Riviera Taylor

To Any Software Users:

Collective Identity and Software-Based Creative Practitioners

Thesis submitted to: the Department of Experimental Publishing, Piet Zwart Institute, Willem De Kooning Academy, in partial fulfillment of the requirements for the final examination for the degree of: Master of Arts in Fine Art & Design: Experimental Publishing

Adviser: Marloes de Valk

Second Reader: Michael Murtaugh

Word Count: 8000 words (max)

Introduction

People who use Linux find themselves in a double bind. On the one hand, such people are committed to a technology which espouses freedom, privacy, security, knowledge sharing and the promotion of open standards. On the other hand, the same technology which promotes these ideals is complicit with systems that dominate. How can strengthening relations amongst like-minded people be a prophylactic against this disheartening contradiction? This thesis responds to this question by way of several activities of interest to Linux-based creative practitioners.

I begin by reflecting on the task of installing Linux on various devices. In asking what installing Linux anew might imply for oneself and others the text illustrates that the process is not merely technical. Safety and security are some reasons people may draw from when choosing to use Linux instead of proprietary software. Through repeatedly installing Linux on devices I realised the topic of data encryption offers a way of navigating conversations about these terms. The advantage of running Linux is that it permits a liberating approach to computing. To that end, installing Linux can be a springboard for bringing creative, technically sophisticated possibilities to life. The associated outcomes and processes can inspire people to connect with one another. That is a step towards coming to terms with the predicament outlined above.

The discussion leads me to consider the role of server maintenance in bringing people together. For some months, I maintained a server which ran several services including a public calendar. People and organisations posted to it as a way of sharing information about upcoming events online. The statistical insights of the calendar were informative in terms of understanding the reach of the project. While maintaining the infrastructure I also wrote newsletters on a monthly basis to update people about the status of the server. The aim of this undertaking was threefold. I sought to create a public, offer small-scale infrastructure to others and strengthen my computing skills through practice. The endeavour led me to consider the potential ways in which servers can connect audiences with creative spaces. What are the social implications of running a server?

The third chapter considers affinities between software-centric performance practices and Linux based practices. Taking up these practices can reduce the challenges which software-based creative practitioners may face. What might be gleaned from live coding, performative exercises on the theme of software and codework in terms of developing community sentiment?

Installing Linux

"Trans*Feminist servers ... take the risk of exposing their insecurities and do not confuse security with safety."

(BAK and authors, 2022)

My trust in Big Tech has deteriorated totally. It is imprudent for anyone to rely on the proprietary software or cloud services associated with huge corporations. Simultaneously, it is difficult to divest oneself from those technologies. Increasingly, collectives and individuals recognise that many online platforms are neither safe nor secure. Ideating about how to organise in view of this is one matter. In practice, the technology underlying proposed alternatives, which is often Linux, poses idiosyncratic challenges.

The challenges range from thresholds of participation to security and safety considerations. For instance, Hypha are a group of people based in Romania who share a common interest in how technology shapes social and political realities. They contributed to a research project called *A Traversal Network of Feminist Servers* which sought to connect European creative practitioners working with Linux. Interactions with a nomadic, feminist server named *rosa* formed the basis of that research endeavour. Consider the following assertion made by this collective in the publication which accompanied the research:

There is a need for safe digital services, from chat groups, storage, to shared document writing, but it's also very important to be easy to use. ... To have safe digital services, advocacy and policy changes can be used to impose more legislative control of the extractivist platforms. Self-hosting can't always substitute the technologies that everyone is familiar with; it's more of a path towards learning and working collaboratively, and towards sharing a common digital space. (Ștefănescu, Bucur, Nisioi *et al.*, 2022, p. 57)

Hypha's statement outlines the background against which the following discussion takes place. It is important to analyse their train of thought as there is much to unpack here. The group's call for improvements in online safety are important. To some extent this topic is a matter of data encryption. Still, setting up safe services is complicated and I hope to illustrate this through a discussion of the Linux installation process. The relationship between regulatory bodies and online platforms is a growing talking point. People who use Linux and practise self-hosting should not aim to provide infrastructural services for the vast majority of people. Collaboration can be crucial in terms of attending to potentially isolating choices which can accompany alternative software usage.

Linux on a laptop

Hard drive encryption is important from a computer security point of view. It has a role to play in the event that a device is lost or stolen. It plays an integral role in setting up a Linux system. As such, there are several ways of setting up hard drive encryption on Linux systems. For example, there is a difference between plain encryption and so-called Linux Unified Key System encryption. With the latter, the hard drive creates a so-called header which states that the disk is encrypted. Without the passphrase, the drive cannot be decrypted. However, the use of encryption is still noticeable due to the presence of the header. Under some conditions, someone could spot this and attempt to guess the password to decrypt the drive. People might even succeed in this process if the password is not strong. With plain encryption, by contrast, the header section is absent. Therefore it looks as if the drive is full of random data. The passphrase to decrypt a plain encrypted laptop might be stored as a file on USB stick. The passphrase will likely consist of randomly generated data. As such, nobody can know the passphrase to decrypt the hard drive as, in a sense, there is not one. Instead, the USB stick works as a key. If the device is powered on without the USB stick inserted, the computer might enter the vendor's firmware interface. There are, of course, further steps which could be taken to augment the security of a laptop encrypted in such a way.

What would happen if the key were to get lost or if the USB stick were to suffer damage? Accessing the data on that computer could become difficult. For that reason, it is sensible to make a back-up of the key file on another USB drive if opting for this encryption method. This is essential because it would be impossible to recreate the data in the key file if the file is lost. However, this backup would only get one so far in terms of regaining access to the laptop if something were to happen. In opting for such an encryption method, the administrator of the laptop takes on a risk: they could inadvertently lose access to the data on the device. Many people are not concerned about losing access to all the data on their computer. Such people do not use USB keys to unlock their devices and often have backups of their files in cloud storage. Accordingly, it is necessary to create multiple backups of the data on the computer. Doing so permits moving away from cloud storage. It also mitigates some concerns about losing the key. Understanding this process and its implications involved a lengthy, private meditation on installing Linux in a secure and sustainable way. Figuring out how to configure plain encryption took many hours. This discussion illustrates some of the complexity about getting started with self-hosting in a secure way.

