Computers & Laws II: Software and Content Licensing

Kaido Kikkas
SPEAIT, Autumn 2023

2023 Kaido Kikkas. This document is dual-licensed under the GNU Free Documentation License (v l.2 or newer) and the Creative Commons Attribution-ShareAlike (BY-SA) 3.0 Estonia or newer license

Intro

- Recap from last week:
  - The two weird sentences in Latin
  - The WIPO universe (in particular, licenses and copyright)
- Software: free accessory => standalone product => …
- The Internet Age: a lot of software with a wide variety of legal regulations is available online…
- … and not just software (Scribd, Flickr, Slideshare, DeviantArt...)

“Do I have to pay for it, or not?”

- The typical question for most ordinary users
- The common typology: “commercial” vs “freeware”
- Actually, the rights that are given to the user are equally important – so the better distinction would be
  - **Proprietary** – software distributed as a product under well-defined, strict licenses (but not necessarily for money!)
  - **FLOSS** (Free/Libre and Open-Source Software) – distributed under equally well-defined but more liberal licenses (often only stating authorship), as a service rather than a product and usually (but not always) for free

“Yabba dabba doo!”

- Long ago, in the *Software Stone Age*,
  - scarce resources remained in commons (just as in primitive societies)
  - software (and IT in general) was elitary, for the elect few
  - no market existed for software
  - many key people were influenced by counterculture
  - many IT projects had military connections => no place for generic business
- So they mostly purchased/rented software together with the computer

Seventies: change of paradigm

- 1969 – for the first time, IBM sold software as a separate item
- AT&T started to license Unix to universities (v5; 1973) and companies (v6; 1975); access to source code started to get throttled
- 1979 – AT&T Unix v7, mostly closed source
- Mid-1970s – Apple and Microsoft born, both consider their software their property
- 1983 – IBM systems went closed-source; the Apple vs Franklin court case affirmed copyright as applicable also on compiled software (not just source)

William H. Gates III

- 1972: Traf-O-Data (Lakeside, Seattle) - Bill Gates and Paul Allen
- 1973: Bill goes to Harvard and drops out soon (too much computers and poker)
- 1974: MITS Altair. Bill and Paul found Micro-Soft
- 1975: Altair BASIC, the first widespread proprietary application. *Open Letter to Software Hobbyists* - proprietary software born together with warez

The Golden Years of Microsoft, Part I

- 1981: IBM PC. MS obtained QDOS
- Bill Gates licensed MS-DOS to IBM
- 1983/84 Apple Lisa ==> Windows
- 1990: Windows 3.0 (and 3.1/3.11)
- Conquering the market

Some success factors

- Very good at catching the moment (throughout the history)
- The first to really target “the ordinary user"
- Risky, legally borderline and obnoxious style of marketing
- More or less defined the current proprietary software licensing

The 90s setback

- Several factors that challenged the position of MS in early 90s:
  - Internet moves within reach for everyone
    - Source of warez
    - Old-school ideas of software surface again
    - People learn that there is more than just MS
  - Missing the early momentum of the Web
  - Alternative players start to appear
  - Growing interest of anti-monopoly forces

GNU and Linux

- 1984: Richard M. Stallman founded FSF with a goal to develop GNU (Gnu's Not Unix), a free alternative to Unix. Has not fully ripened yet - but developed a lot of useful software (the GNU utilities) and, most importantly, the GNU General Public License
- 1991: Linus Torvalds released the first version of Linux kernel (“just a hobby”) that soon moved to the GPL – the beginning of FLOSS

Another ‘right time, right place’

- There were Unix for large enterprise, Microsoft and Apple for smaller companies, but no proper hacker system for serious hobbyists, schools/universities and startups
- Overlapped with the first Web explosion – birth of online, community-based development models
- 1992-93 – three free BSD Unixes; 1995 – the LAMP bundle
- An “Internet server” started to mean a machine that was cheaper by an order than before

The Browser Wars (revisited)

- 1994: Netscape Navigator
- January 1995: MS browser developed by four people
- July 1995: MS proposed division of the browser market, Netscape refused
- MS acquired Spyglass and developed IE out of it, using a lot of power play to get rid of the competitor. The wars ended in 1997 with Netscape being defeated and bought up by AOL
- Netscape prototype called Mozilla open-sourced in 1997
- Alternate browsers developed fast, IE stagnated for 3 years (6.0) – the wars continued

The Golden Years of Microsoft, Part II

- 1997: MS had won the first browser war, PC software market ruled by Win + Office
- IE tied to Windows ==> almost impossible to dethrone (later, the same was tried with MS Media Player)
- Uncle Sam (and later, EU) did not like it

Uncle Sam and other angry folks

- 1995: MS forced to give up swallowing Intuit (a company making financial software)
- 1997: MS found guilty in monopolism (Windows + IE), 1 MUSD daily fines
- November 1998: An inner document leaks from Microsoft, pointing at Linux as a serious threat (as opposed to dismissal by official rhetorics) and suggesting (quite dirty) measures. The scandal that followed stopped MS marketing campaigns for half a year
- http://www.catb.org/~esr/halloween/

