Three Takes on Taking Care

The second issue of *Networks of One's Own* was initiated by Varia, a Rotterdam based initiative which focuses on working with, on and through everyday technology.

This publication is an occasion to revisit three very different projects that have been important for the emergence of Varia, and for its individual members. These projects give insights into how and why we engage with collaborative practices. While we were gathering in different formations to initiate, develop, organise or present these projects, Varia was being conceived, first as an idea, then as a group and later as a physical space.

As artists, designers and researchers working with and around free/libre open source technology, the authorship of the tools and research that we develop is shared, not just amongst us at Varia but also with a wider international network. Multiple agents are involved at different moments, with varying intensity and for a range of different reasons. This results in intricate interrelationships of ownership which complicate documentation and long-term maintenance. Relational projects need flexible support structures that can handle and hold fragmentation and clustering.

For the projects in this publication, a chain of interrelated events provided an infrastructure for ongoing reflection, sharing and collaborative attention to such projects. Our path crossed at Varia in Rotterdam but also during 35c3 in Leipzig, at the Computer Grrrls exhibition in Paris, at Constant worksessions in Brussels and Antwerp, in conversation with XPUB tutors and students or at AMRO in Linz. While a lot of work was done by bringing these projects to different meeting points, we felt there was a need for concentrated rumination and active documentation. By collapsing maintenance work with publishing, we found a form that enabled this collective care work to happen and to spend the necessary concentrated time together. The making of this publication created a space to interconnect multiple ways of taking care. As a result, we hope that the projects have become more clearly articulated and are now ready to be released back into the networks to form new constellations.

Bibliotecha

Bibliotecha (https://bibliotecha.info/) proposes an alternative model of distribution for digital texts. It allows specific communities to form and share their collections, through a single-board computer running free software to share books over a local WIFI hotspot.

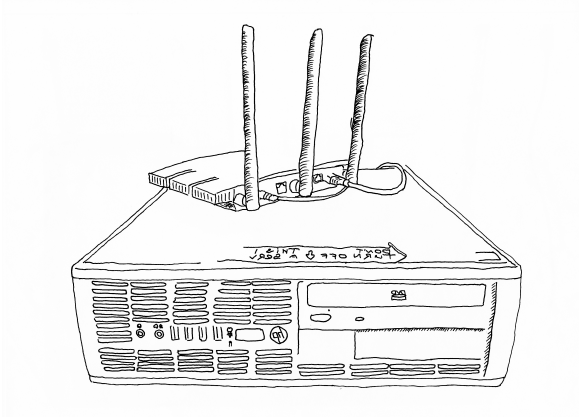
Bibliotecha's history begins in the Piet Zwart Institute, where there was an active culture of reading and sharing study material made available via a common bookshelf. In the same convivial way the students took, shared and returned books from that bookshelf, they sent each other PDFs over the internal mailinglist. That happened until the moment they were asked to stop, since sharing these books might involve copyright violations. While digital formats and the internet should make it easier than ever to share books, digital rights management and repressive copyright systems make the physical book paradoxically easier to share than its digital counterpart. In response to this we came up with what would eventually be Bibliotecha, a digital library that was available via its own off-line network. The project started at the *'Free Libraries For Every Soul'* (http://impakt.nl/archive/2013-festival/program-2013-festival/special-projects/free-libraries-for-every-soul/) hackathon at Impakt 2013. Since then, it has been presented as an installation and a workshop at different events and venues. Introducing the project to new audiences provided an opportunity for the contributors to travel with the work and gain experience and exposure as cultural practitioners. While Bibliotecha addressed a practical need at the time, its real importance has been to open up the forms and possibilities of a shared practice on the one hand and the computer network on the other.

For this issue we interviewed different people using Bibliotecha to get a sense of how the project has been used. In addition we updated the software and rewrote the installation manual. A *source code repository* (Bibliotecha/

bibliotecha-install) is included in this publication.

Start reading: *interviews* (Bibliotecha/index.html)
Publication tool: mkdocs

Homebrewserver.club



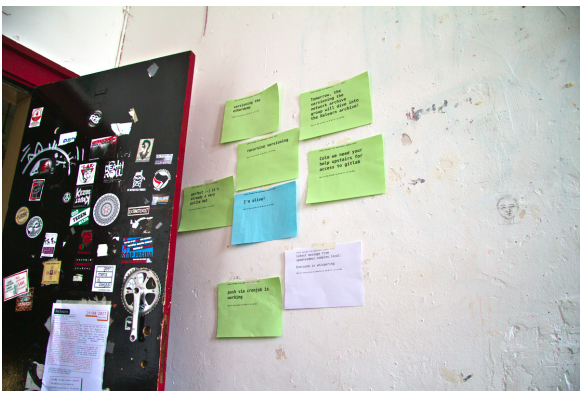
The homebrewserver.club started in 2014 after attending a few editions of the Rotterdam Crypto Party. While Crypto Parties were focused on encryption and privacy — essentially offering cryptography as a solution to surveillance and corporate dominance — there was a parallel interest to look at the more systemic issues of corporate platforms. Out of this interest the homebrewserver.club was founded as a way to learn about hosting one's own on-line services rather than relying on corporate ones. The homebrewserver.club takes the 'home' in homebrewserver.club literally and the 'self' in self-hosting figuratively. That means its members host from their homes rather than from datacenters, for and with their communities rather than just for themselves. The club has a history of recurring but irregular meetings, first at the Wunderbar in Worm, later at the Piet Zwart Institute (after closing hours) and finally in Varia. The club meetings tended to be thematic in nature, focusing on a particular topic (VPN, Mail, Web Servers) and opening up the collective knowledge that existed in the group of participants. In the early days there was also a roaming server, used as a 'club house'. This server, which would be hosted by different participants after each meeting, served as an on-line social space. Later, the focus in the club shifted to more in-depth and contextual tutorials that were published as series on *http://homebrewserver.club* (http://homebrewserver.club). The club has thus worked as place for collective learning and skill building, where technological choices get

contextualized on the axes political-economy and DIY amateurism.

We re-articulated the HBSC manifesto and added several tutorials on how to get started with self-hosting. A *redesign of the website* (https://git.vvvvvvaria.org/varia/homebrewserver.club) is part of this second edition of NOOO.

Start reading: *manifesto* (Homebrewserver.club/index.html)
Publication tool: Pelican

Ruminating Relearn



Relearn is a collective learning experiment with as many teachers as it has participants, week-long gatherings that have been taking place since 2013. In 2017 a group of people now involved in Varia decided to organise an edition of Relearn in Rotterdam.

For this publication, we ruminated the traces of that event. By actively documenting the experience of Relearn, we highlighted and spent time with questions that continue to circulate in the network of participant-teachers. We *annotated sources* (Ruminating-Relearn/relearn-2017.sources/) to generate *reflections* (Ruminating-Relearn/Ruminating-Relearn.html) and to open up the discussions to a wider audience with the help of a *glossary* (Ruminating-Relearn/Ruminating-Relearn.html#call-for-tracksintroduction-introduction).

Start reading: *preamble* (Ruminating-Relearn/index.html)
Publication tools: Etherpump & Distribusi & EXIF-image-editor

Colophon

This issue was written, edited, designed and produced in the summer of 2019 between Rotterdam and Brussels.

With contributions from Varia (Manetta Berends, Yoana Buzova, Cristina Cochior, Dennis de Bel, Silvio Lorusso, Luke Murphy, Lídia Pereira and Roel Roscam Abbing), Constant (Femke Snelting) and Colm O'Neill.

Thank you: distribusi, etherpad, etherdump, etherpump, mkdocs 1.0.4, mkdocs-alabaster, EXIF-image-editor, Pelican and Pandoc

Edition: 100 + *https://networksofonesown.varia.zone* (https://networksofonesown.varia.zone)

This event is made possible with the kind support of the Creative Industries Fund NL.

creative
industries
fund NL

A READ/WRITE version of this document can be found here: *https://pad.vvvvvvaria.org/nooo2.episodeintro* (https://pad.vvvvvvaria.org/nooo2.episodeintro)

This file can be created with the following command:

```
$ curl https://pad.vvvvvvaria.org/
  nooo2.episodeintro/export/txt | pandoc
  -f markdown -t html -s -c assets/three-
  takes-on-taking-care.css -o three-
  takes-on-taking-care.html && echo "
  Written at $(date '+%A %d %B %Y (%H:
  %Mh)')." >> three-takes-on-taking-
  care.html
```

Written at Thursday 26 September 2019 (13:00h).

Table of Content

Bibliotecha

Homebrewserver.club

Ruminating Relearn

README files

Bibliotecha Interviews

Since Bibliotecha is built around the idea of offline networks which are hosted by disparate communities to host their digital libraries, it is hard to know how and by whom the project is used. To be able to share some of the stories using Bibliotecha, we reached out to people who have used the project and documented their stories.

Bibliotecha parasiting on the bookshelf in Varia

Keeping it on and off

A text chat with Vedran Gligo.

Bibliotecha was hosted in *Hacklab01.org* (https://Hacklab01.org/) in Zagreb, Croatia in 2016. This after Vedran Gligo, a member of the lab attended a Bibliotecha workshop that took place during the *2016 Dan D Festival* (http://2016.dan-d.info/en/bibliotecha-2/).

Are you wondering who Vedran is? Well, Vedran Gligo is a DIY artist / hacker based in Hacklab01.org in AKC Attack in Zagreb, Croatia. He dabbles in glitch art and loves wrecking havoc on various formats. Real life passions include but are not limited to

frontend web development, GNU/Linux promotion and system administration, and holding free public workshops from those fields.

To find out about Vedran’s experience with Bibliotecha we interviewed him. In the end it becomes a chat with Vedran over instant messaging, since he is on the move in Zagreb at the time. On top of that, he is interrupted by a big summer storm and he has to hide until the rain stops. We talk from the safe, dry shelter he found somewhere in the streets of Zagreb.

At the time Vedran set up the Bibliotecha in the Hacklab01, he had some issues with the installation, and had to manually copy files, otherwise they would get corrupt downloads. That issue was known to us and fixed at a later stage, but perhaps Vedran didn't follow on that since what he tells us is that they actively needed the hardware. Meaning they were ‘keeping Bibliotecha on and off’, using the Raspberry Pi for other works at the lab.

After he tells us a lightning bolt strikes 100 meters from Vedran he runs off to a tram, trying to leave the storm and escape the danger of unwanted time travel. As the conversation continues he tells us casually that the Bibliotecha was installed on 'top of a bunch of other servers'. Blending in with the locals. The library it hosted contained material collected by the Hacklab01 community over the years, ranging from ‘science stuff to anarchy stuff’ as Vedran puts it.

They are all contributions from the lab members. They had a note next to it, giving a short description of Bibliotecha and how to use it, intending to stimulate people to share books. Vedran has the opinion that, rather than a librarian to curate the content, the library is in more use of a 'sysadmin' to keep up the system running (we imagine him almost laughing about this one). Directly afterwards however, he gently mentions that caring about the content actually sounds nice too. The choice of words here is important. 'Caring', not 'curating'.

Finally he shares his long-standing desire to set up a Bibliotecha at an anarchist bookstore in Zagreb, called *Stocitas* (https://www.stocitas.org/). So far, it seems that the new Bibliotecha is a candidate for being present in two locations in Zagreb. And as we talk about removing the whole setup external Wi-Fi dongles and their drivers, since the newest build is for boards with on-board Wi-fi,

Vedran warns us we don't need the newest Raspberry pi, unless we need 4K video and I can almost hear both us and him, laughing again, in between the lines.

A nomad archive

A conversation with *Martin la Roche* (http://www.martinlaroche.nl/).

Martin la Roche got introduced to Bibliotecha after he and Kym Ward had the idea to initiate a digital library project at the Jan van Eyck in Maastricht in 2015. During their residency they invited Bibliotecha contributors to give a workshop. That digital library project started off using the Bibliotecha framework and was called the *White Elephant Biblioteque*. The WEB was meant to serve as a communal art space, shared with a group of artists, researchers, writers, designers, etc. The group met every week and the plan was to have two Bibliotecha nodes. One to stay at the Jan van Eyck and another one that Martin carried with him. The Bibliotecha library hosted at the Van Eyck disappeared mysteriously.

To this day no-one knows exactly what happened to it. There was a person in charge of taking care of it, but gradually neglecting that responsibility resulted in a disappearance. The second Bibliotecha that Martin carried around with him however, was a clone of the missing one. During this carrying around he developing it as a 'nomad archive'. His basic methodology was, to travel with the library to different places and activate it once in the location. The offline library provided space for a more intimate, personal encounter every place it visited. The WEB was however also an on-line archive, and Martin mentions that bringing an actual physical object, would be pointless, and serve more as an advertisement, a sort of gimmick.

From each location the library travelled to, it got contributions to the collection. However, the distribution of the accumulated archive was visibly more common than people upload new content. That led Martin to another direction in the use of the library. It organically became a vessel, a carrier for disseminating information relevant to his own practice. Not so much as an ornament to it, or a way of promoting it, but as an extension. Until then, the work he made was supported by a website or publication. Since having developed the White Elephant Biblioteque he also made sure it is present in a digital format and available in the library. As he puts it, he

started thinking of it as a space not only for the developing archive, but also as a space for his practice, a hybrid between both, the space of coexistence of the two collections.

Since the start in 2015, counting four years at the time of writing, the library has travelled to Madrid, Sao Paolo, Santiago and Amsterdam, where it currently resides. It is not permanently on, and not in a pubic space. Martin activates it when reaching a new destination, where it becomes available and public for the duration of him being there. It is a travelling library.

He never removed anything from the archive, and the collection now counts around 100 titles. As Martin believes he privileges quality, every invitation for contributions is in context of the project. The library circulate to promote the use of it as well as facilitate a dialogue, rather than to distribute the 100 titles, because the library is also available on-line.

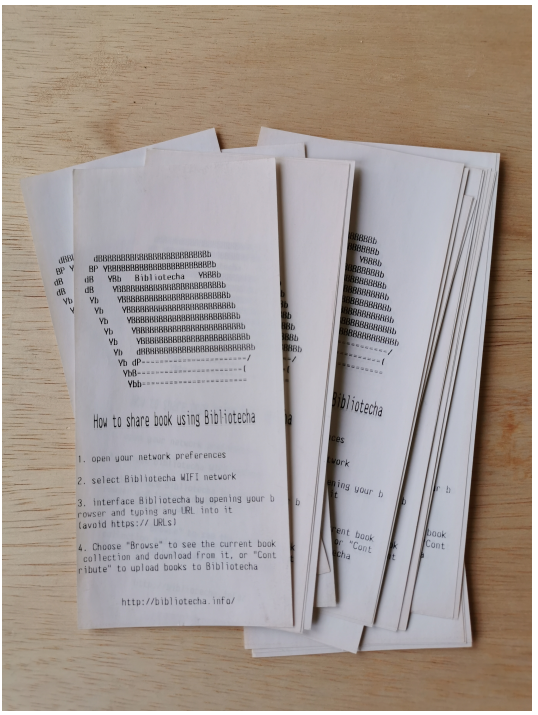
He shares with us an interesting thought he had about the project: He said if he is to be allowed to dream he would want to keep track of the physical book collection he travels with along with the digital library. Thinking of a hybrid librarian tool, allowing him to note down the books he borrowed out to people. He is curious about the mix of the concrete printed matter with the digital books and as a way of organisation, even though it may be mainly symbolic.

