What is the relation between software licenses, CO_2 emissions, energy consumption and e-waste?

December 22, 2024

Preface

Most of us don't suspect there could be a relation between software licenses, CO_2 emissions, energy consumption, and e-waste.

The aim of this issue is to provide the reader with an idea of why that relation could be.

Before we do that, since I don't know how familiar you are with how much the ICT (Information and Communication Technology) industry pollutes, I will list some numbers.

-Roman

ICT, CO_2 emissions, energy consumption & e-waste

- CO_2 emissions of the ICT industry is estimated to be between 1.8% and 3.9% of global CO_2 emissions[1] - at par with the aviation industry.[2]
- many times since 2007, e-waste has been found to be the fastest growing segment of waste.[3]
- e-waste has grown 5 times faster than it can be recycled since 2010, the research arm of the UN wrote in 2024.[4]
- "since 2012, the amount of compute used in the largest AI training runs has been increasing exponentially with a 3.4-month doubling time [...]."

  In other words, "since 2012, this metric has grown by more than 300,000x [...]."[5]

  This number is so big that it is hard to grasp; the Association for Computing and Machinery summarised the increase:

  > it's exploding.[6]

That's it for numbers.

Back to our premise:

> what do software licenses have to do with CO_2 emissions, energy consumption, and e-waste?

What is the relation between software licenses, CO_2 emissions, energy consumption and e-waste?

To get an idea of why paying attention to software licenses matters when trying to reduce CO_2 emissions, energy usage and e-waste, we can look at these 3 questions:

1. can licenses of sofware we rely on hinder our ability to reduce energy consumption?
2. can licenses of software we use prompt us to increase the pile of e-waste when a device could otherwise carry on being used?
3. can licenses of devices hinder people's ability to keep using devices once purveyors no longer support them?

Can licenses of sofware we rely on hinder our ability to reduce energy consumption?

The German government environmental agency[7] published a report[8] where they compared energy usage of a proprietary word processor, presumably Microsoft Word (they don't say), with an Open Source word processor (Libre Office maybe).

The test shows that the proprietary word processor requires 3.87 times more energy than the Open Source word processor, for "standard usage" and "perform[ing] the same tasks".[9]

I am not writing about this example to tell you that all proprietary software consume more energy than Open Source software.[10] That is not the point.

I am mentioning this study to bring up this question:

> if a software consumes too much energy, do we have the agency (the ability) to do something about it?

To answer this question, we can look at software licenses.

> Why?

While anyone can study the source code of Libre Office, understand what it does and optimize the programme for energy usage, for example by killing frivolous tasks; no one, other than Microsoft, knows what Microsoft Word does under the hood, or can modify its source code. Only Microsoft's people can.

That's not to say that Microsoft can't reduce the energy its software consume, they would certainly like to do so.

However, I see two caveats:

1. we know we can only stick to one priority; can reducing energy consumption become the priority of purveyors of (proprietary) software given the incentive mechanisms in place for their organisations to sustain?
2. Can legislators legislate the activities of purveyors of (proprietary) technologies?

   Which is actually asking:

> how can the law apply to technologies whose code is distributed as binaries (i.e. secret to legislators and the law)?
>
> Or in other words, how can we write (legal) code, to legislate, or apply the law to, technologies whose code no one other than its purveyors can read and write?[11]

Each of these questions should certainly be discussed on their own, unfortunately I can't do this here if I want to keep this draft concise. However, the reason I am mentioning these questions is to raise the idea that binaries probably hinder our ability, individually or collectively, to have agency over what a software does, and so:

> if we want to have agency over the energy technologies consume, shall we first gain agency over technologies themselves by using software which source code is freely distributed?[12]

I am not saying licenses are a panacea.

I am suggesting proprietary licenses, as well as the structure and incentive mechanisms of organisations writing proprietary software, might jeopardize our ability to reduce energy consumption. Therefore, if we want to have agency over the energy software consume, I am suggesting that we might be better off using technologies whose souce code is distributed freely, and probably think of how the governance of software infrastructures and software development we use is organised.[13]

That is it for question one. Next we look at e-waste. For this, we try to clarify the relation between licenses and hardware churn.

Can licenses of software we use prompt us to increase the pile of e-waste when a device could otherwise carry on being used?

Users of Apple's products will know that after a few years, the operating systems of their iPhone (or Macbook) won't receive updates from Apple.

As a result, users will no longer be able to update the software they use, or install the newest software. The software updater will complain that the operating system is too old or no longer supported.

