copyright rights** to others
- apply to anything that shows individual **creative** expression
- copyrights attaches automatically to anything you create, as soon as it is “*fixed in a tangible medium of expression*”

Copyrights — limitations

- duration: usually in the 90–150 years range
- not as strong as patent protection. In particular: copyrights do not limit usage of **other expressions of the same idea**
 - ▶ extreme situation: 2 authors independently create 2 identical works; they both enjoy copyright on their respective creations
 - ▶ that wouldn't be possible for patents, where the first to obtain patent protection will “win”
- copyright laws have built-in protections to defend freedom of speech, freedom of citation, etc. (“*fair use*”).
- copyright (under common law) is applicable to software as *non-dramatic literary work*. Copyright does not protect purely functional expressions, but most source code has enough creativity in it to be eligible for copyright protection

- Trademarks protect the *association* of a word, phrase, symbol, or design used *with* the provider of goods or services
- Trademarks are meant to **protect consumers** from confusion as to whom they are buying a product or a service from
 - ▶ hence: using a mark to truthfully identify a product/service is permitted by trademarks (this is known as **nominative [fair] use**)
- Trademarks can be obtained *de facto* by mere use, but formal registration to a trademark registration system (e.g., the USPTO in the US) gives additional rights
- A fundamental notion in trademark enforcement is that of “brand strength”, the more a trademark is perceived as “strong”, the easier it is to defend (possibly limiting its use by others)
 - ▶ see: **trademark dilution** and **naked licensing**

Trademarks (2)

Trademarks are peculiar in comparison to other branches of “IP”:

- trademarks can **last forever**, as long as they are used and are strong on the market
 - ▶ e.g., how long will the McDonald's or Coca-Cola trademarks be around?
- trademark protections must **not overlap**: different trademark owners cannot use the same mark (i) in the same market (= types of product + geographical territory), (ii) for the same good
 - ▶ accepting this possibility would *confuse customers*
 - ▶ note: this means that it might be possible to register a bitten apple as a logo for a product in a category other than computer hardware/software
- in some countries (and most notably in the US) trademarks **must** be defended, otherwise they will be lost

- The oldest branch of intellectual property
- Information that:
 - ▶ has business value
 - ▶ is not generally known to the public
 - ▶ is actively maintained secret
- Under trade secrets, there are several goods such as chemical or pharmaceutical formulas, but also software. E.g.:
 - ▶ the Coca-Cola recipe
 - ▶ the source code of proprietary software
- Trade secret protection is obtained by declaring that the details of a subject are secret
- Trade secrets last as long as their secret status is actively protected. Disclosure, reverse-engineering, or independent invention may destroy it.

**Why would you choose trade secret protection
instead of patent protection?**

Why would you choose trade secret protection instead of patent protection?

possible reasons:

- not enough novelty to be patent protectable
- patent duration is not enough for our time-frame
- no interest in selling licenses

Discussion

Pour ou contre la "PI" ?

Lawrence "Larry" Lessig

Auteur:

- *Code and Other Laws of Cyberspace* ("code is law")
- *The Future of Ideas* ("ideas are not property")
- *Free Culture* ("free culture is a commons")
- *Republic, Lost* ("How Money Corrupts Congress—and a Plan to Stop It ")

(Co)Créateur des licences *Creative Commons*



<free culture>

oscon
July 24, 2002 *

{<<} {::} {>>}

- **Utilitarian Defense of IP:** Laws and policies that maximize “wealth” or “utility” (Common Law)
- **Natural-Rights arguments:** creations of the mind are entitled to protection just as tangible property is (Continental Law)

What are the differences?

Types of Property

- Immovable property (realty, land, houses...)
- Moveable property (chairs, cars, clocks...).

Types of Property

- Immovable property (realty, land, houses...)
- Moveable property (chairs, cars, clocks...).

Tangible rights

Types of Property

- Immovable property (realty, land, houses...)
- Moveable property (chairs, cars, clocks...).

Tangible rights

Property (tangible, corporeal) attributes:

- **Antagonistic** (if I take your car, you don't have it any more)
- **Exclusivity** (if you close your car, I can't enter)
- **Scarcity** (conflict over resources, scarce == ownable)

IP vs. tangible property

If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas.

— George Bernard Shaw (1856–1925)

If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas.

— George Bernard Shaw (1856–1925)

- Not antagonistic (if I take your idea, you still retain it)
- Not exclusive (your use does not exclude my use)
- Not scarce (ideas abound)

Intellectual Property ≠ Tangible property

If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas.

— George Bernard Shaw (1856–1925)

- Not antagonistic (if I take your idea, you still retain it)
- Not exclusive (your use does not exclude my use)
- Not scarce (ideas abound)

Intellectual Property ≠ Tangible property

Intellectual Property == Intangible rights **oxymoron?**

Other differences between Intellectual Property and Tangible (“physical”) Property:

- Expiration date.
- When you copy the IP resource, you don’t harm the owner of the copied object.
- It is often difficult to draw the line between “copy” (plagiarism) and “inspiration”.

- The term itself (Stallman): analogy with physical property distorts and confuses.
- Heterogeneity among the 4 branches of IP.

The expression is very well-established in lawyer circles though.

Recommendation: use “IP” only if you *really* want to talk about multiple branches of it at the same time. Otherwise be specific.

IP Criticism — effectiveness

- A long tradition of opposition to patent and copyrights.
- Econometric studies don't conclusively show net gains in wealth or increase in innovation
 - ▶ See Michele Boldrin and David K. Levine, "Against Intellectual Monopoly" <http://levine.sscnet.ucla.edu/general/intellectual/againstfinal.htm>
- Costs of the patent system (legal advice, registration, courts, lawyers, patent trolling...)
- Property rights in "ideal objects" necessarily requires violation of other individual property rights!
- Hard to draw a line between plagiarism and inspiration
 - ▶ See for instance Chihuly's lawsuit: http://seattletimes.com/html/localnews/2002686721_chihuly16m.html



[https://commons.wikimedia.org/wiki/
File:GlamCamp_NY_Kippelboy_nina_
Paley_\(2\).JPG](https://commons.wikimedia.org/wiki/File:GlamCamp_NY_Kippelboy_nina_Paley_(2).JPG)

- Ethical issues: does fostering more innovation and creativity justify restricting individual freedoms?
- It's not coherent: protects only certain types of creations
- Distinction between creation and discovery is not clear or rigorous
- An alternative? Contract theory (S. Kinsella)
 - ▶ N. Stephan Kinsella, *Against Intellectual Property*, [https://mises.org/sites/
default/files/15_2_1.pdf](https://mises.org/sites/default/files/15_2_1.pdf)

Copyright on software

Un peu d'histoire du Droit d'Auteur

1662 Licensing of the Press Act (Angleterre, roi Charles II, **restraint l'auteur**, plutôt que le protéger)

2. compl. Paris (1896), rev Berlin (1908), compl. Berne (1914), rev Rome (1928), Brussels (1948), Stockholm (1967), Paris (1971), amend. 1979 (Berne Union)

3. pour la protection des artistes interprètes ou exécutants, des producteurs de phonogrammes et des organismes de radiodiffusion

Un peu d'histoire du Droit d'Auteur

- 1662 Licensing of the Press Act (Angleterre, roi Charles II, **restraint l'auteur**, plutôt que le protéger)
- 1709 Statute of Anne (protège l'auteur sur une durée limitée de **14 ans renouvelable une fois**)

-
- 2. compl. Paris (1896), rev Berlin (1908), compl. Berne (1914), rev Rome (1928), Brussels (1948), Stockholm (1967), Paris (1971), amend. 1979 (Berne Union)
 - 3. pour la protection des artistes interprètes ou exécutants, des producteurs de phonogrammes et des organismes de radiodiffusion

Un peu d'histoire du Droit d'Auteur

- 1662 Licensing of the Press Act (Angleterre, roi Charles II, **restraint l'auteur**, plutôt que le protéger)
- 1709 Statute of Anne (protège l'auteur sur une durée limitée de **14 ans renouvelable une fois**)
- 1791/93 Droit d'auteur (et des brevets) introduit en France, pendant la révolution, sous l'impulsion de Beaumarchais, qui fonde la première société d'auteurs (SACD); l'idée est de protéger les auteurs des exploitations abusives de leurs œuvres par les théâtres ou les imprimeurs; **droit à vie, puis 5 ans après la mort**

-
- 2. compl. Paris (1896), rev Berlin (1908), compl. Berne (1914), rev Rome (1928), Brussels (1948), Stockholm (1967), Paris (1971), amend. 1979 (Berne Union)
 - 3. pour la protection des artistes interprètes ou exécutants, des producteurs de phonogrammes et des organismes de radiodiffusion

Un peu d'histoire du Droit d'Auteur

- 1662 Licensing of the Press Act (Angleterre, roi Charles II, **restraint l'auteur**, plutôt que le protéger)
- 1709 Statute of Anne (protège l'auteur sur une durée limitée de **14 ans renouvelable une fois**)
- 1791/93 Droit d'auteur (et des brevets) introduit en France, pendant la révolution, sous l'impulsion de Beaumarchais, qui fonde la première société d'auteurs (SACD); l'idée est de protéger les auteurs des exploitations abusives de leurs œuvres par les théâtres ou les imprimeurs; **droit à vie, puis 5 ans après la mort**
- 1886 Convention de Berne (reconnaissance internationale, **sans dépôt préalable**, exclusivité traduction).² Note: ratifié par les US en 1989

2. compl. Paris (1896), rev Berlin (1908), compl. Berne (1914), rev Rome (1928), Brussels (1948), Stockholm (1967), Paris (1971), amend. 1979 (Berne Union)

3. pour la protection des artistes interprètes ou exécutants, des producteurs de phonogrammes et des organismes de radiodiffusion

Un peu d'histoire du Droit d'Auteur

- 1662 Licensing of the Press Act (Angleterre, roi Charles II, **restraint l'auteur**, plutôt que le protéger)
- 1709 Statute of Anne (protège l'auteur sur une durée limitée de **14 ans renouvelable une fois**)
- 1791/93 Droit d'auteur (et des brevets) introduit en France, pendant la révolution, sous l'impulsion de Beaumarchais, qui fonde la première société d'auteurs (SACD); l'idée est de protéger les auteurs des exploitations abusives de leurs œuvres par les théâtres ou les imprimeurs; **droit à vie, puis 5 ans après la mort**
- 1886 Convention de Berne (reconnaissance internationale, **sans dépôt préalable**, exclusivité traduction).² Note: ratifié par les US en 1989
- 1961 Convention de Rome³

2. compl. Paris (1896), rev Berlin (1908), compl. Berne (1914), rev Rome (1928), Brussels (1948), Stockholm (1967), Paris (1971), amend. 1979 (Berne Union)

3. pour la protection des artistes interprètes ou exécutants, des producteurs de phonogrammes et des organismes de radiodiffusion

Un peu d'histoire du Droit d'Auteur

- 1662 Licensing of the Press Act (Angleterre, roi Charles II, **restraint l'auteur**, plutôt que le protéger)
- 1709 Statute of Anne (protège l'auteur sur une durée limitée de **14 ans renouvelable une fois**)
- 1791/93 Droit d'auteur (et des brevets) introduit en France, pendant la révolution, sous l'impulsion de Beaumarchais, qui fonde la première société d'auteurs (SACD); l'idée est de protéger les auteurs des exploitations abusives de leurs œuvres par les théâtres ou les imprimeurs; **droit à vie, puis 5 ans après la mort**
- 1886 Convention de Berne (reconnaissance internationale, **sans dépôt préalable**, exclusivité traduction).² Note: ratifié par les US en 1989
- 1961 Convention de Rome³
- 1970s on élargit le champs d'application: jeux vidéo, logiciels, BD

2. compl. Paris (1896), rev Berlin (1908), compl. Berne (1914), rev Rome (1928), Brussels (1948), Stockholm (1967), Paris (1971), amend. 1979 (Berne Union)

3. pour la protection des artistes interprètes ou exécutants, des producteurs de phonogrammes et des organismes de radiodiffusion

Un peu d'histoire du Droit d'Auteur

- 1662 Licensing of the Press Act (Angleterre, roi Charles II, **restraint l'auteur**, plutôt que le protéger)
- 1709 Statute of Anne (protège l'auteur sur une durée limitée de **14 ans renouvelable une fois**)
- 1791/93 Droit d'auteur (et des brevets) introduit en France, pendant la révolution, sous l'impulsion de Beaumarchais, qui fonde la première société d'auteurs (SACD); l'idée est de protéger les auteurs des exploitations abusives de leurs œuvres par les théâtres ou les imprimeurs; **droit à vie, puis 5 ans après la mort**
- 1886 Convention de Berne (reconnaissance internationale, **sans dépôt préalable**, exclusivité traduction).² Note: ratifié par les US en 1989
- 1961 Convention de Rome³
- 1970s on élargit le champs d'application: jeux vidéo, logiciels, BD
- 1995 WTO's TRIPS (world-wide, presque)

2. compl. Paris (1896), rev Berlin (1908), compl. Berne (1914), rev Rome (1928), Brussels (1948), Stockholm (1967), Paris (1971), amend. 1979 (Berne Union)

3. pour la protection des artistes interprètes ou exécutants, des producteurs de phonogrammes et des organismes de radiodiffusion

Quelques organisations

WIPO/OMPI : association privée sans financement, affiliée aux UN

WTO/OMC : Organisation Mondiale du Commerce

INPI : institut français pour la Propriété Intellectuelle

EPO : Office européen des brevets

USPTO : US Patent and Trademark Office

Copyright on software: History (1)

- Software came first as part of a hardware system (*bundling*)
- In 1969, IBM “unbundled” software and services from hardware sales (due to antitrust issues).
- Portable languages (C, Unix): software began to be distributed in an independent manner (1970s).
- At first, there was a big debate about if software should be protected by patents or by copyright.

The goals of copyright protection on software were:

- Protect investments in the development
- Promote distribution of works
- Protect the creative human activity by providing incentives
- Protect a technology very prone to be copied

Copyright on software: Reasons

Copyright (rather than other IP branches at first) was finally chosen because of following characteristics:

- Simplicity (no registration, no formalities, ...)
- Automatic
- Inexpensive
- No novelty, just originality (it may be state of the art!).
- Also covers documentation
- International (Berne, TRIPS, etc.)
- Harmonization with other works.

Adapting the concept of copyright to software is not an easy task as there are many exceptions and special circumstances.

Protected by copyright on software:

- The computer program (i.e., instructions, in any form): source code and object/binary code!
- The description of the program (e.g., specification, UML diagrams)
- Additional material (user manuals, guides, etc.)
- Interfaces (graphics, sound, fonts, ...)
- (Databases) ⇒ *sui generis* database rights

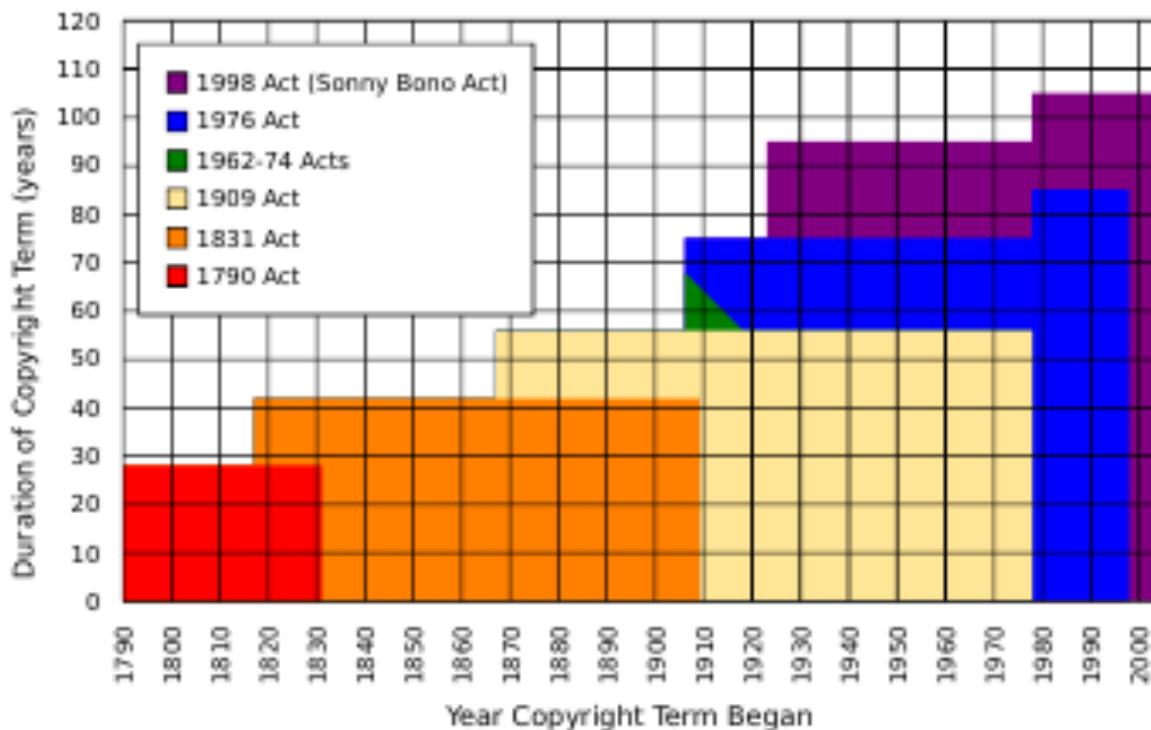
Outside the scope of copyright on software:

- Algorithms
- Procedures
- Techniques used for development
- Raw, factual data
- Very very short *and* obvious code snippets

Why Do I Need a License?

- Copyright covers software.
- Copyright is “closed by default”.
- As author: if you don’t license your code, **it can’t be used by others** (legally).
- As user: if you don’t have a license for some code, you can’t use it (legally).