About librarianship, he mentions The White Elephant Bibliotheque started as a communal project, curated by the group. It then blended with his own archive and without having removed the previous collection, he still maintains both. Thus the library is in a situation of both having and not having a librarian. There is urgency for someone maintaining the library. By using it, promoting it, keeping it actively accessible, even when not not at all times, then at very specific times and places. Martin likes to treat digital books the same way he treats physical ones. 'I want to give them special attention and space as well as talk about them, because sometimes, we treat digital books too much as pure information, and pure information is gone, you look at it when you need it but hen you just leave it there, so I think there is a need for transforming this digital information into a unit we use and care about.' He fantasises about a folder that limits its content to just ten files, and thus hosts the ten most important digital books at that moment. Every time a

new book must enter, he must dismiss another. He thinks about a Bibliotecha that does not host a vast collection but rather allows just ten items and every time a new item enters, it is a decision. Again he notes, to give attention, to have space. 'I have the feeling that in the end it is not just about accessibility but also attention and giving time. It is the same with printed books. You can have a very huge library and then you never read.'

Currently, using an old version of Bibliotecha, Martin shares an issue he faces. With every upload, he has to manually upload the content to Calibre. This obstacle though, has become a tool, a tool that puts him in the position to have to review every entry. And even though he always uploaded all uploads and never dismissed a contribution, this created an interesting moment, because he was forced to do it, it was not automatic.

Martin concludes our conversation by the sharing that even though his use of Bibliotecha is very specific, he appreciates the project being not only a digital offline library, but a tool to initiate and facilitate dialogue and to ask questions.



A thermal receipt print manual on how to use Bibliotecha

At the Interface Lab in Kunstuniversität Linz

A conversation with *Andreas Zingerle* (http://www.andreaszingerle.com/).

Andreas participated in the Bibliotecha workshop during Art Meets Radical Openness

in Linz in 2014. He recounts how during the workshop, he went through the whole process of setting up, rather than not just copying the image and in the process creating documentation together for future users, which he thought was really good.

He participated at the AMRO festival with his bachelor students from the *Timebased and Interactive Media Department* from the Arts University in Linz. Back then he was teaching a class called 'introduction in physical computing'. One of the homework assignments was to participate in AMRO. The students could choose any event during the festival to participate in themselves and after having noticed they hadn't attended that workshop, he used a later classes to introduce them to the project.

He got interested in using Bibliotecha during his classes and uploaded relevant materials, texts, tutorials. Whenever he wanted to share a text with them, before passing around a USB drive, he would urge them to try Bibliotecha and get the files from there. While not having planned it, he says it turned out to work very well within his teaching. While the class was only once a week, he left his Bibliotecha installation in the lab so it could have a fixed space making it more or less permanent. On the door of the interface lab there was a poster so that students would be aware of its existence. The poster also contained information on how to use it. Throughout the duration of the course kept coming back to it as the place where people could find class material. However since the classes didn't take place in the lab itself a lot of students were asking: 'So what does this mean, in the break I have to go to the Interface lab to download it, why cannot I have it now?'. "So it seemed to be a bit complicated for them.", Andreas says.

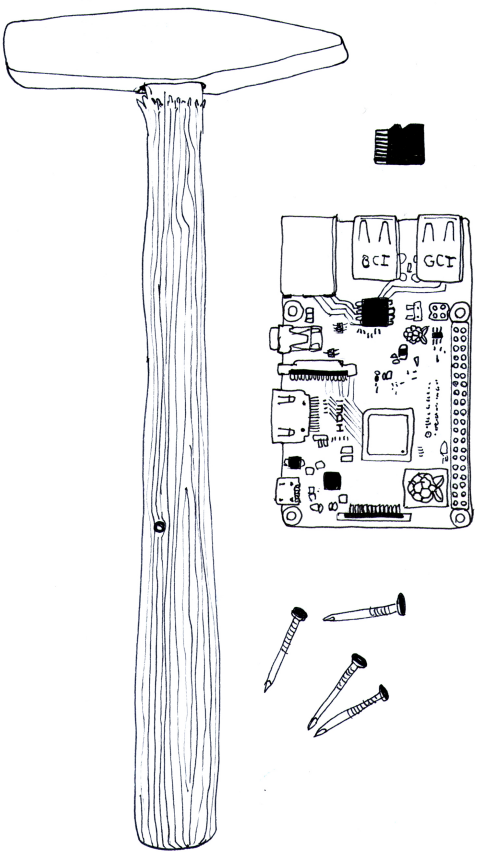
The Bibliotecha node was kept in the Lab, making the library accessible to all bachelor students, and they continued to use it. Though Andreas doesn't know for sure how much, there was a steady stream of files being added.

Later the Bibliotecha installation was used for other occasions like during a workshop series on Internet crime and fraud where the sensitive materials that Andreas didn't want to appear on the internet where shared through Bibliotecha instead. For him it became an additional tool that helped him during his workshops, and as he used it to share other texts relevant to the workshops it became more relevant to the audience as well.

In the end the Interface lab was a set of different working stations for the students and functioned as a meeting space where many different people would come. The collection consisted mostly of books on physical computing, programming, some media theory papers and shared study material. How effective Bibliotecha was in such a situation was doubtful, since he thinks most people want to share specific books with specific people. The students were aware of the project but he thinks a lot of students eventually preferred physical books over the digital ones.

Bibliotecha Manual

Bibliotecha is a framework to facilitate the local distribution of digital publications within a small community. It relies on a microcomputer running open-source software to serve books over a local wifi hotspot. Using the browser to connect to the library one can retrieve or donate texts. Bibliotecha proposes an alternative model of distribution of digital texts that allows specific communities to form and share their own collections.



Introduction

Welcome to the Bibliotecha manual! This guide serves as a human-friendly document for setting up an offline-first local library for yourself and your community.

Prerequisites

Bibliotecha is made specifically for use on the cheap and widely accessible *Raspberry Pi* (https://www.raspberrypi.org/) single board computer and Debian based *Raspbian* (https://www.raspberrypi.org/documentation/raspbian/) operating system.

You should follow the *official setup documentation* (https://www.raspberrypi.org/documentation/setup/) on the Raspberry Pi website in order to get your board up and running. You will need to have access to the internet in order to download the necessary packages as well as access to the command-line.

The current latest *Raspberry Pi 3 B+* (https://www.raspberrypi.org/products/raspberry-pi-3-model-b/) model is recommended. This model is chosen because it offers a built-in wireless card for convenient networking and a sufficient memory allowance of 1GB. It is possible to use other models of board but they should at least provide these guarantees.

Raspbian Buster (https://www.raspberrypi.org/blog/buster-the-new-version-of-raspbian/) is the current latest recommended version of the standard Raspberry Pi operating system. *Etcher* (https://www.balena.io/etcher/) is a useful and simple tool for flashing the operationg system onto the SD card which you will plug into your Raspberry Pi.

Pre-installation

Before getting started, we need to perform some preparatory steps. These steps must be completed successfully before moving on with the rest of the guide.

You should run the following commands at the command-line interface of your Raspberry Pi.

Firstly, we switch our user to the root account:

```
$ sudo -i
```

We then perform the initial system update and upgrade:

```
$ apt update
$ apt upgrade
```

We should then perform a number of steps within the *raspi-config* (https://www.raspberrypi.org/documentation/configuration/raspi-config.md) tool:

```
$ raspi-config
```

- Change the user password
  - Choose the `Change User Password` option.
  - It is important to configure your Raspberry Pi with a secure passphrase. A *diceware passphrase* (https://www.rempe.us/diceware/#eff) is a recommended

approach for choosing a sufficiently strong passphrase.

- Choose a hostname
  - Follow the `Network Options > Hostname` options.
  - The hostname will be the name that identifies the Raspberry Pi on the local network.
- Configure predictable network interfaces
  - Follow the `Network Options > Network interface names` options.
  - It is important to enable predictable network interface names so that the automatic installation script can detect which network interfaces are in use.
- Configure the localisation
  - Follow the `Localisation Options > Change Locale` options.
  - It is recommended to ensure that the `en_GB.UTF-8 UTF-8` locale is selected. This is the default. Once this is selected, select `<Ok>` on the two following dialogs to generate the locale.
- Expand the SD card partition
  - Follow the `Advanced Options > Expand filesystem` options.
  - This allows more space on the SD card to be used. This is important for when you will start to place more and more digital books in your Bibliotecha.

The Raspi-config interface then ask you to restart the Raspberry Pi which you should do. If not, you can also run this from the command-line:

```
$ reboot
```

Remember, you will need to use your new user passphrase to access the Raspberry Pi after rebooting it. Make sure you store this passphrase somewhere safe!

Automated installation

It is now time to run the automatic installation. This script will install and configure all the necessary moving parts of Bibliotecha.

If you're interested in doing this process manually (for the purpose of learning, for example), a *Manual installation* (#manual-installation) guide is provided.

If you would first like to read the script, the *source is available* (https://git.vvvvvvaria.org/varia/bibliotecha-install/src/branch/master/bibliotecha.sh).

On the Raspberry Pi once again, run the following commands:

```
$ sudo -i
$ curl https://
install.bibliotecha.info | bash
```

The script will automatically reboot your Raspberry Pi when it is finished.

If you run into any issues, please see the *Troubleshooting section* (#troubleshooting).

Post-installation

After rebooting, there should be a Wifi hotspot available with the name `Bibliotecha`. You should wait a few minutes for this hotspot to become available. This Wifi access point is being served from the Raspberry Pi. You should be able to connect to this Wifi. It is not password protected.

Once connected you should be directed to the so-called "captive portal" of the Bibliotecha where it explains how to enter the library and use it. The library should be available at *http://bibliotecha.library* (http://bibliotecha.library).

It is recommended to customise your captive portal page to suite your own needs. This file is available in the `/var/www/bibliotecha/index.html` location. You should also take a look at customising your `/etc/motd` SSH welcome banner.

You will be required to configure the *Calibre-web* (https://github.com/janeczku/calibre-web) installation. Extended configuration documentation is available from the *Calibre-web wiki* (https://github.com/janeczku/calibre-web/wiki/Configuration). The most important part is the "Location of Calibre database" for which you can enter the following path:

    /opt/calibre-database

This is the default installation path used by the installation script.

You may also want to look at the "Feature Configuration" section where you can decide whether to allow uploading, anonymous browsing and allowing public registrations. These depend on your context and to whom you will serve the library to.

Click "Submit", "Login" and you will be redirected to the library login page. The default adminstration password login details are:

> Username: admin
>
> Password: admin123

You should change these details to secure your adminstration account.

Maintaining a Community Library

Once your Bibliotecha is configured you can start to think about how you and your community would like to maintain the library. You should ask yourself some questions:

- Who will be the digital librarians? The catalogue will need care.
- Will you allow public registrations? Will you allow public uploads?
- How will you publicise the library within the local context?
- What kind of library do you want to create? What are the themes?
- Who will be responsible for maintaing the system?

Understanding Bibliotecha Networking

Bibliotecha uses standard, venerable and stable GNU/Linux networking tools and configuration to enable the *local-first* (https://www.inkandswitch.com/local-first.html) networking setup. Installing, running and maintaining a network configuration is no easy topic! However, it is a useful skill to have. Overall, Bibliotecha is made up of the following programs and configurations:

- */etc/network/interfaces.d/* (https://manpages.debian.org/buster/ifupdown/interfaces.5.en.html): The network interface configuration
- */etc/hosts* (https://manpages.debian.org/buster/manpages/hosts.5.en.html): The hostname definitions
- *Hostap* (https://wiki.debian.org/hostap): The Wifi access point provider
- *Dnsmasq* (https://wiki.debian.org/HowTo/dnsmasq): The DNS and DHCP server
- *Dhcpcd* (https://manpages.debian.org/buster/dhcpcd5/dhcpcd.8.en.html): The DHCP client
- *Calibre* (https://calibre-ebook.com/): The library database

- *Calibre-web* (https://github.com/janeczku/calibre-web/): The library web application
- *Lighttpd* (https://www.lighttpd.net/): The web server

When your Bibliotecha is setup and running, it is doing a number of things. It is first serving a Wireless access point (Hostap) which your devices can connect to. After you connect, your device is given an IP address on the local network (Dnsmasq and Dhcpcd) as well as a local DNS entry (mydevice.library, for example). Once you open a web browser, it will indicate that you need to log into the network, but in fact you are brought to a web page (Lighttpd) which shows you how to reach the library web application (Calibre-web).

Troubleshooting

Because Bibliotecha is made up of a number of moving parts it is not feasible for this manual to cover all the possible issues. However, we try our best here to provide context, background, useful tips and tricks to help you become familiar with fixing your Bibliotecha. We recommend a DIWO (Do It With Others) approach!

Please make sure to take some time to read *Understanding Bibliotecha Networking* (#understanding-bibliotecha-networking) so that you are familiar with all the moving pieces. When troubleshooting, it is important to narrow down which piece of the puzzle is responsible. To do this, you need to know what the pieces are.

If all else fails, please send an email to the public *mailing list* (https://we.lurk.org/postorius/lists/bibliotecha.we.lurk.org/).

All of the following commands should be run as the root user.

I cannot connect to the internet from the Raspberry Pi

If you are connecting an Ethernet cable to your Bibliotecha in order to connect it to your local router and have access to the internet, then you might notice that the requests do reach their destination.

Bibliotecha is configured to capture all the network requests it receives and point them to the library interface. You will need to temporarily disable the `dnsmasq` service:

```
$ systemctl stop dnsmasq
```

It should then be possible to connect to the wider internet now.

The wireless access point is not available

The access point is responsibility of the `hostapd` program. You should check the status of the running service with:

```
$ systemctl status hostapd
```

If there are errors, you can see the logs with:

```
$ journalctl -u hostapd
```

You may also attempt to restart this service afterwards:

```
$ systemctl restart hostapd
```

You may also want to inspect the `/etc/hostapd/hostapd.conf` configuration.

I do not receive an IP address when I connect

Providing IP addresses is the responsibility of the `dnsmasq` and `dhcpcd` service. You should check the status of `dnsmasq` with:

```
$ systemctl status dnsmasq
```

You should make sure that `dhcpcd` is running with:

```
$ dhcpcd5
```

How to upgrade Calibre-web

You'll need to re-connect your Bibliotecha to the internet with an ethernet cable and then run the following commands:

```
$ systemctl stop cps
$ cd /var/www/calibre-web
$ git pull origin master
$ .venv/bin/pip install -r
requirements.txt
$ systemctl start cps
```

Manual installation

It is possible to install Bibliotecha manually. This can be useful and fun if you would like to learn more about GNU/Linux networking tools and interfaces. These skills are generally useful but especially so when considering community networks.

The following guide follows the steps of the automatic installation script.

Changing to Root

All commands should be run as the root user.

Change the user to the root user with:

```
$ sudo -i
```

Update the System

We should update the system before going further:

```
$ apt update
```

This makes sure that we have the latest package listing from the online Debian package list.

Install Networking Packages

We then need to install the networking packages that we will need:

```
$ apt install -y \
    dhcpcd \
    dnsmasq \
    dnsutils \
    hostapd \
    wireless-tools
```

Afterwards, we'll make sure to stop these services running while we work on the installation right now. We can do that with:

```
$ systemctl stop dnsmasq
$ systemctl stop hostapd
```

We do want these services to be enabled when we reboot though:

```
$ systemctl unmask hostapd
$ systemctl enable hostapd
$ systemctl enable dnsmasq
```

We'll also want to disable and stop the `avahi-daemon` which we won't be using since we rely on `dnsmasq` to handle our DNS configuration and serving:

```
$ systemctl stop avahi-daemon
$ systemctl disable avahi-daemon
```

Configure Network Interfaces

We now need to configure our network interfaces. The network interfaces correspond to the Ethernet port and the Wireless card. These are how the Raspberry Pi connect to other devices for networking uses.

We need to learn the names of our network interfaces:

```
$ ip a
```

The ethernet interface is the name beginning with "en" and the wireless interface is the one beginning with "wl". These are the predictable inteface naming conventions which we rely on.