Users can muddle through for a bit, and try to keep their devices, but sooner or later they will run into security or compatibility issues.

Eventually, they will be forced to discard their devices and buy a new one; possibility increasing the pile of e-waste.

You'll know this is not just a problem for Apple's users. Microsoft's users are in the same boat.

When Microsoft moved from Windows 10 to Windows 11, some users might have found that their computers did not meet the requirements to run Windows 11. This is a recurring problem. Each time there is an update, some machines won't meet the new requirements.

If we think of the billions of devices in use, and the many updates to come, then we can start thinking about the number of machines that might increase e-waste each time there is an update: how many users will get stuck because their machines can't support an update? And how many computers will become e-waste?

A research firm estimated at 240 millions the

number of computers that could go to waste due to the update from Windows 10 to 11.[14]

I don't know what this number is worth. The problem I want to remind us of, is that - each update - potentially nudges people to discard an otherwise functional machine.

We know about this, and it is not hard to imagine that sooner or later the devices produced will end up as e-waste, so one might ask:

> what are the alternatives? How could we do otherwise? How could we even blame users or the ICT industry?

We live in a world where there is a race to provide users with the best experience, with devices and applications that can do more, faster, better, *etc.* while energy usage, or the e-waste these machines will become, isn't really part of most people's consideration: most of us say we care, yet, at the end of the day, most of us have at least two devices[15] and, so far, most of us have kept adopting devices as the industry feeds us more.

In other words, we think that we would like to do something, but we keep tagging along as product cycles prompt us to ditch old devices and adopt the latest.

Furthermore, most of us are unable to imagine how we could do otherwise.

And how could companies which survival is tied to constant releases of new programmes or devices could think of slowing down, or extending the lifespace of devices they distribute? Is the underlying structure, or the incentive mechanisms of these organisations, the root of the issue?

I don't know, but looking at software licenses might be a start.

Could licenses help us gain agency on the duration of the life cycles of our devices, and produce less e-waste? Could licenses help us imagine how we could organise software infrastructures and its development differently?[16]

Could we imagine using a device for 10, 20, 30 years? Shall extending the time we keep our devices become what we mean by progress?

In other words, what would it take to gain agency over the lifespan of technologies?

We discuss this question in the next section where we see how we could keep using devices longer.

Can licenses of devices hinder people's ability to keep using devices once purveyors no longer support them?

I don't know how much you are familiar with operating systems based on GNU/Linux, but oftentimes computers which are no longer powerful enough to run MacOS or Windows are still powerful enough to run GNU/Linux. Users can give their computers a "second life" by installing GNU/Linux on their machines.

I myself use a laptop that was released in 2012, which I assume ran Windows for a few years, and then was no longer able to do so confortably. I bought it second-hand in 2022 and I've been using it since as my main computer.

So those 240 millions computers I wrote about earlier, that will eventually no longer be able to run Windows, might not necessarily become e-

waste (at least for another few years), if we can install GNU/Linux on them.

That said, installing GNU/Linux, or another third-party operating system, isn't a given.

Purveyors don't always make it easy for users to keep using devices once support ends.

> Why is that? Why can we extend the life of our devices by installing the operating systems of our choice once our devices can no longer run MacOS or Windows?

I will give a few examples, and explain why these issues are related to licenses.

Some iPod aficionados are able to carry on using iPod generations 1, 2, 3, 4, 5 by porting (adapting) Linux onto these devices, even though Apple deemed these products obsolete.[17]

Unfortunately, that only worked up until to a point.

People can't port Linux onto Ipod generation 6.

> Why is that?

Ipod generation 6 shipped with encrypted firmware.[18] In other words, no one else other than Apple can see or understand how the device works. This restriction makes it near impossible, or very complicated and time consuming, for others to write bits of programmes that would extend the life of the device. So far, software programmers have not been able to port (adapt) Linux on these iPods.[19]

That is the worst case scenario; when purveyors

lock out a device they distribute so no one else can tinker with it.

It is not always as bad as the whole device being locked out though.

Sometimes users are able to re-use or repurpose a device, still, not all features of the device function.

For example if you try to install GNU/Linux on a computer, you might find that the sound, the Wifi, some keyboard keys, or ports, won't function. For most of us, these issues make a device useless, and so possibly prompt us to discard this device, and a buy a new one.

Again, that is a licensing issue.