Copyright Termination



Assumptions: work created when 35-year old; life duration: 70 years.

https://commons.wikimedia.org/wiki/File:Copyright_term.svg

Copyright expires after:

- Automatic copyright when the work is published.
- Minimum: 50 years after death of the copyright holder (Berne).
- In general (USA, Europe): 70 years after death.
- When copyright expires, a work enters public domain (\approx free for all, more on this later).

Periodic extension of copyright law duration.

- **Fair Use** (USA) permits limited use of copyrighted material without acquiring permission from the rights holders.
 - ▶ Examples: commentary, search engines, criticism, news reporting, research, teaching, library archiving and scholarship.

Le **droit d'auteur** est fondé sur les textes contenus dans le *Code de la Propriété Intellectuelle*⁴ (CPI)

Un droit de propriété exclusif, et automatique

Art. L111-1 :

L'auteur d'une oeuvre de l'esprit jouit sur cette oeuvre, du seul fait de sa création, d'un droit de propriété incorporelle exclusif et opposable à tous.

4. <http://www.legifrance.gouv.fr/affichCode.do?cidTexte=LEGITEXT000006069414>

Deux facettes : droit moraux et droit patrimoniaux

droit moral (L121-1...L121-9)

L'auteur jouit du droit au respect de son nom, de sa qualité et de son oeuvre. Ce droit est attaché à sa personne. Il est perpétuel, inaliénable et imprescriptible. (Art. L121-1)

droit patrimonial (L122-1...L122-12)

Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. (Art. L122-4)

Les logiciels relèvent de ce droit...

Art. L112-2

Sont considérés notamment comme œuvres de l'esprit au sens du présent code :

...

13. *Les logiciels, y compris le matériel de conception préparatoire ; ...*

Donc, il y a bien un droit moral et un droit patrimonial sur le logiciel, comme sur tout autre *œuvre de l'esprit*.

Mais avec des restrictions aux droit patrimoniaux *pour les salariés*

Un logiciel n'est pas un livre, donc on introduit des exceptions au faveur des entreprises :

Art 113-9

Sauf dispositions statutaires ou stipulations contraires, les droits patrimoniaux sur les logiciels et leur documentation créés par un ou plusieurs employés dans l'exercice de leurs fonctions ou d'après les instructions de leur employeur sont dévolus à l'employeur qui est seul habilité à les exercer.

L'exception à l'exception : le cas des stagiaires

Pourtant, le logiciel que l'un de vous (Master à Paris 7) développe entièrement dans le cadre d'un stage *lui appartient!*

La raison est que si :

- Le stage s'inscrit dans le cursus pour l'obtention d'un diplôme ou d'un titre
- Le stagiaire reste sous la responsabilité juridique de l'établissement d'enseignement bien que hors de celui-ci
- Il y a une convention de stage
- Il n'y a pas de rémunération (!= gratification)

Alors le stagiaire n'est pas considéré comme un salarié.

Voir : circulaire ministérielle du 30/10/1959, et celles n. 22 du 26/03/1970 et n. 86.065 du 13/02/1986.

Et des exceptions aux droits moraux

Art. L121-7

Sauf stipulation contraire plus favorable à l'auteur d'un logiciel, celui-ci ne peut :

- ① *S'opposer à la modification du logiciel par le cessionnaire des droits mentionnés au 2^o de l'article L. 122-6, lorsqu'elle n'est préjudiciable ni à son honneur ni à sa réputation ;*
- ② *Exercer son droit de repentir ou de retrait.*

Le droit à la paternité est maintenu : n'oubliez pas de marquer votre nom dans le code que vous écrivez !

Démarche à suivre au moment du recrutement

Si vous travaillez déjà sur un projet logiciel (libre ou pas) à vous avant d'être recruté, *préservez vos droits* en le faisant marquer en annexe du contrat du travail.

Si vous contribuez à du logiciel libre, essayez de négocier une clause spéciale qui vous en reconnaîsse entièrement les droits moraux et patrimoniaux.

Quelques réflexions :

- ContractPatch : Everything Is Negotiable⁵
- ContractPatch : Understanding the power balance⁶
- ContractPatch : It's never too late⁷

5. <https://sfconservancy.org/blog/2016/aug/04/everything-is-negotiable/>

6. <https://sfconservancy.org/blog/2016/sep/26/contractpatch-step2/>

7. <https://sfconservancy.org/blog/2016/nov/30/contractpatch-step3/>

Exemple: contrat de travail Abilian

ARTICLE 11 : PROPRIÉTÉ INDUSTRIELLE - INVENTIONS ET DROITS D'AUTEUR

En application des dispositions de l'article L.611-7 du Code de la Propriété Intellectuelle, LE SALARIÉ reconnaît que toute invention, programme, amélioration ou découverte faite ou révélée par lui pendant la durée de son contrat de travail avec LA SOCIÉTÉ, et survenant à l'occasion ou en relation avec son emploi ou utilisant directement ou indirectement des éléments sources, travaux de LA SOCIÉTÉ ou de ses clients, sera considérée comme étant intervenue dans le cadre de l'exercice normal de son travail et pour le compte de LA SOCIÉTÉ, sauf autorisation expresse de LA SOCIÉTÉ.

L'invention, l'amélioration ou la découverte et leurs bénéfices appartiendront exclusivement à LA SOCIÉTÉ et LE SALARIÉ devra aussitôt communiquer à LA SOCIÉTÉ toutes particularités de ladite invention, amélioration ou découverte.

LE SALARIÉ devra, sur demande de LA SOCIÉTÉ, tant pendant la durée de son contrat qu'après sa rupture, prendre toutes les mesures nécessaires afin d'obtenir que soient accordées à LA SOCIÉTÉ des brevets ou toutes protections des droits de propriété intellectuelle portant sur l'invention, l'amélioration ou la découverte.

En application de l'article L.113-9 du Code de la Propriété Intellectuelle, tout logiciel et leur documentation qui serait créé, adapté ou modifié par LE SALARIÉ dans l'exercice de ses fonctions ou d'après les instructions de LA SOCIÉTÉ appartiendra à LA SOCIÉTÉ à laquelle sont dévolus tous les droits patrimoniaux.

En contrepartie de sa rémunération au titre du présent contrat, LE SALARIÉ cède, par les présentes, la propriété ainsi que l'ensemble de ses droits dans tous les autres droits de propriété intellectuelle relatifs à l'activité de LA SOCIÉTÉ résultant du travail individuel ou collectif de LE SALARIÉ dans l'exercice de ses fonctions au sein de LA SOCIÉTÉ. LE SALARIÉ s'engage à signer ou fournir tout document nécessaire à l'effectivité de cette cession.

ARTICLE 12 : PROTECTION ET TRAÇABILITE DES LOGICIELS

L'utilisation des logiciels informatique au sein de LA SOCIÉTÉ est soumise à l'obtention de licences d'utilisation accordées par diverses entreprises extérieures. LE SALARIÉ s'engage à utiliser et à reproduire ces logiciels dans le respect des termes de leur licence d'utilisation.

Dans le cadre de ses fonctions de développement logiciel, LE SALARIÉ s'engage également à ne pas utiliser de logiciels, de fragments de logiciels, de bibliothèques ou de *frameworks* logiciels d'origine douteuse ou dont les licences ne permettraient pas une intégration avec les produits de LA SOCIÉTÉ dans le respect des choix juridiques effectués par celle-ci.

LE SALARIÉ reconnaît qu'il a pleinement connaissance qu'en vertu des dispositions légales, la reproduction illégale de logiciels constitue une infraction susceptible de poursuites judiciaires civiles et/ou pénales. La réalisation, l'acquisition ou l'utilisation de copies illégales de logiciels par LE SALARIÉ à l'occasion de son activité professionnelle pourra entraîner son licenciement immédiat.

ARTICLE 18 : LOGICIEL LIBRE

LA SOCIÉTÉ ABILIAN est un acteur du logiciel libre, à ce titre elle a signé la “charte libre emploi” annexée à ce contrat de travail.

Elle encourage notamment à ce titre la collaboration et notamment, par exception à l'article 17, le partage de certaines informations, dont la nature est précisée par la direction de l'entreprise, avec des communautés externes à l'entreprise.

Elle encourage également la participation de ses salariés à des projets de logiciel libre - s'ils ne sont pas susceptibles de faire concurrence à ceux de LA SOCIÉTÉ - par exception à l'article 16.

Le cas d'oeuvres ayant plusieurs auteurs

- oeuvre de collaboration : plusieurs auteurs identifiés, chacun associé à des parties spécifiques de l'oeuvre globale
- oeuvre collective : une seule entité édite, publie, diffuse une oeuvre où les contributions individuelles ne sont plus clairement identifiables

Durée des droits (France) :

- individuelle : 70 ans après la mort
- collaboration : 70 ans après la mort du dernier auteur
- collective : 70 ans après la publication

La notion de licence

Ce qu'on appelle généralement licence logicielle est plus précisément un contrat de mise à disposition de logiciels.

contrat accord privé entre des parties

Ce qu'on appelle généralement **licence logicielle** est plus précisément un **contrat de mise à disposition de logiciels**.

contrat accord privé entre des parties
mise à disposition il ne s'agit pas d'une “vente” ou “cession de droits”, mais une simple “mise à disposition” dans des conditions précisées par le contrat lui-même.
Ces conditions sont limitées seulement par l'accord des parties et par le droit national applicable.

Important : on peut avoir multiplicité de licences

Le détenteur des droits peut distribuer son logiciel sous une variété de licences différentes : la licence est un contrat de droit privé, et je suis libre d'en signer autant que je veux.

Cela est très utilisé dans le monde du logiciel libre

- Mozilla/Firefox : code sous licence MPL/GPL/LGPL, au choix libre de l'utilisateur
- OCaml compiler : LGPL pour tous, mais autre licence possible via le Consortium

Mais aussi dans le logiciel propriétaire : le même Windows peut être OEM ou pas.

Comment on valide un contrat?

Les droits nationaux diffèrent

- signature manuscrite
- signature électronique
- ...

Comment on valide un contrat?

Les droits nationaux diffèrent

- signature manuscrite
- signature électronique
- ...

Pour les logiciels, on trouve souvent

- shrink-wrap : on accepte en ouvrant le cellophane
- click-through : on accepte en cliquant un bouton

Comment on valide un contrat?

Les droits nationaux diffèrent

- signature manuscrite
- signature électronique
- ...

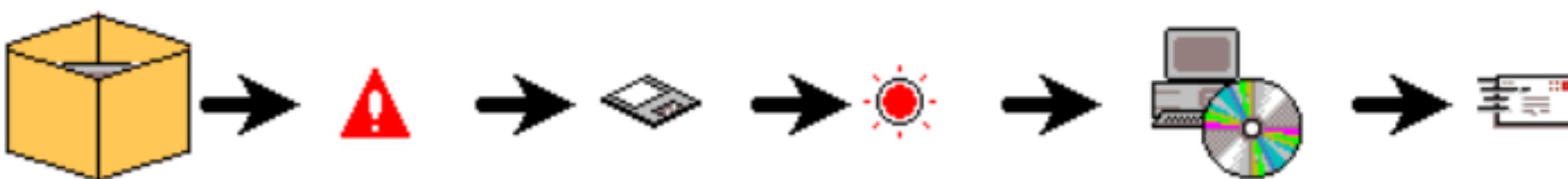
Pour les logiciels, on trouve souvent

- shrink-wrap : on accepte en ouvrant le cellophane
- click-through : on accepte en cliquant un bouton

N.B. : on peut le recuser, mais *rien* n'autorise l'utilisation du logiciel dans ce cas !

Exemple : le cas de la detaxe Windows

1. Déballez...
2. Sans allumer!
3. Insérez la disquette de VOTRE OS
4. Allumez!
5. Installez votre OS... et
6. Demandez le remboursement de Windows.



On peut trouver écrites des clauses interdisant la revente de la licence pre-installée...

Plus d'info : <http://non.aux.racketiciels.info/>

Les **limites** qu'on trouve *dans les contrats propriétaires* portent le plus souvent sur

- l'**interdiction** de la copie (dans la limite du droit qui autorise une sauvegarde)
- des **limitations d'usage** (nombre de processeurs, machine, utilisateurs simultanés, etc.)
- l'**interdiction** de la **décompilation** (dans la limite du droit en vigueur, quand même)

Reverse engineering

1991 Directive EU sur les logiciels (durée = 50 ans)

Article 6 Decompilation

*1. The authorization of the rightholder shall not be required where reproduction of the code and translation of its form within the meaning of Article 4 (a) and (b) are indispensable to obtain the information necessary to achieve the **interoperability** of an independently created computer program with other programs, provided that the following conditions are met :*

*(a) these acts are performed by the licensee or by another person having a **right to use a copy** of a program, or on their behalf by a person authorized to do so;*

(b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in subparagraph (a); and

*(c) these acts are **confined to the parts** of the original program which are necessary to achieve interoperability.*

2. *The provisions of paragraph 1 shall not permit the information obtained through its application :*

(a) to be used for goals other than to achieve the interoperability of the independently created computer program;

(b) to be given to others, except when necessary for the interoperability of the independently created computer program; or

(c) to be used for the development, production or marketing of a computer program substantially similar in its expression, or for any other act which infringes copyright.

3. *In accordance with the provisions of the Berne Convention for the protection of Literary and Artistic Works, the provisions of this Article may not be interpreted in such a way as to allow its application to be used in a manner which unreasonably prejudices the right holder's legitimate interests or conflicts with a normal exploitation of the computer program.*

Clean Room/Dirty Room reverse engineering

Pour produire un logiciel B concurrent du logiciel A, ce qui est interdit *par le droit d'auteur* est :

- la copie (de parties significatives) du code source de A
- la décompilation (de parties significatives) du code objet de A
- sauf si indispensable pour l'interopérabilité

Par contre vous pouvez :

- observer le fonctionnement de A
- réécrire *indépendamment* B avec les fonctionnalités de A

A priori, clean room/dirty room sont équivalents à ce niveau,

mais

le fait de ne pas copier, ou de réécrire totalement un logiciel ne mets pas forcément à l'abris de poursuites.

Un exemple concret

Le cas Softimage vs. Synx Relief http://www.legalis.net/jurisprudence-decision.php3?id_article=1039 de 2003 montre une décision de justice où la réécriture de code, tout en étant reconnue comme ne pas violant le droit d'auteur, est caractérisée comme un acte illicite.

Considérant que l'utilisation par un tiers à des fins commerciales du travail de recherche et d'efforts intellectuels importants comme du savoir-faire d'autrui s'analyse, indépendamment de tout risque de confusion, comme un agissement parasitaire fautif;

Dans le monde du logiciel, il n'y a pas que le droit d'auteur qui s'applique !

Comme dans tout contrat privé, on peut trouver des clauses abusives, qui sont réputées non écrites dans les pays où le droit les interdit.

Deux exemples connus

décompilation même si la licence l'interdit, en Europe on peut le faire pour l'intéropérabilité

revente de licence la plupart des licences dites OEM de produits Microsoft interdisent la revente à un tiers pour utilisation sur un autre ordinateur, et plusieurs licences logicielles interdisent la vente à un tiers tout court

Brevets logiciels

Définitions

Un brevet confère un monopole sur une *invention*, par la voie de deux droits :

Droit de Faire Art. L. 611-1 CPI : “Le brevet confère à son titulaire ou à ses ayants cause un droit exclusif d’exploitation”

Un brevet confère un monopole sur une *invention*, par la voie de deux droits :

Droit de Faire Art. L. 611-1 CPI : “Le brevet confère à son titulaire ou à ses ayants cause un droit exclusif d’exploitation”

Droit d'Interdire à tout tiers :

- la fabrication, l'offre, la mise dans le commerce,
- l'utilisation ou l'importation ou la détention du **produit** objet du brevet
- l'utilisation du **procédé**, objet du brevet, ...
- l'offre, la mise dans le commerce ou l'utilisation ou l'importation ou la détention... .
- du produit obtenu directement par le procédé objet du brevet

L'atteinte à ces droits constitue le délit de **contrefaçon**.

Les origines

En 1474, le Sénat de Venise vote un texte connu comme “Parte Veneziana”, qui énumère les principes qui sont à la base des *brevets d'invention* :

Motivations

- On concède à l'inventeur une *exclusivité* sur l'invention, ...
- limitée dans le temps (9 ans) et dans l'espace (la république de Venise), ...
- en échange de la *révélation* du contenu de l'invention, ...
- ce qui est supposé accélérer l'activité inventive dans l'intérêt du plus grand nombre.

Les origines

En 1474, le Sénat de Venise vote un texte connu comme “Parte Veneziana”, qui énumère les principes qui sont à la base des *brevets d'invention* :

Motivations

- On concède à l'inventeur une *exclusivité* sur l'invention, ...
- limitée dans le temps (9 ans) et dans l'espace (la république de Venise), ...
- en échange de la *révélation* du contenu de l'invention, ...
- ce qui est supposé accélérer l'activité inventive dans l'intérêt du plus grand nombre.

Modalités

- on dépose une demande qui *doit* présenter une invention *nouvelle*, ...
- auprès d'un office administratif qui définit précisément le contenu de l'*exclusivité* demandée, ...

Evolutions jusqu'à la révolution industrielle

Ensuite, on assiste à un foisonnement de notions de “monopole”, “licence”, “brevets” qui appellent à une réglementation :

- 1623 Statute of Monopolies en Angleterre
- 1790 Patent Act aux Etats Unis
- 1791 Brevet en France (en même temps que le droit d'auteur)
- 1883 Convention de Paris, première convention internationale (entre Belgique, Brasil, France, Guatemala, Italie, Hollande, Portugal, Salvador, Serbia, Espagne et Suisse) ; établissement du droit d'antériorité (*prior art*) dans les applications internationales ;
création de l'Union Internationale pour la Propriété Industrielle, ... qui fusionne en 1893 avec le bureau issu de la convention de Berne et devient le BIRPI.