For the following steps, I assume the following:

- Ethernet: enx78e7d1ea46da
- Wireless: wlp2s0

We then configure the ethernet interface. We put the following in `/etc/network/interfaces.d/enx78e7d1ea46da` :

```
auto eth0
allow-hotplug enx78e7d1ea46da
iface enx78e7d1ea46da inet dhcp
```

This configuration allows the default behaviour for the Ethernet interface. When you plug in an ethernet cable, typically coming from your local network router, you will receive a dynamic IP from that router. This makes it easy to reach the internet later.

We then configure the wireless interface. We put the following in `/etc/network/interfaces.d/wlp2s0` :

```
auto wlp2s0
iface wlp2s0 inet static
  address 10.0.0.1
  netmask 255.255.255.0
```

This configuration sets up a static IP address for the wireless interface. We do this so as to put the IP address within the range of the addresses which we allow in the local network. We will configure this range in the following step.

Configure Dnsmasq

In the `/etc/dnsmasq.d/wlp2s0.conf` file, we put the following:

```
bogus-priv
server=/library/10.0.0.1
local=/library/
address=/#/10.0.0.1
interface=wlp2s0
domain=library
dhcp-
range=10.0.0.50,10.0.0.200,255.255.255.0,12h
dhcp-option=3,10.0.0.1
dhcp-option=6,10.0.0.1
dhcp-authoritative
```

This configuration sets up a local `.library` domain and a range of IP addresses that can be assigned in the `10.0.0.50` -

`10.0.0.200` range. It also makes sure to resolve all unknown domain requests to the `10.0.0.1` IP. This is useful for the purposes of the captive portal configuration later on.

Configure Hostapd

We need to set up the wireless access point too. In the `/etc/hostapd/hostapd.conf` we add:

```
interface=wlp2s0
ssid=Bibliotecha
hw_mode=g
channel=11
auth_algs=1
```

We also need to make sure that `hostapd` uses this configuration. We have to ensure that the following is uncommented and present in the `/etc/default/hostapd` file:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Configure the Hosts file

We need to register the library on the network. In the `/etc/hosts` file we put the following at the end of the file after all the other entries:

```
10.0.0.1        bibliotecha.library
```

Install and Configure Lighttpd

Moving on, we should install the web server which will respond to network requests and return the web pages of the library interface. We can install Lighttpd with:

```
$ apt install -y lighttpd
```

When then need to make sure that the following configuration is available in the `/etc/lighttpd/lighttpd.conf` file:

```
server.document-root = "/var/www"
server.modules += ("mod_proxy",)
include "bibliotecha/bibliotecha.conf"
```

When then create the Bibliotecha configuration with:

```
$ mkdir /etc/lighttpd/bibliotecha
```

And then make sure the following is in the `/etc/lighttpd/bibliotecha/bibliotecha.conf` file:

```
server.error-handler-404 = "/bibliotecha/
index.html"
$HTTP["host"] ==
"bibliotecha.library" {
```

```
  proxy.server = ("" => (("host" =>
"127.0.0.1", "port" => "8083")))'
}
```

This ensures that all unknown requests are pointed to the captive portal page of Bibliotecha. And when we request the `http://bibliotecha.library` domain, we are then pointed to the Calibre-web installation.

Install and Configure Calibre

Calibre is responsible for maintaining the underyling database of the library. We can install it with:

```
$ apt install -y calibre
```

This will take some time as there are man required packages. Once it if finished, you will then need to create a new database for the Calibre-web installation to use.

We should run the following:

```
$ mkdir /opt/calibre-database
$ /usr/bin/calibredb restore_database
--really-do-it /opt/calibre-database
```

Configure the Captive Portal

When we connect to the Bibliotecha wireless access point, we will be directed to a splash page where we are introduced to the library. We need to set that up:

```
$ mkdir /var/www/bibliotecha
```

And then we download the default page into location:

```
$ apt install -y wget
$ wget https://git.vvvvvvaria.org/
varia/bibliotecha-captive-portal/raw/
branch/master/index.html -O /var/www/
bibliotecha/index.html
```

We should also ensure that the correct ownership permissions are configured:

```
$ chown -R www-data: /var/www/bibliotecha
```

Install and Configure Calibre-web

We now setup the Calibre-web installation. We first get a copy of the source with:

```
$ mkdir /var/www/calibre-web
$ git clone https://github.com/
janeczku/calibre-web /var/ww/calibre-
web
$ cd /var/www/calibre-web
```

We then need to install the Python dependencies with:

```
$ python3 -m venv .venv && .venv/bin/pip
install -r requirements.txt
```

And finally configure the service to be run by Systemd. In the `/etc/systemd/system/cps.service` we need to add:

```
Description=Calibre-Web

[Service]
Type=simple
User=root
ExecStart=/var/www/calibre-web/.venv/
bin/python /var/www/calibre-web/cps.py
WorkingDirectory=/var/www/calibre-web

[Install]
WantedBy=multi-user.target
```

And we also enable this to run on reboot:

```
$ systemctl enable cps.service
```

We should also ensure that the correct ownership permissions are configured:

```
$ chown -R www-data: /var/www/calibre-web
```

Setup a new MOTD

When you SSH into the Raspberry Pi, you can enable a nice welcome message.

This is possible bu putting the following in the `/etc/motd/` file:

```
 ____  _ _     _ _       _            _
|  _ \(_) |   | (_)     | |          | |
| |_) |_| |__ | |_  ___ | |_ ___  ___| |__   __ _
|  _ <| | '_ \| | |/ _ \| __/ _ \/ __| '_ \ / _` |
| |_) | | |_) | | | (_) | ||  __/ (__| | | | (_| |
|____/|_|_.__/|_|_|\___/ \__\___|\___|_| |_|\__,_|

      Digital books need libraries too
```

Conclusion

That's it! You should now reboot your Raspberry Pi with:

```
$ reboot
```

You can now follow the *post-installation* (#post-installation) steps.

Contributing

Bibliotecha is made up of the following projects:

- *bibliotecha-install* (https://git.vvvvvvaria.org/varia/bibliotecha-install)
- *bibliotecha-manual* (https://git.vvvvvvaria.org/varia/bibliotecha-manual)
- *bibliotecha-captive-portal* (https://git.vvvvvvaria.org/varia/bibliotecha-captive-portal)

All contributions are welcome!

You can also find us on the *mailing list* (https://we.lurk.org/postorius/lists/bibliotecha.we.lurk.org/).

Acknowledgements

- The *Calibre* (https://calibre-ebook.com) project
- The *Calibre-web* (https://github.com/janeczku/calibre-web) project

Contributors to Bibliotecha have been Yoana Buzova, Lasse van den Bosch Christensen, Andre Castro, Lucia Dossin, Max Dovey, Michaela Lakova, Luke Murphy and Roel Roscam Abbing

Powered by mkdocs 1.0.4 (http://www.mkdocs.org) *& mkdocs-alabaster* (https://github.com/iamale/mkdocs-alabaster)

The homebrewserver.club principles

Mon 17 June 2019
By hbsc & friends

Promote approaches, not apps

We privilege general approaches over particular software applications. We try to contextualize our technical choices socially, politically and economically to provide in-depth understanding and prevent The Best Way™ solutionism. For these reasons we like free and open source software as a starting point and try to provide documentation for others to learn as well.

Take the ‘home’ in homebrewserver.club literally and the ‘self’ in self-hosting figuratively

That means we try to host from our homes rather than from data centers - a.k.a. ‘the cloud’ - and we try to host for and with our communities rather than just for ourselves.

Make space for learning together

Primarily the homebrewserver.club wants to be a space for learning together. This either means that you are knowledgable about a topic and willing to share or that you are curious and willing to learn. We are all about the long route that provides grounded understandings instead of mindlessly copy pasting things into the terminal to install software via $current_hip_framework.

Yes, We’re Config™

We try help each other out out but we can’t do the work for you. We’re Config, meaning we’re comfy with figuring things out and making mistakes. We take pleasure in researching configurations and in the struggle of getting things working. We gladly lend you a helping hand, but we won’t be able to hold your hand through the whole process.

Serve from constraints

Having a homebrew server means owning up to the fact that you are serving from constraints. You don't have the fastest connection. You use as little power as possible. Your server is some spare laptop you had lying around. You want as little maintenance or worrying as possible. You're not on 24/7 stand by if there is an issue. All of that is perfectly ok. Knowing this influences our choices in terms of what to run and how.

Be an amateur

What works for data centers and the industry does not necessarily work for homebrew servers. As such, the homebrewserver.club documentation won't be exhaustive. We rather intend to add to existing on-line knowledge, particularly from the perspective of the homebrew server admin.

Embrace the Feminist Server Manifesto

We subscribe to the principles of the *Feminist Server Manifesto* (https://areyoubeingserved.constantvzw.org/Summit_afterlife.xhtml) forwarded by our friends at Constant, a non-profit, artist-run organisation based in Brussels active in the fields of art, media and technology.

Aspire to broaden participation

By publishing guides and tutorials, we hope to facilitate the means for diverse communities to learn how to set up and run their server according to their needs, goals and principles. For us, this means our channels of communication are an harassment-free experience for everyone, regardless of gender, gender expression, race, ethnicity, religion, sexual orientation, sexual characteristics, disability, age or technological ability. Individuals promoting bigotry will be removed from the channels.

Setting up a web server

Wed 14 August 2019
By misschienaasappel

Introduction

Ever wanted to host your own website from the comfort of your house? Ever wondered how to achieve this? Search no further! This guide will help you with the installation and configuration of web server software, which is what allows a computer to start handling HTTP requests and serve web content in response.

Besides helping you with the installation, this guide will help you getting the right certificates, configuring your server and publishing your homebrew served website.

Some background knowledge

First off: what is the web, what is a web site and what is a web server?

The web is the single most known part of the internet. Because of that, it often happens that ‘the web’ and ‘the internet’ become conflated. Therefore, it often becomes a bit hazy to state what the difference is between the internet and the web.
Generally speaking ‘the web’ is only the part of the internet that we interact with with a web browser. More technically speaking, the web is the part of the internet that runs on port 80 and port 443 and that uses the HTTP and HTTPS protocols.

Websites are text documents that are formatted through HTML, CSS and JS. These three technologies tell the web browser what the structure of the page is, how it should be laid out and what kind of interactions are possible. Websites are transmitted using Hyper Text Transfer Protocol, which is why we usually type them like so `http://homebrewserver.club`.

A web server is a piece of software which listens for and responds to HTTP requests.

So in essence the web is a network of webservers which runs on top of the internet and through which websites can be retrieved.

Requirements

- A spare computer.
- A basic understanding of the command line.
- An *ssh server and client* (demystifying-ssh.html) installed.
- A *registered domain name* (https://computer.howstuffworks.com/dns.htm).
- Have an available power socket next to your router.
- An ethernet cable to connect your server to the router.

The instructions on this guide were run on a Debian Stretch distribution.

Installing Apache

The Apache HTTP server is a free and open-source web server software and it has been around since 1995, being the most widely used server software in the world. Because of this, documentation is plentiful and the support community is very large, meaning that help is quite easy to get for any of your web server issues.
For this reason, Apache has been selected for this guide.

There are, of course, other web server software available, the most popular of which being Nginx. Nginx, which is also free and open-source software, arrived on the scene circa 2004, and it also became a favourite for its resource efficiency.

If you want to geek out further about the differences between Apache and Nginx, *this article* (https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations) will give you an overview.

So, without further ado, open a terminal window and let's get started:

First, make sure you update your packages list:

```
$ sudo apt update
```

Then, install the Apache HTTP server software:

```
$ sudo apt install apache2
```

If all went well, Apache should have been started immediately after installation. To double check this, run:

```
$ sudo systemctl status apache2
```

Example output:

```
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/
  apache2.service; enabled; vendor preset:
     Active: active (running) since Sat 2019-06-22
  21:29:51 UTC; 6s ago
   Main PID: 18398 (apache2)
        CPU: 573ms
     CGroup: /system.slice/apache2.service
             ├─18398 /usr/sbin/apache2 -k start
             ├─18402 /usr/sbin/apache2 -k start
             ├─18403 /usr/sbin/apache2 -k start
             ├─18404 /usr/sbin/apache2 -k start
             ├─18405 /usr/sbin/apache2 -k start
             └─18406 /usr/sbin/apache2 -k start

  Jun 22 21:29:50 supermuch systemd[1]: Starting The
  Apache HTTP Server...
  Jun 22 21:29:51 supermuch systemd[1]: Started The
  Apache HTTP Server.
```

Configuration Time

You can find Apache's configuration files in the following location: /etc/apache2/sites-available.

The 000-default.conf file should look a little something like this:

```
ServerAdmin webmaster@localhost
  <VirtualHost *:80>

          # ServerName example.org

          ServerAdmin webmaster@localhost
          DocumentRoot /var/www/html

          ErrorLog ${APACHE_LOG_DIR}/error.log
          CustomLog ${APACHE_LOG_DIR}/access.log
  combined

  </VirtualHost>
```

For ease of use, and in case you would like to have several websites/services running behind a single server, copy this file into another, easily identifiable one, for example, calling it something like "mydomain.conf".

```
$ sudo cp 000-default.conf mydomain.conf
```

Using your favourite text editor, uncomment the ServerName line and change it to reflect your domain name:

```
ServerAdmin webmaster@localhost
  <VirtualHost *:80>
          ServerName mydomain.org

          ServerAdmin webmaster@localhost
          DocumentRoot /var/www/html

          ErrorLog ${APACHE_LOG_DIR}/
  mydomain.error.log
          CustomLog ${APACHE_LOG_DIR}/
  mydomain.access.log combined

  </VirtualHost>
```

Enable this configuration by running:

```
$ sudo a2ensite mydomain.org
```

Restart Apache to load the new configuration:

```
$ sudo service apache2 restart
```

HTTPS

HTTPS, which stands for *hypertext transfer protocol secure*, is an extension of the HTTP protocol. As its name suggests, it adds a layer of security to the data exchanged between client and server. By adding an encryption layer to the exchanged packets, it seeks to avoid man-in-the-middle attacks, eavesdropping, etc. While HTTP uses port 80 by default, HTTPS uses port 443.

As part of its bigger goal to "encrypt the entire Internet", the *Electronic Frontier Foundation* (https://certbot.eff.org/about/) developed Certbot, a free and open source tool for automating the server-side deployment of *Let's Encrypt Certificates* (https://letsencrypt.org/), thus enabling HTTPS.

Let's get down to it! Again, these instructions are specific to Debian 9 (Stretch), but detailed instructions for installation on other distros can be found on *Certbot's website* (https://certbot.eff.org/instructions).

