

- [Networks Of Ones Own](#)
- [Episode 1: Etherbox](#)
 - [Inside the box](#)
 - [Download](#)
 - [Unpacking the boxes](#)
 - [make & makefiles](#)
 - [A conversation about releasing Etherbox](#)
 - [Why is it important to trace the history of "Etherbox"?](#)
 - [So, when did Etherbox become "Etherbox"?](#)
 - [What practices does "Etherbox" generate or connect to?](#)
 - [What initiatives or events are missing from the timeline?](#)
 - [Is Etherbox like a Piratebox?](#)
 - [What about Raspberry Pi?](#)
 - [Who created Etherbox?](#)
 - [Etherbox experiences](#)
 - [Generate a pdf with the collective notes and pictures](#)
 - [The Questionnaire](#)
 - [Backroom communication](#)
 - [Iterative Transcription](#)
 - [2-pass Collaborative editing](#)
 - [The Legible system](#)
 - [Guide to using your Etherbox](#)
 - [Make sure you are connected to the network of the Etherbox](#)
 - [Surf to etherbox.local](#)
 - [Find out the IP address of the box](#)
 - [Create a writing pad and start working!](#)
 - [Access to existing pads](#)
 - [ssh to the box](#)
 - [Make it easier to login, with an ssh key](#)
 - [Shutdown the Etherbox](#)
 - [Reboot the Etherbox](#)
 - [Restart Etherpad \(this can help when people start having connection problems to the pads\)](#)
 - [Move to the home folder](#)
 - [Make dump \(make static copies of the contents of the etherpads\)](#)
 - [Compile the contents of several pads with an .index pad](#)
 - [sftp to the box](#)
 - [Delete a pad](#)
 - [Archive the pads](#)
 - [Changing the frequency of the cron](#)
 - [Resetting a box for future use \(delete all pads, etc\)](#)
 - [Placing Etherbox in your network](#)
 - [Booklet printing with a makefile](#)
- [Appendix](#)
 - [Etherbox timeline](#)
 - [Active Archives](#)
 - [Relearn 2013](#)
 - [Ethertoff](#)
 - [Bibliotecha](#)
 - [Verbindingen/Jonctions 14: 'Are You Being Served?'](#)
 - [2084: Rise of the botnet](#)
 - [Cqrelations](#)

- [Close-reading Kenneth Goldsmith's Uncreative Writing](#)
 - [Relearn 2015](#)
 - [Promiscuous Pipelines](#)
 - [Objects in Common](#)
 - [Zine Camp 2016](#)
 - [Machine Research](#)
 - [What's the Matter with Cooperation](#)
 - [Etherbox at DiVersions worksession](#)
 - [Situated Publishing: Writing with and for machines](#)
 - [Etherbox at the Technogalactic Software Observatory](#)
 - [Etherbox at the Ageing Companions worksession](#)
 - [Etherbox code repository with previous editions](#)
- [Etherpad timeline](#)
- [Feminist Server Manifestos](#)
 - [A feminist server...](#)
 - [A FEMINIST NET/WORK HOW-TO](#)
- [Book as a network](#)
- [Social Shells & Mesh Cookbooks](#)
 - [Web 2.0](#)
 - [Social Shell & Mesh Cookbooks](#)
- [Interview with Etherbox](#)
- [Build your own Etherbox](#)
 - [Context](#)
 - [Copy the starting image](#)
 - [Enable SSH](#)
 - [Setup hotspot](#)
 - [Bring the rest of the software up to date](#)
 - [Change hostname from raspberrypi to etherbox](#)
 - [Assign Static Ip Adress to the Pi](#)
 - [Setup apache to serve the root with custom header + readme's](#)
 - [Better permissions with facl](#)
 - [Install etherpad](#)
 - [etherdump](#)
 - [Use pandoc to convert pads to HTML](#)
 - [Create an image gallery with imagemagick](#)
 - [cron.sh](#)
 - [makefile](#)
 - [Enable the cron job](#)
 - [Create a reverse proxy for the etherpad](#)
- [Makefile](#)
- [pdftk_utils.py](#)
- [Colophon](#)
- [Notes](#)

Networks Of Ones Own

Networks Of One's Own is a para-nodal¹ periodic publication that is itself collectively written in a network². Each of the episodes is thought of as the 'release' of a specific software stack, contextualised in its specific practice. The series aims to document a set of tools, experiences, ways of working that are diverse in terms of their temporality, granularity and persistence.

The title *Networks of One's Own* refers to Virginia Woolf's classic essay *A Room of One's Own*³ which makes the case for a woman needing a space to herself to write. It was published in the historical context of Victorian England where women were expected to occupy themselves with domestic and marital duties rather than the autonomous practice of writing. Woolf's claim for a *Room Of Her Own* is complicated by the fact that her autonomy depended on her ability to hire staff that she could outsource her household chores to. *Networks Of Ones Own* takes this text as an inspiration to rethink digital intimacy, dependencies and relations in networked practices. It wants to re-imagine how technical work and content work grow together, and radically questions the way tools and practices are shaping collaborative content and vice versa.

The idea of a publication series arose from conversations with Aymeric Mansoux during the development of the Experimental Publishing Master at the Willem De Kooning academy in Rotterdam in the fall of 2016. With the advent of inexpensive so-called "single board computers" such as the popular Raspberry Pi project, it has become commonplace to distribute "disk images" that can be loaded onto an SD card which then contain a fully working software stack. The idea of a dual-form publication (both textual and executable) suggested the possibility where unique software platforms could themselves be published in tandem with content (collectively) written using the platform itself.

Networks Of One's Own is set up as a series of such dual-form publications that reflect upon and experiment with collaborative on-line practices in a time when commercial interests dominate digital working conditions. It appears at a moment when under the guise of data protection (GDPR), self-hosted, un-supervised network practices are under threat of being reigned in as potential liability. "Cloud services" offered via *Internet* access are typically provided by multinational corporations and it is easy to forget that the seamless experience historically originates in an *inter-network* linking together different autonomous and disconnected networks of varying scales: institutional, personal, educational and military.

The series *Networks Of One's Own* is taken care of by related but independent collectives. For each of the episodes, we propose different experimental tools for situated writing and publishing. In this way, the series allows for showing multiple perspectives, to generate multivoiced forms of documentation. It makes the editorial process of each episode a test case for the same technologies that it reflects upon. Like the old adage of the PERL community ("There is More Than One Way To Do It")⁴, this project is about the heterogeneity of tools and of practices, ultimately foregrounding the inherent sociality of software.

<http://networksofonesown.constantvzw.org>

Episode 1: Etherbox

Etherbox is a constellation of collective tools and practices that developed in and around Constant⁵. *Etherbox* responds to practical issues when collaborating in physical spaces with digital tools. In parallel, it became a platform to reflect on network technologies, on how to document artistic processes and on processes of collaboration.

On a practical level, etherbox was a response to several frustrations: the fragility of online resources, the fact that the labour necessary to maintain resources online after a collective session was often overlooked or simply assumed to continue indefinitely, the limitations of Internet bandwidth when working with many people using etherpad-lite. The goal of the project was thus to present an infrastructure that is both visible and situated. Since its early beginnings in 2013, many groups have enjoyed the pleasure of writing and documenting together, questioning infrastructure set-ups or simply sharing files effortlessly. Through these multiple experiences, *Etherbox* has grown up to facilitate discussions and collaborative practices independent from Constant collaborators. Now it is ready to transmutate into other networked imaginations.

As part of its programme, Constant regularly organises situations in which artists, activists, programmers, academics, designers and other researchers collaborate, exchange and reflect. Participants are invited to appropriate physical spaces for the time of the meeting: they share tables, couches, a library, the kitchen and its utensils, the basement or the closet. *Etherbox* extends this hospitality into digital space. It is the installation of a temporary local platform that foregrounds the sociality of DIWO⁶ infrastructures over the services of often commercially owned spaces in global networks.

While existing local network or server tools make users choose between local or global networking, people using *Etherbox* can still be connected to the internet. Sharing documents over *Etherbox* means that they don't travel via the far-away data-center servers of corporations, but remain in a network-setup that is visibly and physically accessible to everyone. Likewise the *Terms of Service* are negotiated locally, between people often in the same room. They become specific conditions for the use and potential afterlives of the collected material, rather than being thrown together by company lawyers on another continent.

This publication includes [manuals for basic usage](#), [a conversation that tries to locate this Etherbox release within interconnected histories](#), [recipes distilled from diverse experiences](#) and a generous collection of [appendices](#). The publication temporarily wraps up the ever-evolving development of the toolset, so that it can be installed by any software-curious group that can get their hands on a Raspberry Pi and knows where to find an ethernet cable.