- 1951 Création de l'INPI
- 1957 Code de la Propriété Intellectuelle en France (modifié jusqu'en 2006)
- 1970 PTC (Patent Cooperation Treaty) de Washington
- 1973 l'EPC (European Patent Convention) de Munich (révisée en 2000) crée l'Office Européen des Brevets, et précise le champ de la brevetabilité (Article 52)

Inventions brevetables (Art. 52)

- 1 Les brevets européens sont délivrés pour

Inventions brevetables (Art. 52)

- 1 Les brevets européens sont délivrés pour
les inventions

Inventions brevetables (Art. 52)

- 1 Les brevets européens sont délivrés pour
les inventions nouvelles

Inventions brevetables (Art. 52)

- 1 Les brevets européens sont délivrés pour
les inventions nouvelles impliquant une activité inventive

Inventions brevetables (Art. 52)

- ① Les brevets européens sont délivrés pour
les inventions nouvelles impliquant une activité inventive et susceptibles d'application industrielle.

Inventions brevetables (Art. 52)

- ① Les brevets européens sont délivrés pour
les inventions nouvelles impliquant une activité inventive et susceptibles d'application industrielle.
- ② Ne sont pas considérés comme des inventions au sens du paragraphe 1 notamment :
 - a les découvertes ainsi que les théories scientifiques et les méthodes mathématiques ;
 - b les créations esthétiques ;
 - c les plans, principes et méthodes dans l'exercice d'activités intellectuelles, en matière de jeu ou dans le domaine des activités économiques, ainsi que *les programmes d'ordinateurs* ;
 - d les présentations d'informations.

Inventions brevetables (Art. 52)

- ① Les brevets européens sont délivrés pour
les inventions nouvelles impliquant une activité inventive et susceptibles d'application industrielle.
- ② Ne sont pas considérés comme des inventions au sens du paragraphe 1 notamment :
 - a les découvertes ainsi que les théories scientifiques et les méthodes mathématiques ;
 - b les créations esthétiques ;
 - c les plans, principes et méthodes dans l'exercice d'activités intellectuelles, en matière de jeu ou dans le domaine des activités économiques, ainsi que *les programmes d'ordinateurs* ;
 - d les présentations d'informations.

Cela paraît limpide, mais...

- Les dispositions du paragraphe 2 n'excluent la brevetabilité des éléments énumérés auxdites dispositions que dans la mesure où la demande de brevet européen ou le brevet européen ne concerne que l'un de ces éléments, *considéré en tant que tel*.

La raison

Cette nuance est là pour permettre de breveter, par exemple, l'ABS, qui contient *aussi* du logiciel, et dont l'information sur le procédé est en partie codée dans le logiciel.

Exceptions

L'art. 53 énumère un certain nombre d'exceptions explicites (plantes, animaux, procédés chirurgicaux...)

Article 82

Unité d'invention

La demande de brevet européen ne peut concerner qu'une invention ou une pluralité d'inventions liées entre elles de telle sorte qu'elles ne forment qu'un seul concept inventif général.

Article 82

Unité d'invention

La demande de brevet européen ne peut concerner qu'une invention ou une pluralité d'inventions liées entre elles de telle sorte qu'elles ne forment qu'un seul concept inventif général.

Article 83

Exposé de l'invention

L'invention doit être exposée dans la demande de brevet européen de façon suffisamment claire et complète pour qu'un homme du métier puisse l'exécuter.

Article 84

Revendications

Les revendications définissent l'objet de la protection demandée. Elles doivent être claires et concises et se fonder sur la description.

Article 84

Revendications

Les revendications définissent l'objet de la protection demandée. Elles doivent être claires et concises et se fonder sur la description.

Article 85

Abrégé

L'abrégé sert exclusivement à des fins d'information technique; il ne peut être pris en considération pour aucune autre fin, notamment pour apprécier l'étendue de la protection demandée et pour l'application de l'article 54, paragraphe 3.

Aujourd’hui, un certain consensus s’est dégagé sur ce qui est nécessaire pour l’obtention d’un brevet sur une invention :
subject-matter relever d’un domaine brevetable

Europe : Art. 52 (Japon assez similaire)

US : “everything under the sun”

unité de l’invention présenter une unique invention

révendications indiquer clairement le/les procédés dont on demande la protection

Aspects plus substantiels

nouveauté ne doit pas déjà être connue

Europe : primo-déposant (*first-to-file*), *obligation de secret* (18 mois)

US : ~~primo inventeur~~, first-to-file (depuis 2013)

activité inventive (non-obviousness aux US) ne pas être triviale

applicabilité industrielle doit pouvoir être industrialisée (US : “*useful*”)

suffisance de la description (*disclosure*)

...

A noter la différence profonde entre deux termes trop souvent mélangés

invention un objet ou une technique nouvelle

innovation introduction d'une idée nouvelle (qui peut être largement connue ailleurs) dans un contexte qui ne l'appliquait pas (selon Schumpeter, "une invention mise en pratique")

La brevetabilité du logiciel en Europe

pre-1986 : Art. 52 exclue les logiciels

1986-1998 : relâchement progressif de l'EPO :

"programs for computers" [...] with] a technical character [...] are] not excluded from patentability"

EPO technical guidelines

1986 Vicom (T208/84), 1.7.1986 : "programme avec effet technique"

1988 IBM visual indication (T115/85), 5.9.1988 : "programme qui résout un problème technique"

1994 Sohei case (T769/92), 31.5.94 : "programme qui nécessite des considérations techniques"

1998 IBM Computer program product (T1173/97). 1.7.1998 :
"programme qui peut avoir un effet technique"
Insécurité juridique, donc...

2002 la Direction du Marché intérieur de la Commission européenne (dirigée par Frits Bolkestein) soumet la proposition de directive 2002/0047

- 2002 la Direction du Marché intérieur de la Commission européenne (dirigée par Frits Bolkestein) soumet la proposition de directive 2002/0047
- 2002 en analysant le document informatique de la Commission, on découvre qu'il a été rédigé avec l'aide matérielle de la BSA, une association privée regroupant essentiellement des grandes industries du logiciel nord-américaines

- 2002 la Direction du Marché intérieur de la Commission européenne (dirigée par Frits Bolkestein) soumet la proposition de directive 2002/0047
- 2002 en analysant le document informatique de la Commission, on découvre qu'il a été rédigé avec l'aide matérielle de la BSA, une association privée regroupant essentiellement des grandes industries du logiciel nord-américaines
- 2003-2005 s'ensuit une longue bataille qui se termine le 6 Juillet 2005 avec le rejet massif de la directive (après les multiples modifications, personne en voulait plus).

US : ce que vous voulez, ex. Amazon one-click, ou
l'opérateur “non”

EU : flou... l'EPC exclue les logiciels, l'EPO les a petit à petit inclus ; sa position se veut stricte <http://www.epo.org/news-issues/issues/software.html>, mais en pratique, elle a délivré à Apple en 2010 le brevet qui a été utilisé pour interdire la vente du Galaxy S II en Hollande (on y revient)

Autant les connaître !

voyageur pressé USPTO 5,249,290

Amazon's one-click USPTO 5,960,411

défilement d'une galerie photo EP2059868, le brevet Apple contre Samsung !

Quelques exemples II

A portable electronic device with a touch screen display for photo management is disclosed. One aspect of the invention involves a computer-implemented method in which the portable electronic device displays an array of thumbnail images corresponding to a set of photographic images. The device replaces the displayed array of thumbnail images with a user-selected photographic image upon detecting a user contact with a corresponding thumbnail image in the array. The user-selected photographic image is displayed at a larger scale than the corresponding thumbnail image. The portable device displays a different photographic image in replacement of the user-selected photographic image in accordance with a scrolling gesture. The scrolling gesture comprises a substantially horizontal movement of user contact with the touch screen display.

Coût d'un brevet

En France :

depôt à l'INPI : 320+15 euros

redevance 25 euros de la 2eme à la 5eme année, 135 euros de la 6eme à la 10eme année, 270 euros de la 11eme à la 15eme année, 530 euros de la 16eme à la 20eme année

conseil et frais de procédure \$\$\$ (total moyen entre 7000 et 8000 euros)

Extension à l'étranger :

américain 15 000 euros

japonais 15 à 20 000 euros

européen avec 10 traductions 30 000 euros

idem, plus 20 ans de redevance 100 000 euros

Qui peut obtenir un brevet

inventeur celui qui trouve

il est toujours mentionné (en Europe, Art. 62)

detenteur celui qui obtient le brevet

aux US : est l'inventeur, mais le brevet est assigné au déposant

en Europe : presque jamais l'inventeur

Le brevet étant une question d'argent, le “droit moral” ne va guère plus loin que la mention de l'inventeur.

Les cas de figure le plus courants

L'inventeur est :

indépendant il détient le brevet

salarié c'est presque toujours l'employeur

invention de mission = employeur : on vous a demandé
de chercher

hors mission attribuable = employeur : vous trouvez
dans un domaine de l'entreprise, ou en
utilisant moyens et/ou savoir faire de
l'entreprise

hors mission non attribuable = employé : dans les
autres cas (lesquels?)

Dans ce cas, un "juste prix" est dû au salarié

A l'université et au CNRS, on partage les bénéfices entre
salarié et institution

Droit d'auteur vs brevet

Elle semble difficile, mais elle ne l'est pas.

droit d'auteur : Agatha Christie *ne peut pas s'approprier* les aventures de Sherlock Holmes, i.e.

- copier de toute pièce le livre de Arthur Conan Doyle
- faire un search/replace “Sherlock Holmes” avec “Hercule Poirot”
- remplacer Arthur Conan Doyle par Agatha Christie comme auteur
- revendre le livre à son compte

mais, Arthur Conan Doyle *ne peut pas interdire* à Agatha Christie de reprendre l'idée d'un détective qui résout des cas difficiles avec son intelligence déductive, si Agatha Christie ne fait pas une copie éhontée des aventures de Sherlock Holmes

Brevet : Arthur Conan Doyle *peut interdire* à Agatha Christie de reprendre l'idée d'un détective qui résout des cas difficiles avec son intelligence déductive, Agatha Christie ne peut donc pas créer Hercule Poirot, sauf si Arthur Conan Doyle accepte de lui conceder une licence d'exploitation, à ses conditions.

Pour qui pense être Arthur Conan Doyle, le brevet est beaucoup plus intéressant, mais il ne faut pas faire ses plan sans Edgar Allan Poe !

Voir <http://www.dicosmo.org/Papers/up4-3DiCosmo.pdf>

Pour produire un logiciel B concurrent du logiciel A, ce qui est interdit *par le brevet* est :

Pour produire un logiciel B concurrent du logiciel A, ce qui est interdit *par le brevet* est :

tout

Le brevet donne pouvoir d'interdiction absolue et discrétionnaire⁸

- *réfuser une technologie essentielle à un concurrent*
- attaquer en justice, et l'accusé doit prouver son innocence

Cela transforme les brevets en armes de *guerre économique*.

En générale, cela avantage les plus grands, qui s'immunisent entre eux par cross-licensing gratuit et écrasent les petits (Ex : Gif (Unisys/IBM), Hyperlinks (British Telecom vs. Prodigy (ISP)).

Pour le logiciel libre, le brevet est une entrave majeure : la liberté de distribution du Logiciel Libre rend impossible de payer des licences à la copie !

Actions des grands

Cela a été utilisé comme arme contre le Logiciel Libre, surtout quand utilisé par d'autres grands :

2007 Microsoft affirme que Linux viole 235 de ses brevets

2011 Microsoft fait payer LG, Samsung sur des brevets non mieux déterminés

2011 Microsoft attaque Barnes&Nobles sur le Nook et Android

En réponse à ces actions, on a vu se développer des réponses diverses : IBM a offert 500 brevets de son portefeuille aux développeurs Logiciel Libre ; IBM, Novell, Philips, RedHat et sony ont créé en 2005 l'Open Invention Network,⁹ qui accumule un portefeuille de brevets défensif.

Actions des trolls et des pétits

Mais la réalité a évolué, et aujourd’hui, même les grands groupes, qui pensaient être les seuls bénéficiaires du système, sont mis à mal par des nouveaux acteurs très aggressifs :

patent trolls parasites purs qui accumulent des brevets uniquement pour attaquer en justice, et ne produisent aucun bien ; les stratégies de défense face à eux sont bien faibles

- a fun take on patent trolls : Patents : Last Week Tonight with John Oliver (HBO)¹⁰

pétits contre les grands

- Eolas (patent n. 5,838,906) vs. Microsoft pour 521M\$
- BTG contre MS et Apple

La position des acteurs du Logiciel Libre sur les brevets

- Fedora et Debian refusent de distribuer des logiciels couverts par des brevets
- Red Hat a fait une promesse publique sur les brevets logiciel http://www.redhat.com/legal/patent_policy.html
- Plusieurs licences de Logiciel Libre contiennent des clauses explicites contre les brevets (cf. prochain cours sur les licences libres)
- Plusieurs associations liées au Logiciel Libre suivent le processus législatif et combattent le brevet logiciel (en Europe, la FFII).

Lectures intéressantes

- “The Anatomy of a Trivial Patent”
<http://www.linuxtoday.com/storage/20000526004040PLF>
(Richard Stallman);
- économie des brevets
<http://www.researchoninnovation.org/patent.pdf> (J. Bessen, E. Maskin)
- The case against patents
<http://research.stlouisfed.org/wp/2012/2012-035.pdf>
(Boldrin, Levine)
- l'ingenieur de base <http://www.smh.com.au/articles/2004/07/30/1091080437270.html?oneclick=true>
- “Let’s Limit the Effect of Software Patents, Since We Can’t Eliminate Them” <http://www.wired.com/2012/11/richard-stallman-software-patents/> (Richard Stallman)

Les licences (de logiciels libre)

Tableau sur la compatibilité entre licences

Lecture du tableau : Peut-on, à partir d'une licence A (licence d'origine), distribuer sous une autre licence B (licence de distribution)

		Licences B : utilisée pour la distribution																		
		Propriétaire	Copyleft										Permissif							
			Affero GPL	GPL V3	GPL V2	LGPL V3	LGPL V2.1	CeCILL	CeCILL-C	MPL	OSL	EUPL	CPL	EPL	BSD	BSD non Modifiée	Apache	Latex	Academic Free Licence	CeCILL-B
Licences A : d'origine		Propriétaire	*	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
		Affero GPL	x	o	*	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
		GPL V3	x	*	o	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
		GPL V2	x	v	v	o	x	x	x	x	x	x	x	x	x	x	x	x	x	
		LGPL V3	x	*	o	x	o	x	x	x	x	x	x	x	x	x	x	x	x	
		LGPL V2.1	x	v	v	o	o	x	x	x	x	x	x	x	x	x	x	x	x	
		CeCILL	x	x	o	o	x	x	o	x	x	x	x	x	x	x	x	x	x	
		CeCILL-C	x	x	?	?	x	x	o	x	x	x	x	x	x	x	x	x	x	
		MPL	x	x	x	x	x	x	x	x	o	x	x	x	x	x	x	x	x	
		OSL	x	x	x	x	x	x	x	x	x	o	x	x	x	x	x	x	x	
		EUPL	x	x	x	o	x	x	o	x	x	o	o	o	x	x	x	x	x	
		CPL	x	x	x	x	x	x	x	x	x	x	x	o	x	x	x	x	x	
		EPL	x	x	x	x	x	x	x	x	x	x	x	x	o	x	x	x	x	
		BSD	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	
		BSD non Modifiée	*	?	?	x	?	x	?	x	x	x	x	x	*	*	*	*	*	
		Apache	*	*	*	x	*	x	x	x	x	x	x	x	*	*	*	*	*	
		Artistic Licence	*	*	*	x	*	x	x	x	x	x	x	x	*	*	*	*	*	
		Academic Free Licence	*	x	*	x	x	x	x	x	x	x	x	x	*	*	*	*	*	
		CeCILL-B	*	x	*	x	x	x	o	o	x	x	x	x	*	*	*	*	*	

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - Public domain
 - Scope-limited reciprocal licenses
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - CAA/CLA
 - License popularity
 - Free riding in the “cloud”

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - Public domain
 - Scope-limited reciprocal licenses
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - CAA/CLA
 - License popularity
 - Free riding in the “cloud”

User point of view

- Copyright covers software.
- Copyright is “closed by default”.
- As user: if you don’t have a license for some code, you can’t use it (legally).

No License Required?

Author point of view

- Copyright kicks in as soon as someone creates a “tangible” (expressible) work.
- In absence of any licensing declarations, don’t allow any use (“all rights reserved”).
- Without a license your (potential) users can’t use your software.
- You need to offer at least *some rights*.

FOSS licenses are legal hacks

- FOSS licenses are **legal hacks**: they behave as other copyright licenses, but instead of restricting user rights, they **grant more (and very specific) rights**
- in particular: FOSS licenses grant enough rights to ensure users enjoy the **4 freedoms** (run, study, copy, modify)
- that does *not* mean “do anything you want”; FOSS licenses can (and do) impose specific **conditions**
 - ▶ if you do not respect them, the license does not apply to you and you fallback to copyright default: “all rights reserved”

Note: FOSS is hence **not against “IP”**. In fact, FOSS licenses *use* copyright law to guarantee software freedom.

Licenses are constitutions for FOSS communities

- Software licenses are **social contracts** just as much as they are legal documents
- When you choose a license, you are charting a course for the future
- You are often establishing a relationship to a larger community
- Not purely about mechanical and legal choices
- It is very difficult to change later: it is worth to invest time into understand licensing before choosing

Implementing a basic Free Software license might be very easy:¹

Example

Copyright © 2019 Foobar Developers.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the redistribution of source code retain the above copyright notice.

That's it!

FOSS licenses implement software freedom

FOSS licenses are the legal mechanism used to implement the 4 freedoms for software users

When you receive a Free Software you get:

- Freedom #0, to **run** the program, for any purpose
- Freedom #1, to **study** how the program works, and change it
- Freedom #2, to **redistribute** copies
- Freedom #3, to **improve** the program, and **release** improvements

Note: *all* four freedoms must be granted *at the same time* for the software to be considered FOSS.