Linux on smartphones

With few exceptions smartphones ship with one of two operating systems: iOS or Android. The former is "a proprietary and closed operating system ... specifically designed by Apple for its mobile products" (Ippolita, 2013, p.10). The latter "is an operating system built on a Linux kernel and distributed under the open source Apache license" (Ibid.). Android was developed by Google and

has become a *de facto* standard, in a similar way to how the Windows operating system for desktop became a standard. The only basic difference is that, thanks to the open source license, manufacturers can create custom flavors of Android for their hardware (Ibid.)

Vanilla Android is an example of Linux in a bastardised form. However, there are alternatives. For example, at the time of writing, PostmarketOS, a non-Android operating system for mobile phones which is also based on Linux, supports over 800 devices. An eighth of these devices are well supported; the remainder are supported in a limited way. That PostmarketOS offers good support for over 100 devices is impressive. Replicant, a free Android operating system, supports 13 devices. The scale of these projects indicates that these operating systems serve different communities with different needs and aspirations.

One commonly invoked reason to run Linux on a mobile phone is that it can extend the lifespan of a device. It might seem counterintuitive to use Linux on a phone because many apps are not available. This entails a different way of staying up to date with what is going on in the world. It also involves finding different ways of communicating with others. On the subject of staying up to date, mobile phone manufacturers often decide to stop providing software updates for older devices. Consequently, users of those phone models find themselves in less than optimal positions regarding security. This is something PostmarketOS aims to address.

As the quote which opens this chapter indicates: Security is one matter, safety is another. The former pertains to the reinforcement of defence mechanisms in anticipation of threats, malevolence, saboteurs, and so on. The latter pertains to matters of well-being such as self-continuity, psychological boundaries and the limits of what is comfortable. To act on the desire to extricate oneself from proprietary technology is coherent with the aims being articulated by proponents of alternative software. Thus, running Linux on ones phone is similarly coherent. However, the experience might be isolating, strange and disorientating, which is to say unsafe. This is partly because non-free software is normalised in and beyond the mobile phone ecosystem. It is also partly because smartphones

are utilised in different ways to how laptops are utilised. For example, even after installing Linux on a mobile device, it may be impossible to speak with other people when making telephone calls. Thus, an issue mobile Linux users may face includes an inability to communicate with emergency services. Furthermore, it is improbable that the phone will be able to receive alerts issued by governments. Ultimately, one never knows when an emergency may arise. So, whereas receiving long-term security updates is beneficial, running Linux on a mobile phone could pose risks which outweigh the gains.

Linux on routers

In terms of unlocking the full capabilities of a device, wireless routers are more numerously supported by Linux distributions than mobile phones are. Routers are wifi enabled boxes with ethernet ports on them which connect devices such as laptops and smartphones to the internet. Reasons to move away from proprietary router firmware range from improved security features and more comprehensive online documentation to creative possibilities. Routers compatible with Linux often provide a convenient interface for installing Linux on the device. Doing so opens up a world of technical possibilities; it becomes possible to re-configure the router in surprising ways. For example, the router could serve websites, be involved in a virtual private network or configured as a backup machine for a server.

However, understanding where to begin with installing non-proprietary firmware on a router can also be tricky. This is reflective of a more general dilemma around participating in software communities which can be articulated as follows. On the one hand, "[t]he first characteristic of a F/OSS community consist [sic.] of adopting working methods that are open to the collaboration of all comers" (Ibid., p. 51). If the collaborative platforms such communities put into action are technical in nature then the first hurdle newcomers face is understanding those technical systems. Thus, community initiators must take into account the social implications of the systems which are implemented to ensure effective conditions for participation. In some software communities, there are more complex dynamics at play. Take the example of Linux. In principle anyone can contribute changes to the source code. However "the reality is more subtle" (Kelty, 2008, p. 219). Linux development involves "a hierarchy of contributors, maintainers, and "trusted lieutenants"" (Ibid.). Due to such biases it is more difficult, in practice, to take part in writing code that would be widely distributed. This is complicated by the fact that "Linux includes gendered and classed practices which usually remain unremarked" (Mackenzie, 2005, p. 82). Thus, participating in some software communities, particularly those consisting almost entirely of educated men, can be difficult.

Suppose someone has a question about why something is not working correctly, which is often an issue. Perhaps it is a line of code that doesn't work as expected, or an error that doesn't make sense. In the context of online forums, issue trackers and mailing lists, it may be unclear where to begin with resolving the matter. Indeed, this is most problematic when starting out with software. It might feel necessary to trawl through the internet looking for answers or debug impenetrable error messages rather than asking about the specific issue at hand. Silencing oneself or not participating in public discussions is a direct consequence of this. This can be isolating. The result is a lose-lose situation. The person facing the issue loses out on connecting with others who may know how to solve the problem. Likewise, if a solution is found but not shared, then other people might feel lost when facing the same dilemma. Of course, there are communities which, for a host of reasons, people may not want to participate in. When this is not the case, software practitioners should take the initiative to be vocal about problems which they are facing. Suffering in silence is never desirable.

Linux on servers

In early 2024 the W1555 housing collective, located in Charlois, Rotterdam, put out an open call. The collective sought community driven projects to make use of two Do-It-Together spaces. "W1555's vision and goals for these spaces focus on art, education and culture in all its diversity", the open call stated (W1555, 2024, p. 2). In response, sound artist Niek Hilkmann proposed to start a school around sound: Klankschool. Furthermore, the application was approved. The expected moving-in date was April 1st. Therefore, in March, he arranged some meetings at Varia, a space for everyday technology also located in Charlois. Approximately 20 people attended these meetings.