First, add backports to your packages list and update it:

```
$ echo deb http://deb.debian.org/debian stretch-
  backports main | sudo tee -a /etc/apt/sources.list
  && sudo apt update
```

Now, install Certbot:

```
$ sudo apt install certbot python-certbot-apache -t
  stretch-backports
```

Run Certbot to get the right certificates for your domain:

```
$ sudo certbot certonly -d myserver.org
```

After following the process, and if all went well, you should now see the following message:

```
- Congratulations! Your certificate and chain have
  been saved at:
     /etc/letsencrypt/live/mydomain.org/fullchain.pem
     Your key file has been saved at:
     /etc/letsencrypt/live/mydomain.org/privkey.pem
     Your cert will expire on 2019-09-24. To obtain a
  new or tweaked
     version of this certificate in the future,
  simply run certbot
     again. To non-interactively renew *all* of your
  certificates, run
     "certbot renew"
   - If you like Certbot, please consider supporting
  our work by:

     Donating to ISRG / Let's Encrypt:   https://
  letsencrypt.org/donate
     Donating to EFF:                    https://
  eff.org/donate-le
```

Now, it is time to edit your /etc/apache2/sites-available/mydomain.conf file accordingly:

```
<VirtualHost *:80>
         ServerName mydomain.org

         ServerAdmin webmaster@localhost
         DocumentRoot /var/www/html

         ErrorLog ${APACHE_LOG_DIR}/
  mydomain.error.log
         CustomLog ${APACHE_LOG_DIR}/
  mydomain.access.log combined
  </VirtualHost>

  #NEW CONFIG STARTS HERE
  <IfModule mod_ssl.c>
  <VirtualHost *:443>
         ServerName mydomain.org

         ServerAdmin webmaster@localhost
         DocumentRoot /var/www/html

         ErrorLog ${APACHE_LOG_DIR}/
  mydomain.error.log
         CustomLog ${APACHE_LOG_DIR}/
  mydomain.access.log combined

      SSLEngine on
    #PATH TO YOUR CERTIFICATES (note: don't forget
  to replace mydomain.org with your actual domain
  name!)
      SSLCertificateFile /etc/letsencrypt/live/
  mydomain.org/fullchain.pem
      SSLCertificateKeyFile /etc/letsencrypt/live/
  mydomain.org/privkey.pem
  </VirtualHost>
  </IfModule>
```

In case you didn't notice, there is now an if statement that evaluates true in case a certain module is present. In this case, it evaluates to true if *mod_ssl* (http://www.modssl.org/) is present. *Apache modules* (https://en.wikipedia.org/wiki/List_of_Apache_modules) can be installed as following:

```
$ sudo a2enmod modulename
```

To verify which modules are already running on your server, type:

```
$ sudo apache2ctl -M
```

If the required ssl_module is not listed, run:

```
$ sudo a2enmod ssl
```

Certificate renewal

Your certificates expire after a period of time. You can, however, automate renewal by adding a *cron job* (https://www.ostechnix.com/a-beginners-guide-to-cron-jobs/) that schedules when the specific renewal command should be run.

Start by running:

```
sudo crontab -e
```

Add the following:

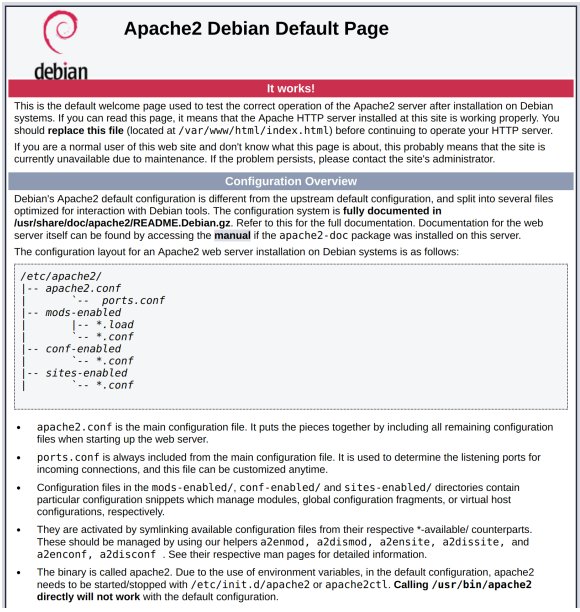
```
5 55 0 * 5  /usr/bin/certbot renew
```

This means the certificates will be renewed every week on Friday at 05:55. You can of

course edit these times according to your preferences! Save your changes and exit the editor.
Time to restart Apache and load all of these changes!

index.html

At this point, when typing https://mydomain.org into your browser, you should be greeted with a page that looks a little something like this:



If you cd into your /var/www/html folder, you will find this default index.html. As recommended by this page itself, you should edit this file before continuing operations on your webserver.
Open it on your favourite text editor and let's get started on a bare-bones "Hello Homebrew World"! webpage.

```
<!doctype html>

  <html lang="en">
  <head>
    <meta charset="utf-8">
    <title>My first homebrewed webpage</title>
  </head>

  <body>
    <h1>Hello Homebrew World!</h1>
  </body>
  </html>
```

Open your browser again and savour the fruits of your hard work.

That was it! Now you are ready to have hours of endless fun sailing the vast sea of HTML, CSS, JavaScript, etc.

Port Forwarding configuration for your home router

Mon 14 January 2019
By hbsc & friends

Introduction

The whole premise of the homebrewserver.club is the simple — yet often overlooked — fact that your home internet subscription theoretically also allows you to host services. The internet is in its essence a bi-directional medium. Anyone with an internet connection can not only look up online content but also host it!

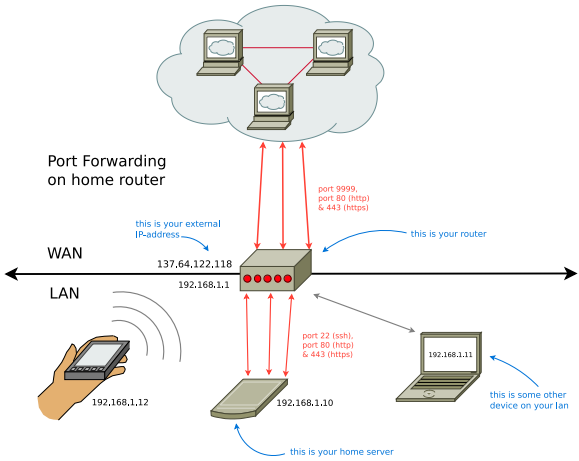
In times of ‘cloud providers’ and ‘virtual private servers’ it is an easy thing to forget, and internet service providers don’t make it easy on you either, but a homebrew server can be as simple as an old laptop connected directly to your home router. However, you do need to change some settings on the router to make that happen!

Requirements

To begin serving from home you need the following:

- Make sure you have physical access to your home router.
- Get to know the password of the admin user (this is usually provided in the box or written on the label on the underside of the router).
- Have an available power socket next to your router.
- Have a homeserver with a web server and open SSH server running on it.
- An ethernet cable to connect your server to the router.

Port forwarding theory



By default home routers have configured the firewall so that the devices behind your router are inaccessible to the internet. This is to prevent your private network from being public.

Machines behind your router (called your local area network or `LAN`) can make connections to the wider internet (known as `WAN`), but not the other way around.

However, when hosting a server at home, we do want that server to be reachable from the internet. In order to do that we need to open so-called *network ports*.

Ports are logical 'gates' that are open or closed to connections. These ports have numbers and are *standardized* (https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#Well-known_ports) for specific protocols or applications. For example, HTTP traffic from a website would default to port `80`. HTTPS defaults to `443` and SSH defaults to port `22`.

To make our server accessible over the internet we need to open the ports on the router and forward them to our server. This is called port-forwarding.

The exact method (and terminology) of port-forwarding differs from router to router. However, it always follows a similar scheme where you designate inbound traffic on a certain port to be forwarded to the your server's IP-address and port on the local area network.

For this you need to have access to the administrative panel of your router.

Find your router

To access the administrative panel of your router you need to find it's IP-address. You can do this by connecting to that router via Ethernet or Wi-Fi and then finding out what your own IP-address is.

On Debian based systems this is done like this in the terminal:

```
$ ifconfig
```

If you get a command not found warning try this:

```
$ ip address
```

This will return information on your network connection. Look for the line saying `inet`

```
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
  1500 qdisc mq state UP group default qlen 1000
  link/ether ac:ab:00:00:ac:ab brd ff:ff:ff:ff:ff:ff
  inet 192.168.1.11/24 brd 192.168.1.255 scope global
  wlp3s0
     valid_lft forever preferred_lft forever
  inet6 fe80::eab1:fcff:acab:374e/64 scope link
     valid_lft forever preferred_lft forever
```

In this case the IP-address is `192.168.1.11` as a rule of thumb you can then change the last digit to either `1` or `254` to find the router.

Log in to your home router and get to know your LAN

Using a web browser navigate to the IP-address you found above to reveal the router's admin panel. It should provide you with a log in field where you can enter the router's admin details to get access to the control panel.

There you will see a lot of possible settings. Look at the options "LAN", "DHCP Leases" or "Network" to get an overview of all the devices.

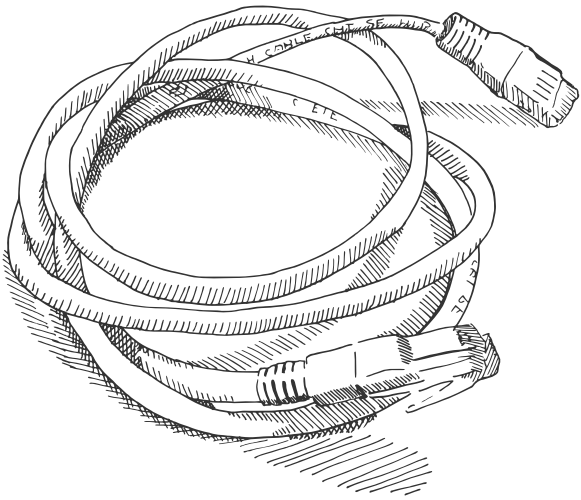
Connect your home server

Use an ethernet cable to connect your home server to your router. In case that it has ethernet ports in different colors/markings make sure you take something that says either `LAN` or `INET` .

Have a look at your router's interface again and look for the IP-address that your server was assigned. In this guide I'll assume it was 192.168.1.10.

Next try to find an option called "Static (DHCP) Lease" or "DHCP Binding" or something similar in your LAN view. Then make sure to assign your server a static DHCP

lease. This will make sure that the server is always reachable under the same IP-addres.



Forward the ports

Once you've set up a static lease to your home server you can start port forwarding. Depending on the make of the router it can be also be called *Port Sharing* or *Traffic Forwarding*. Again, depending on the make of the router, it can be found under the tabs 'security', 'access control' or 'internet'.[1] (#fn:1)

The basic process is to determine which external port to open and to which IP address on the LAN and which port to forward it to.

You might be asked a few things, including the name of the rule, the protocol (TCP, UDP or both), the external port and the internal port. Sometimes you are given the option to open a range of ports.

To open the ports for the web server, we're opening two separate ports, one for plain HTTP and one for secure HTTP.

Open the external port `80` for plain HTTP and redirect it to the local IP-address of the homeserver:

```
Name: "HTTP Homeserver"
  Protocol: TCP
  Device: 192.168.1.10
  External Port: 80
  Port to device: 80
```

Open the external port `443` for HTTPS and redirect it to the local IP-address of the homeserver:

```
Name: "HTTPS Homeserver"
  Protocol: TCP
  Device: 192.168.1.10
  External Port: 443
  Port to device: 443
```

Lastly we will open a port for `SSH`. The change here is that we open the external port `9999` and map that to `22` internally.

Setting SSH on a non-standard port is a low-level way to prevent automated scripts gaining access to your homeserver. TODO add link to basic security

```
Name: "SSH Homeserver"
  Protocol: TCP
  Device: 192.168.1.10
  External Port: 9999
  Port to device: 22
```

Concluding

Now that you have opened the corresponding ports you should be able to type your external IP-address in your browser and should be automatically redirected to the website on your home server.

How to find out which ports to open?

While a majority of applications will work on `80` and `443` you might need to open different port for different applications. For example in the series describing *self-hosted chat over XMPP* (https://homebrewserver.club/configuring-a-modern-xmpp-server.html#set-up-firewall-and-dns) ports `5000`, `5222`, `5269` and `5281` are opened and forwarded.

Most installation guides for software will tell you whether you need to open ports. However it is also possible to see what applications are listening to what port using:

```
$ netstat -tulp .
```

Notes:

1. *https://portforward.com/* (https://portforward.com/router.htm) has a large list of routers and visual instructions on how to set up port forwarding on them. ↩ (#fnref:1)

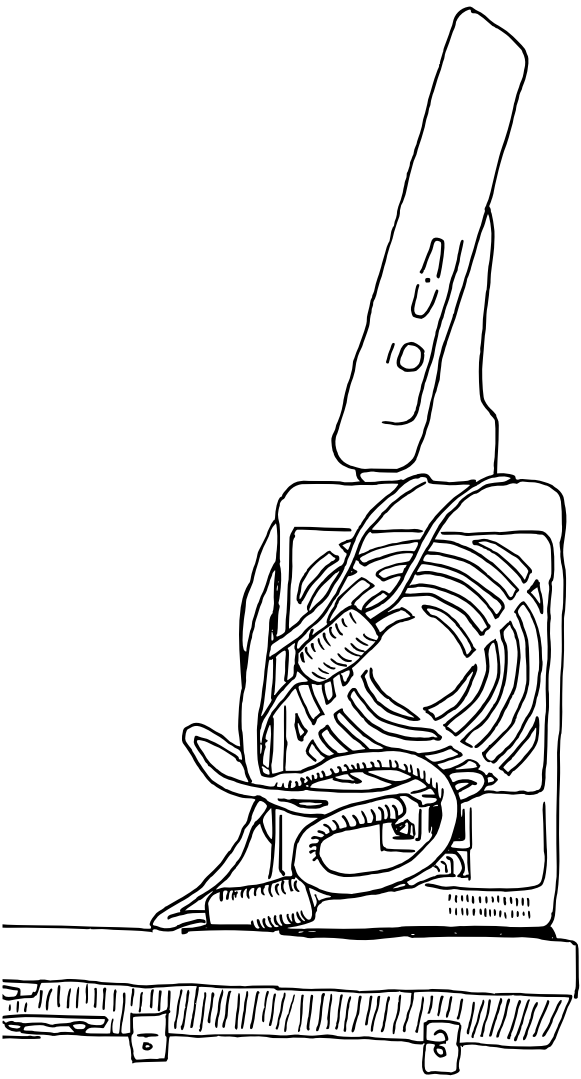
Considerations for server hardware

Wed 10 July 2019
By hbsc & friends

Introduction

You want to get started with self-hosting. This means you will need a computer that will be your server. But what makes a good server?

First, while dedicated server equipment does exist, in the case of the homebrew server it is more helpful to think of a ‘server’ as a function rather than as a special machine.



Why? Because dedicated servers are expensive, loud, specialized and power hungry devices. At the same time any spare computer with a network port running GNU/Linux can *become* a server.

It really depends what exactly you plan to do with it, but the odds are that 10 year old laptop you have lying around is more than up

to the task of hosting light personal websites, a chat server, mail and more!

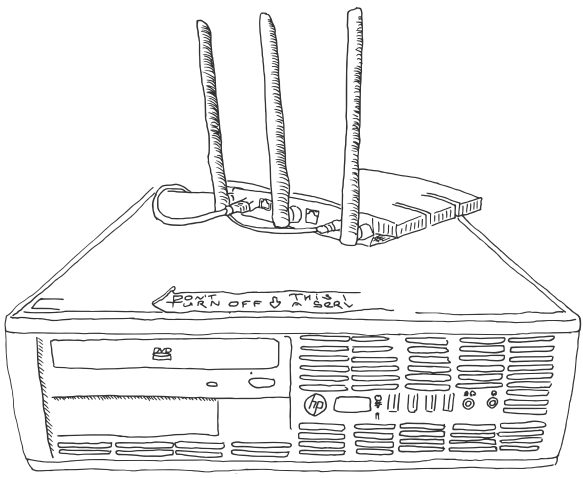
Considerations for servers

One of the main strategies of the homebrewserver.club is re-use. You re-use your existing internet connection at home and likewise you can also re-use old equipment you have as a server.

In general there are a few things you want to take in consideration when determining what hardware to use.

Power usage

All computers use power, but some use more than others. The function of the server is to be on-line and on the network 24/7 (*although that is not always necessary* (https://solar.lowtechmagazine.com/2018/09/how-to-build-a-lowtech-website.html)). That means power usage is an important consideration, not only because it will cost you money but also because it will have an environmental impact.



You can get an approximation of power usage of a machine by looking at the power rating on the power supply or AC adapter. This value is usually expressed in Watt (or W). That number represents the theoretical *maximum* draw of that system, it might use less but not more. The laptop this article is written on is rated 65W but average consumption is about 10W.