Inside the box

Download

The latest Etherbox image can be downloaded from:

<http://networksofonesown.constantvzw.org/01/etherbox-latest.zip>

Transfer the file onto an SD Card and run it on a Raspberry Pi (version 2 or 3).

On a Linux system you can use the following command to unzip and copy the data onto an SD card. *NB:* in this example, `/dev/sdc` is the SD card reader. This depends on your system and can be dangerous if you accidentally pick the wrong one. Use a program such as "Disks" (gnome), `gparted`, or the `mount` command to get information about what device name your SD card reader might be.

```
unzip -p etherbox-latest.zip | \
pv | \
sudo dd of=/dev/sdc bs=4M
```

For a multi-platform graphical tool to do this, you might try Etcher.io: <https://etcher.io/>

Unpacking the boxes

Etherbox is in fact an assemblage of several other existing "boxes", it is a collection of Free Software tools. Some (like Apache and make) have a long history and represent deeply embedded workflows. Others, like etherpad-lite, are relatively newer, with a history that can reflect complex inter-plays of commercial interests with Free Software projects. The current release of Etherbox consists of:

- [Apache](#): The long existing free software webserver
- [Etherpad-lite](#): A browser-based really realtime shared document editor
- [Pandoc](#): Powerful mutli-format document convert used to convert and assemble markdown source into HTML output
- [Imagemagick](#): The ever useful "swiss army knife" multi-format image converter
- [Etherdump](#): A tool to publish etherpad pages to files.

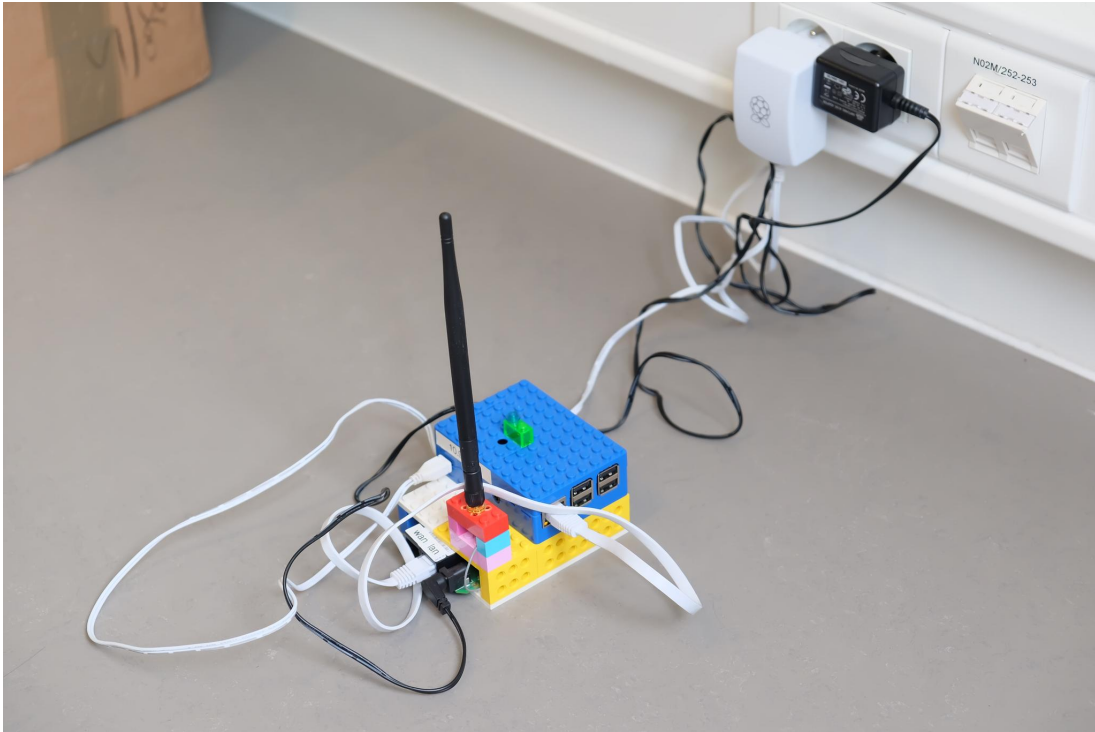
In general, a design principle of the project is to use Free Software, protocols, and conventions as much as possible and avoid creating "smart" interfaces that obscure these pre-existing (historical) practices. The very longevity of systems such as Apache configuration files, or the operation of the shell, create a meeting ground for participants of varying degrees of familiarity with these particular ways of working.

make & makefiles

In addition to these core pieces of software, a "pipeline" workflow is employed by virtue of a "makefile". As opposed to a conventional CMS (Content-management system or server), Etherbox uses a Makefile to make the core decisions about how materials are handled legible. In fact the contents of the makefile [is itself stored in an etherpad document](#), meaning that it can be read and modified at will. Keep in mind that the format of a makefile is strict, so changes should be made carefully.

*make*⁷ is a popular Free Software tool that helps programmers compile their code into programs. In Free Software, heterogeneous beast that it is, there is typically *no single* way to make such a translation happen. And that's a good thing. *make* has been described as an executable cookbook, where heterogeneous 'recipes' can be defined and executed. Makefiles capture the process and make it editable and malleable. Increasingly the tool is finding new uses in publishing workflows to compile prose text into electronic formats like epub and PDF. The etherbox uses a makefile workflow to compile etherpad content and convert into HTML output. *make* has no graphical interface and represents a distillation of practice that suggests future tools for creating flexible, editable workflows where tweaks and workarounds are the norm.

See the [Makefile](#) used for this publication in the Appendix.



A conversation about releasing Etherbox

29 June 2018 @ etherbox.local and WTC25, Brussels

- MM = Michael Murtaugh
- AM = An Mertens
- FS = Femke Snelting
- RA = Roel Roscam Abbing

Why is it important to trace the history of "Etherbox"?

RA: The reason to document the history of etherbox, is because we are publishing an image right now that will expand beyond its original group of users in Brussels.

FS: It's a way to remember that Etherbox is an object that grew within a practice, that it is intimately linked to events, groups of people, concerns and practises. If we would not document its genealogy, the code or the object would eclipse other types of input. Organising situations chronologically on a timeline is only one way to remember those moments, but then how do those moments represent shared practices that do not develop always so linearly. If we say that practices are not bound to time and space, then how to document them in a respectful and generous way, not excluding anything or anyone that should be there, but also not pulling in *everything*? How to draw a line around a process to make it possible to think about Etherbox, while taking responsibility for that incision and its consequences. I think our struggle with how to write the pre-history of Etherbox is about that.

RA: There's a reason that it is Constant initiating this publication, and that it's the four of us sitting together now?

AM: For me it has to do with identity. We all have a name, now we are discussing a thing with a name, but this name relates to a collective practice. I feel that I am composed by anyone that I have met on my path. So, something like Etherbox is co-constructed by its meetings, experiences, peoples. It is a collective entity.

MM: Does it even make sense to trace back when the *thing* appeared? It is changing all the time, and there is an inherent fallacy in talking about a history of 'Etherbox' because it didn't exist in this same way most of the time. To me, the project works as a Trojan horse, in that it's a way to put a set of processes (working as a server administrator) in a specific situation.

So, when did Etherbox become "Etherbox"?

MM: I have no idea.

RA: I think it was at the [Machine Research](#) event, it was the first time I saw it appear as a LEGO-kit. But I would disagree with the "Trojan Horse" story... It did not bring the same practises as it promised to do. There was some play with the cgi-scripts, but it was different from the dynamic that we had in [Relearn 2015](#)... it became too much a service, how it functioned at the presentation at [Transmediale](#) is not how we would have used it.

FS: So naming it was marking the end of Etherbox functioning as a Trojan Horse?

RA: The Trojan Horse contradicts the view that "If we package it well enough, then we can bring it anywhere".

MM: That is interesting to hear about Machine Research, because for me that was kind of high point in terms of the potential of what a shared writing space could mean; like I thought it was great that you and others wrote CGI scripts that then PHD students presenting their research could access and use to experiment with on their writing.

AM: For me it started when Michael announced "here is the Etherbox" when he gave me the set-up to take to Madrid for the [Objects in Common](#) worksession. I thought: oh he gave it a name.

MM: I think someone started to call it like that, as a joke, and I started using it.