During the first meeting, people shared what they wanted from Klankschool. For example, somebody wanted to run a radio station where the sound of the space would be constantly broadcast on the internet. Other people spoke about devising a system for booking the space. Some people wanted to run their own websites on a domain associated with Klankschool. At the time, it seemed to me that a server would be necessary. Sadly, after the first meeting, it transpired the space which would house Klankschool was in need of renovations. A leak in the floor was damaging the building. A legal dispute began between third parties to determine who was responsible for financing the repair works. The circumstances were beyond the control of Hilkmann and moving-in plans were postponed indefinitely. Notwithstanding, a group of enthusiasts attending the March meetings wished to keep the energy going and proposed to do so in a different social space. Myself and others set about strengthening connections between people by putting our energy into this context. We kept the name of Klankschool.

In the process of setting up Linux machines described above, with the exception of a workshop about installing Linux on routers, I collaborated with others very little. This changed when it came to installing Linux on a server. In September 2024, I organised a discussion about starting a community server. A dozen people attended this meeting. The discussion followed the format of a Long Table, an idea created at the turn of the century by theatre maker Lois Weaver. The Long Table seeks to create an atmosphere "where the anecdote, the personal story, the theoretical point and the statistical piece of data have a place [and] one thing's not privileged over another" (McAvinchey, 2014, p. 29). Through a combination of feminist practice and non-moderated, etiquette-driven performance, the Long Table aims to strike up conversation amongst community members. The performance group Weaver is a member of emphasise that "[t]he Long Table is an open-source format" (Split Britches, 2019). The outcome of the conversation was that a server could be useful for a while. There was no expectation that it would become a permanent feature, nor that the people occupied with its maintenance would be irrevocably involved. To that end, I found Weaver's observation that community "doesn't last forever" reassuring and appropriate to the circumstances (McAvinchey, 2014, p. 30).

I was keen to facilitate with materialising the ideas which people had vocalised during the meetings at Varia back in the Spring. I planned to use software to generate collective activity. In the context of feminist server maintenance practices, "[f]ree software aligns politically with feminist servers" core values, such as sharing knowledge, empowering each other and working against power hierarchies based on gatekeeping, access to resources, tools and knowledge' (Wessalowski and Karagianni, 2023, p. 196). I wanted to know what people might want from a someone maintaining a machine based on such technology.

Server maintenance

Thus, an aspect of conducting this research involved maintaining a server, a second-hand computer running Linux. It hosted several services for the duration of some months. This included a SFTP server (for sharing files), a Git interface, a kanboard (for collaborative project management) and a collaborative live coding environment. There was a public calendar and I wrote newsletters to people on a monthly basis. Many of these services had public websites as counterparts. Here I focus on my efforts to contribute to a collective organisation through maintaining a server and writing a newsletter. What positive difference was made by these endeavours?

About the newsletter

The purpose of the newsletter was to update people about the status of the server, how it was changing over time and which services were running. I challenged myself to talk about technical processes using non-technical language to keep people engaged. The publishing of newsletters is related to the concept of a counterpublic. Archival scholar Cait McKinney (2020, p. 60) writes that "[n]ewsletter networks share some characteristics with online networks: for example, they connect distant others to support counterpublic work, non-hierarchical collaboration, and high levels of engagement". The newsletters I wrote were distributed via an online discussion list, rather than through the medium of print. The decision to run a discussion list, rather than a mailing list, was to encourage multidirectional communication amongst interlocutors. The discussion was quite limited, however. In the main, the list consisted of me writing to others on a monthly basis.

Some people are marginalised, disenfranchised or otherwise excluded from having a say in the running of software communities. That feminist practitioners find it necessary to carve out independent spaces to examine and practice with software reflects an endemic problem. To a significant extent the stymieing of 'imaginative practices' amongst software practitioners is attributable to a pervasive ethos of 'technocratic ... meritocracy' in these settings (Snelting, 2015). As discussed above, the people involved in producing software can create hostile atmospheres. Perhaps writing a newsletter about maintaining a server was an ineffective way of stimulating conversation amongst others.

About the calendar

The calendar, by contrast, was more successful. It was active online and utilised software known as Gancio. Gancio is a federated, public calendar with a low threshold of participation. Federated, in this context, means that the calendar is part of the Fediverse. Multiple apps are part of the Fediverse and serve different purposes. Mastodon, which was developed in "response to dissatisfaction with hate speech and white supremacy on Twitter", is perhaps the best known Fediverse application (Gehl and Zulli, 2023, p. 3277). It is a micro-blogging platform where people share updates in the form of toots. Another example of a Fediverse application is PeerTube, a video sharing platform. What differentiates Mastodon from X, PeerTube from YouTube? Unlike the social media created by Big Tech, which is centralised, federated platforms are decentralised. Fediverse apps are interoperable thanks to a common protocol called ActivityPub. The protocol "allows a video platform to connect not just with other PeerTube platforms, but with all social networks and other media instances based on the same protocol" (Wessalowski and Karagianni, 2023, p. 198). Thus, what is shared on PeerTube can be re-circulated on Mastodon.

Moreover, the angle which federated platforms take on protecting communities stands in contrast to centralised social media. The software is free and can be self-hosted which enhances flexibility, agency and self-determination on multiple levels. Instances are able to make choices about others instances with which they federate. Codes of conduct and terms of service often act as a basis for moderating online activity which in turn reinforces cohesive relationships amongst users. "These documents are used to shape the community, make their ideological alignment clearer, and try to prevent manipulation and misunderstanding around vague notions like openness, transparency, and freedom" (Mansoux and Roscam Abbing, 2020, p. 139). Such policies also minimise the amount of harmful content.