So how to calculate what that costs for a year of usage? First let's asume that this value is indeed the Watt per hour (Wh). Multiply that number by 24 to get the usage in a day.

```
10 Wh * 24 hours = 240Wh or 0.24KWh
```

Having the KWh figure will let you use that value to compare against your utilities bill.

Multiply it by the number of days in a year to get a sense of the costs on a yearly basis.

```
240 Wh * 365 days = 87600 Wh or 87.6 KWh
```

The price for electricity where I am is €0,24 per KWh so on a yearly basis that is €21 euros. On a monthly basis that is €1,75.

Considering power consumption at extreme ends.

A PC tower made for gaming that needs a 300 Watt supply can use up to 2,628 KWh of electricity a year, which would cost up to €630. That is only a little less than the average inhabitant of this planet uses *in total during a year*[1] (#fn:1). At the same time a Single Board Computer (*like this one* (https://solar.lowtechmagazine.com/2018/09/how-to-build-a-lowtech-website.html) uses a maximum of 1.3 Watt (without peripherals). That comes down to 11.3 KWh or €3,20 on a yearly basis. Now these are both rough measures and power usage will likely be lower. However, this calculation does indicate that the choice of hardware and resulting power consumption do matter.

Energy consumption vs embodied energy.

While older equipment will use more power for the same (or less) performance as newer equipment this should not be your only consideration, especially when running on renewable power sources. In fact, the overwhelmingly largest part of energy usage of digital technology comes from *manufacture and not usage* (https://solar.lowtechmagazine.com/2009/06/embodied-energy-of-digital-technology.html). Environmentally, it can make more sense to reuse an older, less energy efficient, device rather than investing in a newer energy efficient device.

Benefits and disadvantages of laptops as servers

Laptops make good homebrew servers since they are widely available, relatively powerful and energy efficient. Aside from that they have the benefit of having a screen, keyboard and battery.

If you are just starting out with running a server and are not yet so comfortable with the *command line and ssh* (https://homebrewserver.club/demystifying-ssh.html) it can be a real benefit as you can work on the

machine directly. Aside from that the laptop battery can mitigate sudden power outages, so that the laptop can shut itself down gracefully if power runs out.

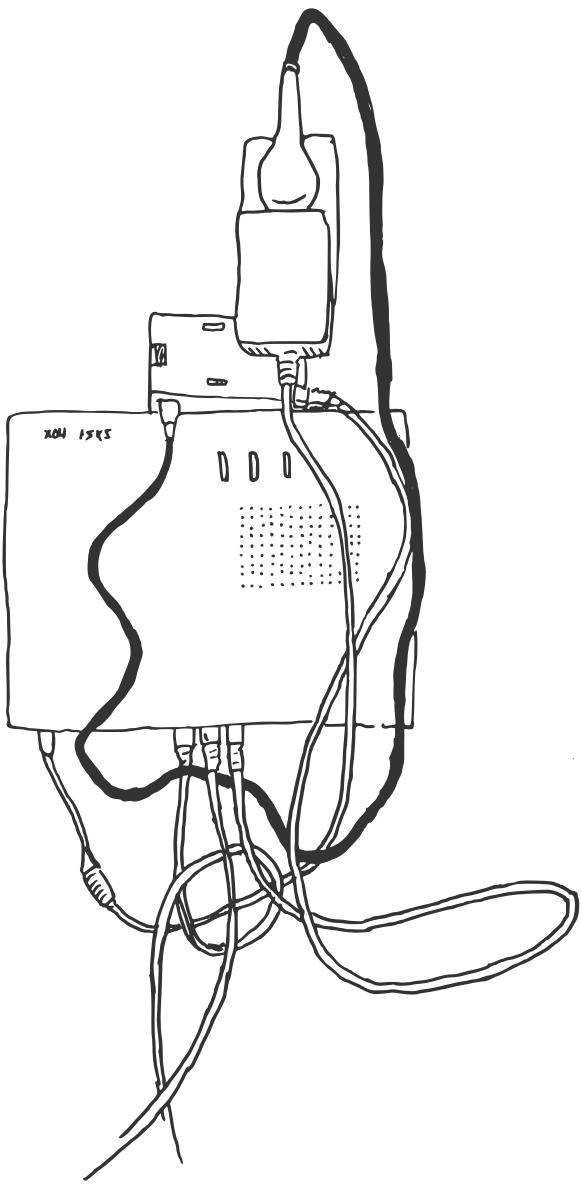
Laptops will also have the ability to hold one or more harddisks and plenty of USB ports for connecting multiple peripherals.

It is not unlikely that you have one lying around or will be able to find one in a thriftstore or on the second hand market.

Disadvantages can be size, sound, heat generation and power consumption (in particular if you don't tune the power management to turn off the screen).

Single Board Computers

Arguably the ideal homebrew server hardware is the Single Board Computer (SBC). These are often very small computers on a single chip. They are aimed at hobbyists and meant for prototyping. Popular brands are RaspberryPi, BeagleBone, Hardkernel and Olimex.



They typically use the same ARM processors as used in smart phones meaning they are extremely energy efficient, using only a few Watts of energy under full load. Most models don't use active cooling so they make no sound, which is a boon in domestic environments. They can also be very cheap, ranging from €10 to €50.

The disadvantages are that, compared to a laptop, they are relatively 'incomplete'. You will need at the very least get an SD card and 5V charger to use them. They don't have a battery meaning that a power outage will shut it off without warning, possibly corrupting the SD-card. In terms of performance they usually have less RAM available than a laptop. Their bus bandwith to read/write disks they can also be lacking.

Having said that there are some boards which are really well optimized for homebrew server usage, such as the Olimex boards. These have SATA ports (for connecting HDDs), are able to use Li-Po batteries, Gigabit Ethernet ports and decent amounts of RAM. You will also find many SBCs on the second hand market.

In general there are quite some Free Software distributions optimized for single board computers such as *Armbian* (https://armbian.com).

Virtual Private Server

Sometimes it is not possible to take the 'home' in homebrewserver.club literally because your Internet Service Provider may be blocking certain ports or types of traffic.

A possible alternative is then to consider a Virtual Private Server (VPS). These are servers, that for all practical purposes (including root access) are usable in the same way as a physical server. These VPSes are usually located in datacenters and offered by hosting companies. They are software only, hence virtual, and usually multiple VPSs share a single physical machine.

They can be found for as little as €3 a month.

The great advantage of a VPS is that you won't have port restrictions, traffic shaping and generally a fast uplink. Aside from that, a VPS means no hardware maintenance and it's not your personal IP-address.

The downsides are that you are essentially renting, storage space tends to be relatively expensive, you might not always have full control over the kernel and OS. For the

paranoid it might mean that since your entire system is running in a physical machine's RAM, it is possible to retrieve your sensitive data without you knowing.

Costs of the homebrew server

One of the things that might hold one back are the costs involved in the whole undertaking.

As an example one can take my personal setup which I have been running since 2014. At the time I paid around €75 for a new Olimex Micro (2ghz, 2 cores, 1GB ram) including peripherals, such as a high performance SD card and power supply. Later I spent another €80 for a 2TB HDD. In terms of energy usage the upper bound is at 10Wh meaning €21 yearly.

So in total

```
€75 + €80 + €105 (€21x 5 years) = €260 .
```

Per year this means €52 or about €4,30 a month maximum. So while the upfront cost can be substantial, the cost over time is really manageable. Especially if you share it with a bunch of friends!

If you can reuse old equipment the costs would be only for electricity, since you've already been paying for your ISP connection and reuse hardware.

As for a VPS, if one compares the processing power of lower-tier VPS providers one can get 1 core 2Ghz and 2GB RAM for about €3. This is equivalent to low-end SBCs but if one starts to add storage to the VPS, the price quickly ramps up.

Notes:

1. See per capita electricity consumption for the world population: *https://en.wikipedia.org/wiki/List_of_countries_by_electricity_consumption* (https://en.wikipedia.org/wiki/List_of_countries_by_electricity_consumption) ↩ (#fnref:1)

Demystifying SSH

Mon 17 June 2019
By decentral1se

Introduction

Some of the essential things that separate a server from other computers is that first they are usually not where you are and second that often come without screen and keyboard. With homebrewservers this is particularly the case when using Single Board Computers (SBC).

In order to use a server you need to connect to it over the network using the command-line interface or shell. This is usually done with a program called SSH which stands for Secure Shell.

One of the more important and foundational skills needed for experimenting and maintaining servers is understanding, using and troubleshooting SSH.

Setting up and using SSH can be a challenge at first. There are many moving parts to consider. Working with SSH means knowing something about user accounts, permissions, public key cryptography, protocols, clients, servers and agents. And yet despite so much to consider, SSH is praised as something easy to use and quite often presupposed knowledge between peers.

With this in mind, there is a need to demystify SSH. In this guide we aim to show how to setup an SSH client and server as well as discuss common issues and approaches for day to day use and troubleshooting.

Prerequisites

A basic understanding of *the command-line* (http://write.flossmanuals.net/command-line/introduction/) is required.

The SSH ecosystem is very well established. It is available on all modern GNU/Linux distributions, MacOS and Windows. You can use your home server or if you don’t have one yet you can use your own personal laptop to experiment (in this case, your laptop will play the role of both the server and client as explained later).

The commands shown in this guide were run on a *Debian Stretch* (https://en.wikipedia.org/wiki/Debian_version_history#Debian_9_.28Stretch.29) distribution but the actual tool

names should be the same on other distributions.

SSH and OpenSSH

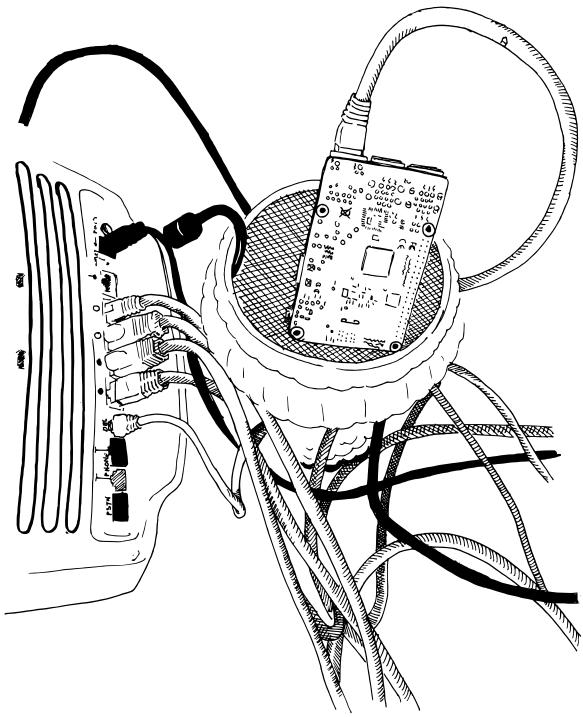
The term “SSH” can refer to a number of different but related things.

SSH stands for the *Secure Shell* (https://en.wikipedia.org/wiki/Secure_Shell) which is a protocol for describing how to provide a secure channel of communication from a client to a server. However, it more typically refers to the *OpenSSH suite of tools* (https://en.wikipedia.org/wiki/OpenSSH) which are the programs you will install and use when dealing with SSH.

The OpenSSH tools are an implementation of the SSH protocol. This means that these tools follow the behaviour described in the documents of the protocol.

Installing the SSH server and client

It is important to understand the client/server architecture of SSH. If you are remotely connecting to your home server from your laptop, then your laptop is the client and the home server is the server.



There are two packages which contain all the tools that the OpenSSH tool suite provides. The *openssh-server* (https://packages.debian.org/stretch/openssh-server) and *openssh-client* (https://packages.debian.org/stretch/openssh-client) packages.

To install the SSH server on your home server, run:

```
$ sudo apt install -y openssh-server
```

And on your client, run:

```
$ sudo apt install -y openssh-client
```

Creating a user account on the server

In order for your client to connect to the server, a user account must be created on the server. This can be done with the following command (where `homebrewer` is the new user account):

```
$ sudo useradd --create-home --shell /bin/bash
  homebrewer
```

You should then set a password for this new user:

```
$ sudo passwd homebrewer
```

This password will be used in the following step.

Logging in for the first time

You can now SSH into the server from your client. Assuming that you have a DNS entry which points to your home server (`myhomebrewserver.com`) then you can log in with:

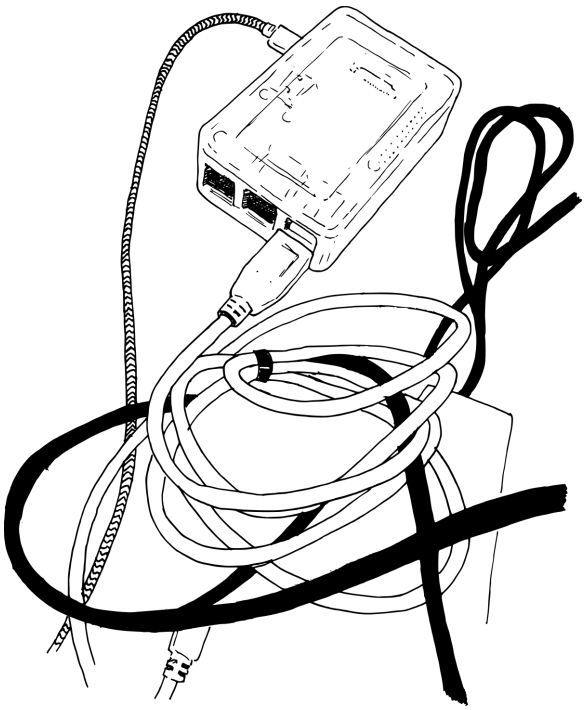
```
$ ssh homebrewer@myhomebrewserver.com
```

On first connection to a new server, you will be shown a fingerprint and asked if you would like to continue connecting. For now, choose to continue without validating the fingerprint.

You should then be asked for the password that you entered when creating the account. If you have any issues at this point, please see the *Troubleshooting SSH* (#troubleshooting-ssh) section.

Passwords, keys and security

As we have seen so far, connecting to an SSH server using password authorisation is relatively simple. However, password authorisation is typically recommended against due to *security considerations* (http://docs.hardentheworld.org/Applications/OpenSSH/#disable-password-authentication).



Security is relative and you may not be concerned with defending against a *brute-force attack* (https://en.wikipedia.org/wiki/Brute-force_attack). However, since other methods of authorisation are so commonly used and often the source of of problems when dealing with SSH connectivity, we will also cover this topic also.

The alternative to password authorisation is *public key cryptography* (https://en.wikipedia.org/wiki/Public-key_cryptography). The idea is to generate two keys, one public and the other private. The public key part can be shared publicly and will be registered with the SSH server. The private key part cannot be shared with anyone else and will be used on the client side for authentication.

For those not familiar with public key encryption, a brief and pragmatic introduction is provided by *this ssd.eff.org guide* (https://ssd.eff.org/en/module/key-concepts-encryption#1).

Validating server host keys

When logging into your server you were asked to validate a key fingerprint:

```
The authenticity of host 'myhomebrewserver
  (666.666.666.666)' ed.
  ECDSA key fingerprint is
  SHA256:lUhBmQXjkOL0zyoDNarUIbhd3RXo7X
  Are you sure you want to continue connecting (yes/
  no)?
```

When an SSH server is installed, key pairs are generated on the server. The design of SSH is such that when you make a secure connection

you should be sure that where you are connecting to is the server you expect.

On the server, you can validate that this is the fingerprint you expect:

```
$ sudo ssh-keygen -lf /etc/ssh/ssh_host_ecdsa_key
  256 SHA256:lUhBmQXjkOL0zyoDNarUIbhd3RXo7X
  root@myhomebrewserver (ECDSA)
```

These values should match the ones you were first shown. Where the `ECDSA` in the first message corresponds to the key file name in `/etc/ssh/`. The SSH server will generate key pairs for each algorithm it supports.