Recurring concepts in FOSS licensing

- **Use:** The right to use (run) the program, for any or some purposes.
- **Redistribution:** The act of copying the program and giving it to others.
- **Derivative work:** A program based on other programs, reusing parts of it (in binary or source code form)
- **Authorship attribution:** The obligation of recognizing the authorship of a work when using it or applying any change, such as deriving or redistributing it.
- **Licensing:** the act of choosing a (FOSS) license for a specific copy of a software
 - ▶ it is a privilege of the copyright holder(s)
 - ▶ note: different copies of the *same* software might be distributed under different licenses

Software remains “owned” (in the “IP” sense) by the copyright holder.
Users only get specific rights, determined by the license.

Are there permissible restrictions in FOSS licenses?

Are there permissible restrictions in FOSS licenses?

Yes: everything that does not get in the way of the 4 freedoms and/or the Open Source Definition is acceptable.

In practice, deciding what is OK and what is not is not always clear cut, is often not codified in guidelines, and the decisions may vary across gatekeepers (FSF/OSI/Debian/etc).

Commonly accepted restrictions are:

- **mandatory attribution** of authors (as long as attribution does not impede normal use of the work)
- **transmission of freedoms** (see copyleft later)
- **protection of specific freedoms** (e.g., access to source code or prohibition of “technical measures”, DRM)

FOSS license categories

FOSS licenses can be [classified according to the conditions they impose](#) in exchange of software freedom.

We identify the following macro-classes of FOSS licenses:

- Lax permissive (AKA “permissive”)
- Scope-limited reciprocal (AKA “weak copyleft”)
- Reciprocal (AKA “strong copyleft”)

Note: “more strict” licenses are not “less free” than others. Even the most strict FOSS licenses are incomparably more permissive than proprietary software licenses

Exercise

Compare any FOSS license to the EULA (End User License Agreement) of Microsoft Windows.

- Historically relevant subset of lax permissive licenses
- The simplest licenses: very few restrictions
- Mandating only attribution (keep names and copyright notice)
- Available for all uses, including [use in proprietary software](#)
- Originally written for and popularized by universities

Examples: MIT, BSD, ISC

- Include explicit grant of patent license (in modern variants)
- Available for almost all uses, including use in proprietary products

Examples: Apache (the license)

Reciprocal licenses

- Require that derivative works maintain the same license
- Usually require binary distributions to be accompanied by complete and corresponding source code (CCS)
- Also known as “strong copyleft” or just “copyleft”
- Historically called “viral licenses”, as a denigration tactic
 - ▶ If reciprocally licensed code is incorporated, then the application is “infected” and must be released as a whole under the same license

Examples:

- GPL, AGPL
- CC BY-SA (for non-software works)

Scope-limited reciprocal licenses

- Like reciprocal licenses, but with **limitations on the scope** of which parts of a derived work fall under the license terms
 - ▶ changes to the “**main work**” falls under the license terms
 - ▶ “**additional works**” that happen to be used with/added to/embedded with the main work do not
- They vary in the way the boundaries between main and additional works is defined
- According to the denigratory analogy: “virality” is limited to the main work
- Also known as: “weak copyleft”

Examples: MPL, CDDL, LGPL

What is copyleft?

Copyleft is a strategy of utilizing copyright law to pursue the policy goal of fostering and encouraging the equal and inalienable right to copy, share, modify and improve creative works of authorship.



Copyleft (as a general term) describes any method that utilizes the copyright system to achieve the aforementioned goal. Copyleft as a concept is usually implemented in the details of a specific copyright license, such as the GNU General Public License (GPL) and the Creative Commons Attribution Share Alike License.

Copyright holders of creative work can unilaterally implement these licenses for their own works to build communities that collaboratively share and improve those copylefted creative works.

— <http://copyleft.org/>

What is copyleft? (cont.)

- Granting the four freedoms is enough to guarantee users will get them **only for a specific copy of the work**
 - ▶ how about further downstream redistribution?
 - ▶ how about derived works?
 - ▶ (how about future versions?)
- Copyleft makes sure that all users receiving a copy of the program, no matter how modified, also enjoy the four freedoms
- The **copyleft clause** might have diverse implementations but all of them share the same concept: distribution of any version of this program must preserve user freedoms.
- Copyleft is also an **industrial strategy**: it ensures a level-playing field (contrary to lax permissive licenses), that promotes co-operation
- On the other hand copyleft does preclude *some* business models, and for that reason it might get backlash

License compatibility

- Two licenses are **compatible** if a **joint derivative work** (i.e., a work containing code released under each license) could be legally distributed
 - ▶ ideally as FOSS, although the notion of compatibility is general
- Compatibility is determined by comparing restrictions imposed by all involved licenses
 - ▶ e.g., GPL and MPL version 1.1 are incompatible (i.e., it is impossible to integrate code released under the two licenses without violating the terms of at least one of them)
 - * GPL: user must convey the entire program's source code "under the terms of *this* (= GPL) License"
 - * MPL 1.1: modifications are "governed by the terms of *this* (= MPL) License"
- A dependent variable, that does not affect compatibility *per se*, is the **resulting license** under which the joint derivative work will be redistributed
 - ▶ e.g., GPL and BSD licenses are compatible, but the resulting joint work will be under the terms of GPL only

Dual- (or multi-) licensing

Distribute software under two (or more) different sets of licenses.
The expression is unfortunately overloaded to express different notions:

- **license segregation**: different licenses apply to different copies of the same program (e.g., for proprietary relicensing business models)
- **user choice**: different, alternative (OR-ed) licenses apply to the same copy of the software; the user choose the license
 - ▶ degenerate case: “version N or above” clauses. The user can choose *which version* of the license apply to them

Motivations:

- License compatibility (e.g., Perl, Firefox)
- Business models based on market segregation (e.g., MySQL, OCaml)
- Future-proof license-based strategies

Should I write my own license?

Should I write my own license?

NO.

Bad *ad-hoc* licensing example: ipfilter (2000)

```
/*
 * Copyright (C) 1993–2000 by Darren Reed.
 *
 * The author accepts no responsibility for the use of this software
 * and provides it on an ‘‘as is’’ basis without express or implied
 * warranty.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that this notice is preserved and due credit is given
 * to the original author and the contributors.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 *
 * I hate legalese, don't you?
 */
```

ipfilter license “clarification” (2001)

```
/*
 * Copyright (C) 1993–2000 by Darren Reed.
 *
 * The author accepts no responsibility for the use of this software
 * and provides it on an ‘‘as is’’ basis without express or implied
 * warranty.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that this notice is preserved and due credit is given
 * to the original author and the contributors.
 *
 * Yes, this means that derivative or modified works are not
 * permitted without the author’s prior consent.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 *
 * I hate legalese, don’t you ?
 */
```

Theo de Raadt announces ipfilter replacement

Date: Tue, 29 May 2001 19:13:11 -0600

From: Theo de Raadt <derraadt@cvs.openbsd.org>

Subject: ipf

sometime in the next 20 hours, i will be removing ipf from the source tree since it does not meet our freedom requirements, as have been outlined in policy.html and goals.html since the start of our project.

we will have to work on an alternative.

<https://groups.google.com/d/msg/fa.openbsd.tech/q3b--naHTF0/iERRvuKkTFEJ>

- The real problem is that code with a non-free license was incorporated into the core of a free operating system
- Carelessness with licenses invites trouble

Why you should not write your own license

Many people have attempted to write their own FOSS licenses, especially in the early days, but:

- You will probably get it wrong (e.g., Artistic License 1.0)
- Your license will not immediately be approved or recognized by FOSS license gatekeepers—OSI, FSF, Debian—limiting adoption (of both the license and *your code*)
- You will contribute to **license proliferation**

License proliferation

- Vanity licenses: known problem in the community in the early years
- A growing number of licenses increases quadratically the possible combinations and interactions
- That, in turn, makes difficult to merge code from diverse sources, both for incompatibility issues and unacceptable clauses
- It introduces juridical uncertainty requiring lawyers, which is what “public” licenses were trying to avoid in the first place
- It favors FUD (Fear, Uncertainty, Doubt) about FOSS complexity

See: Open Source Initiative, “*The License Proliferation Report*”, 2006,
<http://opensource.org/proliferation>

Which license should I choose then?

Two main situations: contributing to an existing FOSS project
v. creating a new one from scratch

- a) If you contribute to an existing FOSS project: just use the current license of the project for your contributions (often you don't get to choose anyhow).

- b) If you create a new FOSS project: choose a license that is
 - ① approved by *both* OSI² and FSF³, and
 - ② popular, and
 - ③ matches the target community/**strategy** of the project

2. OSI license list: <https://opensource.org/licenses>

3. FSF license list: <https://www.gnu.org/licenses/license-list.en.html>

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - Public domain
 - Scope-limited reciprocal licenses
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - CAA/CLA
 - License popularity
 - Free riding in the “cloud”

Popular and noteworthy licenses

- **Lax permissive** (AKA “permissive”)

- ▶ BSD 3-Clause “New” or “Revised” license
- ▶ BSD 2-Clause “Simplified” or “FreeBSD” license
- ▶ Apache License 2.0
- ▶ MIT license
- ▶ ISC License

- **Scope-limited reciprocal** (AKA “weak copyleft”)

- ▶ GNU Lesser General Public License (LGPL), versions 2.1 and 3
- ▶ Mozilla Public License (MPL), version 2.0

- **Reciprocal** (AKA “strong copyleft”)

- ▶ GNU General Public License (GPL), versions 2 and 3
- ▶ GNU Affero General Public License (AGPL), version 3

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - Public domain
 - Scope-limited reciprocal licenses
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - CAA/CLA
 - License popularity
 - Free riding in the “cloud”

- **BSD** (Berkeley Software Distribution) is a Unix flavor developed by University of Berkeley (CA).
- BSD Unix was released under the terms of a “minimalistic” license, which permits both source and binary redistribution, with or without modifications, without any other restriction.
- Historical origin of the most **liberal tradition in FOSS**, opposing the use of copyleft as a strategy to liberate more software
 - ▶ intuition: favor “**developers' freedoms**” over “**users' freedoms**”
- Several revisions of the license exist
- Each revision is in fact a **template**, where copyright notices should be properly instantiated

- Descendants of the original BSD license
- Very popular (BSD userland, PF, TCP/IP, OpenSSH, TCL/Tk...)
- You may redistribute the work, in any form (source or binary), as long as you preserve [copyright notices](#)
- Includes “as is” and “no warranty” clauses
- “Liberal (= libertarian) license”: no control over software evolution

- BSD places minimal restrictions on developers and future software evolutions
- This allows BSD code to remain FOSS or become integrated into proprietary solutions
- Little legal complexity (unlike the *GPL family of licenses)
- It allows developers and companies to spend their time creating and promoting good code rather than worrying about license violations

Prior BSD License (1988)

Copyright (c) <year> <copyright holder>. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the <organization>. The name of the <organization> may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ‘‘AS IS’’ AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

- *de facto* obsolete
- approved by: FSF, Debian
- not approved by: OSI (because better recent alternatives exist)
- GPL compatible

Digression — warranty and disclaimer

- Software by itself is not a consumer product
- When software is (combined into) a consumer product, disclaimers are ineffective
- “As Is”: you are accepting item in the actual state **with all its faults**

Example — BSD warranty disclaimer

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 HOLDER 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.

(yes: it's all ALL CAPS)

4-clause / Original BSD license (1990)

*Copyright (c) <year>, <copyright holder>
All rights reserved.*

*Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:*

- 1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.*
- 2. 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.*
- 3. All advertising materials mentioning features or use of this software
must display the following acknowledgement:
This product includes software developed by the <organization>.*
- 4. 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.*

[as-is + no warranty disclaimer]

- infamous **advertisement clause** (AKA: “badgeware”)
 - ▶ advertisement notices escalation, up to 70 in NetBSD
 - ▶ further restriction
- *de facto* obsolete
- approved by: FSF, Debian
- not approved by: OSI
- GPL incompatible

3-clause / Revised / New BSD License (1999)

*Copyright (c) <year>, <copyright holder>
All rights reserved.*

*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.*
- * *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.*

[as-is + no warranty disclaimer]

- intuition: 4-clause without advertisement clause
- popular permissive license
- approved by: FSF, OSI, Debian
- GPL compatible

2-clause / FreeBSD / Simplified BSD License

*Copyright (c) <YEAR>, <OWNER>
All rights reserved.*

*Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:*

- 1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.*
- 2. 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.*

[as-is + no warranty disclaimer]

*The views and conclusions contained in the software and documentation
are those of the authors and should not be interpreted as representing
official policies, either expressed or implied, of the FreeBSD Project.*

- intuition: 3-clause without the non-endorsement clause
- used by the FreeBSD project
- approved by: FSF, OSI, Debian
- GPL compatible

Popular BSD-like licenses

- ISC
- MIT

ISC: the shortest FOSS license

Copyright (c) 4-digit year, Company or Person's Name

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

- Functionally similar to the 2-clause BSD
- Language made “unnecessary” by Berne Convention removed.⁴
- BIND, DHCP, and preferred license by the OpenBSD project
- approved by: FSF, OSI, Debian
- GPL compatible

4. according to <http://www.openbsd.org/policy.html>

The MIT License

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.
[as-is + no warranty disclaimer]*

- Functionally similar to the 2-clause BSD.
- It doesn't contain an explicit notice prohibiting the use of the name of the copyright holder in promotion
- It states more explicitly the rights given to the end-user
- Very popular: originally X.org, now web frameworks (e.g., Node), 45% of GitHub projects (2015)
- approved by: FSF, OSI, Debian
- GPL compatible

- Old versions: 1.0 (original) and 1.1 (ASF, 2000)
 - ▶ <https://www.apache.org/licenses/LICENSE-1.0>
- An extension of the 3-clause BSD license
- Permits integration into closed source projects
- Apache License 2.0 (January 2004): permissive license.
 - ▶ Make the license easier for non-ASF projects to use
 - ▶ Explicitly grants patent rights where necessary to operate, modify and distribute the software (§3)
 - ▶ Patent retaliation (terminating the license upon the initiation of a lawsuit)

<https://www.apache.org/licenses/LICENSE-2.0>

- popular license
 - ▶ ≈150 projects hosted by the Apache Software Foundation (2015)
 - ▶ over 8'000 non-ASF projects located at SourceForge are available under Apache License (2012)
 - ▶ 25% of Google Code projects, including Android user space (2008); 11% of GitHub (2015)
- approved by: FSF, OSI, Debian
- compatible with GPLv3
- incompatible with GPLv2

§3. *Grant of Patent License*

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable, patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work , where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) [...].

§3. *Grant of Patent License*

[...] *If You institute **patent litigation** against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.*

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - **Public domain**
 - Scope-limited reciprocal licenses
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - CAA/CLA
 - License popularity
 - Free riding in the “cloud”

How far can we go with liberal licensing?

i.e., maximizing user freedoms and minimizing constraints

An example: WTFPL License

*DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
Version 2, December 2004*

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

*DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION*

0. You just DO WHAT THE FUCK YOU WANT TO.

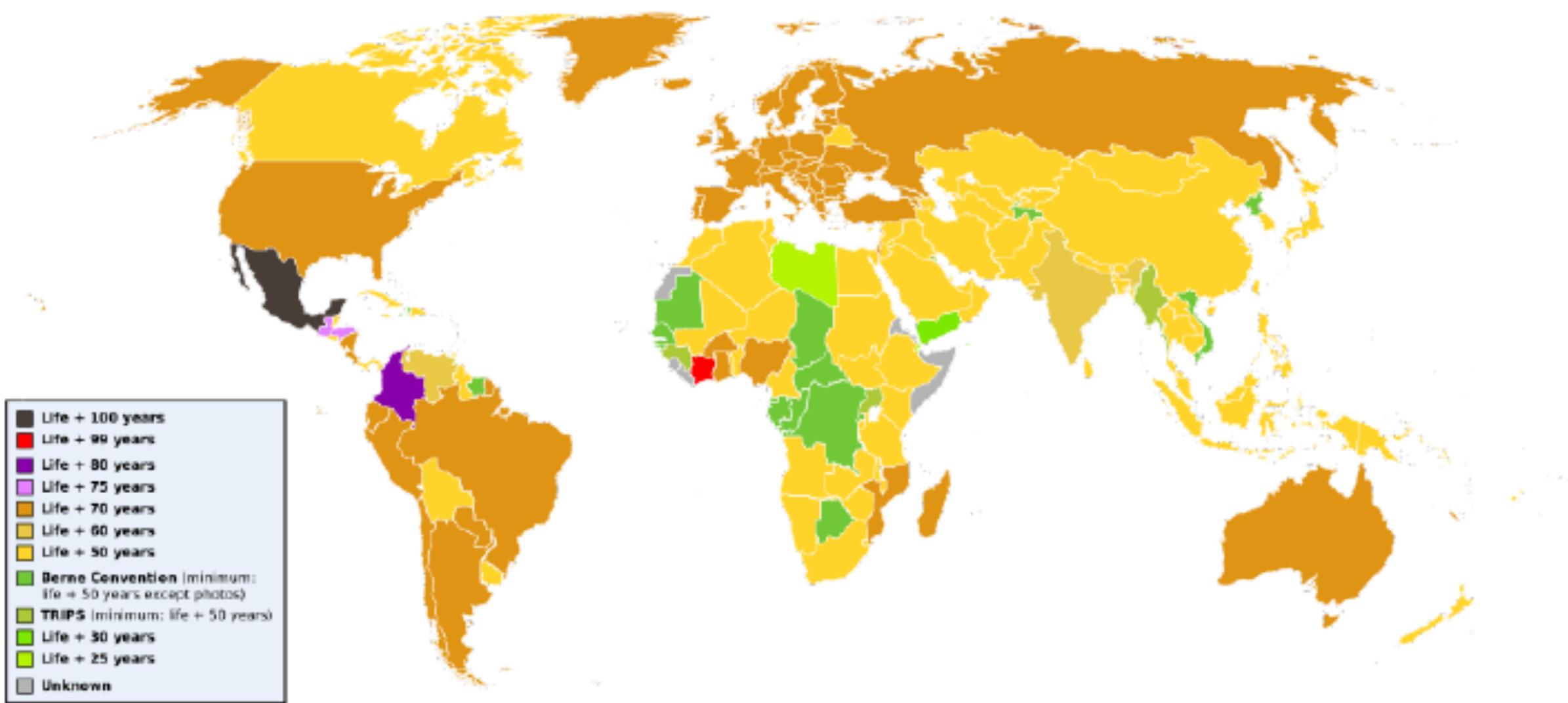
- Irreverent text, on purpose
- Licensees are encouraged to do what the @*%#!#* they want to
- Not very popular, not a good choice for software
- Approved by: FSF, Debian
- Not approved by: OSI

Works in the **public domain** are those works for which copyright rights have expired, forfeited, or not applicable.