Over the course of six months, more than 75 events taking place at 16 different locations in Rotterdam appeared on the calendar. In March alone there were over 25 events. The events ranged from hang-outs and film-screenings at low-key locations to performances and festivals at mainstream cultural venues. It was the policy of the moderators to disallow events promoting offensive ideologies. There were eight actors across 13 known fediverse instances following the calendar. This statistical information offers some insight into the reach of the project. Although Gancio is federated that represents only one way of staying up to date with what is posted on the instance. Anonymous means of staying in-the-loop include visiting the website, subscribing to an RSS feed, ICS feed, or embedding an HTML widget in a webpage. With five ways of getting information about upcoming events, the calendar likely reached more people than the statistics above concretely indicate towards.

Performing with software

"Collective identity is created by, or related to, a movement culture comprised of a relatively autonomous network of interactions between different individuals, institutions, and organizations. In order to be effective, movements have to be goal-oriented, and act as strategic collectives that always strive toward social change."

(Szczepanska, Bergquist and Ljungberg, 2005, p. 432)

This chapter focuses on performance practices having to do with software. Three practices are discussed. In view of the quote above, I am curious about strategic affinities detectable between these practices and in relation to the topics discussed so far.

Codework

The term codework was introduced in a 2001 issue of American Book Review. The editor of the issue, Alan Sondheim, articulated an umbrella term under which thematically related works were grouped. Sondheim fleshes out the subject with examples from several practitioners, co-contributors and artists. He situates his own works alongside those by Mez (Mary Anne Breeze), Antiorp (Netochka Nezvanova) and Jodi (Joan Heemskerk and Dirk Paesmans) amongst others. Works by these people diverge in several ways. For instance, Jodi are well known for creating experimental websites. Mez, on the other hand, rose to notoriety by sending emails to discussion lists in a captivating idiolect. In view of such differences, a three-part codework "taxonomy" is offered (Sondheim, 2001). It includes "[w]orks using the syntactical interplay of surface language, with reference to computer language" (Ibid.). It furthermore extends to "[w]orks in which submerged code has modified the surface language" and "[w]orks in which the submerged code is emergent content" (Ibid.). Put broadly, codework consists of "poetic and artistic work in programmable media which engage with, reference, and use computer code" (Engberg, 2005, p. 5). Sondheim's short text sparked a debate about the constitution and implications of codework amongst commentators.

Execution is an organising concept in debates about codework. Actual and successful execution amounts to the production of machine-readable data. It is a technical process mediated by the relationship between a programme and the platform it runs on. Software art critic Inke Arns (2004, p. 187) writes that codeworks "oscillate between supposed executability ... and non-executability". In Arns' argument, codework acts as a basis for wider

conclusions reached about software art. Execution remains a speculative and tantalising possibility for the codeworks featuring in Arns' discussion. Arns is emphatic about the status of "so-called "Codeworks" consisting predominantly of ASCII-Code (not being *executables*)" (Ibid., p. 179). Other researchers also think in terms of whether or not a given codework is executable, or more accurately, machine readable. (Karagianni, Diakrousi and Gryllaki, 2022; Ledesma, 2015).

While Arns does not explicitly discuss the tension between the readability of code and mode of address, others do. According to literary scholar Rita Raley (2006, p. 8), "John Cayley has endeavored to call our attention to the material difference between code addressed to a reader (pretend) and code addressed to the machine (genuine)". Source code, because it is both human and machine readable, puts into question the gap between these modes of address. Take the example of Perl poetry, a practice which takes the "ambiguity of address" discussed by Cayley (2002), a writer and artist, to the extreme. Perl poems are short-form texts written in the Perl programming language. Examples of Perl poems include *Black Perl*, Sharon Hopkins' *Rush* and Craig Counterman's *Ode to my Thesis*. The limitations of the writing style include the requirement that the code is a valid Perl script. It should execute successfully and possibly return output. It is undeniably "kitsch" (Mackenzie, 2015, p. 27).

As art critic Clement Greenberg (1989, p. 11) observes "[k]itsch is deceptive. It has many different levels, and some of them are high enough to be dangerous". What is dangerous about Perl poetry is the pattern of thinking it promotes. Perl poets articulate themselves in code which implies that they write in

the code of the Other, and that is something quite different from what is in question in the message, since it is from this code that the subject is constituted, which means that it is from the Other that the subject receives even the message that he emits. ... Code messages or message codes will be distinguished in pure forms in the subject of psychosis, the subject who is satisfied with that previous Other (Lacan, 2001, p. 232)

Put differently, people who write executable poems adopt a programming language, written by others for a different purpose, as their chosen communicative medium. "Perl was developed by system administrators to allow programs to be quickly written for ad-hoc purposes, such as renaming a large number of files or tidying up the formatting of a text document" (Mackenzie, 2015, p. 26). As code is intended to be read by machines, diminished self expression is characteristic of this practice. The content of their poetry is dominated by the code and this is fundamentally off-putting.

Live coding

Live coders create music or graphics for audiences with code in real time. The importance of software in this context makes discussing the performance practice pertinent. This art form epitomises software-centric performance and has long occupied a place in the culture around alternative software advocacy. The international activities of live coding communities attests to practitioners' efforts to generate a buzz around the practice. In live coding performances it is crucial that "algorithms are written in real time" (Magnusson, 2014, p. 10). Performers position themselves behind a laptop while writing code in front of an audience. The artists will often connect their laptop to a projector to display an enlarged version of their screen. The thrown image is an important feature because it allows the audience to see any graphics being created. When there are graphics, these are often audio-reactive: the visuals move in synchrony with sonic rhythms. Simultaneously, "projected screens allow ad-hoc software distribution" (McLean, 2008, p. 229). Accordingly, as non-academic researcher Alex McLean points out, audiences can see the code being written and executed and are free to do what they want with the material. Although projecting screens is a frequent feature of the performance genre, "[I]ive coding does not have a particular unified aesthetic in terms of musical or visual style" (Magnusson, 2014, p. 8). As performers' movements are minimal the audience are engaged on a cognitive and rhythmic plane.