As you have accepted your server host key fingerprint, the public key of the server will be placed in your `$HOME/.ssh/known_hosts` file on your client where it will be remembered for future connections.

If the host key ever changes, you will see something like:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  @    WARNING: REMOTE HOST IDENTIFICATION HAS
  CHANGED!     @
  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING
  NASTY!
  Someone could be eavesdropping on you right now
  (man-in-the-middle attack)!
  It is also possible that a host key has just been
  changed.
  ...
```

This simply means that that the host key you accepted for this host has changed and you should first figure out why this is the case before making a secure connection.

Dealing with SSH keys

Generating a public and private key pair

Now that we understand the idea of host keys, we should generate client keys. On your client, run the following:

```
$ ssh-keygen -t ed25519
```

This uses the *currently preferred algorithm* (https://wiki.archlinux.org/index.php/SSH_keys#Generating_an_SSH_key_pair) for generating keys. The key generation will ask you to enter a passphrase. It is typically recommended use a strong passphrase (see *diceware* (https://www.rempe.us/diceware/#eff)) for the reason that if someone else gets your private key part, they still will still not be able to use it because they also need to know the passphrase.

You should now have two keys in your `$HOME/.ssh` folder. A `id_ed25519` (private

key part) and a `id_ed25519.pub` file (public key part).

Your public key part might look something like:

```
$ cat $HOME/.ssh/id_ed25519
  ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOTWIy+Em89QR/
  Xs6RUmr+0U0F3GVXY/UA2MXs9MoqdY myuser@myhost
```

You can then start an `ssh-agent` instance and register the private key part:

```
$ eval "$(ssh-agent -s)"
  $ ssh-add $HOME/.ssh/id_ed25519
```

The `ssh-agent` should ask for your passphrase. The agent is useful because it will store the passphrase of your key and re-use it the next time you log into the server. This can be particularly useful later on when you need to use multiple SSH keys.

You can confirm that the `ssh-agent` has loaded this key by running:

```
$ ssh-add -L
  ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOTWIy+Em89QR/
  Xs6RUmr+0U0F3GVXY/UA2MXs9MoqdY myuser@myhost
```

And finally, you can then configure your client to use this key. Create a `$HOME/.ssh/config` file and put something like the following configuration into it (remember to replace with your server details):

```
Host myhomebrewserver
    Hostname myhomebrewserver.com
    User homebrewer
    Port 22
    IdentityFile ~/.ssh/id_ed25519
```

This is useful because it allows you to only type:

```
$ ssh myhomebrewserver
```

And the correct hostname, user, port and key will be chosen. However, before connecting, you must register your public key part on the server.

Register the public key with the server

On the server, you should register the public key. This is achieved by putting the public key part of the key into the `/home/homebrewer/.ssh/authorized_keys` file.

You can do this like so (replace the public key part with the one you generated):

```
$ sudo -su homebrewser
  $ cd $HOME && mkdir .ssh
  $ echo "ssh-ed25519
  AAAAC3NzaC1lZDI1NTE5AAAAIOTWIy+Em89QR/
  Xs6RUmr+0U0F3GVXY/UA2MXs9MoqdY myuser@myhost"
  > .ssh/authorized_keys
```

You must then ensure that your SSH server is accepting keys as a method of authentication. You should edit the `/etc/ssh/sshd_config` file which is the main configuration file for the SSH server.

You need to ensure the following is present:

```
PubkeyAuthentication yes
```

After editing the file, save your changes and then validate the configuration:

```
$ sudo sshd -t
```

No output should be shown if everything is validating correctly. This is important to do before restarting the SSH server in case you lock yourself out of the server.

You can restart the SSH server with:

```
$ sudo systemctl restart sshd
```

Now it should be possible to connect from your client with:

```
$ ssh myhomebrewserver
```

If you have any issues, please see the *Troubleshooting SSH* (#troubleshooting-ssh) section.

Troubleshooting SSH

Despite our best intentions, we are often confronted with a failed login attempt:

```
$ ssh myhomebrewserver
  Permission denied (publickey).
```

This is not the most helpful message. In order to go about solving this issue, we need to focus our detective work on the client/server architecture and try to discover which side is responsible for the problem. Hopefully the following tips can help you in this process.

On the server

Here are some questions to ask yourself:

- Is your public key registered on the server in the `$HOME/.ssh/authorized_keys` folder?
- Are the `$HOME/.ssh` permissions correct? (see *here* (https://superuser.com/questions/215504/permissions-on-private-key-in-ssh-folder))
- Is the SSH server running?
- Is the `/etc/ssh/ssd_config` correct?
- What does `sudo tail -f /var/log/auth.log` say?

On the client

Here are some questions to ask yourself:

- What does `ssh -vvvvv myhomebrewserver` tell you?
- Are the `$HOME/.ssh` folder permissions correct? (see *here* (https://superuser.com/questions/215504/permissions-on-private-key-in-ssh-folder))
- Is the SSH server available at the port you expect?
- Is your `$HOME/.ssh/config` correct?
- What is registered with the local `ssh-agent` ?

Conclusions

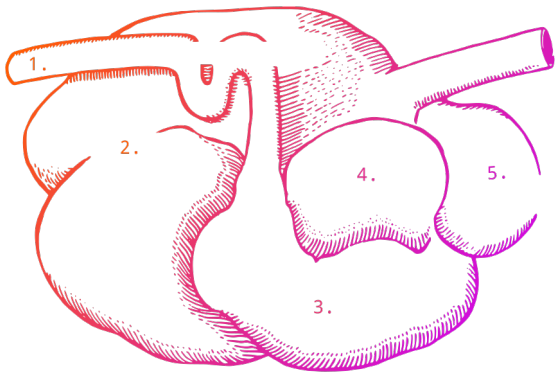
As we can see, SSH is no simple topic! There are many moving parts and a number of topics which require familiarity in order to be able to get remote access to your server.

Further Reading

- */etc/ssh/sshd_config reference* (https://man.openbsd.org/sshd_config)
- *SSH server hardening* (http://docs.hardentheworld.org/Applications/OpenSSH/)

Ruminating Relearn

Table of Contents



Last edited: 2019-09-25 18:25:10
Revisions: 6417
publication.sources/nooo.relearn.preamble (publication.sources/nooo.relearn.preamble.diff.html)

Preamble

This text appears out of order. Not only because of how we've organised it, but because we are writing about a collective learning experiment called Relearn, almost two years after it took place. The documentation of Relearn 2017 is of interest on its own, but it is also ressurected as part of the *Network of One's Own: Three Takes on Taking Care* publication made at Varia.

As part of the collection of *Networks of One's Own* editions, we want to document a network of people shaped by multiple editions of Relearn, a week-long collective learning experiment that has been taking place since 2013 mainly during the summer; a recurring

moment for us to meet new or known free software curious peers.

The following text's choral voice comes from the reactivated core of people who initiated and organised Relearn back in 2017. We are Manetta Berends, Cristina Cochior, Silvio Lorusso and Colm O'Neill, but instrumental to organising Relearn 2017 were a lot more people, including Giulia de Giovanelli, Max Franklin, Roel Roscam Abbing, and the members of the Poortgebouw residency in Rotterdam. And let us not forget of the remaining 30 participants. This text, however, is written from our point of view and as a consequence does not reflect all of the opinions and outcomes of the session itself. We don't want to see this as a shortcoming. Instead, we deliberately decided to embrace the act of documenting an event in fragments, which might be the only way to reveal what Relearn 2017 as a whole was.

The group of Relearn 2017.

In the early months of 2017, the idea to bring Relearn to the city we live(d) in sparked up. Why organise it at all? Looking back, we never once stopped to ask this question. It seemed self-evident that there was a collective desire, if not a need, for previous encounters to continue happening. We became excited at the idea of organising an edition as a way to give something back to the Relearn community, but also to bring it to Rotterdam for the first time, to connect it to people around us, and to contribute to the shape Relearn could take.

This process of documentation is less about reporting and sharing files—although we've selected a representative collection of images and files—than it is about reflecting about what was sucessful and what wasn't, particularly in regard to the new format that Relearn has adopted in 2019.

We began this documentation with one of the first texts that was collectively written: the call for tracks. It became our first point of interest

as it actually introduces a lot of Relearn notions. We began annotating this call and realised how the comments alone produced a different understanding of the text. This vernacular Relearn glossary has come to be populated with history, context and opinions, so it is our point of departure.

We spent some time reflecting on what it means to actively document Relearn 2017, thinking of ourselves as ruminants. Next, we detailed how we responded to the proposals that were sent following the call for tracks. In the publication itself, we're interlacing elements of the documents produced at Relearn such as contextual images. We've chosen to manipulate this documentation with a few tools that we've created, modified or extended (more about this in the colophon) to serve the purpose of re-entering a relearn way of thinking.

We end this text by pointing to the current experiments that are taking place in 2019 while Relearn is travelling on its curve between Rotterdam, Brussels, Paris and wherever else there's enthusiastic relearners.

Our aim with this publication is not to focus on tools and methods, as Relearn usually would, but instead to emphasise the experiments we tried in 2017 in relation to Relearn's overarching patterns and the contextual influences that arose from organising this week in Rotterdam.

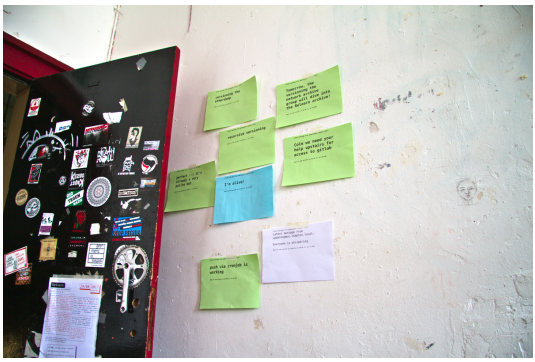
Last edited: 2019-09-25 20:29:22
Revisions: 275
publication.sources/nooo.relearn.methods-of-active-documentation (publication.sources/nooo.relearn.methods-of-active-documentation.diff.html)

Methods of active documentation

What's the active part of documenting Relearn? What does it mean to re-activate a collection of files? Does it mean to execute them again? To read them again? To simply look at them again? Does the material change while we observe it? Do we change while observing the material?

Relearn is intentionally self-reflective. Almost every year there have been attempts to re-examine and revise it. 2017 was no exception. In the Networked Archive track of that year,

the word *rumination* came up a lot. The track participants had the image of a cow's stomach in their minds, with the four different chambers that it consists of: Rumen, Reticulum, Omasum, Abomasum. The first two chambers are the major site of microbial activity that break down the material. After the cud is regurgitated, then swallowed again, until completely mixed (or activated) with saliva, it is then sent to the omasum, where it's broken down into the smallest elements possible to be pushed into the abomasum, the 'real' stomach of the cow. Further digestion takes place here. If Relearn were a ruminant, what would these chambers look like? Would the material ever reach the 'real' stomach? Is there any 'real', 'final' stomach? Do we ever finish digesting what happens during Relearn or are we stuck in a kind of gastric purgatory?



The messages on the group chat are piling up.

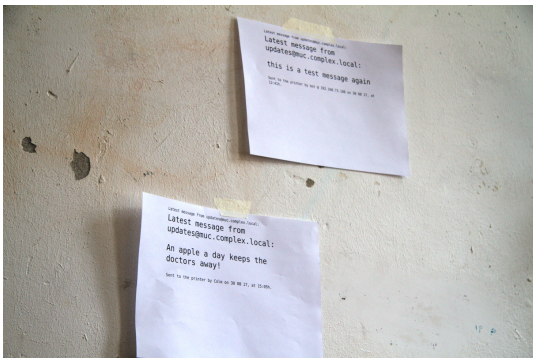
The files and traces that are left after every Relearn could be seen as extensions of a collective memory: notes that were taken in passing are now the only marks left of the thoughts that passed through, snippets of code that lay dormant in the repository, pictures of unidentified gestures which now seem alien, or group photos reminding of who was there. They sit in different hard drives: the hard drive of the server hosting the Gitlab repository and the hard drives of the participants' day-to-day computers. They are in that sense not passive at all. But you could say that they're not active either, as nobody is paying attention to them or remembers that they exist.

How to approach the Relearn 2017 *dump* of images, scripts, etherpad traces and more? We are in a kind of chicken-and-egg situation: how to introduce Relearn while taking into account what Relearn is, or better yet, how Relearn works, in the very way of doing this? Our intuition is that a lack of 'essence' might be a defining feature of Relearn. As the diversity among the various editions demonstrates, Relearn can be seen as a process more than as

a ‘thing’. A radical anti-essentialist approach to Relearn can be formulated as a question: what’s the process of describing a process?

With this question in mind, we started looking at the traces left in various hard drives, servers, and online pages… We took an investigative, forensic approach. We started tagging, looking at metadata and producing more of it, conscious that while the material is unable to speak for itself, it is also a trigger for us to recall memories stored somewhere in our wetware hard drives.

One of the consequences of activating a previous event is that we spend time together again. Reading the *call for tracks* is a way for us to pick up conversations where we left them off in 2017, thus reframing the question of learning/relearning in a collective setting. It turns it into a reroam-over-time (where *reroam* has been a recurring name for moments of meta-reflection on Relearn, during Relearn). What were the motivations, curiosities or urgencies that led us to organising and initiating Relearn in 2017? Are they still the same? How different are they now?



Relearn participants giving advice to each other via XMPP.

Activation can also be an act of mediation. An important element of Relearn is its temporality: short bursts of togetherness anticipated by longer periods of organisational work. There is no stable institution or structure behind Relearn to mediate between people or over time. When there is no one to organise, the spam emails grow in numbers, like wild weeds in an unkept garden. We hope that activating and mediating Relearn through this document format enables thinking about it in a perspective of years instead of one of weeks. By writing about a previous edition of Relearn we hope to continue the conversation around collective practices, of which rumination is one.

There are three kinds of addressees for this document. First, diving into the files lets us

take into account what permeated Relearn before 2017 and what was passed on afterwards. We hope this will reignite the subjects within our own networks, including the Relearn network.

Second, we revisited the Relearn 2017 conversations and inscribed them into a document that could speak to people with similar affinities, who may decide to organise their own experiment.

Last but not least, one of the constants of many editions is the difference in the way participants experience this time together. As it happens when someone joins a new group, for first-time relearners it can be a bit intimidating, a bit awkward, to decipher the habits of Relearn and to share their process with someone else. We hope that this document might feel like having a conversation with a previous relearner, while being outside in the sun, feet in the weeds.

Last edited: 2019-09-25 20:12:03
Revisions: 499
publication.sources/nooo.relearn.call.for.tracks

Call for Tracks[1]

***Relearn**[2] is a collective learning **experiment**[3] with as many teachers as it has **participants**[4] . It is motivated by the possibility to displace parameters of/for research, studying and learning. During the week of Relearn, **we**[5] work with a set of case-studies, observations, questions or stacks of study material that we call "**tracks**"[6] . Relearn outlines the idea of resetting thinking modes, for a diverse set of approaches that we can reconsider, that come out of the development of our cultures towards and through digital entities.*

*While Relearn is an experiment in collective learning, it grows from an interest in **Free / Libre Open Source Software culture and practices**[7] as a way to address and acknowledge the production processes and frameworks[8] involving technology and culture.*

*For the upcoming Relearn, we are looking for people that are willing to research together by initiating a track around a specific subject, methodology or issue they/you are currently working on or would like to investigate during this week. A key element for this experiment is that **tracks need not be fully determined**[9] ,*

and a concrete end-product is not required at all.

This year's Relearn is loosely connected to the ***possibilities to reconsider forms of organisation***[10] *and to create* ***co-autonomies***[11] *in regard to both* ***machinic systems***[12] *and social formations. We hope to make use of the proposed tracks as a looking glass to reconstruct notions of* ***agency***[13] *and consider forms of maintaining them. Particularly within the current political and economic vortex, we would like to discover* ***ways in which co-autonomies can be re-explored, re-shared and re-created*** *from many different angles*[14] *.*

We're allowing ourselves to prefix this word, co-autonomies, because the idea of autonomy alone doesn't emphasise the cross-seeding aspects we hope to stimulate. We mean co-autonomy as the conscious collective need for multiple autonomous systems and practices, that may also be alternative, simultaneous, off grid, self-standing or self-defining, but not operating inside a vacuum closed off to social, political and economical relations.

Feel free to get in touch with us to discuss ideas or possible forms of contribution to Relearn this year. We are looking forward to create a collective study moment in the summer of 2017 in Rotterdam!

Practical info
Relearn will take place between ***Tuesday the 29th of August and will end on Saturday the 2nd of September***[15] *. We are currently working with the community at* ***Poortgebouw***[16] *to see if this could be our Relearn location of this year. Following a tradition from previous years, we will try to arrange a* ***hosting network***[17] *in the city and find a place to stay for each participant. Depending on our subsidy requests, we aim for the lowest participation fee possible for the week, with a strong preference for* ***none***[18] *.*

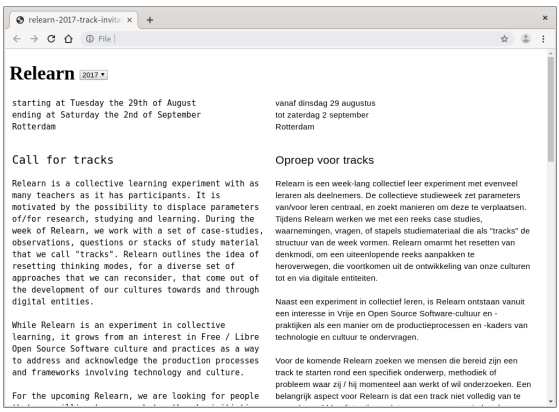
How to submit a track
The reader is invited to submit a proposal around a subject they have been investigating or would like to pursue in a collective setting. In order to propose a track, please send an email to registration@relearn.be with a short text where you informally describe the subject you would like to investigate during Relearn. Please send your proposals before the 15th of June, we will reply at the 25th of June the latest.

Resources & contact
→ email: registration@relearn.be
→ subscribe to the Relearn mailinglist
→ Relearn's gitlab repository
→ IRC channel: #relearn on Freenode
→ Photographs of previous Relearn editions can be found in Constant's gallery

---

Notes:

1. **introduction** This text is the *Call for Tracks* of Relearn 2017. The text sometimes includes some 'stiff', convoluted terminology, which perhaps does not very well represent the informal setting of Relearn, nor does it properly reflect the conviviality and enjoyment of writing this text as a group. Rereading it after a couple of years, we ask ourselves "why did we write it like this?". We recognize a tendency not to define, not to close-off, sometimes at the expense of clarity. Is vagueness just the result of collective writing or a strategic means to welcome different strands of thoughts? Maybe an open call text acquires its full meaning only in hindsight. Because of time constraints, the discussions we had were taking place while writing the text, and not before. More than the expression of a fully formed concept, the open call functioned as a window into an unraveling process. Furthermore, it was also always complicated to write a text that proposed an experiment to an undefined group of actors, all the while channeling the voice of a network with a history. Looking back at the text after two years allows us to rethink its mode of address, and the way it guided the developments of the event itself.



relearn-2017-track-invit

Relearn 2017

starting at Tuesday the 29th of August
ending at Saturday the 2nd of September
Rotterdam

vanaf dinsdag 29 augustus
tot zaterdag 2 september
Rotterdam

Call for tracks

Relearn is a collective learning experiment with as many teachers as it has participants. It is motivated by the possibility to displace parameters of/for research, studying and learning. During the week of Relearn, we work with a set of case-studies, observations, questions or stacks of study material that we call "tracks". Relearn outlines the idea of resetting thinking modes, for a diverse set of approaches that we can reconsider, that come out of the development of our cultures towards and through digital entities.

While Relearn is an experiment in collective learning, it grows from an interest in Free / Libre Open Source Software culture and practices as a way to address and acknowledge the production processes and frameworks involving technology and culture.

For the upcoming Relearn, we are looking for people

Oproep voor tracks

Relearn is een week-lang collectief leer experiment met evenveel leraren als deelnemers. De collectieve studieweek zet parameters van/voor leren centraal, en zoekt manieren om deze te verplaatsen. Tijdens Relearn werken we met een reeks case studies, waarnemingen, vragen, of stapels studiemateriaal die als "tracks" de structuur van de week vormen. Relearn omarmt het resetten van denkmodi, om een uiteenlopende reeks aanpakken te heroverwegen, die voortkomen uit de ontwikkeling van onze culturen tot en via digitale entiteiten.

Naast een experiment in collectief leren, is Relearn ontstaan vanuit een interesse in Vrije en Open Source Software-cultuur en -praktijken als een manier om de productieprocessen en -kaders van technologie en cultuur te ondervragen.

Voor de komende Relearn zoeken we mensen die bereid zijn een track te starten rond een specifiek onderwerp, methodiek of probleem waar zij / hij momenteel aan werkt of wil onderzoeken. Een belangrijk aspect voor Relearn is dat een track niet volledig van te

2. **relearn** Relearn is an experiment in self-organised education that started as a summer school in 2013, originally organised

by the Open Source Publishing group in Brussels, thereafter taking care of by Constant again in Brussels and later in 2016 organised in Calafou. Relearn takes place each time someone finds the resources and collaborators to make it happen. Since 2013 and up to 2017, there have been week-long editions each year, each time organised by someone different. The work we, the annotators here present, did was to facilitate the organisation of Relearn Rotterdam in 2017. To start the process, to make sure the mailing list was informed of the newest developments, that no email was left unanswered, to get in contact with catering groups, to find a space and make sure it was equipped to welcome Relearners. Further than that, the organisation was shared with participants and track initiators alike in regards to the content and caring for the space. This is also why we may refer to Relearn as an action by using the verb relearn, each edition is another opportunity to relearn what Relearn is. We've also been referring to it by saying: Relearn editions, Relearn issues, Relearn iterations, Relearn years, Relearn instances, Relearn sessions; often a mix of nouns and rarely a standalone.

3. **experiment** There are several horizontal educational experiments out there, various critical summer schools or self-organised festivals. Relearn 2017 is influenced by several of these experiments, such as the Anti-University festival in England, Scuola Open Source in Italy or the traveling Parallel School, but it tries to apply what it knows to the local conditions in which it operates. Relearn is considered an experiment because it understands education as an experimental process. It is perhaps not a coincidence that many of the attendants have a background in teaching. Some aspects of the ethos which have guided the relearning experiments until now are:

   - a decision to move away from the teacher -

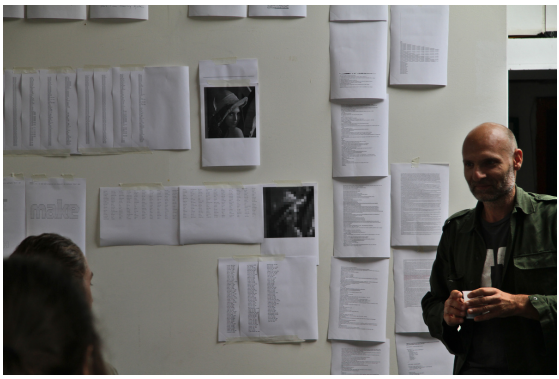
student model and replace those terms with participant and, in some editions, track initiators
- an attitude towards organisational tasks, where we share the day-to-day roles and care/use of the spaces
- an emphasis on process over outcome. This may not be as important in the later iterations of Relearn but in the earlier years, it was important to make clear that Relearn did not have to produce any tactile objects. We originally felt that using the words 'workshop' or 'summer school' meant (text/book/object) production sprint (the hackathon model). This perspective was questioned because it does not allow time for researching and probing (production) processes, which for us became the focus.

Questions that drive Relearn as an experiment:
- How does the group steer this learning situation collectively ?
- What does re-learning depend on in terms of group dynamics ?

4. **participant-as-teacher** One of the foundational aims of Relearn is to erase the duality of learners and teachers and create a space where pedagogy is exercised as a group, where hierarchies are frequently restructured.

5. **we** "We" is intended here both as the whole Relearn 2017 group, and as a smaller group that organised Relearn 2017. Finally, there is the "we" that is writing this documentation. "We" is also used as a collective noun, meant to be all inclusive and inviting to attendants old and new. While that current "we" wrote the call and invitation, that group was in no way static. We mean to invite people in by calling ourselves with a pronoun that includes the reader, however it is never meant to be understood as a monolithic voice speaking for the entirety of the Relearn network.

6. **tracks** Whatever might be going on during Relearn, we tend to focus on processes over outcomes. However, research thematics are important to provide entry points into collective research. Thematic tracks (usually simply called tracks) are subject proposals which are put forwards during the week in different ways, some more technical, some more discursive, some more exploratory. → *Read further*



7. **F/LOSS** From the beginning of the Relearn editions, F/LOSS (*Free/Libre Open Source Software*) has been an underlying principle of the sessions. Inspired by the F/LOSS community, we rely on free software to carry out our experiments and to set up the infrastructure that makes self-organisation possible. In the same spirit, we release and make public the findings, discussions and code that emerged during the week and try to encourage multiplication through the distribution of materials.
8. **software-as-critique** Relearners see software not only as a tool or a medium, but as a large web of organisations, protocols, people and machinic agents. In this sense, software is more an environment than a thing. From this perspective, using and inhabiting software means taking a stance, often critical, sometimes radical, towards default decisions.
9. **programme** Relearn started with the recurring *Install Yourself Into Relearn* session, where introductory tables to *XMPP*, *Etherpad*, *Local Network* and *Collective Practices* were set up. Participants could join a different table each round to find out about the ways in which we were using these practices. This was followed by another table carousel, where the four tracks were introduced: *Ransom Headlessnesss in the Design of Language* was looking at the interplay between design and (programming) languages; *Versioning the Networked Archive* was exploring the link between networked communities and archives starting from practical examples; *Not-So-Utopian Open-Source Pedagogies* was combining a critical examination of utopian conceptions in open source software with constructive methods for the organisation of groups; *Collective Care Transmission Forms* worked with forms of networked collectivity that emerge around certain health situations. The tracks are

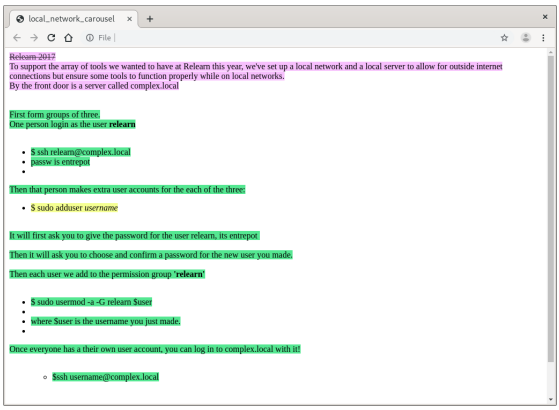