There are thus various ways for copyrightable works to **enter the public domain**:

- ① copyright is **not applicable** to this kind of works (facts, theories, short snippets, ...)
- ② the copyright has **expired**
 - ▶ copyright expired *tout court*
 - * see Public Domain Day, January 1st each year
 - ▶ the copyright owner failed to follow copyright renewal rules, where/when applicable (e.g., the novel *Anthem* by Ayn Rand)
- ③ the copyright owner **deliberately places** it in the public domain

World copyright terms



https://commons.wikimedia.org/wiki/File:World_copyright_terms.svg

Public domain dedication I

To deliberately put a work in the public domain, authors should use a **public domain dedication**, e.g.:

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors.

Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law,

Public domain dedication II

whether vested or contingent, in the Work. Dicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

<http://creativecommons.org/licenses/publicdomain/>

- Is it actually possible *before* copyright expiration?

- Is it actually possible *before* copyright expiration?
- Several legal systems (and most notably in Europe) effectively prohibit any attempt by the owners to surrender copyright rights automatically conferred by law
 - ▶ Particularly moral rights (perpetual, unwaivable, inalienable)
- A solution: the CC0 license by Creative Commons—waive all copyright rights [to the fullest extent allowed](#) by law

[...]

2. **Waiver.** *To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.*

3. **Public License Fallback.** *Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the*

Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not invalidate the remainder of the License, and in such case Affirmer hereby affirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claims and causes of action with respect to the Work, in either case contrary to Affirmer's express Statement of Purpose.

[...]

<http://creativecommons.org/publicdomain/zero/1.0/>

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - Public domain
 - **Scope-limited reciprocal licenses**
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - CAA/CLA
 - License popularity
 - Free riding in the “cloud”

1998 version 1.0, as a successor of the NPL (Netscape Public License)

1999 version 1.1 by Mozilla Organization

- public feedback/review process on how to improve version 1.0
- allow for dual-/multi-licensing

2012 version 2.0

- public process again
- GPL-compatible

Mozilla Public License (MPL) 2.0

- weak (or partial) copyleft license, with file-based boundaries on the reach of copyleft requirements
- separation between covered software and larger work

Covered Software

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof

Larger Work

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software

See <http://opensource.org/licenses/MPL-2.0>

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <https://mozilla.org/MPL/2.0/>.

i.e., license notice

Covered software

① copyleft clause

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License

② source code requirement

If You distribute Covered Software in Executable Form then [...] such Covered Software must also be made available in Source Code Form

Mozilla Public License (MPL) 2.0 (cont.)

- Larger work: its own license with carve out for MPL-covered code
You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software.
- explicit multiple-licensing support:
If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, [...], this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

Mozilla Public License (MPL) 2.0 (cont.)

- Patent provisions: patent license + patent retaliation (similar to Apache2)
- approved by: FSF, OSI, Debian
- version 2.0 of the license is compatible with the GPL
- version 1.1 is *incompatible* with the GPL
 - ▶ A module covered by the GPL and a module covered by the MPL version 1.1 cannot be linked together.
 - ▶ For this reason, Firefox has been relicensed under multiple licenses (MPL, GPL, LGPL).
 - ▶ MPL 1.1 can be specifically amended to allow combining with GPL and others (§13, “Multiple-licensed code”).

- The Common Development and Distribution License (CDDL) is based on the MPL, version 1.1
- Produced by Sun Microsystems for the OpenSolaris projects (kernel, userland, ZFS, DTrace, NetBeans, GlassFish, ...)
- approved by: FSF, OSI, Debian
- It tries to amend GPL-incompatibility issues in the MPL 1.1
 - ▶ without succeeding (according to FSF and Debian)
- Some non-compliance issues with European law system in the MPL have been corrected in the CDDL
- approved by: FSF, OSI, Debian
- GPL-incompatible

1991 GNU *Library General Public License*, version 2 (for uniformity with GPL version)

1999 GNU *Lesser General Public License*, version 2.1

- name change to emphasize that it is inferior (from a copyleft POV) to the GPL, rather than the recommended variant of the GPL for software libraries

2007 GNU LGPL, version 3

- reimplemented as GPLv3 + additional permissions
- very popular license for libraries (and more)
- approved by: FSF, OSI, Debian
- GPL compatible

- §4. You may copy and distribute *the Library* (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, [...]

- §4. You may copy and distribute *the Library* (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, [...]
- §5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “*work that uses the Library*”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

Note the lack of explicit file boundaries (contrary to, e.g., MPL)

- §4. You may copy and distribute *the Library* (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, [...]
- §5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “*work that uses the Library*”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

Note the lack of explicit file boundaries (contrary to, e.g., MPL)

- §3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library [...] This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

- A “*Combined Work*” is a work produced by combining or linking an *Application* with the *Library*. The particular version of the *Library* with which the *Combined Work* was made is also called the “*Linked Version*”.
- The “*Minimal Corresponding Source*” for a *Combined Work* means the *Corresponding Source* for the *Combined Work*, excluding any source code for portions of the *Combined Work* that, considered in isolation, are based on the *Application*, and not on the *Linked Version*
- The object code form of an *Application* may incorporate material from a *header file* that is part of the *Library*.

- 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 [...]
- [provided that] you do one of the following:
 - Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version [...]
 - Use a suitable shared library mechanism for linking with the Library. [...]

things start to get quite technical for a legal document...

Intuition: enable users to enjoy the 4 freedoms on the “library”.
Technically, that means being able to replace the library with a modified version of it.

1 Free Software licensing

2 License bestiary

- Lax permissive licenses
- Public domain
- Scope-limited reciprocal licenses
- Reciprocal licenses

3 Selected licensing topics

- GPL v. linking
- CAA/CLA
- License popularity
- Free riding in the “cloud”

GNU General Public License (GPL)

- considered to be the most popular Free Software license
- approved by: FSF, Debian, OSI

1989 version 1 (by RMS), as a generalization (hence the name) of licenses already used by the GNU project for: Emacs, GDB, GCC

1991 version 2 (by RMS)

- “liberty or death”; early ex. of defense against patents and similar threats to user freedoms

2007 version 3 (by RMS with counsel from E. Moglen/SFLC)

- public review process
- software patents clauses
- DRM clauses (anti “tivoization”)
- license compatibility provision
- internationalization
- self-defense against further restrictions

What makes the GPL so special?

- First implementation of copyleft
- Highly influential on all subsequent copyleft/share-alike licenses, including Creative Commons
- Without the GPL, copyleft would have been just an abstract idea
- Designed to prevent proprietary relicensing of Free Software code
- Popularity

§3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, [...]*

§3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, [...]

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.

§3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, [...]

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.

However, as a special exception,⁵ the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

5. the so called “system library exception”

§2. You may *modify* your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) 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*.
- derived works fall under the terms of the GPL themselves, hence *their* source code must be distributed as well
 - (a) is a local requirement
 - (b) only triggers upon distribution

§5. 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.

§5. 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.

How about permission to use the Program?

§5. 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.*

How about permission to use the Program?

§0. [...] Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, [...]

Recommended way to apply the GPL to source code:

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

- part of the copyright/license notices, not of the license itself
- individual software authors *can* leave the “or later” clause out
- other licenses include implicit “or later” requirements in the license text itself (e.g., MPL)

For best practices on how to manage copyright/license notices see:
Software Freedom Law Center, *Managing copyright information within a free software project*

<https://softwarefreedom.org/resources/2012/ManagingCopyrightInformation.html>

Why you shouldn't use the Lesser GPL for your next library
<https://www.gnu.org/licenses/why-not-lGPL.html>

*The GNU Project has two principal licenses to use for libraries. One is the GNU Lesser GPL; the other is the ordinary GNU GPL. [...] using the Lesser GPL permits [use of the library in proprietary programs](#); using the ordinary GPL for a library makes it available only for free programs.
[...]*

*Which license is best for a given library is [a matter of strategy](#) [...].
[...] Free software developers need to make advantages for each other. Using the ordinary GPL for a library gives free software developers an advantage over proprietary developers: a library that they can use, while proprietary developers cannot use it. [...]*

There are reasons that can make it better to use the Lesser GPL in certain cases. The most common case is [when a free library's features are readily available for proprietary software through other alternative libraries](#). In that case, the library cannot give free software any particular advantage, so it is better to use the Lesser GPL for that library.

- Written by Richard Stallman and the FSF, published in 1991.
- The most popular Free Software license: estimated to cover 50-70 % of all Free Software projects (at the time)
- It's more than a software license: it is a social contract, imposing that all players have the same rights and obligations

Why update it?

- Written by Richard Stallman and the FSF, published in 1991.
- The most popular Free Software license: estimated to cover 50-70 % of all Free Software projects (at the time)
- It's more than a software license: it is a social contract, imposing that all players have the same rights and obligations

Why update it?

After 15 years, needed updating in order to remain effective against [new threats](#) to user freedoms.

Intuition: the GPL is a mean to an end. It is an implementation that might have bugs (or grow them over time), which need to be fixed in further releases of the license.

Public consultation process:

- very relevant and the social responsible thing to do: given the abundance of “or later” software, the effects of the release of GPLv3 might be huge
- It lasted 18 months: from January 16, 2006 (first draft) to June 29, 2007 (final version)
- Invited participants from high-profile Free Software projects
- 4 drafts
- 5 International Conferences (Boston, Porto Alegre, Barcelona, Tokyo and Brussels)

§3. Protecting Users' Legal Rights From Anti-Circumvention Law.

[...]

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

- does not *forbid* to implement DRM & co. in software
- but allows to write interoperable software and bypass restrictions
- neutralize laws that get in the way of user freedoms (e.g., DMCA, EUCD)

- Together with

§6. Conveying Non-Source Forms.

[...]

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source.

it also neutralizes “tivoization”, i.e., the circumvention of the GPL by using cryptography to disallow the installation/execution of modified versions of a GPL’d program

Protection against patent threats is implemented by GPLv2 only in the “**Liberty or Death**” clause:

§7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all.

[...]

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. [...]

GPLv3 adds (i.e., “liberty or death” remains) stronger protection against patent threats through legal-engineering:

§11. Patents.

[...] *Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.*

§10. Automatic Licensing of Downstream Recipients.

[...] *you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.*

§7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. [...]

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. [...]

All other non-permissive additional terms are considered “further restrictions” [...] If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term.

- **§7. Additional Terms.**

[...] for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms: [...]

 - (e) Declining to grant rights under *trademark* law for use of some trade names, trademarks, or service marks; or
(and other similar permissions for adding warranties/“as is” disclaimers, limiting the use for publicity purposes, etc.)
- *Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU [Afferro General Public License](#) into a single combined work, and to convey the resulting work.*

§15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

§16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The Application Service Provider (ASP) loophole

Exercise

- ① *obtain a copy of some GPL'd program*
- ② *modify it*
- ③ *offer remote access to your modified version over the Net (e.g., web app, remote API, etc.)*

Does the GPL force you to redistribute the code of your modified version?

The Application Service Provider (ASP) loophole

Exercise

- ① *obtain a copy of some GPL'd program*
- ② *modify it*
- ③ *offer remote access to your modified version over the Net (e.g., web app, remote API, etc.)*

Does the GPL force you to redistribute the code of your modified version? No.

- GPL (both v2 and v3) copyleft clauses trigger upon distribution of the modified copy, in either source or binary form
- if you do not do any of that, copyleft does not kick in
- from copyleft POV, this is very problematic for web/network apps
“GPL is the BSD of Web applications” — Bradley Kuhn
- but in an increasingly more connected world, the problem is more general

- Based on the GPL
- Pioneered by Affero, Inc. (version 1, 2002)
- Published by the Free Software Foundation (version 3, 2007)
- It contains the extra **Affero clause** that requires distribution of modified source code of applications to users **interacting remotely** over the network with the program
- The clause has initially been considered for inclusion in GPLv3, but then relegated to a separate license
- approved by: FSF, Debian, OSI
- GPL compatible (explicitly so)

§13. *Remote Network Interaction* [...]

if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software.

Bonus slides

Outline

1 Free Software licensing

2 License bestiary

- Lax permissive licenses
- Public domain
- Scope-limited reciprocal licenses
- Reciprocal licenses

3 Selected licensing topics

- **GPL v. linking**
- CAA/CLA
- License popularity
- Free riding in the “cloud”

Derivative works and the GPL

GPL copyleft propagation applies to (GPLv2 language):

a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.

Exercise (linking and the GPL)

Can you link a GPL program/library with a non-GPL program/library, without applying the GPL to the obtained binary?

Derivative works and the GPL

GPL copyleft propagation applies to (GPLv2 language):

a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.

Exercise (linking and the GPL)

Can you link a GPL program/library with a non-GPL program/library, without applying the GPL to the obtained binary?

Answer:⁶

- FSF (and popular) answer: no; the GPL applies
- some corporate lawyers’ answer: yes; the GPL doesn’t apply
- court cases/tribunal answer: none (yet)

6. according to some, the *actual* answer is thus “we don’t know”

Derivative or collective works?

[US law language]

- a **derivative work** is a “work based upon one or more preexisting works”, which requires some transformation or adaption of the original
- a **collective work** is created when a person brings together “preexisting materials... in such a way that the resulting work as a whole constitutes an original work of authorship”
 - ▶ individual parts remain under their individual licenses
 - ▶ a separate license apply to the collection

Does linking create a derivative or a collective work (or both)?

Linking and the GPL — FSF position

License text (redux):

a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.

From the GPL FAQ:⁷

Q: Does the GPL have different requirements for statically vs dynamically linked modules with a covered work?

A: No. Linking a GPL covered work statically or dynamically with other modules is making a combined work based on the GPL covered work. Thus, the terms and conditions of the GNU General Public License cover the whole combination. [...]

Q: Can I release a non-free program that's designed to load a GPL-covered plug-in?

A: [...] Using shared memory to communicate with complex data structures is pretty much equivalent to dynamic linking

Linking and the GPL — arguments

- dynamically linked executables contains “annotations and elaborations” on a base binary
 - ▶ does a Linux **kernel module** contains annotations and elaborations on the base expression of the kernel?
 - ▶ if yes, then it might be a derived work of the kernel (GPLv2)

Linking and the GPL — arguments

- dynamically linked executables contains “annotations and elaborations” on a base binary
 - ▶ does a Linux **kernel module** contains annotations and elaborations on the base expression of the kernel?
 - ▶ if yes, then it might be a derived work of the kernel (GPLv2)
 - ▶ how about **user programs** that run on Linux?
 - ▶ according to Linus and kernel developers:⁸

*NOTE! This copyright does *not* cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does *not* fall under the heading of “derived work”.*

Linking and the GPL — arguments

- dynamically linked executables contains “annotations and elaborations” on a base binary
 - ▶ does a Linux **kernel module** contains annotations and elaborations on the base expression of the kernel?
 - ▶ if yes, then it might be a derived work of the kernel (GPLv2)
 - ▶ how about **user programs** that run on Linux?
 - ▶ according to Linus and kernel developers:⁸
- NOTE! This copyright does *not* cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does *not* fall under the heading of “derived work”.*
- ... but are they right?
- the legal principle of **usage of trade** might play a role too

Linking and the GPL — arguments (cont.)

- arguments to the contrary (often by corporate lawyers) claim that linking only creates **collective works**—not subject to the GPL as a whole—because there is no substantial difference between two executables on disk and two in memory

Linking and the GPL — arguments (cont.)

- arguments to the contrary (often by corporate lawyers) claim that linking only creates **collective works**—not subject to the GPL as a whole—because there is no substantial difference between two executables on disk and two in memory
- **header files** might also play a role
 - ▶ during compilation (before linking) you might use header files to prepare your executable for dynamic linking
 - ▶ if the headers used at compile time are GPL'd, then your dynamically linked executable might be a **derived work of the headers**

- arguments to the contrary (often by corporate lawyers) claim that linking only creates **collective works**—not subject to the GPL as a whole—because there is no substantial difference between two executables on disk and two in memory
- **header files** might also play a role
 - ▶ during compilation (before linking) you might use header files to prepare your executable for dynamic linking
 - ▶ if the headers used at compile time are GPL'd, then your dynamically linked executable might be a **derived work of the headers**

Exercise (LGPL header files v. linking)

*Is header inclusion a potential issue for LGPL'd programs too?
Why?*

Linking and the GPL — arguments (cont.)

- there are also other types of “linking”: RPC, RMI, REST API, etc.
When do they constitute “linking” in a sense that would trigger strong copyleft requirements?
 - ▶ no consensus yet
 - ▶ legal folklore seems to suggest that:
 - * loosely coupled and/or popular and/or standardized APIs with several alternative implementations should not trigger the GPL
 - * tightly coupled and/or ad-hoc and/or single-implementation APIs should trigger the GPL
- on the other hand, it seems consensual that static linking produces a derivative work of the GPL part

Linking and the GPL — arguments (cont.)

- there are also other types of “linking”: RPC, RMI, REST API, etc.
When do they constitute “linking” in a sense that would trigger strong copyleft requirements?
 - ▶ no consensus yet
 - ▶ legal folklore seems to suggest that:
 - * loosely coupled and/or popular and/or standardized APIs with several alternative implementations should not trigger the GPL
 - * tightly coupled and/or ad-hoc and/or single-implementation APIs should trigger the GPL
- on the other hand, it seems consensual that static linking produces a derivative work of the GPL part

Ultimately, this GPL linking dilemma is problematic only for those who want to somehow circumvent one of the main goals of the GPL which, *per se*, is very clear.