When the source code of parts of a device isn't distributed, when the firmware is encrypted, when users collectively can't know how a device works, it hinders the work of those trying to port (adapt) Linux to extend the life a device for many. As a result, users get locked out from using the device they've purchased.

Some of us might think that, that's how things are. Maybe we've been used to things being this way - or maybe we've never even thought that we should be able to keep using a device once its purveyors stop supporting it. Yet these possibilities exist. Not all technology producers lock out devices they sell.

Some companies, like MTN Reform, produce laptops whose source is distributed.[20] Anyone can study how these laptops work, manufacture spare parts, and even produce whole laptops for commercialisation.[21]

To conclude discussing this question; if we can see other possibilities exist:

can we move away from using devices which are locked out?

Shall we, as users or companies, expect usage to be unrestricted when we acquire a device?[22]

Conclusion

If we can see that software licenses or goverance models of software development and distribution can hinder our ability to reduce CO_2 emissions, energy consumption or e-waste, and that to reduce these, we might want to have agency over technologies, shall we move away from usage of technologies which is restricted and adopt technologies which we can have agency over?

Of course, to write that the distribution of source code can reduce e-waste, isn't to write that it is a panacea to reduce it all. This isn't also to say that the software industry won't be able to switch to sustainable practices.[23] But, this is to raise the idea that by paying attention to the licenses of software we use, whether we are a technological purveyor, a company or a user, we might foster our ability to reduce pollution that stem from our digital activities.

I thank Joseph P. De Veaugh-Geiss of KDE whose presentation titled Software Licensing For A Circular Economy *prompted me to write this issue. Some examples I got from his presentation. You can watch it there:* `https://media.ccc.de/v/37c3-12047-software_licensing_for_a_circular_economy`, accessed in February 2024.

Postscript on water consumption

As I was writing this issue, each time I was searching for information about how much the ICT industry pollutes, I found data about how much water data centers consume, especially since companies have been running AI models.

I'll just copy two sentences that caught my attention:

- ChatGPT consumes 1/2 of water (to cool down data centers) for every 10 to 50 answers it provides to users.[24]
- "[T]he global AI demand may be accountable for 4.2 [to] 6.6 billion cubic meters of water withdrawal[25] in 2027, which is more than the total annual water withdrawal of 4 [to] 6 Denmark or half of the United Kingdom."[26]

These increases in water consumption also bring us back to the question of whether it is resonable to expect organisations to move away from practices, while their survival is tied to entropic business models (systems or models which existence consumes the environment which makes their existence possible).[27]

Notes

[1]Freitag, Charlotte et al. “The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations.” Patterns (New York, N.Y.) vol. 2,9 100340. 10 Sep. 2021, doi:10.1016/j.patter.2021.100340

[2]According to the International Energy Agency, "[i]n 2022 aviation accounted for 2% of global energy-related CO_2 emissions" `https://www.iea.org/energy-system/transport/aviation`, accessed in October 2024.

[3]according to reports by the UN, the International Labour Organization, the EU, research papers *etc*.

[4]The Global E-Waster Monitor 2024, Baldé, Kuehr, Yamamoto, McDonald, D'Angelo, Althaf, Bel, Deubzer, Fernandez-Cubillo, Forti, Gray, Herat, Honda, Iattoni, Khetriwal, di Cortemiglia, Lobuntsova, Nnorom, Pralat, Wagne

[5]"AI and compute", Dario Amodei, Danny Hernandez, 2018, `https://openai.com/index/ai-and-compute/`, accessed in October 2024.

[6]Tech Brief, November 2021, Issue 1, Association for Computing and Machinery, `https://dl.acm.org/doi/pdf/10.1145/3483410`, accessed in October 2024.

[7]Umweltbundesamt

[8]*Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik* von Jens Gröger, Köhler, Naumann, Filler, Guldner, Kern, Hilty, Maksimov, December 2018

[9]The test finds that the proprietary word processor needs 3.6 watt when the Open Source word processor only needs 0.93 watt to do the same tasks.

[10]They don't always do, for example with browsers; see the same study.

[11]If this question is foreign to you, I can help; or, to get some idea of how proprietary code can impede the legislative apparatus, one can look at the work of Lawrence Lessig, *Code 2.0*, 2001.

[12]Free does not mean gratis; for more information see: `https://yctct.com/free`

[13]That does not mean that we have to come up with new models or DIY, not at all; other governance models of digital infrastructures and developments have already been in place, at least since the 1970s, and we can benefit from them.