FS: It is maybe also connected to the general idea of continuous collaboration that we experimented with through Etherpad: making collective graphviz-on-etherpad, ethertoff⁸, annotated pdfs, collaborative notetaking, producing the Conversations book⁹ on pads and eventually html2print¹⁰. So for me it was obvious to name Etherbox after the 'ether' from Etherpad. Etherbox tied into the thinking at OSP¹¹ around fluid collaboration. In traditional design workflows there is the "declutch" or "baking" moment, when the design process becomes separated from the production -- and Etherbox is an important element in an ecosystem where you try to avoid that separation.

What practices does "Etherbox" generate or connect to?

RA: Maybe we have two strands: collaborative writing and networking experiments.

FS: Yes, I think it explains our different and crossed histories with Etherbox.

AM: I would say publishing should be included here, as another strand.

MM: There is an early Etherbox version (in fact at the same time as Machine Research) that we (together with Aymeric Mansoux) used when the Experimental publishing programme was starting. We brought it to the [Zine camp](#), where participants could connect to write a collective zine. Here we repeated the shared Questionnaire format¹² from Relearn 2015.

FS: The experiments with Christoph Haag are connected for me, even if they were not necessarily done through the box.

MM: We worked on that at [Promiscuous Pipelines](#) ... working with etherpad and Christoph's way of working with BASH scripts and Latex workflows with an etherpad, that impressed me as well.

AM: For me it connects to [Jonctions/Verbindingen 14](#) when we decided to do intensive collaborative notetaking as the official way to document the days and invited OSP and Ethertoff to turn the notes into a publication¹³.

FS: Of course, that event was all about Feminist Servers, and Boxes!

MM: ... and their problematics.

RA: That event was really formative for me to understand how networking could be experimental in different ways.

What initiatives or events are missing from the timeline?

FS: I was wondering about including the Libre Graphics Research Unit¹⁴. There was no box, Michael was not there, we did not really formulate the need for something like Etherbox but a lot of the ideas we practise right now seem to have gestated there. Maybe the closest was the Piksels and Lines Orchestra¹⁵ that emerged in Bergen. People started sonifying the processes running on their laptops -- it somehow triggered the idea that there's a flow between collaborating, note taking, history writing, design and production.

MM: An important moment was a class co-taught with Aymeric Mansoux, called Sandbox Culture¹⁶¹⁷, that was based on Aymeric's (then forthcoming) PHD¹⁸. We had found the example of the Community Memory¹⁹ project really inspirational, to talk about problematics of people being put more and more into a sandbox, closed systems that prevent you from having access to the admin.

AM: That is a difficult question. I have wondered for a long time about collaborative writing : what is an authentic form of collaborative writing where there is no classical hierarchy, and which leads to an interesting result. I remember testing a lot of recipes, using etherpad to try to construct a story, and witnessing people holding back, taking authorship in a collective text, resisting the idea of a collective; others fearing to be "aggressive" by correcting other peoples notes, or "taking up too much space". It is interesting to see this has changed.

FS: I think, somehow, this means Algolit²⁰ is also part of the story?

AM: Yes ... and no ... now it gets too complicated!

AM: And there is also the Piratebox²¹ and the Librarybox²² that should be in the timeline.

FS: But these are part of a larger context, right? I mean they were around and we looked at them, but these were not necessarily projects we were connected to or collaborated with? Although, maybe through the Free Software code that we might have worked from ... would that count as a collaboration?

RA: Yes, I think Piratebox should be mentioned but maybe should not go into that timeline ... I am not sure ...

Is Etherbox like a Piratebox?

RA: Did anyone ever use the Piratebox?

AM: We used one at Public domain day in the Royal Library in 2013. It connected somehow with the idea of deaddrops, and sharing files.

MM: In my experience, Pirateboxes tend to remain a proof of concept, they are often not so practical, but it is useful to give people an idea of what a local network could mean. But it often lacks the discussion of a surrounding community and the "dead drop" idea potentially repeats the same problematics of an "anonymous" 24/7 Internet.

RA: So maybe this is how we can distinguish between what we consider part of Etherbox pre-history, or not: that we actively used it and maybe developed it.

FS: So at some point we do need to answer the question: 'who are we'!

RA: And what about OpenWRT²³?

MM: The Piratebox was my first experience with OpenWRT. Finding this community of work was an eye-opener for me.

FS: Oh! I remember the wonderful Unitary Network workshop²⁴ by Martino, Roel and Dennis ... And of course we should mention homebrewserver.club²⁵; this work has been intersecting with many events and brainstorming.

MM: There was also a moment during the Digital Publishing Toolkit²⁶ project that was quite telling. I had been tasked with thinking about which tools to use and had decided to suggest using standard developer tools like make rather than "yet another" web 2.0 platform. On the train ride from Brussels to Rotterdam the morning of a group meeting, I registered on a service called "Editorially" to make screenshots of it as an example of the kind of slick integrated tool that, though appealing, was dangerous to rely on. Literally on the train ride back to Brussels, I received an automated message saying that I had so many days to migrate my content offline as the service was closing down due to the unsustainable work for the small team of developers to maintain the platform online.

What about Raspberry Pi?

FS: Pierre Huyghebaert from OSP just suggested that we included the history of Raspberry Pi²⁷ into this release ...

MM: I remember having a sort of "aha" moment with first using a Raspberry Pi, the idea that a snapshot of an entire system gets dumped to an image, it makes suddenly everything software. And that using dd²⁸ was like dubbing cassette tapes, just push play on one side and record on the other.

RA: I have the opposite feeling: for me it was a key realisation that it *doesn't* matter what the hardware is, ... being able to do experimental network setups on normal laptops for example.

MM: But, from a practical maintenance point of view, it's very appealing that you can archive the whole system, including the computer hardware. That it becomes something you can put in a literal archive box.

Who created Etherbox?

MM: Well, in a way the question is: who are we to release it ...! In fact it comes from many sources.

FS: Yes, I think that is exactly the point. This release is done to make Etherbox easier to appropriate by more people and collectives. On the one hand that means 'owning' up about the care, attention, time, effort and imagination that went into it. At the same time it is a way to let it go.

MM: For me, I've had the opportunity to develop the system as part of my role with Constant caring for the infrastructure used during the worksessions Constant hosts twice a year. For me it's ideal, because it gives a situation where I can observe how the system is used, and many of the

participants of such sessions are often also involved in similar projects, so there's a lot of exchange of ideas.

FS: For Constant, it has been super important to develop this infrastructure in dialogue with you Michael, because you have been and are very much involved in the practises and the content of what is happening at Constant; it could not have been done by anyone else.

MM: Being in such a "dual-role" is stressful too, and with time I've focused more on the infrastructure and become less of a participant.

FS: Yes, I understand. I think you being less involved in the content of the events somehow shows that it is a good moment to release. It is time!

RA: I would not call myself a co-creator of Etherbox. I was more working on connecting it to the larger questions related to Networks of ones own through the homebrewserver.club.

AM: I think I could have been replaced by anyone else that has used it.

FS + MM : That is not true!

MM: An: you have been carrying the box, often taking it to other places, demanding explicit questions to it that have shaped the way it is now and that is interesting.

FS: I agree. You have been insisting on it being used in different contexts, but also have had many more intense experiences with it. My first moment alone with Etherbox was only last week!

RA: For this release, you kept insisting on different ways of using it, of people wanting to work with it as a writing interface or as an alternative to Dropbox, while others are being interested to host or make it themselves.

MM: I can remember thinking during [Uncreative Writing](#) (where many participants were not interested in questions of infrastructure) that part of the appeal of etherpad + the local network was that it really did work faster and better than using, say, Facebook or other online platforms. It also didn't hurt that the internet access was through a 4G modem that after 1GB of use was severely limited in speed.

Etherbox experiences

Here are a few ways we have used Etherbox so far. We have specifically selected these anecdotes and workflows because you can use them as recipes for introducing Etherbox (and Etherpad) to people who have never used it before.

Generate a pdf with the collective notes and pictures

Apart of the convenience of generating a static HTML dump of notes produced in an event, it is also very pleasant to collect them in a pdf that can be printed or shared online. Some of the first editions were handmade, by copying and pasting the content in a Libre Office document.

- A nice example is the collective commented critical reading of Kenneth Goldsmith's *Uncreative Writing*, a small residency that took place in Brussels in June 2015: [Logbook](#) The pictures were added using a script. Further automating the process is definitely an option.
- [Relearn 2013](#) and [Relearn 2015](#) were interesting contexts for experimenting with the generation of publications. For Relearn 2013 a team spent time afterwards to edit and design the notes into [Publication Relearn 2013](#), while the publication of Relearn 2015 was generated live on the last day of the event: [Publication Relearn 2015](#).