The Halloween Documents 1998

- By Vinod Valloppillil, a then-engineer at MS
- Sent to the top management
- Warns of Linux as a serious contender, due to immunity to the prevalent market strategies of the company
- Existing measures considered ineffective, suggested “embrace, extend and extinguish” and massive propaganda (including baked data and paid-for “independent studies”)

Later developments

- Microsoft and Apple still largely rule desktop, Android (Linux kernel, largely free but includes proprietary stuff) rules on mobile devices, business servers are split, heavy iron runs on penguins (see top500.org!)
- MS: Gates and Ballmer hated FLOSS, Satya Nadella is ambiguous (Azure, Codeplex, Linux Subsystem in Windows); Today's MS is much more Linux-friendly, but still uses dirty tricks (OOXML, UEFI etc)

Software licenses: an overview

- Proprietary software
  - Commercial proprietary software
  - Shareware
  - (proprietary) freeware
- FLOSS
  - Different accents:
    - Free Software (ethical-philosophical approach)
    - Open Source (pragmatic-technological approach)

Free?

- “Free as in freedom, not as in free beer” - Richard M. Stallman
- Not separated by money – RHEL takes money, IE/Edge does not; yet the former is FLOSS, the latter is proprietary
- The point is in user freedom

Commercial proprietary software

- Detailed licenses with as little rights conveyed to users as possible by EULAs. Most of Windows and OSX, some Linux (has made a comeback with Steam)
- Two trends: license packs (wholesale) allow more flexibility, but EULAs go more strict and intrusive
- Main use: professional applications (e.g. CAD, business graphics and multimedia), games. Two earlier bastions (office and development tools) are shaking – FLOSS + Google/cloud!
- Product support and warranty are promoted as an essential part of the deal

Shareware

- Freely copied “try before you buy” software (typically around 2-4 weeks). After the period, one should either stop using it or register it. Types include
  - Adware (not its malicious namesake!) – shows ads in the unregistered form, registering gets rid of them
  - Nagware – constantly reminds, sometimes hinders usage
  - Crippleware – disables some important functions (e.g. saving the work)
  - Demo/trialware – cut-down versions of large commercial applications
- Main use: small apps, utilities. Support does sometimes exist
- Shareware was popular in the 90s, a modern spiritual successor is *freemium*

Proprietary, zero-price (“freeware”)

- “Free at this moment”, may have limitations
- Main use: utilities; a big exception was IE (and later, MS Media Player)
- Different versions can have different licenses (old ones released as freeware)
- Some more interesting types:
  - “send me” software: postcardware, cokeware…
  - “Not free for Some Bad People” - army, psychiatrists...

Free Software vs Open Source

- **Free Software** – social and ethical standpoint: “free because it is a right thing to do”; closed source is an ethical problem
  - Richard M. Stallman and his school of thought
  - Free Software Foundation, https://www.fsf.org
- **Open Source** – more pragmatic, technological standpoint: “free because it makes more sense”; closed source is a suboptimal technical solution
  - Eric S. Raymond, Bruce Perens, Linus Torvalds etc
  - Open Source Initiative. https://opensource.org
- Two schools climbing different sides of the same mountain

The Four Freedoms by FSF

- The freedom to **run** the program as you wish, for any purpose
- The freedom to **study** how the program works, and change it so it does your computing as you wish. Access to the source code is a precondition for this
- The freedom to **redistribute** copies so you can help your neighbor
- The freedom to **distribute** copies of your **modified** versions to others. By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this

GNU GPL: main points

- Starting point: **authorship** (thus, even free software makes use of copyright)
- Four **freedoms** of users
- Redistribution must retain data about previous authors, every contributor adds his/her own. Also to be added is the full text of GPL – the license “sticks”
- A couple of things to consider:
  - No discrimination on usage
  - No need to disclose in-house changes if not redistributed

Things people ask

- Can ask money, yet demands free distribution? The model can actually work (books, music)
- Warranty and support exclusively stated as complementary services ==> most software distributed 'as is'. BUT: both support and warranty are essentially free market for everyone interested (connection to original author is not needed)
- SaaS (*software as a service*) – one can pay for support, know-how, certs, training, manuals, stuffed penguins...