[14]https://www.canalys.com/insights/end-of-windows-10-support-could-turn-240-million-pcs-into-e-waste

[15]Two devices is conservative. I have seven devices at home and at the office: two laptops running GNU/Linux, one which I use as my main computer and another one which I repurposed as a server, an e-reader and four deprecated devices (two phones and two laptops) sitting in my cupboard.

[16]Again, we don't have to start from scratch; many models have been in use since the 1970s; we just don't see them.

[17]See the list: *iPod products obsolete worldwide*, https://support.apple.com/en-us/102772

[18]Vendors often argue that encrypted firmware is for the security of users; some security pundits dismiss this argument and warn us that this is *security through obscurity* ("the practice of concealing the details or mechanisms of a system to enhance its security [...] Security by obscurity alone is discouraged and not recommended by standards bodies. The National Institute of Standards and Technology (NIST) in the United States recommends against this practice: "System security should not depend on the secrecy of the implementation or its components.""–Wikipedia)

[19]http://www.ipodlinux.org/Project_Status/

[20]as in unrestrictedly; not to confuse with gratis, for more explanation see https://yctct.com/free. Also, it is tempting to think that there is no business models if the source code of a technology is distributed freely. This is a mistake. Free distribution and profit aren't oxymoron. I like to use cooking recipes as an example of information that is distributed freely, yet which fosters business (recipes cannot actually be copyrighted). We can discuss this further.

[21]We could also discuss business models of companies which distribute the source code of their technologies; coincidentally, I also think that distributing the source code strengthens a business model.

[22]Is that what people from the movement 'The Right to Repair' are asking?

[23]Microsoft for example first rejected using Open Source

licenses; now they release some of the code they write as Open Source; so, could they in the future evolve further and move to a business model where the source code they write is distributed freely? (I am suggesting Microsoft could do better by distributing source code freely rather than using Open Source licenses because Open Source licenses can restrict usage; for more information see: *Open Source misses the point* `https://agency.yctct.com/open-source.html`)

[24]""GPT-3 needs to "drink" (i.e., consume) a 500ml bottle of water for roughly 10-50 responses", depending on when and where it is deployed." Li, Peng, Jianyi Yang, Mohammad Atiqul Islam and Shaolei Ren. "Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models." ArXiv abs/2304.03271 (2023): p.3

[25]"Water withdrawals, or water abstractions, are defined as freshwater taken from ground or surface water sources, either permanently or temporarily, and conveyed to a place of use." –OECD

[26]*Ibid*, p.1

[27]A term used in that context by philosopher Bernard Stiegler.

Contact

Email
yctct@yctct.com

Public key
https://yctct.com/_pubkey.txt

GPG fingerprint
2E0F FB60 7FEF 11D0 FB45 4DDC E979 E52A
7036 7A88

Languages
English, Français

Time zone
UTC+01:00

TZ identifier
Europe/Berlin

Imprint

PUBLISHER yctct
LAYOUT yctct
COPY EDITING yctct
DISTRIBUTION yctct
PRINTING the local copy shop
PAPER 80 g/m^2 recycled
PRINT RUN 200
FONTS Inter licensed under OFL-1.1
SPELLCHECK aspell licensed under LGPL-2.1+
TYPESETTING LaTeX licensed under LPPL-1.3c
VERSION CONTROL git licensed under GPL-2.0+
TEXT EDITOR vim licensed under Vim [license]
OPERATING SYSTEM GNU/Linux Trisquel which complies with the GNU Free System Distribution Guidelines
COMPUTER second-hand PC from 2012

yctct@yctct.com
https://pub.yctct.com

This document was compiled on December 22, 2024.

GPG signed commit 17846a63ef50d5b2a3de2bd2321be6c3af8625c6

The pdf version of this document is GPG signed. You can verify its authenticity and integrity with the file https://pub.yctct.com/this_is_a_website_4.pdf.sig

Source file https://pub.yctct.com/this_is_a_website_4.tex

Version 0.012

© 2024 *This is a website #4* is licensed under CC BY-ND 4.0.

TL;DR of the license: you can copy and redistribute this booklet verbatim with attribution, even commercially. If you redistribute this booklet electronically, you cannot apply technological restrictions e.g. DRM. To see the full license, check https://creativecommons.org/licenses/by-nd/4.0/.

To order issues, or translations, email yctct@yctct.com.

Free price. *Free price* doesn't mean that this is gratis. *Free price* means that you are free to choose how much you want to pay to support research.