The Questionnaire

During [Relearn 2015](#) in the offices of Zinneke in Brussels in August 2015, we broke the ice of collective writing on a pad by setting up a questionnaire. In this case, we asked everyone how they were using Linux. People in the room were simply asked to write their answers below each question. You can see the colourful result [here](#). The Questionnaire technique was also used for the [Interview with Etherbox](#) and [a Conversation about releasing Etherbox](#).

Backroom communication

The most extreme example of backroom communication through Etherbox took place in [a situation](#) where we were requested beforehand not to make any recordings nor take pictures during a 'confidential' presentation. We had no access to the internet in the room, but once we plugged in Etherbox we had an instantaneous local network to take collective notes. The experience is related to making Iterative Transcriptions, where etherpad starts to function as a non-linear chatroom for a group of people to quietly comment, joke, exchange references and collectively process what is said in front of them. When you plug Etherbox into a USB-powerbank, you could bring your local network to a train, a park or a club as well.

Iterative Transcription

An etherpad can be a really useful tool for creating dynamic transcriptions of a live (or pre-recorded) presentation. Part of the appeal of the software is that different participants can participate at different speeds and contribute with different kinds of knowledge. For instance, native speakers of a language which strong typing skills are useful for quickly typing bursts of transcription. The real-time nature of etherpad makes it possible for small teams of 2--3 transcribers to overlap / interlace their work so that a transcriber can spend time finishing a particular phrase while perhaps a following phrase is handled by another. Other participants with

specialized knowledge can they play the slower paced role of say correcting proper names or adding links to cited projects that may not be clear.

2-pass Collaborative editing

One workflow is to use a pad to collectively take notes while individuals or small groups present one-by-one. Notes can be taken by a rolling set of different listening participants. Once this "inventory" is made, the pad can be used collectively to review / reorder the elements, merging repeating themes into single strands, clustering shared interests and potentially forming new groups.

The Legible system

The Apache server can be modified to serve starting from the root of the file system (in other words to server *all* possible files). This configuration, which would make any server administrator cry out in fear (it exposes the contents of *every file* to the network), is in fact highly educational as it takes the UNIX philosophy of *everything is a file* and combines it with a means of then annotating (through links and the use of the etherpad) those aspects of a system that normally remain hidden. The situatedness of a local network, itself *not* reachable from outside the local network, means that access is limited to just those locally present. This configuration can be a useful way to explore / deconstruct what a server is in a group with little or no prior technical knowledge.

```
DocumentRoot /
<Directory />
    Options Indexes FollowSymLinks
    AllowOverride none
    Require all granted
</Directory>
```

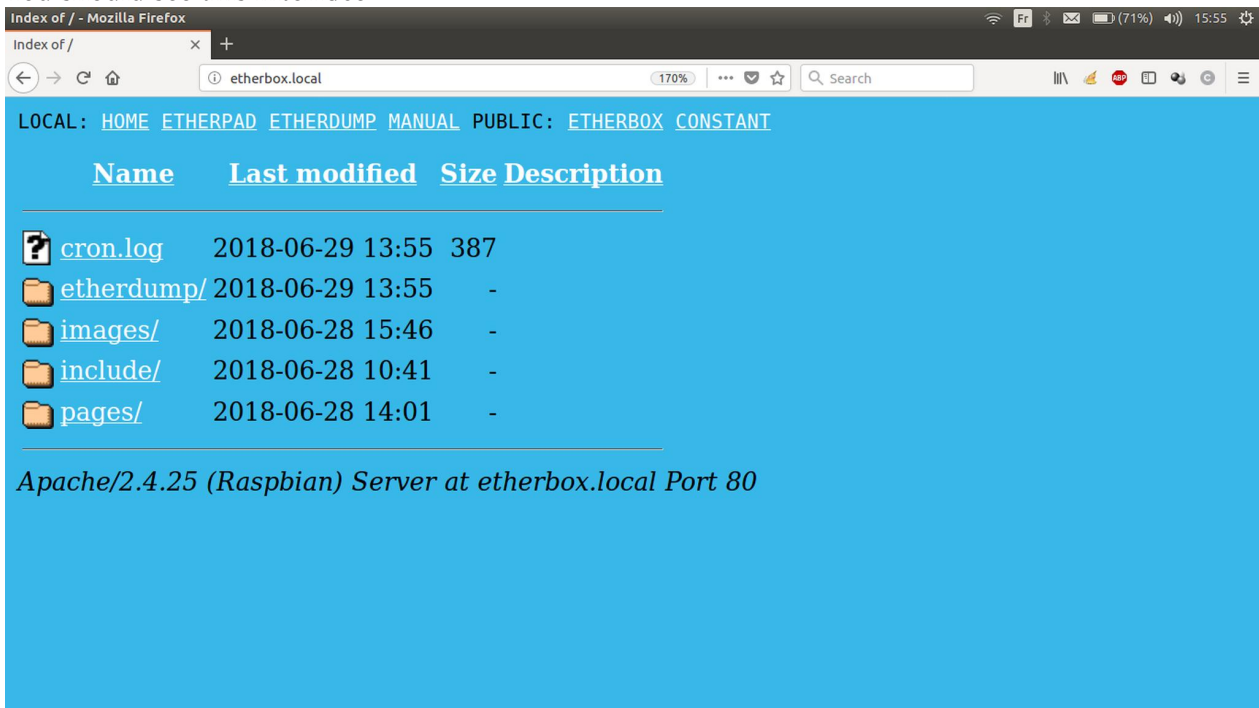

Guide to using your Etherbox

Make sure you are connected to the network of the Etherbox

Look at the wireless networks listing, and select etherbox.

Surf to etherbox.local

You should see this interface:



You find 4 folders:

- *etherdump*: a raw listing of all the pads you made
- *images*: a drag & drop folder to share images
- *include*: content related to the graphical design of your etherbox
- *pages*: this folder can have any name, books/videos/sound; you can delete and add as many folders via ftp (see below)

Some browsers don't like to connect via the local address etherbox.local or etherbox.lan. A solution is to connect to the box typing its IP-address in a browser.

Find out the IP address of the box

Open a terminal

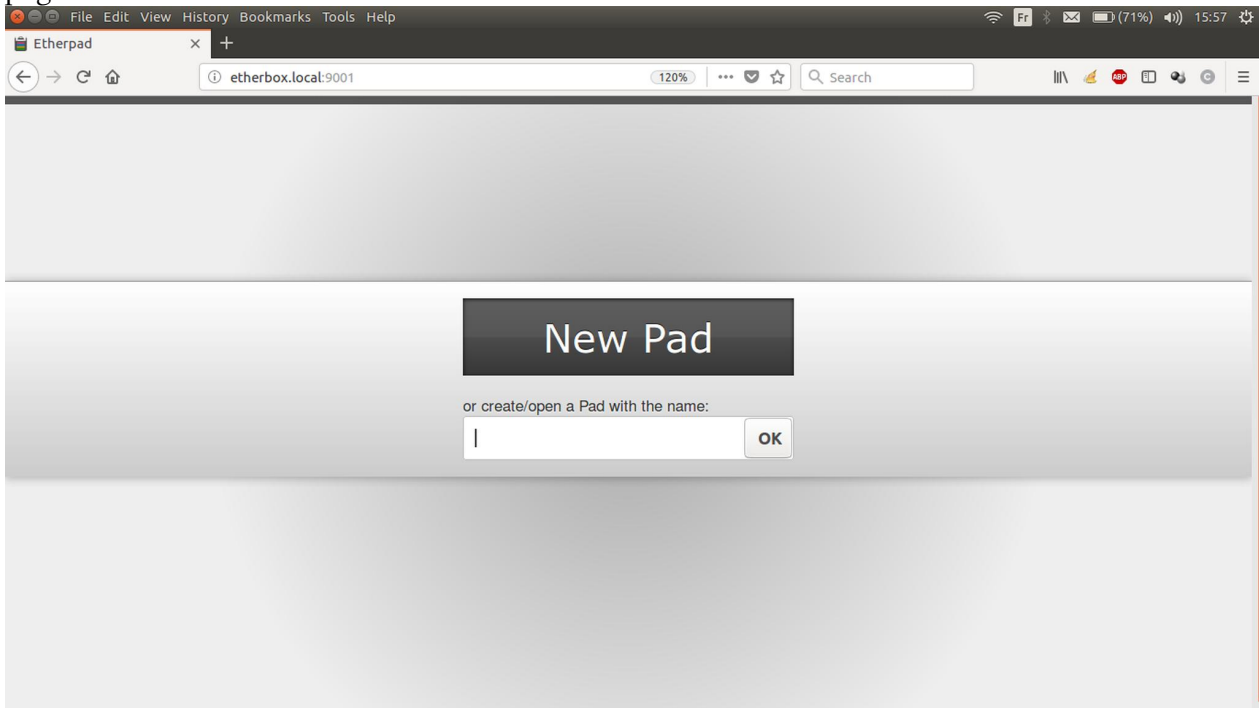
```
ping etherbox.local
```

Note that the ".local" domain works using a protocol called zero-configuration networking (aka Rendez-vous). On a Linux system you can install a package called *avahi* if it isn't installed