Copyleft

- The license transfer clause
  - **None**: derived software does not have to share the original license (can also turn proprietary). E.g. X11, BSD, Apache
  - **Weak**: some derivates do not have to pass on the license. E.g. GNU LGPL, MPL
  - **Strong**: license will go along with all derived works. E.g. GNU GPL
  - **Extra strong**: also covers the 'ASP loophole' or usage over networks (considered as redistribution). E.g. AGPL
- Some math: 0 + 1 = 1
- A general principle: strong copyleft favours users (continuity of freedom), weak favours developers (more choice)

GPL v3

- Summer 2007
- Initially many opponents (incl Linus Torvalds), fewer by time
- Clauses against 'tivoization' and software patents (basically, if you use software under GPL 3 and proceed to enforce your software patents, the act invalidates the license)
- Better compatibility with other free licenses, but it remains an issue

EUPL

- EU Public License
- 2007, most recent amendments from 2021
- Equally valid in all EU languages!
- Designed to be compatible with most common copyleft licenses (from LGPL to AGPL, also both GPL 2 and 3)

A comparison of various licenses

License	Always free	Free to copy	Free to use	Open source	Can be modified	Derivates	Copyleft
Commercial proprietary (EULA)	no	no	depends on the license	no	no	no	no
Various restricted licenses (e.g. MS RSL)	restricted	restricted	depends on the license	limited access	depends on the license	no	no
Shareware	restricted	restricted	restricted	no	no	no	no
Proprietary non-commercial freeware	restricted	yes	no	no	no	no	no
Proprietary freeware	yes	yes	yes	no	no	no	no
Public domain software (only in common law countries)	yes	yes	yes	sometimes, but not required	sometimes, but not required	sometimes, but not required	no
Free software licenses with no copyleft (BSD, X11, MIT, Mozilla etc)	no	yes	yes	yes	yes	yes	no
Free software licenses with weak copyleft (GNU LGPL)	no	yes	yes	yes	yes	yes	somewhat
Free software licenses with strong copyleft (GNU GPL, EUPL)	no	yes	yes	yes	yes	yes	yes
Free software licenses with very strong copyleft (AGPL)	no	yes	yes	yes	yes	yes	yes, including use over networks

Online content

- There is plenty of other things online than software
- Proprietary content is ‘business as usual’ - but how is free content regulated online?
- Stallman proposed the Free Documentation License
  - Initial success (incl Wikipedia)
  - Later criticism (legal ambiguities,, hard to understand)
- In 2002, Prof. Lawrence Lessig founded Creative Commons
- Main point: increasing use of free software principles within the existing legal framework

Creative Commons: the third way

- Middle of the road needed:
  - “IP traditionalists” say “Copyright: all rights reserved”
  - Stallman quips “Copyleft: all rights reversed”
  - Creative Commons suggests “Some rights reserved”
- Two goals
  - Allow the authors to license their work in a simple and flexible manner
  - Create a transparent and easily understood licensing system

Simple steps

- A family of sometimes quite different licenses!
- The easiest way: go to https://www.creativecommons.org and answer a couple of questions:
  - Can your work be used commercially (yes/no)
  - Can others derive new works – three options:
    - Yes
    - Yes, if they use the same license (copyleft!)
    - No
- In addition, the author has a chance to pick jurisdiction and the form of the work (photo, text…)

The CC Trinity: one license in three forms

- Commons Deed - human-readable, 1-page summary
- Full license – lawyer-readable, usable at courts
- RDF/XML metadata, for semantic search online

The Commons Deed for CC BY-SA

The main CC licenses

- CC Zero (0)
- CC Attribution (BY)
- CC Attribution-ShareAlike (BY-SA)
- CC Attribution-NoDerivs (BY-ND)
- CC Attribution-NonCommercial (BY-NC)
- CC Attribution-NonCommercial-ShareAlike (BY-NC-SA)
- CC Attribution-NonCommercial-NoDerivs (BY-NC-ND)

CC licenses used earlier (not anymore)

- Sampling Plus – partial free use, non-commercial use only for the full work
- NonCommercial Sampling Plus – only non-commercial
- CC Music Sharing <= BY-NC-ND
- CC Wiki <= BY-SA
- CC GNU GPL and CC GNU LGPL
- CC Founders' Copyright – 14+14 years
- CC DevNations – CC BY for developing nations (the World Bank list), strict © for others

CC in Estonia

- 2010, by the OER (*Open Educational Resources*) community
- 6 CC 3.0 licenses harmonized with Estonian legal system:
  - Autorile viitamine (BY)
  - Autorile viitamine + jagamine samadel tingimustel (BY-SA)
  - Autorile viitamine + mitteäriline eesmärk (BY-NC)
  - Autorile viitamine + mitteäriline eesmärk + jagamine samadel tingimustel (BY-NC-SA)
  - Autorile viitamine + tuletatud teoste keeld (BY-ND)
  - Autorile viitamine + mitteäriline eesmärk + tuletatud teoste keeld (BY-NC-ND)

Concluding remarks

- The 'intellectual property' approach is out of date
- The fight grows more intense and political – there is a danger that profoundly technological decisions will be increasingly made by politicians (more tinfoil to them!):
  - Technological incompetence
  - Hidden interests
- On the other hand, the only way to stop FLOSS is to ban it
- Hybridization of business models

For further reading

- Web:
  - wipo.int
  - creativecommons.org
  - fsf.org
  - opensource.org
- *Free Culture* by Larry Lessig
- *Information Liberation* by Brian Martin
- + books by Eric v. Hippel, Paul Graham, Steven Levy, Yochai Benkler and others

Thanks

</legal_stuff>