The simultaneous generation of auditory and visual phenomena at moments when code is executed is aesthetically satisfying. A region of code is executed and, in a flash, the audio changes accordingly, for example. The practice appeals to diverse audiences, programmers and non-technical people alike. "Typically performers start with a clean sheet ... and build their compositions from scratch", writes one of the forefathers of live coding Thor Magnusson (2011, p. 21). People who perform live coding are thus skilled in improvisation and rehearse to expand their technical vocabulary. "Free jazz improvisers have even more freedom. But a live coding musician is like an improvising composer, able to transform the whole structure of the piece with a few keystrokes" (Blackwell, Cocker, Cox *et al.*, 2022, p. 5).

Live coding is possible in a range of programming languages from C++ to CSS. Typically, the software is free or open-source. Access to source code enables live coders to customise and edit the software, read and audit the code. Publishing such code is an act of sharing knowledge. As former course director at the Piet Zwart Institute Simon Yuill highlights, live coding involves distributive practices. He argues that live coding "is inherently distributive, for it integrates the distribution of the knowledge of how to

produce into that which it produces" (Yuill, 2008, p.69). In other words, "[I]ive coding makes visible its process of thinking so that others might then modify, build upon, and creatively develop this further still" (Blackwell, Cocker, Cox *et al.*, 2022, p. 221). However, distributive practices are not unique to live coding. Such practices are found also in codework. What, then, beyond aesthetics, is at stake in live coding?

Perhaps the answer lies in community and collective learning. There is an active live coding community in the Netherlands: Netherlands Coding Live. This group organises monthly in-person meet-ups in different cities on a regular basis. In early 2025, I gave a short presentation during a Rotterdam meet-up at v2 . I wanted to reach out to like-minded people and encourage them take part in the activities of Klankschool. I spoke about the server I was maintaining, its connection to live coding practices, and the context in which it was situated. In an effort to knit live coding and server maintenance practices together I installed and ran software known as Flok on the server. Flok is an online platform where multiple people can write code simultaneously in different live coding languages. At the end of the evening, there was an opportunity to perform live coding with others in attendance. We utilised another public Flok instance for this purpose. We wrote code together on a shared web-page. This allowed us to read, see and hear the changes each were making to the code as these changes were made. We could thus learn from one another with mutual attention.

It is possible to live code on a laptop. What if it were possible to live code on a router? As I write this paragraph, I am compiling the Test2::Util Perl module on a router running OpenWrt. OpenWrt is a Linux distribution primarily for routers and network attached storage devices. I hope to perform live coding on the router and create music with it. That is, I intend to execute code which generates audio on the router itself. Some Perl code, or programmers, as I just discussed, demonstrate reprehensible qualities. Not all though. More than twenty years ago McLean (2004) wrote a piece of software called *feedback.pl*. His blog post details that in order to make sounds with the software it is necessary to install the Audio::Beep Perl module. Test2::Util is a dependency of this module so I am attempting to compile it. Compiling the module involves running hundreds of tests. The following command has been executing for seven hours:

perl -MCPAN -e "install 'Test2::Util'" 2>&1 >~/perllog It failed to execute successfully on several previous occasions. This was initially due to the absence of dependencies and subsequently to the constraints of the hardware. It was not possible to execute the command while staying logged-in to the router. So, it was necessary to instruct the router to execute the command to compile the module at a scheduled time. "This might all sound like a rather strange and tortured way of making music, but actually the opposite is true. It's not strange, ... [i]t's not tortured either" (Ibid.). In this scenario I do not agree with McLean. There is something a bit strange about attempting to turn a router into a musical instrument. And unlike a laptop, this hardware is somewhat torturous to work with. Nevertheless, experimentation and pushing the limits of ones technical knowledge in preparation for a performance is what I find engaging in the practice of live coding. It is one thing to perform live coding in front of an audience. How each audience member appreciates the performance is another matter. I want to attempt to make a sonic performance with a router to understand whether or not it is possible to do so.

Performances on the theme of software

Performances which take up the theme of software are related to the field of software studies, which focuses on the cultural aspects of software. "Software Studies proposes that software can be seen as an object of study and an area of practice for kinds of thinking and areas of work that have not historically "owned" software" (Fuller, 2008, p. 2). Embodied performance is one such field. Likewise, such exercises connect to a patenting practice prevalent during the 1960s known as "embodied software" (Con Diaz, 2015).

Attorneys and programmers used this term in reference to the patent-drafting strategy of submitting patent applications claiming a computer in which a particular program serves as the control system. This computer served as an embodiment of the program, and it received patent protection in lieu of the program itself (Ibid., p. 8).

Put differently, programmers would design electrical circuits on which their programmes could run. They would subsequently seek a patent on the design. There are parallels here insofar that physical exercises are designed by practitioners. Both to act as a vessel for the instructions to be carried out and to explore the social implications of software practices from a non-technical vantage point. This practice is marginal in comparison with live coding and codework. However, it has something important to say about the social aspects of software. So I will consider two examples before concluding.

The first performance I would like to briefly discuss is BolwerK's *Home is a Server*. This was an hour-long performative exercise that took place at Constant Variable in December 2013 during *Verbindingen / Jonctions 14*. "In Home is a Server the actors execute in a very precise manner the different tasks a server is given, when one would like to publish the recipe for

pancakes on a wiki" (Van Dessel, 2013, p.164). 32 participants embodied this process with 32 props including, for example, a wall clock, mirror, hard hat, pens and paper. The exercise consisted of booting-up the machine and filling up the battery to connecting via the secure shell protocol and installing a web server. The people with the mirror and wall clock worked together as a BIOS chip. These participants, like the physical components inside computers, play a role in supporting background processes on which wiki instances depend. These processes were in turn performed by other actors with different objects. There was even a dog playing the role of the watchdog, a piece of technology which prevents a computer from rebooting while running. The group effort turns a technical process into a collaborative process where collectivity is foregrounded.