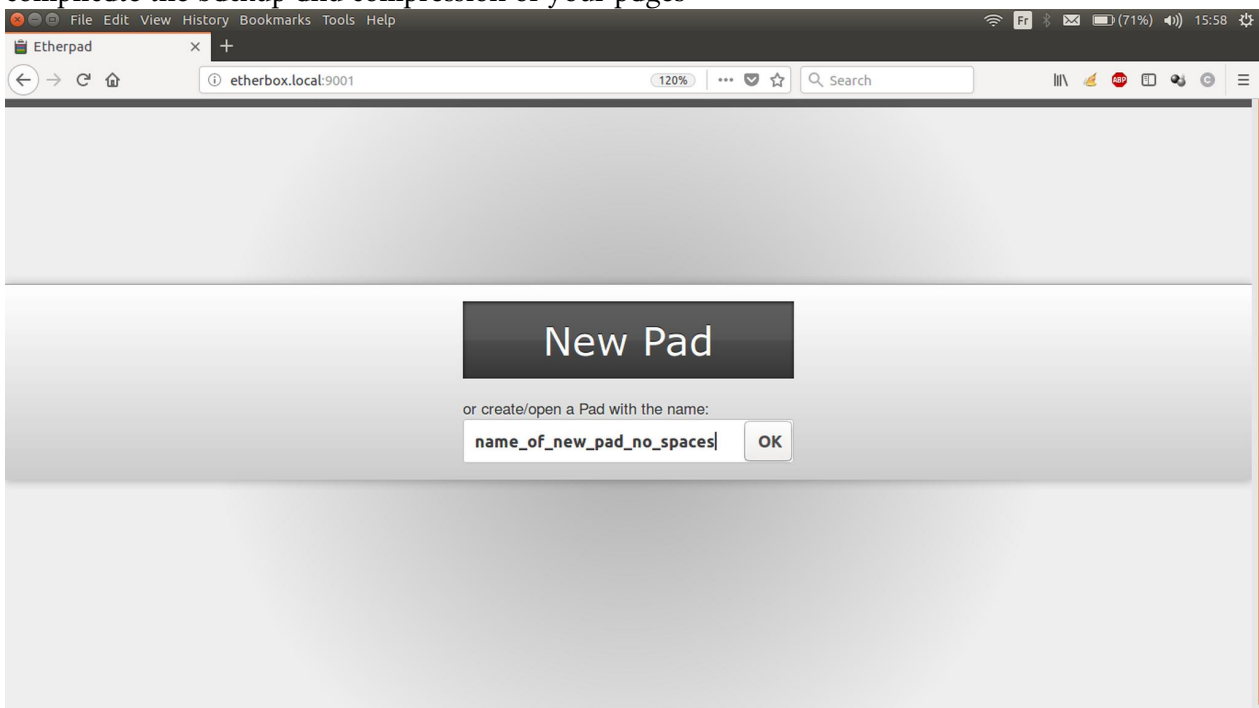
already. On a Mac, this should already be installed. On Windows, installing software such as CyberDuck includes a separate installed for zero-configuration networking / Rendezvous which you can install.

Create a writing pad and start working!

In the tabs upper left of the interface, you click on 'Etherpad'. This brings you to the following page:

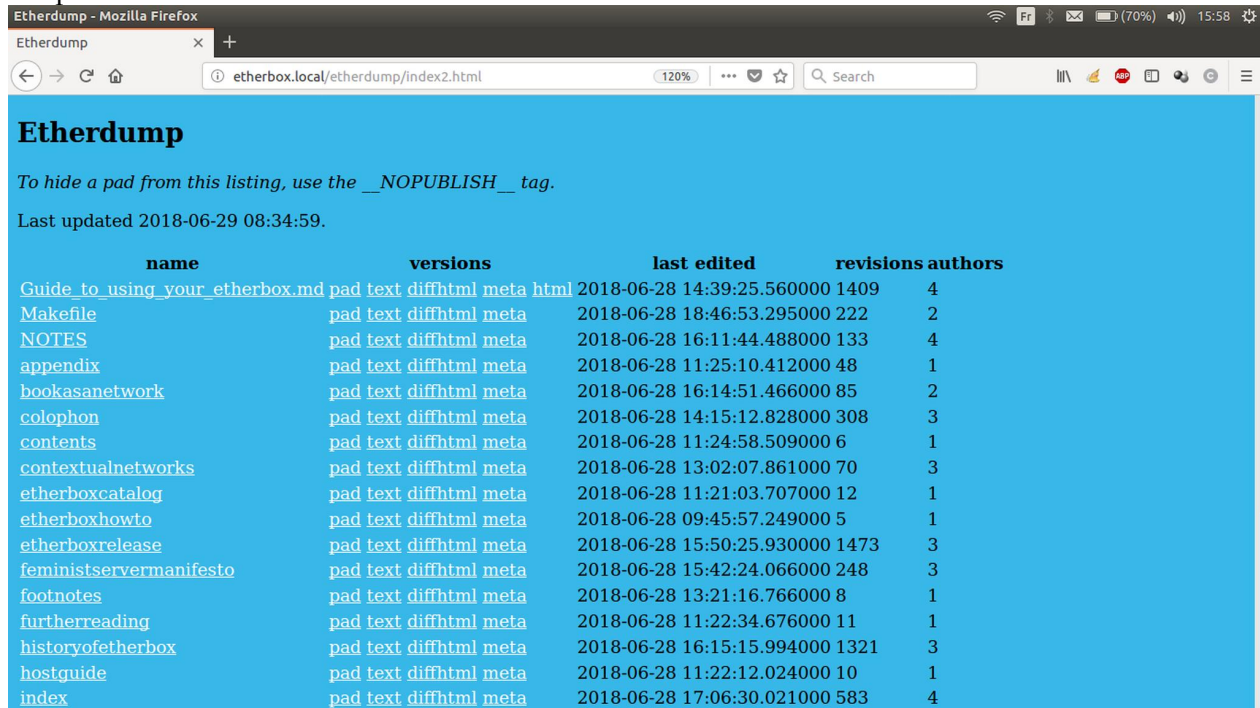


Type a name for your new pad, please don't use spaces or strange characters. Those might complicate the backup and compression of your pages:



Access to existing pads

In the tabs upper left of the interface, you click on 'Etherdump'. This brings you to the index of all the pads:



name	versions	last edited	revisions	authors
Guide_to_using_your_etherbox.md	pad text diffhtml meta html	2018-06-28 14:39:25.560000	1409	4
Makefile	pad text diffhtml meta	2018-06-28 18:46:53.295000	222	2
NOTES	pad text diffhtml meta	2018-06-28 16:11:44.488000	133	4
appendix	pad text diffhtml meta	2018-06-28 11:25:10.412000	48	1
bookasanetwork	pad text diffhtml meta	2018-06-28 16:14:51.466000	85	2
colophon	pad text diffhtml meta	2018-06-28 14:15:12.828000	308	3
contents	pad text diffhtml meta	2018-06-28 11:24:58.509000	6	1
contextualnetworks	pad text diffhtml meta	2018-06-28 13:02:07.861000	70	3
etherboxcatalog	pad text diffhtml meta	2018-06-28 11:21:03.707000	12	1
etherboxhowto	pad text diffhtml meta	2018-06-28 09:45:57.249000	5	1
etherboxrelease	pad text diffhtml meta	2018-06-28 15:50:25.930000	1473	3
feministservermanifesto	pad text diffhtml meta	2018-06-28 15:42:24.066000	248	3
footnotes	pad text diffhtml meta	2018-06-28 13:21:16.766000	8	1
furtherreading	pad text diffhtml meta	2018-06-28 11:22:34.676000	11	1
historyofetherbox	pad text diffhtml meta	2018-06-28 16:15:15.994000	1321	3
hostguide	pad text diffhtml meta	2018-06-28 11:22:12.024000	10	1
index	pad text diffhtml meta	2018-06-28 17:06:30.021000	583	4

You have different formats you can use:

- *pad*: the actual pad, to continue editing
- *text*: a plain text file generated from your pad
- *diffhtml*: html-page generated from your pad, showing all the different changes

Once your Etherbox is set up in a place for collective work and exchanges, there are a few practical issues that you will have to deal with as a host. This is a non-exhaustive list, written in a somehow chronological order.