usually a starting point for the week to unfold. During the time we spend together, other paths are taken. For example, alongside these groups, other types of gatherings took place as well, from impromptu skill-sharing sessions during which chargers were fixed, to one-day satellite workshops, evenings of short presentations of the participants' work, to eetcafe dinners organised by the Poortgebouw. Everyone is encouraged to use their time at Relearn as they see fit.

10. **varia** Parallel to the conversations about organising and hosting an edition of Relearn in Rotterdam, some of us were actively involved in finding a space in the city for Varia, soon to become a *center for collective approaches to everyday technology*, which was founded in Charlois a few months later (October 2017). Varia focuses on the combination of self-organizing in the digital and non-digital . Currently, *Varia* is exploring how self-organized spaces can gain agency over their digital infastructure (amongst other things). At the time of Relearn, this was something that we wanted to discuss and also to put into practice.

11. **thinking-autonomy** The term 'co-autonomy' emerged from the unease we felt to talk about autonomy-as-independence and autonomy-as-full-agency when our own conditions were entangled with those of others, exemplified by the fact that we were guests of Poortgebouw, we had to apply for a small local fund to partly cover costs of catering and the electronics needed to have a functioning local infrastructure, etc. All of these elements made us shift focus towards relational agencies: observing and acknowledging the ties that create the fabric of our self-determined associations. The word autonomy, especially in relationship to the digital, hasn't left our minds, since it brings with itself a baggage of emancipatory traditions. Under the umbrella of Varia or Constant, some of us are still working on it and collecting other terms and expressions that can help capture the entanglement of being related to something, while preserving one's own position.

12. **local-networks** Relearn is also a moment to experiment with the communication tools that we use. In 2013, git, a software for tracking changes in code during development, was a focus, whereas in 2014, the group of Relearners used the IRC protocol to communicate with one another.

In 2015, a local network was set up. In 2016, unfortunately none of the members of this text's writing group was able to go to Calafou. The following year, we continued using Etherpad software for collaborative note taking, next to which we set up XMPP multi-user chatrooms and used XMPP bots to send messages to the whole group or to print content that would be displayed on a physical wall in Poortgebouw. Each edition an attempt is made to rethink the infrastructure on which we rely on, not to fall into defaults, or if we do, to fall consciously.



13. **agency** When discussing this text in 2019, we realised that "agency" does not have a direct translation in many of our native languages. Perhaps the closest to the meaning to the English version is "zelfzeggenschap", "agentschap", "autogestione" or "autodeterminare". Furthermore, depending on the field, agency can mean "the capacity of an actor to act in a given environment" (philosophy), or "the capacity of individuals to act independently and to make their own free choices" (sociology). Whereas the emphasis is on the capacity to act, doing so independently might be the very limit to it.

14. **collective-practices** Digital processes allow us to intervene into a practice in many ways and at different moments. A case in point is this very text: it is being written collectively, in real time. Collective writing reflects Relearn's urge to preserve a multiplicity of voices. Live (online) and collaborative text editors are often a tool of choice as they let us think, read and write together, and keep trace of oral conversations. This collective practice is not easy, it requires some daring, some testing and the vulnerability to show others how one's writing is produced. Collaborative working methods are a point of interest for Relearn because they let us

rethink digital practices that are made for ‘the user’. Practicing together brings context to the foreground.

15. **summer-school** The ‘summer school’ attribute of Relearn has been under discussion in multiple editions. As a result, the 2019 iteration of Relearn started from breaking up the week of activities into concise moments of 2-3 days that could be distributed throughout a longer period of time and across different geographies. The goal was to make Relearn accessible for participants who cannot afford to take a whole week off, but this was also partly an attempt to move away from the summer school format, which has been coopted as another form of commodifying education and device for professional positioning. As Francisco Laranjo puts it, *“design summer schools are overpriced tourism gatherings with good weather and a nice view […] Working on a laptop on Indesign overlooking a lake is not a spiritual retreat to charge creative batteries, it’s just privileged illusion”* (from *A Brief History of the Design Summer School*). Most of the people who make up the Relearn network are based in Western Europe, where summer schools are often inaccessible because of their high fees and in the worst cases become extensions of institutionalised pedagogy lacking self reflection. Relearn emphasises the leisurely, the conviviality and the slow-paced tempo that is generally associated with summer. It puts forward the critical engagement with the social, political and technical structures that make it possible (sometimes to a narcissistic extreme), as an attempt to reimagine what education could mean at a small scale. In 2017, we referred to it as a ‘collective learning experiment’ in order to open up to new formats.

16. **poortgebouw** The Poortgebouw is a Dutch monument built in 1879. After being in a limbo of non-use, it was squatted by a group in 1980 as a protest against an increasing housing shortage. Despite the relatively short duration of this period (only two years), the group managed to organise legally as an association in 1984, and since then has been hosting cultural activities and a co-living community of around 30 people. The space self-organises food-on-donation eetcafes, clothesswap days, alternative music nights, info evenings by activist organisations, among many different do-it-yourself projects and events. The building

particularly resonated with the topic of co-autonomy, as can be seen in the *archival project* that Giulia de Giovanelli & Max Franklin, 2017 co-organisers, coordinated together with others in and around Poortgebouw; it was taken as a case-study for the *Versioning the Networked Archive* track.

17. **hosting-network** The hosting network is one of the tactics that re-appeared in all of the editions: volunteer local participants are matched with visiting participants in order to ensure a place for them to stay during the week. On the one hand, this is a practical solution that helps us keep costs down and on the other, it is a more welcoming introduction to the city and to the group.
18. **finances** The process of organising Relearn in 2017 started out as a voluntary task. We applied for a local fund to cover the catering expenses during the week (breakfasts, lunches and two dinners) and some equipment needed for the local network and the care taking of the Poortgebouw. Being dependent on a grant proposal meant that we didn't know until very late if we would receive the funding - when we did, all of it was spent on lunches. We also asked for a €50 participation fee to cover the most basic needs of the space. After the week was over, the sum that was left was split among three of five continuous organizers, who received a fee of €300 each. The remaining two waived the payment for having a stable income.

Last edited: 2019-09-25 19:56:23
Revisions: 11405
publication.sources/nooo.relearn.call.for.proposals-proposal-footnote (publication.sources/nooo.relearn.call.for.proposals-proposal-footnote.diff.html)

Tracks

Whatever might be going on during Relearn, we tend to focus on processes over outcomes[1] (#fn1). However, research themes are important to provide entry points into collective research. Thematic tracks (usually simply called tracks) are subject proposals which are put forward during the week in different ways, some more technical, some more discursive, some more exploratory.

Tracks is a deliberately loose word used for a subject, an approach or a set of material

allowing participants to influence and orient the direction of collective work. A track is the word that has been used to ensure that everyone involved realises that diversions are encouraged and that one can decide to take alternative routes. Tracks in that sense is meant to resonate with tracks or pathways in a forest. Some of those tracks are more close to being roads and even appear on the map, but others are more serendipitously created as people start to follow the same path more often, pushing the leaves and branches on the floor to the side. They start to leave a trace and create a track for others to follow. Others are less visible, but marked with signages that indicate direction. Some of these tracks carry histories with them, while others are yet to be discovered, thus defined. As such, the strive for this kind of learning can be interpreted as a walk we undertake together, that is short-lived, but whose discoveries are potentially continued in different places, traversing into the day-to-day lives of the walkers.

We start suggesting this metaphorical use of the term track while writing about the 2017 edition. The term has been used since the first edition, but has never been explicitly described. It became a transversal element of Relearn without a fixed understanding of what it could be.

Tracks are prepared in advance. Most Relearn iterations involved an open call for tracks. In some cases these tracks were proposed by people coordinating Relearn itself, but more often than not, track submissions came from people beginning research into a topic and wanting to bring the topic to a wider group of people to develop the subject further. Track organisers are typically asked to bring enough material to make their questions understood, but no more than that. By using the word track instead of workshop, we try to call for proposals of subjects, attitudes or techniques, that can be collectively explored. During Relearn some tracks (borrowing from versioning software vocabulary) have been forked, signifying a split, a deliberate change in direction, a disambiguation or a deeper exploration of a sub-topic.

This openness for input/take-overs from others directly relates to the wish of displacing roles of the teacher and learner (see notes expanding the *call for tracks* text). A secondary, yet key ongoing experiment, is the idea of track travelling: encouraging participants to take part in tracks and then visit other tracks. It emphasises not only the

lack of focus on production, but other aspects too: active documentation, sharing of topics and giving participants a chance to benefit from all the subjects being researched at Relearn. An expression of this—and something that has been done deliberately—is when a track initiator leaves the track they originally started, to join a different one.

Various moments of sharing spring up throughout the length of Relearn. They assist in communicating to the rest of the groups what each track traveler is busy with and become an invitation for external questions. These moments help track travelers move to other focuses, or take ideas and processes from other groups. It is notable that these sharing moments are not scheduled and do not happen regularly. They also tend to happen organically, often the sharing of updates happens throughout meals and breaks.

Reroam discussion on Friday morning.

Relearn 2017 hosted a new type of experiment related to tracks; to put track proposals in conversation with one another and see what type of situation it would yield. As a response to the open call, we received thirteen proposals, of which five were eventually unable to join the week. The remaining eight were at different points in their development. In an attempt not to exclude any tracks, we discussed each proposal at length and noticed some common interests or similitudes in approaches. We then reached out to initiators and asked for their permission—and interest—to being put into dialogue with other initiators.

The motivation to do this was manifold: to provide the space to all those who had submitted a research initiative to find collaborators and to follow curiosities alongside each other, intersecting or multiplying; to make the track preparation process less solitary; to open up some of the research strands in order for more people to be able to join; to already form connections between participants; or serendipitous

research encounters. Indirectly, we were also influenced by the notion that we had put forward in the call: co-autonomies, but this was more of a reaction to the topics that were proposed, not a planned move to enact the term.

We aimed to create an open conversational space where the experiences of those involved could inform the directions of the week. To break open the teacher-student duality by learning about your track partner's practice. Instead of a merge, the idea of having research questions be shaped by each other was a means to open up the organisational work and destabilise norms of knowledge production. This involved some risk taking: different ideas on how to lead the tracks had to be negotiated and the combination could have possibly diluted the focus of the original proposals. Time was also an important constraint; to be able to devote a week to re-learning is already a significant luxury, but to find moments to plan and plot outside of this frame was not as easy for everyone. Collaboration requires togetherness and agendas mirroring each other, and it is rarely free of frictions. In this sense, combining tracks came with more work for everyone involved in the organisation of the week.

We went one step further in this attempt: once we had received approval from all parties in the new partnerships, we opened an etherpad document[2] (#fn2) for each collaboration. In each pad, we included the proposals from each party and proceeded to include our observations and explanations of why we felt it appropriate to join the topics. Our proposal was for the track initiators to use that space to collaborate synchronously or asynchronously on the cross-sections of their subjects.

Four tracks resulted:

- *Ransom Headlessnesss in the Design of Language* looked at the interplay between design and programming languages.
- *Versioning the Networked Archive* explored the link between networked communities and archives starting from practical examples such as the Poortgebouw archive, or the Github archive.
- *Not-So-Utopian Open-Source Pedagogies* combined a critical examination of utopian conceptions in open source software with constructive methods for the organisation of groups.

- *Collective Care Transmission Forms* dealt with healing in cyberspace from a critical feminist materialist perspective, communal care in digital environments and the transferral of aural thoughts and expression.

The outcomes were less ambivalent, we felt: bridging the tracks meant that the organizers had to gain an in-depth understanding of their intentions through the collaboration and to have an oversight on the connections as well as on the digressions of the two topics.

A later, further risk was taken from these resulting tracks (and their pads): the tracks pads, containing track proposals and new information from the established contacts, were made public and used to invite participants to contribute in the secondary call for participation. Those considering to respond to the open call for participation would be able to see the topics in their latest version, and even, if they so wished, to already propose corresponding questions or contribute material. The ideal scenario we were trying to create here was a collective research environment that could traverse the fixed time of collectivity by extending the discussion to before and after.

An observation from the specific track attempts from 2017 is that it was easier to create discussion points between individuals than among groups. Some track proposals were submitted by groups of people. In those instances, the collaborative work done by the initial group was more complex to bridge into a new, bigger group, no doubt due to the already complicated work done in multiple people proposing a single track. Frictions also arose when discussing their common grounds. It is also notable that we were not always able to consider these frictions as points of interest, opportunities to observe collaborative practices. These, as well as the extra quantity of work and coordination it implied, makes us wonder if these experiments worked. Looking back, we would procede with more caution and be more hesitant about pairing up individuals with / and / or groups of multiples without providing a dedicated space (beyond an online editor) for that. Dedicating a full week in the summer to Relearn is complicated enough as it is, investing extra time in preparing a track together can be too exhausting.

When Relearn 2017 ended, no time beyond the practical disassembly of the spaces we used was planned or budgeted. We, as a group of

organisers debriefed lightly, took time to publish (problematically ! cfr. *a-little-too-active-documentation* notes) the etherdumps and information from the local server on shared text & code versioning platforms, but not much beyond that. Relearn 2017 happened on the first week of September, so the traditional September rush ensued.

The Rotterdam edition of Relearn also marked five consecutive years of the summer research project, after which there was a year of silence. Meanwhile, Varia—as an idea, as a group and later, as a space—, was in full development, and various previous Relearners were meeting in multiple different contexts, Barcelona, 35c3, Computer Grrrls, Constant worksessions, etc.

Relearn 2019 was re-initiated with experiences from previous years in mind. A desire not to continue the format, but to address the time and energy requirements to organise an iteration. Relearn 2019 was re-initiated as a curve of short two to three day sessions, spanning several cities, over a longer period of time.

Notes:

1. The goal of Relearn is not to be collectively productive, instead it is left up to each participant to determine the rhythms and patterns in which they want to operate. The session attempts to construct a playful space where if something like work happens, it is for the sake of curiosity and does not require the participants to perform acts of closure, finitude or functionality. The motivation for this can be found in Relearn's history: one of the reasons for starting it in 2013 was a reaction to those *work sprints* that go under the name of *workshops*. Open Source Publishing (OSP – mentioned here as the initiators of Relearn 2013) was and is often invited to give workshops in various art and design schools, but one reoccurring condition of these *workshops* is the desire to produce objects, often in print form: publications, posters or collections. This partially comes from the need to be able to grade and score students' work, but it has become clear that working to these end goals hinders the focus on research questions. Later, this became an idea to address during Relearn. Clearly stating that

making finalised things is not a requirement at Relearn lets participants spend their time exactly where they desire, not constrained by conditions imposed by production.

2. Etherpad is web-based software that allows for multi-handed collaborative writing and editing of texts in real time. The software is heavly adopted by Relearn and many of its supporting organisations, Varia, Constant, OSP. The consistent use of etherpad stimulated two Relearn transversal projects: *Ethertoff* (https://gitlab.constantvzw.org/osp/tools.ethertoff), developed for and used during Relearn 2013) to turn etherpad into a writing interface for a wiki-type website; and *Etherdump* (https://gitlab.constantvzw.org/aa/etherdump) a converting and archiving tool of all pads in an etherpad installation.

Last edited: 2019-09-25 20:42:39
Revisions: 1601
publication.sources/nooo.relearn.colophon
(publication.sources/nooo.relearn.colophon.diff.html)

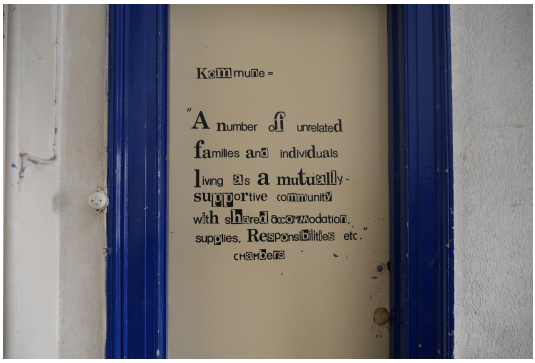
Colophon

Written and designed by Colm O'Neill, Silvio Lorusso, Cristina Cochior and Manetta Berends — June, July, September 2019 — Rotterdam (NL), Carlow (IE).

Relearn 2017 participants included Roel Roscam Abbing, Max Franklin, Giulia de Giovanelli, Amelie Dumont, Annet Dekker, Julie Boschat Thorez, Ruben van de Ven, Gijs de Heij, Thomas Buxó, Quentin Jumelin, Quentin Astié, Hans Lammerant, Cristina Cochior, Manetta Berends, Silvio Lorusso, Colm O'Neill, Marcel Goethals, Miglena Minkova, Kym Ward, Karisa Senavitis (remote), Amy Pickles, Vasiliki Zioga, Desiree Kerklaan, Anita Burato, Dick Reckard, Émile Greis, Adelina Tsagkari, Anne Laforet, Javier Lloret, Dušan Barok, Joca van der Horst, Claudia Roeck, Jim Wraith, Léna Robin, Natacha Roussel, Máté Pacsika, Jeremy Olson.

General Relearn documents can be found at:
http://relearn.be/2017 (http://relearn.be/2017)
https://gitlab.com/relearn/ (https://gitlab.com/relearn/)
https://gitlab.constantvzw.org/relearn (https://gitlab.constantvzw.org/relearn)
Relearn 2017 Etherdump (static archive of the note taking / writing pads): *https://gitlab.com/relearn/relearn2017/tree/master/*

complex.local/home/etherpad/etherdump (https://gitlab.com/relearn/relearn2017/tree/master/complex.local/home/etherpad/etherdump)

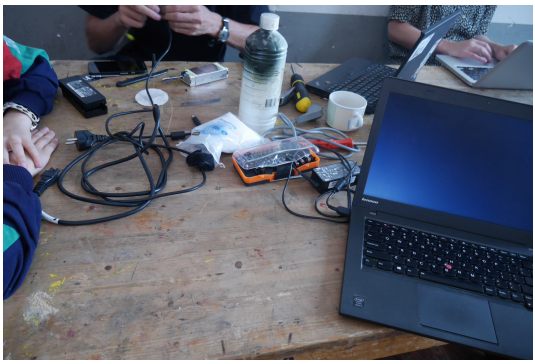


Quote on one of the doors in Poortgebouw.

Tools

To aid this publication, one tool was created and two existing tools got extended: Etherdump and Distribusi.

- Distribusi is used to integrate the photographs of Relearn 2017 into the publication. A command line photograph EXIF comment inserter was built to annotate photo files inside the files themselves. The tool also comes with a reader to go through the annotations in the terminal. *https://git.vvvvvvaria.org/varia/EXIF-image-commenter* (https://git.vvvvvvaria.org/varia/EXIF-image-commenter)
- An EXIF extension has been added to Distribusi that extracts the EXIF 'Comment' metadata before building web indexes based on the contents of a directory. This function is activated by the flag '-c' or '-captions'. The flag can be used in combination with '-nf' or '–no-filenames' which tells Distribusi not to include image filenames. *https://git.vvvvvvaria.org/varia/distribusi* (https://git.vvvvvvaria.org/varia/distribusi)
- New pad-to-publication features were added to the Etherdump software (on the publish-vs-nopublish branch) and were used to make this publication. They allow for *Etherdumping* a selection of pads that include the `__RELEARN__` magic word inside them. We used this occassion to fork etherdump into *Etherpump*, with the idea to stimulate further experimentation with pad-publishing within. You can find the fork and read more about it here: *https://git.vvvvvvaria.org/varia/etherpump* (https://git.vvvvvvaria.org/varia/etherpump).



Fixing chargers and OS installation issues during informal & impromptu skill-sharing sessions.

Last update 2019-09-25 21:15:54.

A morse code message for you

The USB stick can transmit its own name. Have some USB blinking fun in the wee hours of the day. Of course the message is expandable as you see fit.

You can find it in the `assets` folder and run it with the following command: `python blinking-bonus.py`.

Happy blinking!

```
#!/usr/bin/python

import os

message = "-. . - .-- --- .-. -.- ...
--- ..-. --- -. . ... --- -. ."

for char in message:

  if char == "-":
    print("long")
    cmd = 'dd if=/dev/urandom
of=buffer bs=1M count=1 2> /dev/null;
sleep 5'
    os.system(cmd)

  if char == '.':

    print("short")
    cmd = 'dd if=/dev/urandom
of=buffer bs=1M count=1 2> /dev/null;
sleep 3'
    os.system(cmd)

  if char == ' ':

    print("space")
    cmd = 'dd if=/dev/urandom
of=buffer bs=1M count=1 2> /dev/null;
sleep 10'
    os.system(cmd)
```