The second example I would like to discuss is Renée Copraij, Isabelle Jenniches and Stefan Kunzmann's performance *Sensing Presence No. 1.* It took place in 2001 at the anatomy theatre of De Waag, Amsterdam. The performance is split into two parts. In the first part, Copraij lies center-stage on the floor while Kunzmann stands behind a lectern at the edge of the space. During this time he clicks a mouse and Copraij makes nearly indiscernible movements. In the second part, Kunzmann begins stating out loud various body parts. The audience then "recognize Copraij's movements as attempts at connecting the parts of her body, named by Kunzmann, through internal body movements" (Bleeker, 2008, p. 167). In advance of the performance the audience drew variations of a diagram with fixed points connected by straight lines. These formed the basis for the movements made by Copraij. Different nodes in the diagrams aligned with different parts of the performer's body.

Theatre scholar Maaike Bleeker's written account of the performance indicates the following: Copraij "turned what she heard into a stream of energy (internal movements), thus translating the code into a neuromuscular experience" (Ibid.). In other words the code, a combination of diagrammatic algorithms, is embodied through a string of actions. The performance links Copraij's body to a multitude of networked diagrams via speech. Bleeker writes that the piece departs from the nexus of mind and computer and turns to the relation between body and computer. "[U]sually the computer is equated to the mind (or vice versa). Jenniches, Kunzmann and Copraij, on the other hand, proposed to explore the comparison of the computer with the body in order to come up with a Deleuzian-like conception of thinking in terms of movement" (Ibid., p. 167–168).

Bleeker's documentation of the performance at De Waag leaves the reciprocal dynamics between these performers open to the imagination. The

performance could be read as a commentary on how women might practice resistance to power imbalances along gendered axes in computational environments. Whereas Kunzmann is inert and vocal, Copraij is silent, supine yet mobile. Embodying such dichotomies conveys the agency which accompanies intentional movement. *Home is a Server*, by contrast, emphasises the communal aspects of software. The tone of the performance is whimsical and ironic. Without proper training a dog cannot perform the role of a technical watchdog with any amount of precision. Still, the pancakes which were eaten by the participants at the end of the exercise re-iterate the difference between creatures which must metabolise and machines which do not have to. Overall, these performances offer insight into the alternative ways that alternative software might bring people together.

Conclusion

This thesis has discussed topics that creative practitioners working with software grapple with. These range from software-centric performance to maintaining servers and working with Linux. Each chapter focused on a question that has emerged in my practice which involves the holistic use of software. Practising self-immersion in such technology is a choice and not a necessity. It can pose risks and problems as well as advantages and learning opportunities. On the one hand, diversity in terms of gender representation is a point of tension in the milieus surrounding software. On the other hand, the connections between software and Big Tech constitute an imbroglio that rings of appropriation. To that end, the growth of a shatterproof, unresentful software community does not seem plausible. In view of this, as I have sought to highlight, the creative dimensions of software, particularly when informed by feminist practice and research, can bring people together. Such endeavours may involve maintaining small-scale pieces of infrastructure, performing with code in front of crowds or meeting up to talk about Linux.

Reference list

- Arns, I. (2004) 'Read_me, Run_me, Execute_me: Software and Its Discontents, or: It's the Performativity of Code, Stupid', in: Goriunova, O. and Shulgin, A. (eds.) *Read_me: Software Art & Cultures, Edition 2004*. Aarhus, Denmark: Aarhus UP, pp. 176–193.
- BAK and authors (2022) *A Wishlist for Trans*feminist Servers*. Available at: <u>https://archive2</u> <u>.bakonline.org/prospections/a-wishlist-for-transfeminist-servers/</u> (Accessed 16 April 2025).
- Blackwell, A.F., Cocker, E., Cox, G., McLean, A. and Magnusson, T. (2022) *Live Coding: A User's Manual*, Cambridge, Mass: MIT Press.
- Bleeker, M. (2008) 'Performance Documentation 5: Sensing Presence No. 1: Performing a Hyperlink System', in: Bleeker, M. (ed.) *Anatomy Live*. Amsterdam University Press, pp. 165–168.
- Cayley, J. (2002) 'The Code Is Not the Text (Unless It Is the Text)', *Electronic Book Review*, Available at: <u>http://electronicbookreview.com/essay/the-code-is-not-the-text-unless-it</u> <u>-is-the-text/</u> (Accessed 16 April 2025).
- Con Diaz, G. (2015) 'Embodied Software: Patents and the History of Software Development, 1946-1970', *IEEE Annals of the History of Computing*, 37 (3), pp. 8–19. Available at: DOI: <u>10.1109/MAHC.2015.45</u>.
- Engberg, M. (2005) 'Stepping Into the River: Experiencing John Cayley's RiverIsland', *Dich-tung Digital*, 7 (35), pp. 1–9. Available at: DOI: <u>10.25969/mediarep/1768</u>.
- Fuller, M. (2008) 'Introduction, The Stuff of Software', in: Fuller, M. (ed.) *Software Studies: A Lexicon*. Cambridge, Mass: MIT Press, pp. 1–13.
- Gehl, R.W. and Zulli, D. (2023) 'The Digital Covenant: Non-Centralized Platform Governance on the Mastodon Social Network', *Information, Communication & Society*, 26 (16), pp. 3275–3291. Available at: DOI: 10.1080/1369118X.2022.2147400.
- Greenberg, C. (1989) Art and Culture: Critical Essays, Boston: Beacon Press.
- Ippolita (2013) The Dark Side of Google, Amsterdam: Institute of Network Cultures.
- Karagianni, M., Diakrousi, A. and Gryllaki, A. (2022) 'FEMINIST HACK MEETINGS', in: Strete, A., Cochior, C. and Mugrefya, e. (eds.) A Traversal Network of Feminist Servers. Bucharest: Fractalia, pp. 119–153.
- Kelty, C.M. (2008) *Two Bits: The Cultural Significance of Free Software*, Durham: Duke University Press.