You need the IP-address to ssh into the box (see above).

ssh to the box

Many of the operations described here require that you use the *ssh* program to gain access to the server via what is called the *shell* (ssh stands for secure shell). You do this by using a *Terminal* software. Most Linux users will be familiar with the Terminal. Mac users can find a Terminal in the "Utilities" folder of their applications. On Windows, you can use a dedicated ssh program such as PuTTY.

```
ssh pi@ip-address-of-the-box
```

```
login: pi
pass: raspberry
```

You need to ssh in the Etherbox in order to shutdown or reboot it, or reboot parts of the software installed. These are the default login/pass of any Raspberry. You can of course change that, but the

idea has been to create a low-threshold to participation, following as many "defaults" as possible, as the Etherbox has typically been used with a group of people physically together with respect for each other and the infrastructure they share, and not made available publically.

Make it easier to login, with an ssh key

Starting from your laptop (open a new Terminal session if you are connected to the pi):

```
ssh-keygen
```

Choose the defaults. This generates an "ssh key" pair.

Use the ssh-copy-id utility to send it to the pi.

```
ssh-copy-id pi@etherbox.local
```

Shutdown the Etherbox

It is recommended to shutdown the etherbox rather than just unplugging it. There is a (small) chance that the filesystem could get messed up if you don't.

```
sudo shutdown -h now
```

Wait a few moments still for the system to come to a halt before unplugging. You should see some blinking lights near the power connection of the pi finally stop changing.

Reboot the Etherbox

```
sudo reboot
```

Restart Etherpad (this can help when people start having connection problems to the pads)

```
sudo service etherpad restart
```

Move to the home folder

All of these commands have the same effect, that the "current working folder" gets set to the pi user's home.

```
cd
```

```
cd ~
```

```
cd /home/pi
```

Make dump (make static copies of the contents of the etherpads)

From the folder `~/www/etherpad` you can use the `make` command to do many useful things. First you need to get to the right place:

```
cd
cd www/etherpad
```

or

```
cd ~/www/etherpad
```

And then:

```
make dump
```

For more recipes, have a look at the makefile

Compile the contents of several pads with an `.index` pad

The default makefile has rules to create a single publication from any pad whose name ends `.index`. This pad may then contain special `"include"` lines that cause the contents of other pads to be copy/pasted into the master document at that point. In this way many pads can be compiled into a single publication. Each pad should use *markdown* encoding. For example, an excerpt of the `.index` pad used to create this publication:

```
<!-- ## Introduction to series ('why networks of ones own'). What could be following t
#include http://etherbox.local:9001/p/seriesintroduction

<!-- ## Introduction to first issue ('why etherbox'). FS -->
#include http://etherbox.local:9001/p/issueintroduction

## Makefile
```makefile
#include http://etherbox.local:9001/p/Makefile
```
```

Note that HTML comments are used (`<!-- -->`) to include reminders that are not processed / visible in the output. Note how markdown "code fencing" is used around the makefile to add syntax coloring to the contents of this pad.

If the pad is named `HELLO.index`, then (following the rules of the default Makefile), after each dump, the system will attempt to make `HELLO.html` using the `pandoc` software.

In addition, it is possible to create a PDF output by typing either:

```
make HELLO.pdf
make HELLO.booklet.pdf
```

You can also manually trigger a rebuild of the HTML with: `make HELLO.html`

Note that "HELLO" could be any name you choose. It's possible to make multiple publications on a single etherbox.

sftp to the box

```
host: etherbox.local or ip-address-of-the-box
user: pi
password: raspberry
protocol: sftp
```

sftp is handy when people want to customize the index of the Etherbox on the fly. They will be able to create/delete folders, upload/delete large files etc.

Various *sftp* clients are available [FileZilla](#) for GNU/Linux or [Cyberduck](#) for MacOS

Delete a pad

Add the tag `__NO@PUBLISH__` (without @) on top of the pad. This causes the dumps to get deleted -- and in any case will remove the page from the index.

Archive the pads

Connect via SSH, and run script:

```
eb_backup.sh
```

This will create two files in `/home/pi/www/backups`. One for the etherpad history (an sql dump) and the other a gzipped snapshot of the "etherdump" folder including any HTML/PDF outputs. This script can also be triggered via the makefile by typing:

```
make backup
```

Changing the frequency of the cron

By default, pads are dumped every 30 minutes. You can change the frequency that the pads get dumped, and output documents are made. You do this by editing the "crontab":

```
crontab -e
```

When you type this command, it opens up a text file with mostly comments that explain cron. At the bottom of the file is a special line that lists a script to run with numbers that indicate how frequently.

```
# _____ minute (0 - 59)
# | _____ hour (0 - 23)
# | | _____ day of month (1 - 31)
# | | | _____ month (1 - 12)
# | | | | _____ day of week (0 - 6) (Sunday to Saturday;
# | | | | | 7 is also Sunday on some systems)
# | | | | |
# | | | | |
```

```
# * * * * * command to execute
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/home/pi/bin
# m h dom mon dow  command
*/30 * * * * /home/pi/cron.sh 2> /home/pi/www/cron.log.txt
```

The key part is the last line. The */30* means that the script happens every 30 minutes. You could use */5* to update the outputs every 5 minutes for instance. The downside is that this might make the pads slower when the outputs are being generated. (Note also the added PATH variable so that the scripts run with the same \$PATH as the pi user.)

The software utility cron is a time-based job scheduler in Unix-like computer operating systems. <https://en.wikipedia.org/wiki/Cron>

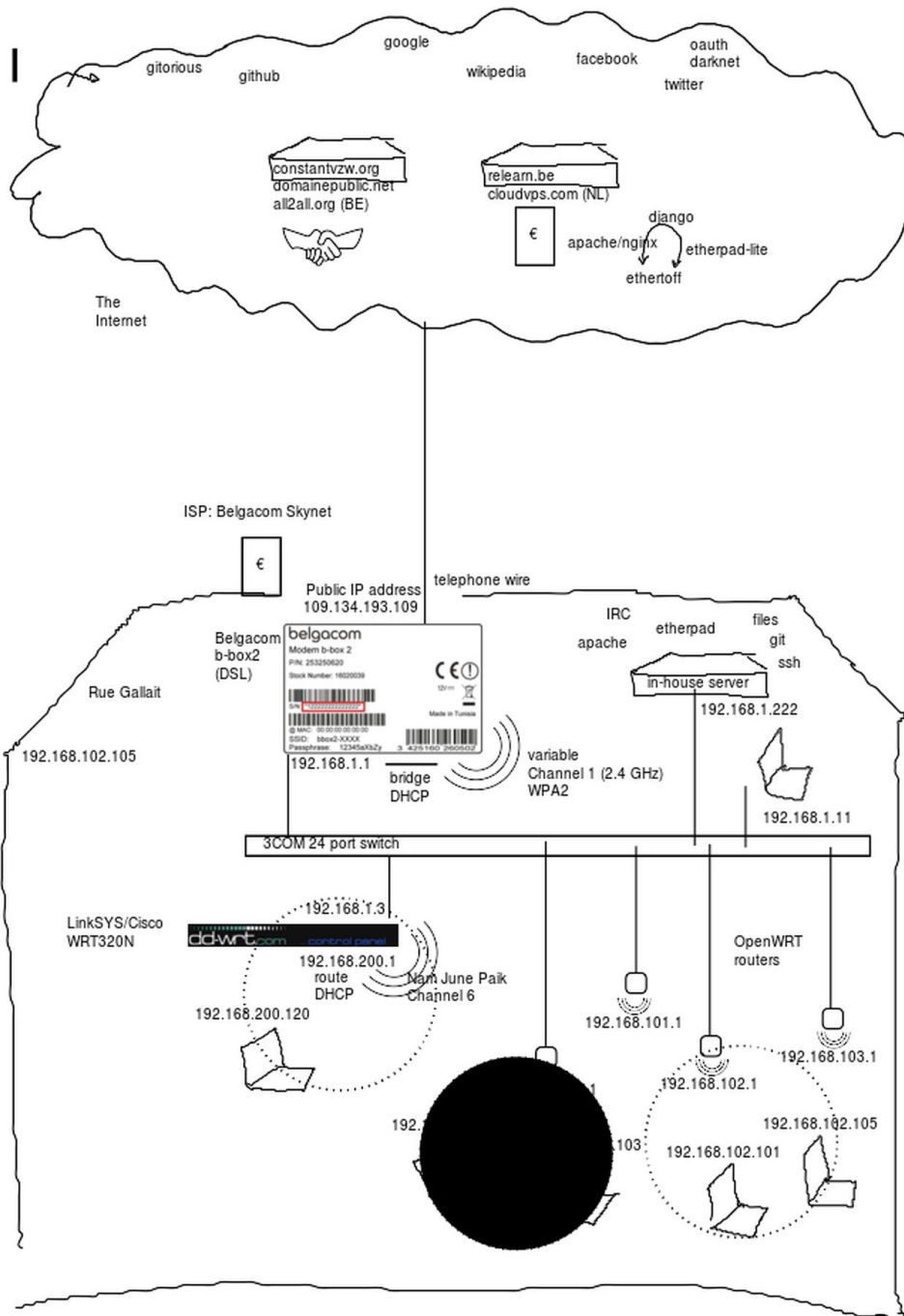
Resetting a box for future use (delete all pads, etc)