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - Public domain
 - Scope-limited reciprocal licenses
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - **CAA/CLA**
 - License popularity
 - Free riding in the “cloud”

Copyright Assignment Agreement (CAA) cession agreement where a copyright holder surrender all their copyright sanctioned rights on some work to another party

Contribution License Agreement (CLA) agreement where a copyright holder gives a license (usually non-revocable, possibly exclusive) to enforce specific copyright sanctioned rights to another party

- on paper, CAA are more powerful than CLA; but they only go as far as the legal system allows them
 - ▶ e.g., in most of Europe moral rights cannot be surrendered
- CLAs can be so broad to be *de facto* equivalent to CAAs
- key copyright right for policy reasons: the **ability to relicense**

If a vendor participating in a FOSS project has, alone, the ability to relicense, strategy considerations based solely on the chosen FOSS license are completely moot.

Not all CAAs/CLAs are born equal

- set of rights surrendered
 - ▶ e.g., enforcement-only agreements
- mandatory vs optional agreements
- to (public benefit) nonprofit vs for profit entities
- safe guards
 - ▶ e.g., “we can relicense, but we will pick within this set of licenses”
 - * common choice: OSI-approved \cap FSF-approved licenses
- alternatives (within limits)
 - ▶ “or later” clauses
 - * possibly with license proxy (e.g., GPLv3, §14)
 - ▶ will

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - Public domain
 - Scope-limited reciprocal licenses
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - CAA/CLA
 - License popularity**
 - Free riding in the “cloud”

Warning

Due to the [free circulation](#) of Free Software, it is very difficult to get hard number about software—and therefore license—popularity.

Nonetheless, many actors of the FOSS ecosystem publish [statistics](#) about those facts. Unfortunately, most of them do so in rather [unscientific](#) ways: without disclosing the details about the dataset they are using, and without liberating the software they use to compile their statistics.

Use [caution](#) in interpreting the data.

Blackduck: “Top 20 Open Source Licenses”

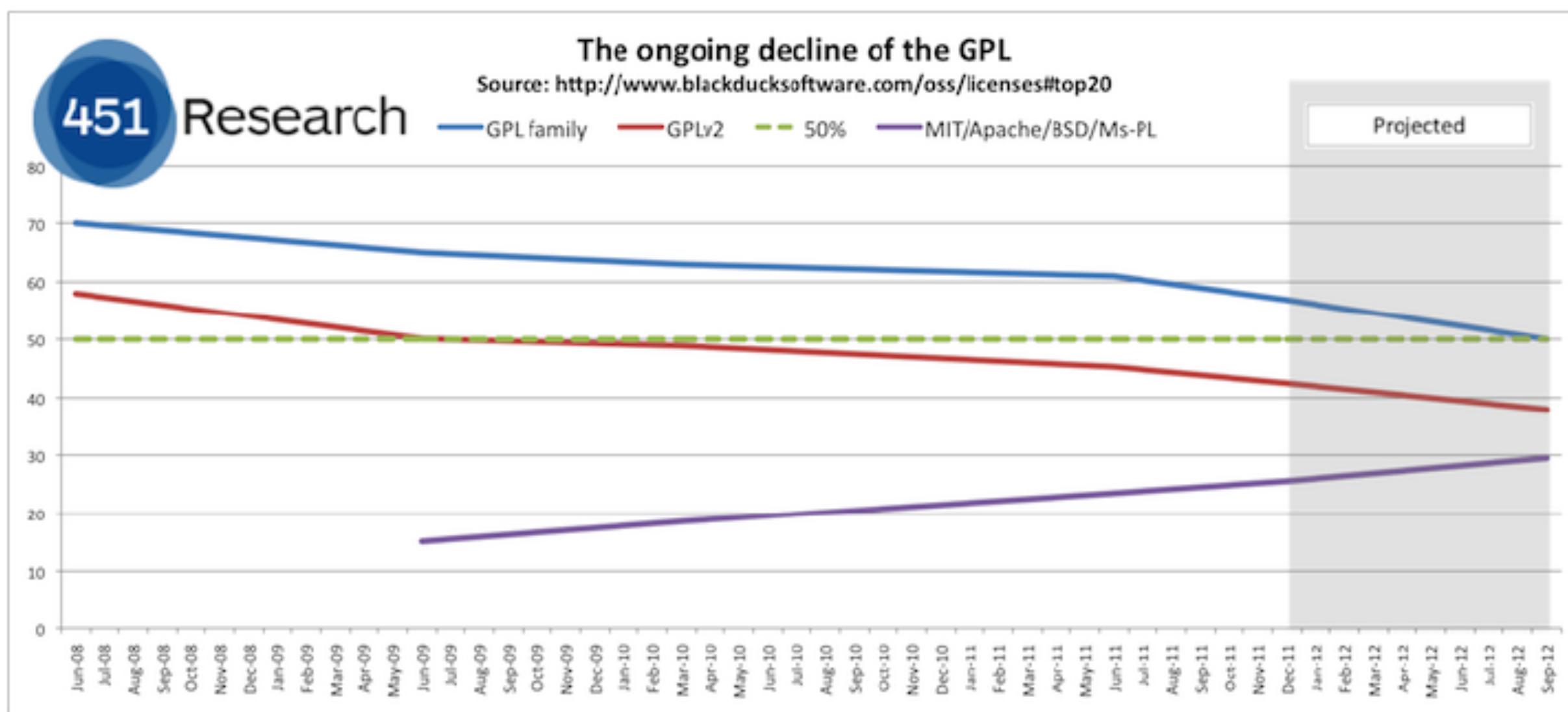
GPL 2.0	25%
MIT	19%
Apache License 2.0	16%
GPL 3.0	10%
BSD 3-clause	7%
Artistic (Perl)	5%
LGPL 2.1	5%
LGPL 3.0	2%
MS-PL	2%
EPL	2%
Code Project Open License	1%
MPL 1.1, BSD 2-clause, CDDL 1.0, GPL, Microsoft Reciprocal License, Sun GPL w/ classpath exc., CDDL 1.1, zlib/libpng, CPL	<1% ⁹

<https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses>

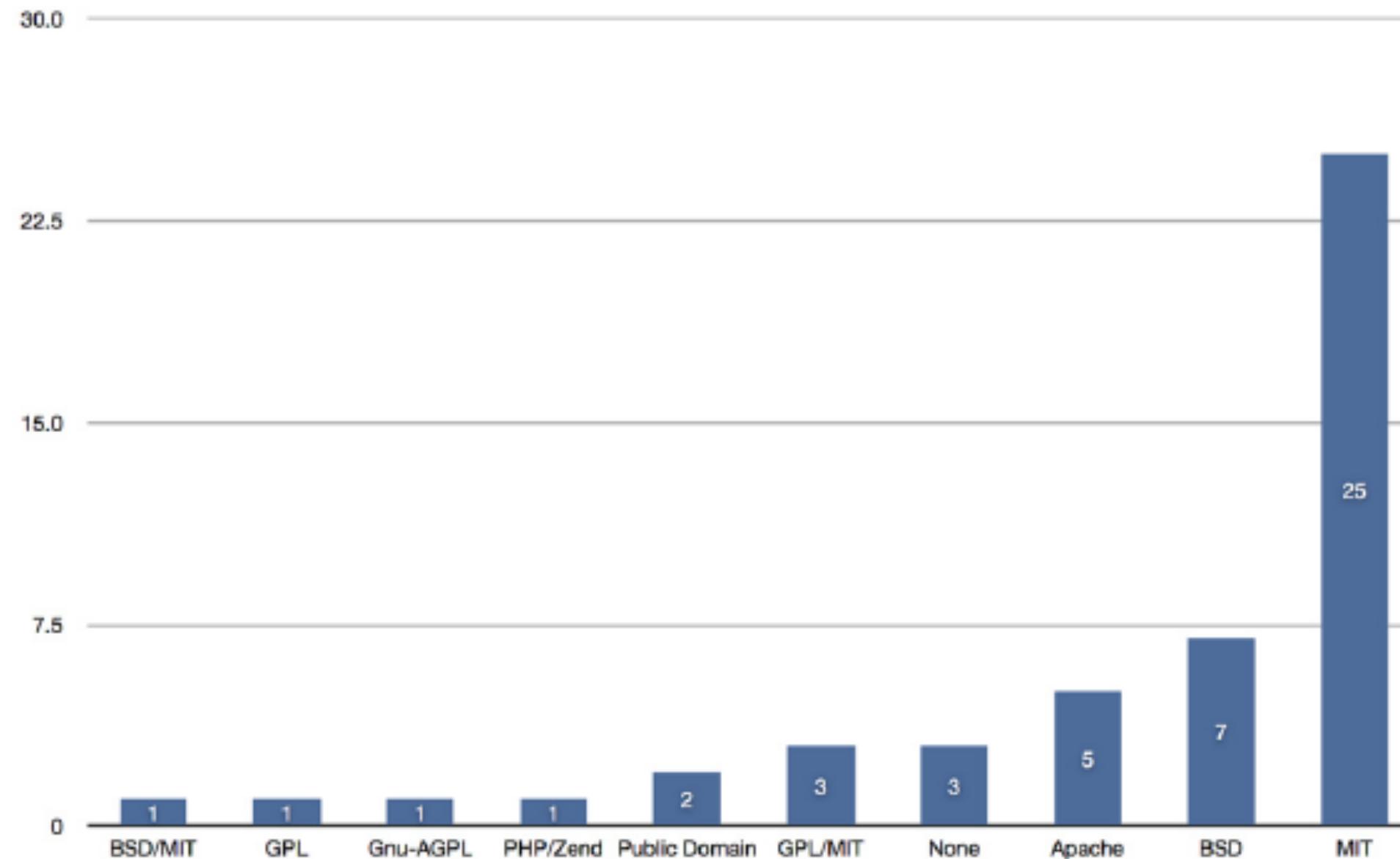
February 2015

Blackduck: the (alleged) decline of the GPL

According to Blackduck, in recent years the GPL is losing ground to permissive licenses:



GitHub: top licenses



<http://ostatic.com/blog/the-top-licenses-on-github>

February 2012

GitHub — a bit more complex than that

- 2013 analysis of GitHub by Aaron Williamson “Licensing of Software on Github: A Quantitative Analysis”¹⁰
- only 14.9% have a top-level license file
- only 3.6% mention a license in README
- for the remaining projects, the breakdown is largely confirmed
- POSS (“Post Open Source Software” debate)

10. http:

//www.softwarefreedom.org/resources/2013/lcs-slides-aaronw/

License popularity in Debian over time



Matthieu Caneill, Daniel M. Germán, Stefano Zacchiroli

The Debsources Dataset: Two Decades of Free and Open Source Software.
Empirical Software Engineering, Vol. 22, 2017.

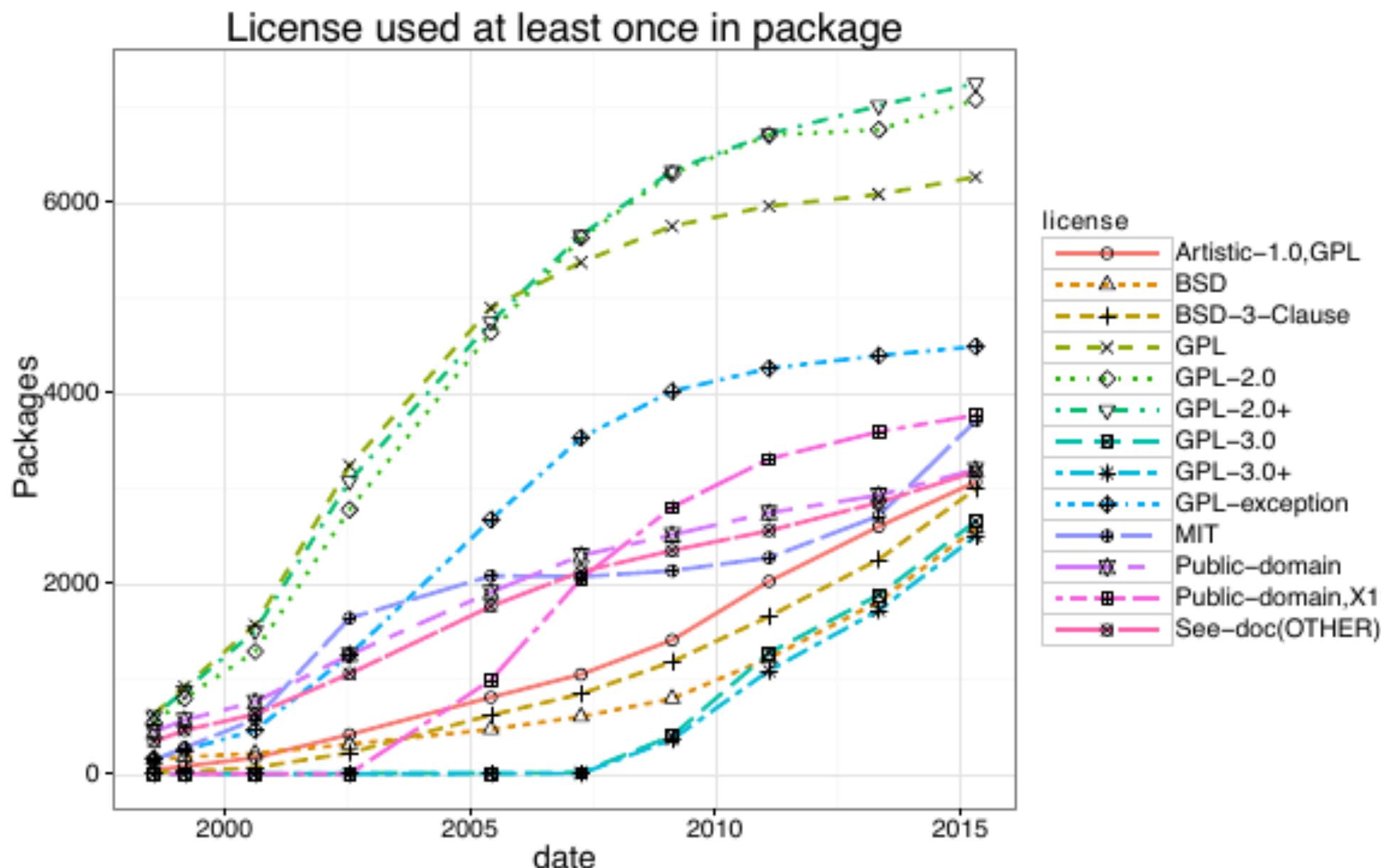


License popularity in Debian over time (cont.)



Matthieu Caneill, Daniel M. Germán, Stefano Zacchiroli

The Debsources Dataset: Two Decades of Free and Open Source Software.
Empirical Software Engineering, Vol. 22, 2017.

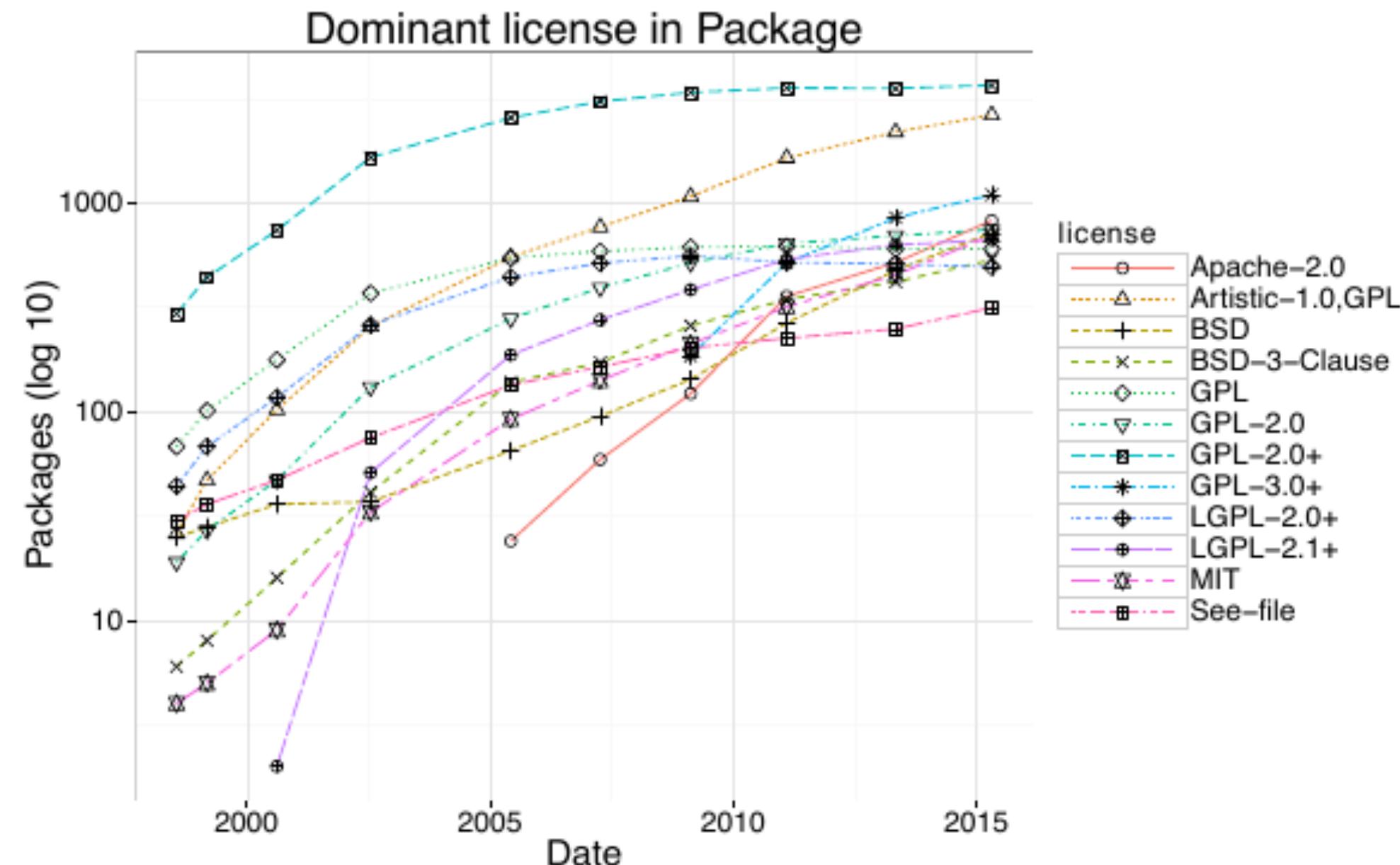


License popularity in Debian over time (cont.)