Lacan, J. (2001) *Écrits: A Selection*, London: Routledge.

- Ledesma, E. (2015) 'The Poetics and Politics of Computer Code in Latin America: Codework, Code Art, and Live Coding', *Revista de Estudios Hispánicos*, 49 (1), pp. 91–120. Available at: DOI: <u>10.1353/rvs.2015.0016</u>.
- Mackenzie, A. (2005) 'The Performativity of Code: Software and Cultures of Circulation', *Theory, Culture & Society*, 22 (1), pp. 71–92. Available at: DOI: <u>10.1177</u> /0263276405048436.
- Mackenzie, A. (2015) Cutting Code: Software and Sociality, New York: Peter Lang.
- Magnusson, T. (2011) 'Algorithms as Scores: Coding Live Music', *Leonardo Music Journal*, 21 (1), pp. 19–23. Available at: DOI: <u>10.1162/LMJ a 00056</u>.
- Magnusson, T. (2014) 'Herding Cats: Observing Live Coding in the Wild', *Computer Music Journal*, 38 (1), pp. 8–16. Available at: DOI: <u>10.1162/COMJ a 00216</u>.
- Mansoux, A. and Roscam Abbing, R. (2020) 'Seven Theses on the Fediverse and the Becoming of FLOSS', in: Gansing, K. and Luchs, I. (eds.) *The Eternal Network: The Ends and Becomings of Network Culture*. Amsterdam: Institute of Network Cultures, pp. 124–140.
- McAvinchey, C. (2014) 'Lois Weaver: Interview and Introduction by Caoimhe McAvinchey', in: McAvinchey, C. (ed.) *Performance and Community: Commentary and Case Studies*. London, England: Bloomsbury Methuen Drama, pp. 21–32.
- McKinney, C. (2020) *Information Activism: A Queer History of Lesbian Media Technologies*, Durham: Duke University Press.
- McLean, A. (2004) *Hacking Perl in Nightclubs*. Available at: <u>https://www.perl.com/pub/2004</u> /08/31/livecode.html/ (Accessed 16 April 2025).
- McLean, A. (2008) 'Live Coding for Free', in: Mansoux, A. and de Valk, M. (eds.) *Floss* + *Art*. Poitiers, France: GOTO 10, in association with OpenMute, pp. 224–231.
- Raley, R. (2006) 'Code.Surface| |Code.Depth', *Dichtung Digital*, 8 (1), pp. 1–24. Available at: DOI: <u>10.25969/MEDIAREP/17695</u>.
- Snelting, F. (2015) *Exquisite Corpse*. Available at: <u>http://www.newcriticals.com/exquisite</u> <u>-corpse/page-8</u> (Accessed 16 April 2025).
- Sondheim, A. (2001) 'Introduction: Codework', *American Book Review*, 22 (6), pp. 1–2. Available at: <u>https://litline.org/ABR/issues/Volume22/Issue6/sondheim.pdf</u> (Accessed 16 April 2025).

- Split Britches (2019) *Long Table*. Available at: <u>http://www.split-britches.com/long-table</u> (Accessed 16 April 2025).
- Ştefănescu, A., Bucur, A., Nisioi, S., Săvoiu, T. and Dobrițoiu, V. (2022) 'HYPHA', in: Strete,
 A., Cochior, C. and Mugrefya, e. (eds.) *A Traversal Network of Feminist Servers*.
 Bucharest: Fractalia, pp. 47–57.
- Szczepanska, A.M., Bergquist, M. and Ljungberg, J. (2005) 'High Noon at OS Corral: Duels and Shoot-Outs in Open Source Discourse', in: Feller, J., Fitzgerald, B., Hissam, S.A. and Lakhani, K.R. (eds.) *Perspectives on free and open source software*. Cambridge, Mass: MIT Press, pp. 431–446.
- Van Dessel, M. (2013) 'Home Is a Server', in: Aktypi, M., Höfmuller, R., Laforet, A., Lakova, M., Mertens, A. and Snelting, F. (eds.) "Are You Being Served?" (Notebooks). Constant, pp. 164–166.
- Wessalowski, N. and Karagianni, M. (2023) 'From Feminist Servers to Feminist Federation', *A Peer-Reviewed Journal About*, 12 (1), pp. 192–208. Available at: DOI: <u>10.7146</u> /aprja.v12i1.140450.
- W1555 (2024) Open Call for Do-It-Together Spaces. Available at: <u>https://www.w1555.org/en/post/open-call-for-dit-do-it-together-spaces</u> (Accessed 16 April 2025).
- Yuill, S. (2008) 'All Problems of Notation Will Be Solved by the Masses: Free Open Form Performance, Free/Libre Open Source Software, and Distributive Practice', in: Mansoux, A. and de Valk, M. (eds.) *Floss* + *Art*. Poitiers, France: GOTO 10, in association with OpenMute, pp. 64–91.

Colophon

This thesis has been produced in the context of the Experimental Publishing Master's course at the Piet Zwart Institute.

You should have received a copy of the Cooperative Non-Violent Public License with this thesis.

Source code is available on request and at:

https://git.xpub.nl/XPUB/project.xpub.nl/raw/branch/master/routers-are-computers/main.tex

Cooperative Non-Violent Public License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS COOP-ERATIVE NON-VIOLENT PUBLIC LICENSE ("LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND ALL OTHER APPLICABLE LAWS. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED. BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED IN THIS LICENSE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN AS CONSIDERATION FOR ACCEPTING THE TERMS AND CONDITIONS OF THIS LICENSE AND FOR AGREEING TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS LICENSE.