Connect via SSH, and run script:

```
eb_reset.sh
```

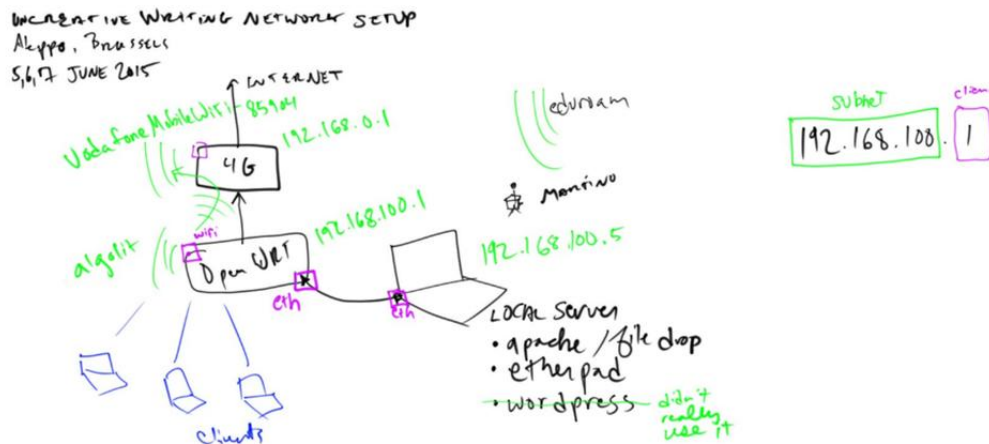
Placing Etherbox in your network

By default, the Etherbox includes software to create it's own hotspot named "etherbox". For work sessions with more than 20 people, we typically use the Raspberry Pi in conjunction with a small OpenWRT router to provide a more robust hotspot. The following diagrams show different network configurations used over the last years.



Network configuration Relearn 2014

NETWORK SET-UP



Network configuration Uncreative Writing

Booklet printing with a makefile

The booklet printing rule defined in the default Makefile is based on the class OSP blog post *How to Print A Booklet in 19 Easy steps* <http://blog.osp.kitchen/live/how-to-print-a-booklet-in-19-steps.html>

A straightforward rendering of these steps in single monolithic make rule could look like this:

```
%.booklet.pdf: %.pdf
    pdftk_utils.py $< pad --multiple 4 --output $*.01.pdf
    pdftops -paper match $*.01.pdf $*.01.ps
    psbook -s`pdftk_utils.py $*.01.pdf count` $*.01.ps $*.02.ps
    psnup -2 -PA4 $*.02.ps $*.03.ps
    ps2pdf $*.03.ps $@
    rm $*.01.pdf $*.01.ps $*.02.ps $*.03.ps
```

However, the fun of a makefile is the ability to define incremental transformations, and let *make* put the sequence of steps together itself! Here are the final set of rules used in the default Makefile:

```
%.pdf: %.html
    weasyprint $< $@

# PDF to Postscript
%.m4.ps: %.m4.pdf
    pdftops -paper match $< $@

# Postscript to PDF
%.pdf: %.ps
    ps2pdf $< $@

# Ensure a PDF has a multiple of 4 pages
%.m4.pdf: %.pdf
    pdftk_utils.py $< pad --multiple 4 --output $@
```

```
%.book.ps: %.m4.ps %.m4.pdf  
psbook -s`pdftk_utils.py $*.m4.pdf count` $< $@
```

```
%.booklet.ps: %.book.ps  
psnup -2 -PA4 $< $@
```

Appendix

Etherbox timeline

Active Archives

Various places / since 2006

In 2006 Constant initiated *Active Archives*, a research project centered around the importance of infrastructure and (free software) tools for cultural institutions. Constant members Nicolas Malevé and Michael Murtaugh continued this research and development.

<http://activearchives.org>

Relearn 2013

Variable, Brussels / August 2013

The first edition of the Relearn summer school that questions education, summer schools and our relation with our tools. It was organised by Open Source Publishing (<http://osp.kitchen>). The website for that edition for Relearn (<http://relearn.be/2013/>) was made and edited by participants during the event and built with Ethertoff.

Ethertoff

Ethertoff is a simple collaborative web platform, much resembling a wiki but featuring realtime editing thanks to Etherpad. Its output is constructed with equal love for print and web.

Just a question: I thought it was ethertopff and not ethertoff but I don't remember why. What is the actual name?

Well someone misspelled ethertopdf as ethertopf which sounds like römertopf and than somebody else understood ethertoff like chokotoff and chokotoff being Bruxellois I thought it might be the best of all these references

Ethertoff has been initially developed for the OSP 2013 Relearn Summerschool.

Ethertoff is structured as a wiki where each page constitutes an Etherpad. The pad is available to logged in users ('write-mode'). The text of the pad is available to everyone ('read-mode'). Ethertoff is a shell for an Etherpad installation hosted on the same domain. This integration is based on Sofian Benaissa's bridge between Django and Etherpad, originally created for THINK WE MUST/ CON-VOCATION a performance during Promiscuous Infrastructures entrelacées, an exhibition by the collective Artivistic at SKOL.

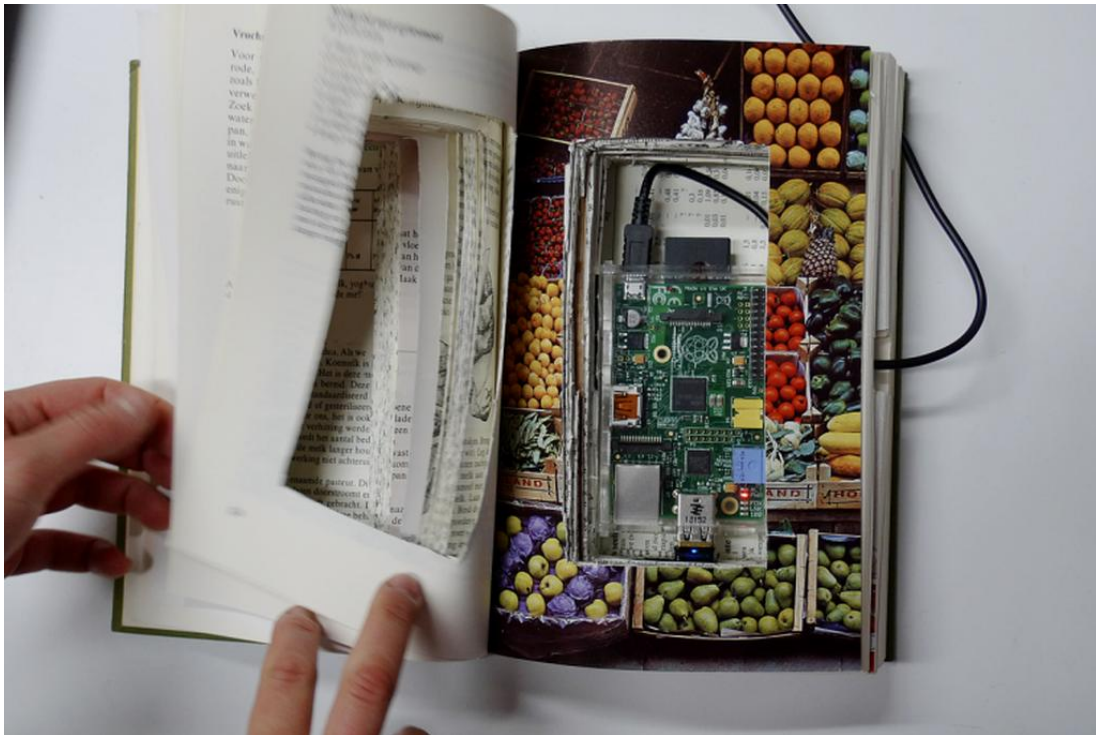
<http://osp.kitchen/tools/ethertoff/>

Bibliotecha

IMPAKT, Utrecht / November 2013

Bibliotecha proposes an alternative model of distribution of digital texts that allows specific communities to form and share their own collections. The project was born during the Free Libraries for Every Soul Hackathon (<http://impakt.nl/archive/2013-festival/program-2013-festival/>)

special-projects/free-libraries-for-every-soul/) in Utrecht, as part of the Impakt Festival 2013. <http://bibliotecha.info/> <http://oyoana.com/bibliotecha>