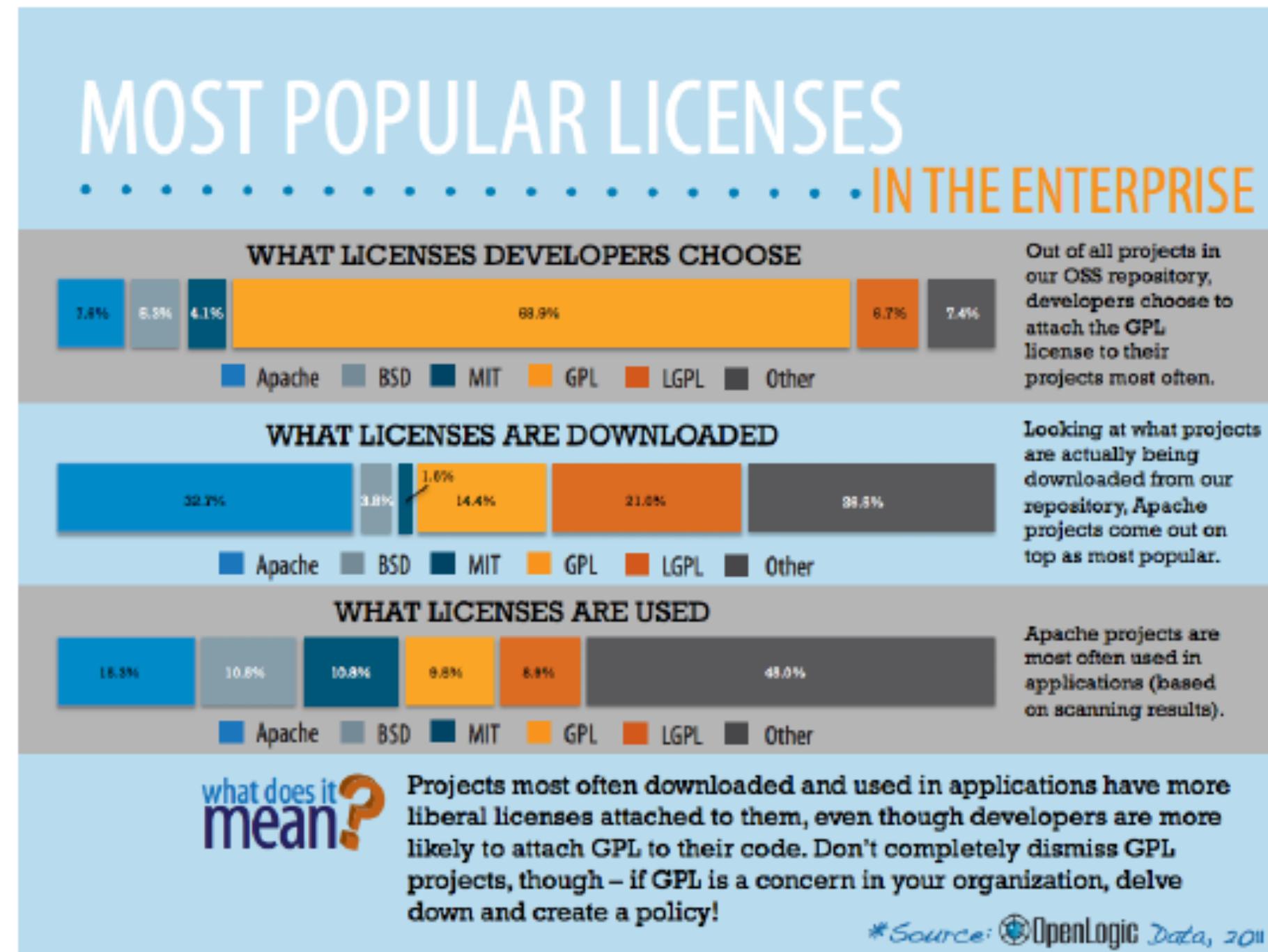


Matthieu Caneill, Daniel M. Germán, Stefano Zacchiroli

The Debsources Dataset: Two Decades of Free and Open Source Software.
Empirical Software Engineering, Vol. 22, 2017.



OpenLogic: most popular licenses in the enterprises



<http://www.openlogic.com/blog/bid/197148/open-source-software-101-understanding-compliance>

July 2012

Discussion: GPL vs BSD

- “BSD code is free, but GPL code stays free”
- copyleft: (non) business friendly?

- 1 Free Software licensing
- 2 License bestiary
 - Lax permissive licenses
 - Public domain
 - Scope-limited reciprocal licenses
 - Reciprocal licenses
- 3 Selected licensing topics
 - GPL v. linking
 - CAA/CLA
 - License popularity
 - **Free riding in the “cloud”**

Definition (Free-rider problem)

The **free-rider problem** occurs when those who benefit from resources, public goods, or services do not pay for them, which results in an underprovision of those goods or services

FOSS is at risk of free-riding, in particular when developers are not sufficiently funded and not enough users contribute back to address developers need.

Definition (Free-rider problem)

The **free-rider problem** occurs when those who benefit from resources, public goods, or services do not pay for them, which results in an underprovision of those goods or services

FOSS is at risk of free-riding, in particular when developers are not sufficiently funded and not enough users contribute back to address developers need.

Large “cloud” providers are prone to FOSS free riding, e.g.:

- take an existing FOSS component, e.g., a server-side DB
- integrate it so that cloud users can automatically deploy it, e.g., in a VM
- charge cloud users for the use of the service and/or the VM, e.g., by uptime, no. of requests, etc.

Anti free-riding licenses

A number of anti free-riding licenses are being experimented with to address the problem.

The balance between addressing the issue and retaining FOSS freedoms (and in particular freedom 0) is hard to achieve though.

To date, no such license has been recognized as free software and/or open source.

Commons Clause

Without limiting other conditions in the License, the grant of rights under the License will not include, and the License does not grant to you, the right to Sell the Software.

[...] “Sell” means practicing any or all of the rights granted to you under the License to provide to third parties, for a fee or other consideration (including without limitation fees for hosting or consulting/support services related to the Software), a product or service whose value derives, entirely or substantially, from the functionality of the Software.

— <https://commonsclause.com>

- clause meant to be added to other licenses, in particular (but not only) Apache
- does not claim to be FOSS; from the FAQ: “Q: is this open source?” “A: No.”
- adoption: extensions for Redis, see
<https://redislabs.com/community/licenses/>

Server-Side Public License I

Based on AGPL, replacing §13 with the following:

§13. Offering the Program as a Service.

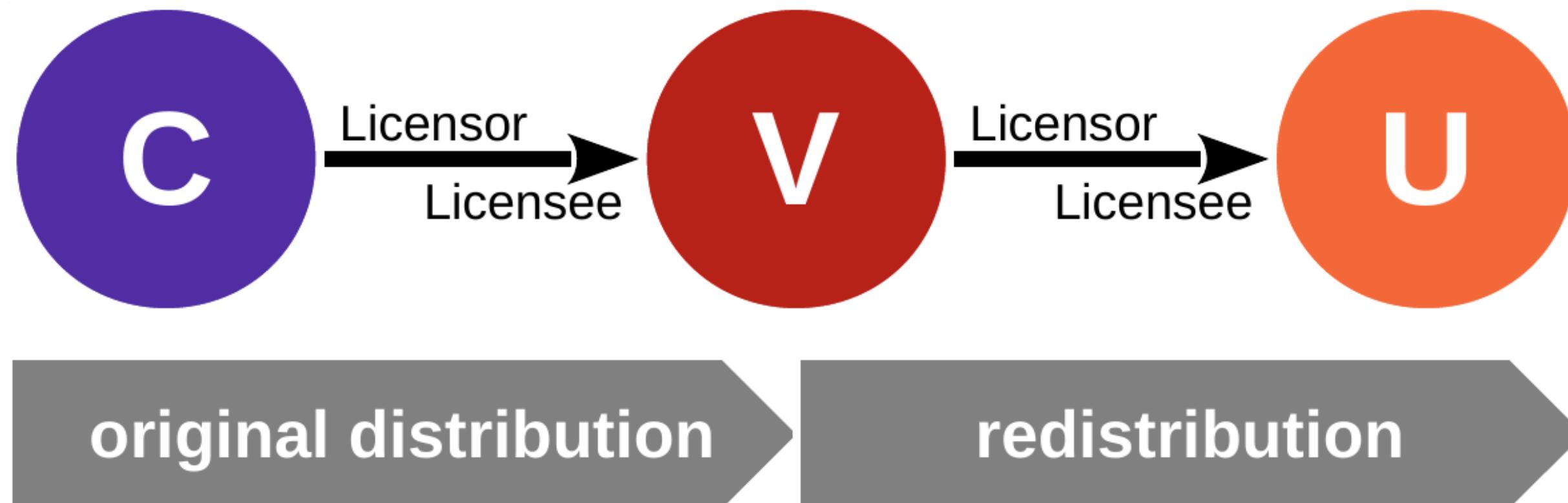
If you make the functionality of the Program or a modified version available to third parties as a service, you must make the Service Source Code available via network download to everyone at no charge, under the terms of this License. Making the functionality of the Program or modified version available to third parties as a service includes, without limitation, enabling third parties to interact with the functionality of the Program or modified version remotely through a computer network, offering a service the value of which entirely or primarily derives from the value of the Program or modified version, or offering a service that accomplishes for users the primary purpose of the Program or modified version.

“Service Source Code” means the Corresponding Source for the Program or the modified version, and the Corresponding Source for all programs that you use to make the Program or modified version available as a service, including, without limitation, management software, user interfaces, application program interfaces, automation software, monitoring software, backup software, storage software and hosting software, all such that a user could run an instance of the service using the Service Source Code you make available.

- <https://www.mongodb.com/licensing/server-side-public-license>
- adoption: MongoDB
- license submitted to OSI for approval, not accepted yet, mostly negative feedback thus far (February 2018)

Copyleft (“Reciprocal”) License

- Copyleft is a copyright-based strategy to ensure original rights cannot be curtailed when passed from licensor to licensee
- The licensee of copyleft-licensed software will have to use the same copyleft license (or later) when passing on the software



Quiz: Charging for Copyleft Code

- Can you charge someone before you give them access to copyleft-licensed software?
 - Yes
 - No

Free as in Freedom, not as in Beer

```
/*
* -----
* "THE BEER-WARE LICENSE" (Revision 42):
* <phk@FreeBSD.ORG> wrote this file. As long as you retain this notice you
* can do whatever you want with this stuff. If we meet some day, and you think
* this stuff is worth it, you can buy me a beer in return Poul-Henning Kamp
* -----
*/
```

- 1. License File**
- 2. File Header**
- 3. More ...**

Example License in File Header

```
/*
 * Copyright (c) 200X-200X by Humpty Dumpty
 *
 * This file is part of the Humpty Dumpty application.
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as
 * published by the Free Software Foundation, either version 3 of the
 * License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see
 * <http://www.gnu.org/licenses/>.
 */

package com.firm.model;
...
```

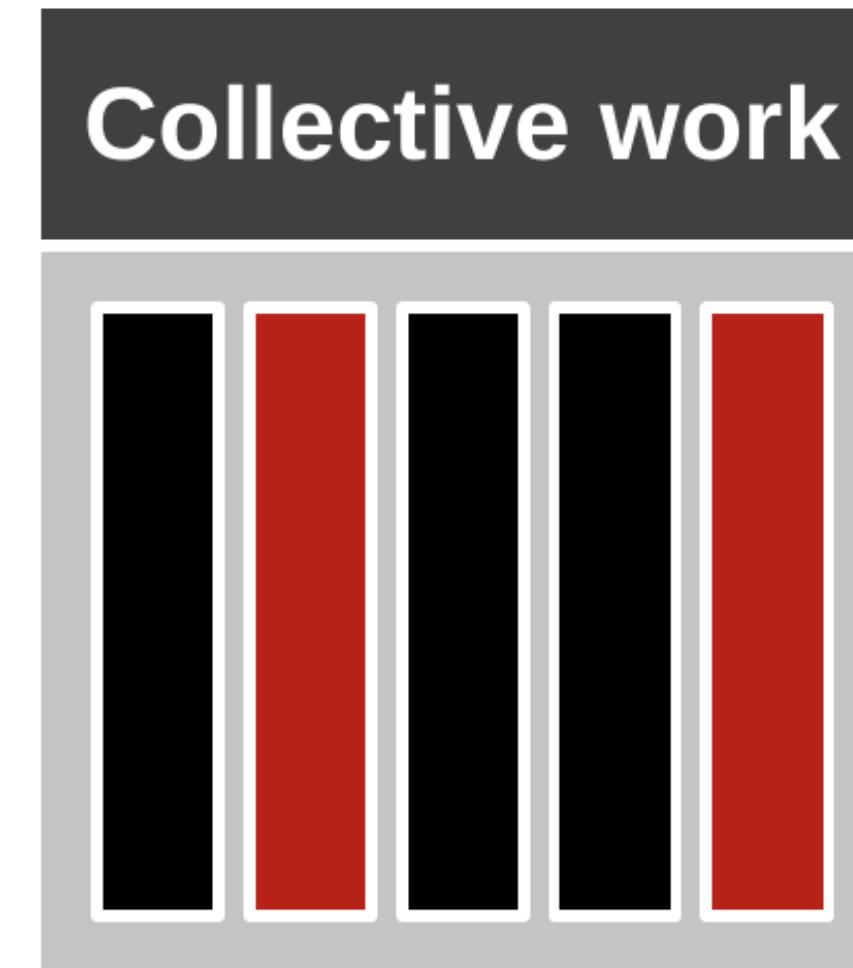
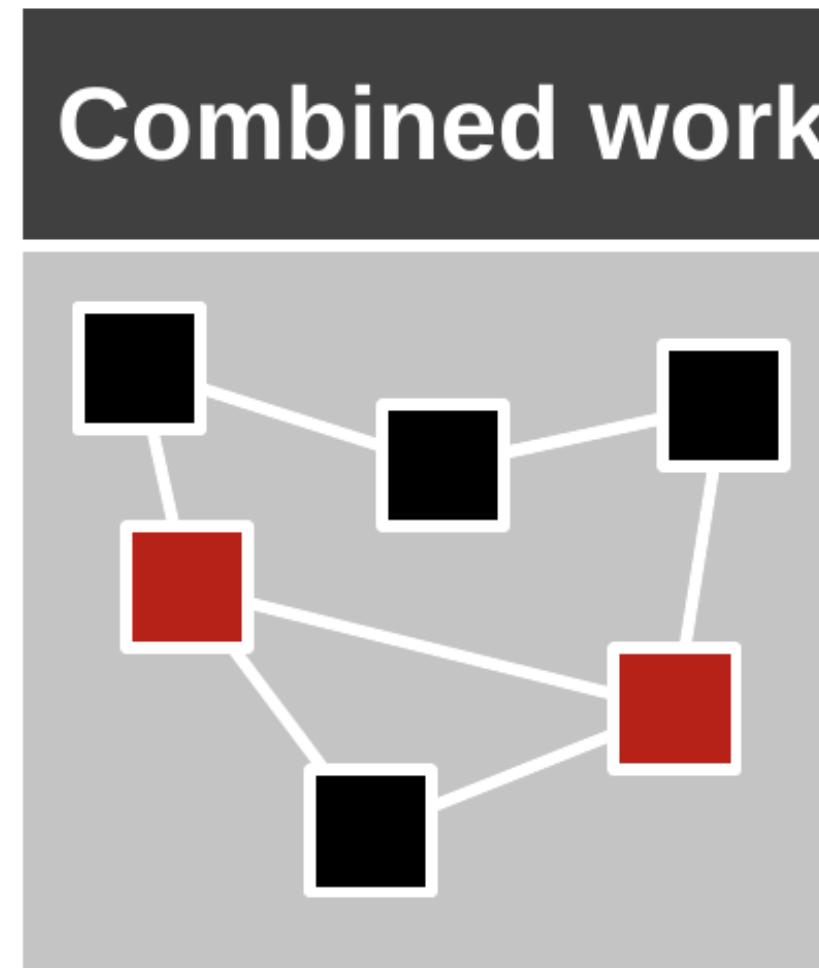
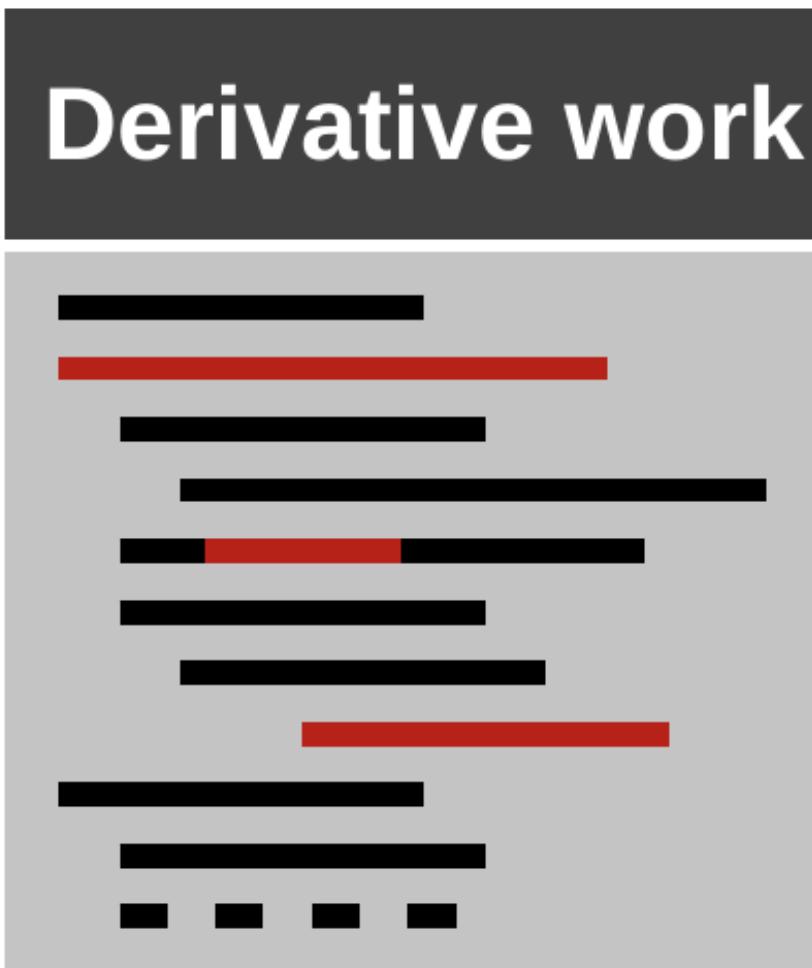
Freedom vs. Freedom