Definitions

An Act of War is any action of one country against any group either with an intention to provoke a conflict or an action that occurs during a declared war or during armed conflict between military forces of any origin. This includes but is not limited to enforcing sanctions or sieges, supplying armed forces, or profiting from the manufacture of tools or weaponry used in military conflict.

An Adaptation is a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose. In addition, where the Work is designed to output a neural network the output of the neural network will be considered an Adaptation for the purpose of this License.

Bodily Harm is any physical hurt or injury to a person that interferes with the health or comfort of the person and that is more than merely transient or trifling in nature.

Distribute is to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale, gift or any other transfer of possession or ownership.

Incarceration is Confinement in a jail, prison, or any other place where individuals of any kind are held against either their will or (if their will cannot be determined) the will of their legal guardian or guardians. In the case of a conflict between the will of the individual and the will of their legal guardian or guardians, the will of the individual will take precedence.

Licensor is The individual, individuals, entity, or entities that offer(s) the Work under the terms of this License

Original Author is in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition

- in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore;
- in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and,
- in the case of broadcasts, the organization that transmits the broadcast.

Work is the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramaticomusical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

You means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

Publicly Perform means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

Reproduce is to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium. Software is any digital Work which, through use of a third-party piece of Software or through the direct usage of itself on a computer system, the memory of the computer is modified dynamically or semi-dynamically. "Software", secondly, processes or interprets information.

Source Code is Any digital Work which, through use of a third-party piece of Software or through the direct usage of itself on a computer system, the memory of the computer is modified dynamically or semi-dynamically. "Software", secondly, processes or interprets information.

Surveilling is the use of the Work to either overtly or covertly observe and record persons and or their activities.

A Network Service is the use of a piece of Software to interpret or modify information that is subsequently and directly served to users over the Internet.

To Discriminate is use of a work to differentiate between humans in a such a way which prioritizes some above others on the basis of percieved membership within certain groups.

Hate Speech is Communication or any form of expression which is solely for the purpose of expressing hatred for some group or advocating a form of Discrimination between humans.

Coercion is leveraging of the threat of force or use of force to intimidate a person in order to gain compliance, or to offer large incentives which aim to entice a person to act against their will.

Fair Dealing Rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

To Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections

To create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."

To Distribute and Publicly Perform the Work including as incorporated in Collections.

To Distribute and Publicly Perform Adaptations. The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include

the right to make such modifications as are technically necessary to exercise the rights in other media and formats. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Non-waivable Compulsory License Schemes, Waivable Compulsory License Schemes, and Voluntary License Schemes in the restrictions.

Restrictions

The license granted in the license grant above is expressly made subject to and limited by the following restrictions:

You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as requested.

If the Work meets the definition of Software, You may exercise the rights granted in the license grant only if You provide a copy of the corresponding Source Code from which the Work was derived in digital form, or You provide a URI for the corresponding Source Code of the Work, to any recipients upon request.

If the Work is used as or for a Network Service, You may exercise the rights granted in the license grant only if You provide a copy of the corresponding Source Code from which the Work was derived in digital form, or You provide a URI for the corresponding Source Code to the Work, to any recipients of the data served or modified by the Web Service.

You may exercise the rights granted in the license grant for any purposes only if:

- You do not use the Work for the purpose of inflicting Bodily Harm on human beings (subject to criminal prosecution or otherwise) outside of providing medical aid or undergoing a voluntary procedure under no form of Coercion.
- You do not use the Work for the purpose of Surveilling or tracking individuals for financial gain.
- You do not use the Work in an Act of War.
- You do not use the Work for the purpose of supporting or profiting from an Act of War.
- You do not use the Work for the purpose of Incarceration.
- You do not use the Work for the purpose of extracting, processing, or refining, oil, gas, or coal. Or to in any other way to deliberately pollute the environment as a byproduct of manufacturing or irresponsible disposal of hazardous materials.
- You do not use the Work for the purpose of expediting, coordinating, or facilitating paid work undertaken by individuals under the age of 12 years.
- You do not use the Work to either Discriminate or spread Hate Speech on the basis of sex, sexual orientation, gender identity, race, age, disability, color, national origin, religion, caste, or lower economic status.

If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made by any Licensor to remove credit from a Collection or Adaptation, keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing:

- the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties;
- the title of the Work if supplied;
- to the extent reasonably practicable, the URI, if any, that Licensor to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and,
- in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author").

If any Licensor has sent notice to request removing credit, You must, to the extent practicable, remove any credit as requested. The credit required by this Section may be implemented in any reasonable manner; provided, however, that in the case of an Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in the license grant of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under the license grant of this License (right to make Adaptations) but not otherwise.

Do not make any legal claim against anyone accusing the Work, with or without changes, alone or with other works, of infringing any patent claim.

Representations Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHER-WISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICEN-SOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. The Sections on definitions, fair dealing rights, representations, warranties, and disclaimer, limitation on liability, termination, and revised license versions will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

Revised License Versions

This License may receive future revisions in the original spirit of the license intended to strengthen This License. Each version of This License has an incrementing version number.

Unless otherwise specified like in the below subsection The Licensor has only granted this current version of This License for The Work. In this case future revisions do not apply.

The Licensor may specify that the latest available revision of This License be used for The Work by either explicitly writing so or by suffixing the License URI with a "+" symbol.

The Licensor may specify that The Work is also available under the terms of This License's current revision as well as specific future revisions. The Licensor may do this by writing it explicitly or suffixing the License URI with any additional version numbers each separated by a comma.

Miscellaneous

Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If the Work is classified as Software, each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a copy and/or URI of the corresponding Source Code on the same terms and conditions as the license granted to You under this License.

If the Work is used as a Network Service, each time You Distribute or Publicly Perform an Adaptation, or serve data derived from the Software, the Licensor offers to any recipients of the data a copy and/or URI of the corresponding Source Code on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.