- Copyleft freedom
 - Freedom for the software, primarily
- Permissive freedom
 - Freedom for the developer, primarily

Collective vs. Derivative Work

- In U.S. copyright law,
 - A “collective work” is a work, such as a periodical issue, anthology, or encyclopedia, in which a number of contributions, constituting separate and **independent works in themselves, are assembled into a collective whole.**
 - A “derivative work” is a work based upon one or more preexisting works, such as a translation, musical arrangement . . . art reproduction, abridgement, condensation, or any other form in which a work may be recast, transformed, or adapted. A **work consisting of editorial revisions, annotations, elaborations, or other modifications** which, as a whole, represent an original work of authorship, **is a “derivative work.”**
- Copyleft effect only applicable to derivative, not collective works

Derivative, Combined, and Collective Work



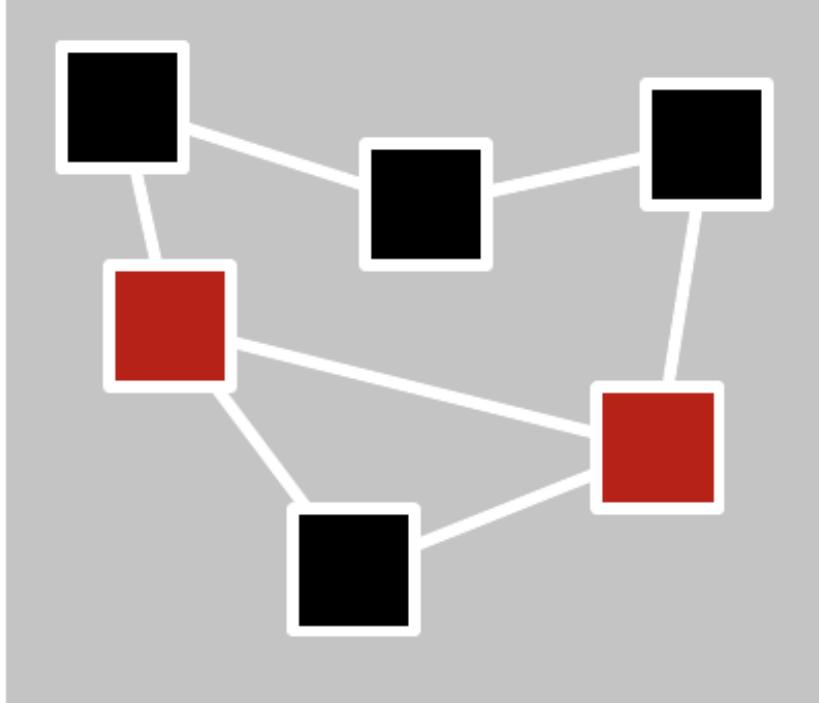
Copyleft Effect and Derivative Work

Licensor perspective on copyleft effect
Freedom of code for receiver of redistribution

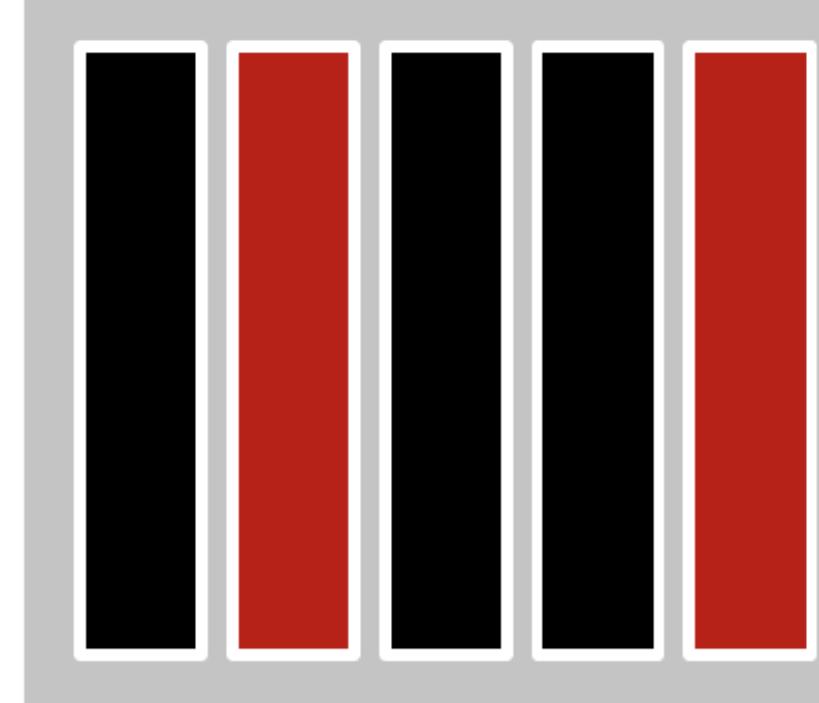
Derivative work



Combined work



Collective work



Licensee perspective on non-copyleft effect
Freedom of choice for licensor how to redistribute

Reciprocal vs. Permissive License

- GNU Public License v2
 - It permits
 - Use of the software
 - It requires
 - **Open sourcing of modifications upon redistribution**
 - Proper attribution
 - Inclusion of license in distribution
 - It forbids
 - Claims of endorsement
 - Use of FSF trademarks
- Apache License 2.0
 - It permits
 - Use of the software
 - **Redistribution of modification without open sourcing**
 - It requires
 - Proper attribution
 - Inclusion of license in distribution
 - It forbids
 - Claims of endorsement
 - Use of ASF trademarks

Critical License (Design) Issues

- **Triggers for having to open source**
 - **Definition of collective vs. derivative work**
 - **Definition of redistribution**
- Other obligations put on redistributions
 - Display of attribution, copyright notice
 - How to treat source code, provision of license files
- Other considerations
 - Provision of exceptions to standard license
 - Also, jurisdiction matters, no single global legal system

Quiz: Collective or Derivative Work?

- In a project, you use file X which is GPLv2 licensed
 - You change file X to become X'
 - Is file X' a collective or derivative work?
 - You create a class A in file Y which subclasses a class from file X
 - Is file Y a collective or a derivative work?
 - You create code in file Z which uses code from file X
 - Is file Z a collective or a derivative work?
 - You statically link file X and file Y into file A
 - Is file A a collective or a derivative work?
 - You dynamically link file Z with file X and deliver the files to your customer
 - Is file Z a collective or a derivative work?
 - You create file B, code of which makes a web service call to file X
 - Is file B a collective or a derivative work?

Decoupling Components Using Shims



The GPL License Family

Use Case	LGPLv1	LGPLv2	LGPLv3
	GPLv1	GPLv2	GPLv3
	-	-	AGPLv3
v1	LGPLv1	LGPLv2	LGPLv3
v2	GPLv1	GPLv2	GPLv3
v3	-	-	AGPLv3
Major Release			

New in the GPLv3 License Family

- Introduction of new use case “web service”
 - Affero GPLv3 license considers use of web service licensing of software
- Introduction of patent clauses, in particular patent retaliation
 - An attempt to stem the tide of (mostly) non-sensical software patents
- Increased compatibility with other open source licenses

Patents and Open Source Licenses

- Patent rights grant
 - §3 of the Apache License 2.0:
 - “Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted.”
- So-called patent retaliation clause
 - §3 of the Apache License 2.0 continued:
 - “If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.”

Contributor License Agreement

- A contributor license agreement
 - Is a contract that legally clarifies any contributions you make to a project
 - Needs to be signed by the owner of the rights (company or developer)
- At least two variants of the contributor agreement
 - **Copyright transfer:** The developer transfers all relevant rights
 - **Relicensing rights grant:** The developer grants relicensing rights
- The (stalled) Harmony project tried to unify these agreements
 - Apparently, unification of contributor agreements wasn't that important
 - Some argue that the open source license is enough

Developer Certificate of Origin [T08]

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

- (a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or
- (b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or
- (c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.
- (d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

Signed-off-by: Random J Developer <random@developer.example.org>
[lucky@maintainer.example.org: struct foo moved from foo.c to foo.h]
Signed-off-by: Lucky K Maintainer <lucky@maintainer.example.org>

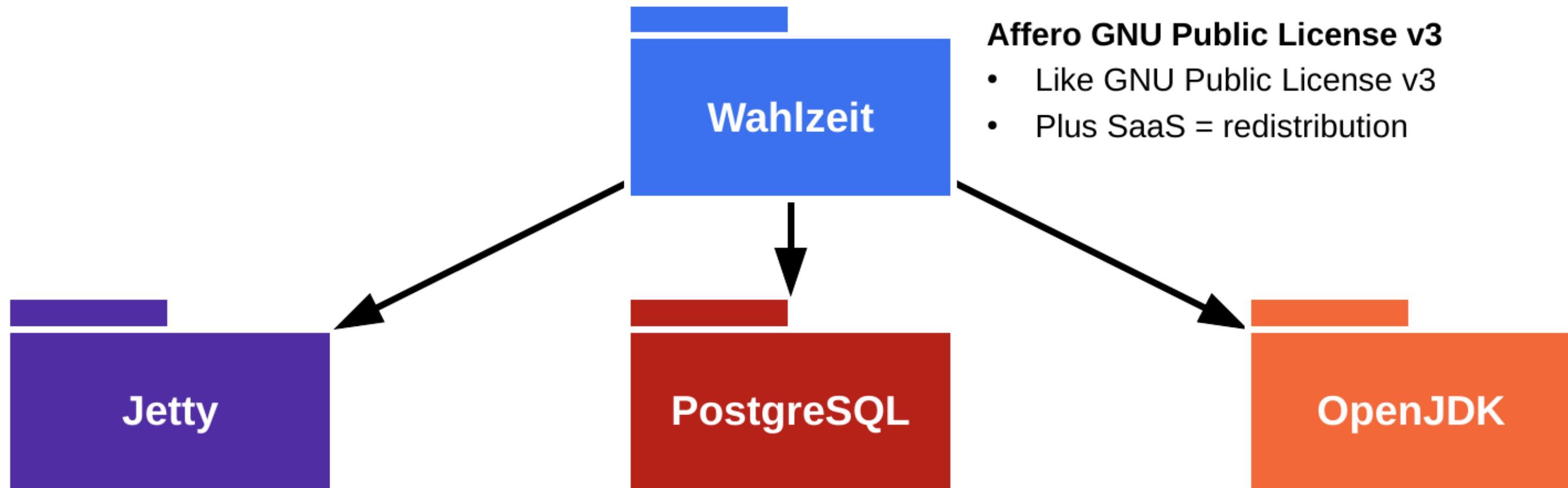
License Evolution

- Licenses often evolve over several versions (GPLv1, v2, v3, ...)
 - Licenses often allow for their replacement by a later version
 - This allows the copyright holder of the license to fix problems
- This allowed Wikipedia to move from the GFDL to CC-BY-SA
 - The FSF and CreativeCommons created special licenses for Wikipedia
 - By evolving through these “later” licenses, Wikipedia switched to CC

Dual and Multiple Licensing

- Some software source code has multiple licenses
- Users can choose the license they like best for their purposes
- Dual-licensed software can act as a shim between two components

Example Package Structure



Apache 2.0 License

- Copyright license [1]
- Patent license [1]
- Provide license, notice
- Provide attribution

PostgreSQL License

- Not reciprocal wrt source
- Requires copyright display
- No implied endorsement

GNU Public License v2

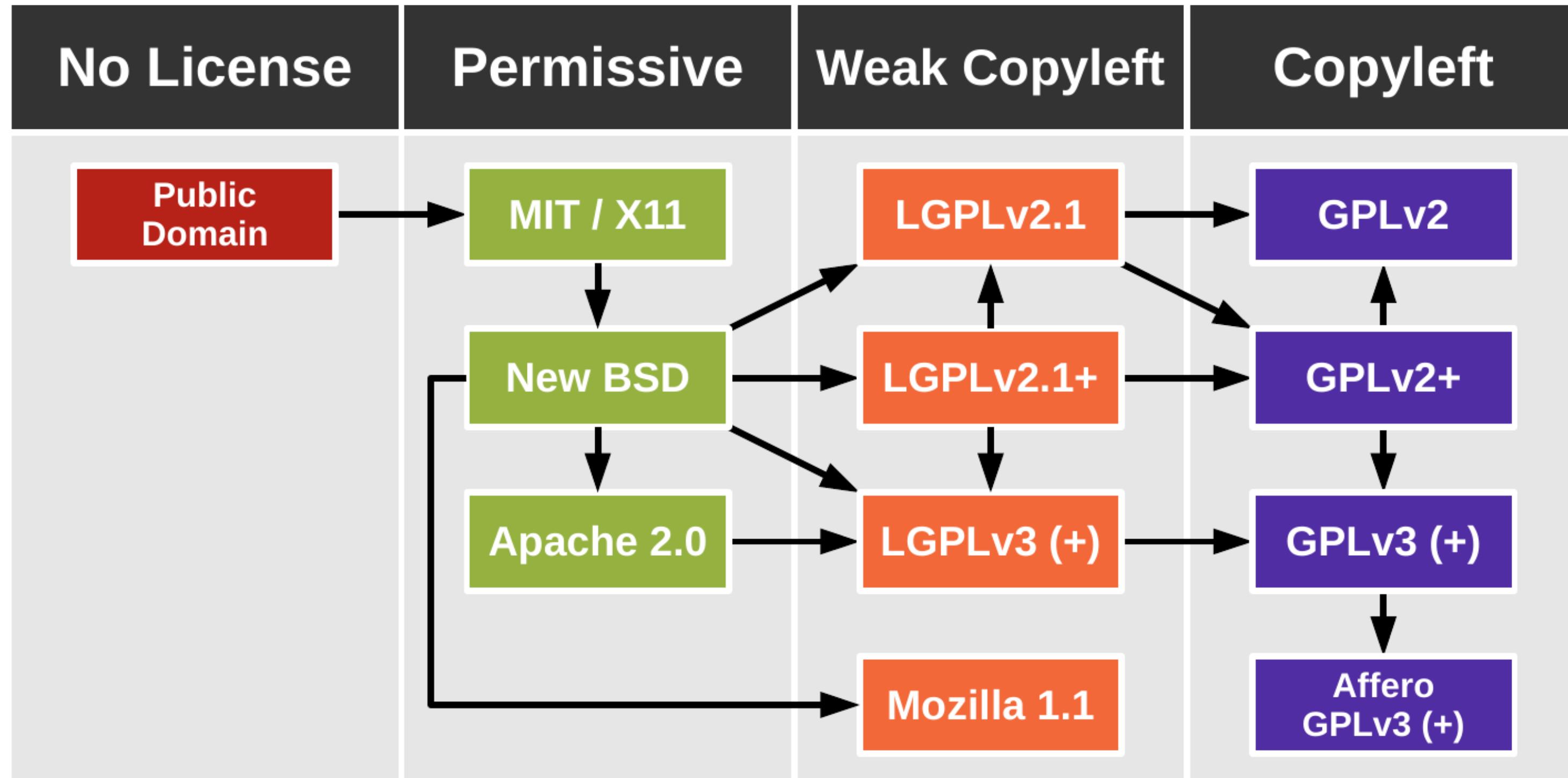
- Reciprocal: Provide source

With Class Path Exception

- Java linking = collective work

[1] Perpetual, worldwide, non-exclusive, royalty-free, irrevocable
308

The License Compatibility Circus [W09]



Crédits



Droit des logiciels

François Pellegrini
Professeur, Université de Bordeaux
francois.pellegrini@labri.fr

Ce document est copiable et distribuable librement et gratuitement à la condition expresse que son contenu ne soit modifié en aucune façon, et en particulier que le nom de son auteur et de son institution d'origine continuent à y figurer, de même que le présent texte.

Logiciel Libre

Cours 3 — Legal aspects

Stefano Zacchiroli
zack@irif.fr

Laboratoire IRIF, Université de Paris

2019-2020

URL <https://upsilon.cc/zack/teaching/1920/loglib/>
Copyright © 2014-2020 Stefano Zacchiroli
© 2013-2014 Roberto Di Cosmo
© 2008-2012 Miguel Vidal
License Creative Commons Attribution-ShareAlike 4.0 International License
<https://creativecommons.org/licenses/by-sa/4.0/>


Logiciel Libre

Cours 4 — Licenses

Stefano Zacchiroli
zack@irif.fr

Laboratoire IRIF, Université de Paris

2019–2020

URL <https://upsilon.cc/zack/teaching/1920/loglib/>
Copyright © 2014–2019 Stefano Zacchiroli
© 2008–2013 Miguel Vidal
License Creative Commons Attribution-ShareAlike 4.0 International License
<https://creativecommons.org/licenses/by-sa/4.0/>


Intellectual Property and Open Source

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

FOSS B02

Licensed under CC BY 4.0 International