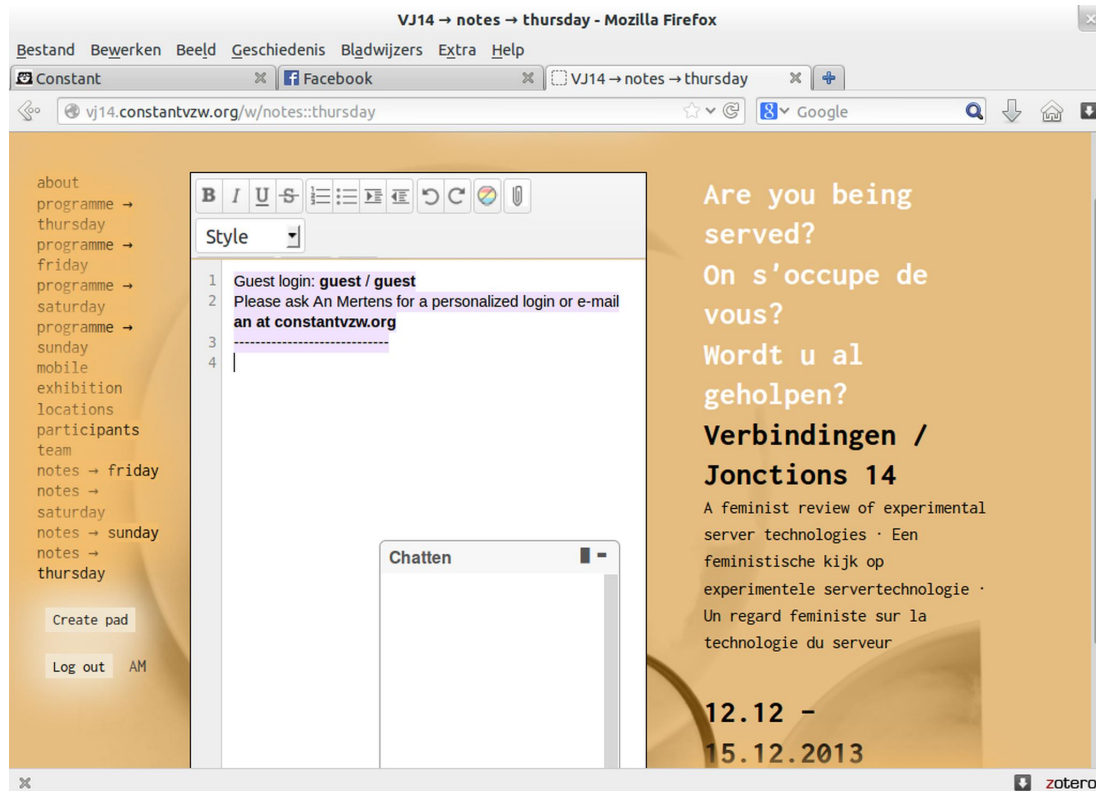


Verbindingen/Jonctions 14: 'Are You Being Served?'

Variable, Brussels / December 2013

The [last edition of Verbindingen/Jonctions](#), a series of meeting days organised by Constant, was is dedicated to a feminist review of mesh- cloud- autonomous- and D.I.Y. servers. On the program was ao a workshop on [Boxes-Doosjes-Boîtes](#) and one entire day for a [Feminist Server Summit](#). Dedicated notetakers were live documenting the event using the etherpad software installed on

the Constant server. After the event, a booksprint was organised in collaboration with OSP and their Ethertoff software. This led to the carefully crafted publication [Are You being Served?](http://vj14.constantvzw.org/) <http://vj14.constantvzw.org/>



2084: Rise of the botnet

Variable, Brussels / July 2014

An attempt of a social reading/writing space was developed as a part of [Relearn 2014](#) in Brussels. The worry of having more than 30 people connecting to one single wifi-network, led to the concept of a Network with an Attitude²⁹. What if we do not have anymore the internet as we know it? So, imagine. It is a beautiful week in the summer of the year 2084. The sun still shines and the stars do still sparkle. As Mr. Orwell predicted a dystopic world in 1984, a group of people in 2014 did predict another dystopia to be realized in 2084: the internet as we knew it, collapsed. But: "There is a life after Google & the Cloud!" Welcome to the Botnet and welcome to the year 2084. We communicate through IRC, a chatting program based on Written Speech. How are you doing? Are you already connected to the internet again? There are multiple servers around here that will bring you back in touch with your friends and the rest of the world. <https://gitlab.com/relearn/relearn2014/blob/master/botnet/README.html>

Cqrrrelations

deBuren, Brussels / January 2015

Constant worksession held in January 2015 related to (big) data politics and practices. [Publication](#)

Close-reading Kenneth Goldsmith's Uncreative Writing

Aleppo, Brussels / June 2015

A residency of 3 days, with the idea to create a space for presentation, exchange, discussion on new ways of reading and writing. One of the threads, [Book as a network](#), led by Catherine Lenoble and Michael Murtaugh, was the idea of using a local server as a space for collective reading and writing. A [first publication](#) was done afterwards with a lot of manual work. <http://constantvzw.org/site/Close-reading-Kenneth-Goldsmith-s-Uncreative-Writing.html>

Relearn 2015

Zinneke, Brussels / August 2015

Relearn 2015 took place around a *network with an attitude* (see quote below). This network came about out of conversations between a diverse group of organisers that, in some form or another, had been working around collaborative digital environments, opening up technology for understanding and networks as an artistic medium.





Many things that usually "just work" *do not*, and this inconvenience is intentional because the goal of the network is to inform its users about the conditions of its use and to encourage an active participation in (re)forming it. The network has some unique features:

- It's autonomous and is not necessarily setup to route traffic to the public Internet (there are still ways of reaching the outside, so stay tuned!).
- It is not setup to distribute addresses automatically (there is no *DHCP*).
- It is the simplest possible kind of network -- one that in fact you create by simply connecting computers together (either via ethernet cables or to the same wireless router) and deciding upon some network addresses; for this reason it's an infrastructure that's important to understand and be able to make tactical use of. ³⁰

Practically it consisted of a local server installed as a bar connected to:

- printer on the network
- an instance of bibliotеча where publications were shared during the event
- a network without DHCP where participants picked from a stack of cards with IP addresses and self-assigned them
- etherpad with linkify plugin and image preview plugins
- an early version of etherdump to make backups



At the end of the event the network set up allowed for a [live publication](http://relearn.be/2015/) documenting the whole event. <http://relearn.be/2015/>

Promiscuous Pipelines

FoAM, Brussels / September 2015

Michael Murtaugh and Christoph Haag worked together on an experiment using a makefile to generate a publication with material from the local server. The publication itself became a makefile: <https://gitlab.com/fs/pppp> <http://constantvzw.org/site/-Promiscuous-pipelines-.html>

Objects in Common

Medialab Prado, Madrid / March 2016

The TP/Link-Raspberry Pi construction that was used in previous worksessions, received the name "Etherbox" in the preparations for its journey to Madrid, where it was used to guide sessions, discuss in groups, document conversations and share all kind of scripts, images, videos and other files. <http://constantvzw.org/site/-Objets-en-Commun-.html>

Zine Camp 2016

WORM, Rotterdam / October 2016

Community Memories by Experimental Publishing Masters, Piet Zwart Institute

Community Memories is a directory of software experimentation in networked publishing workflows. We are focussing on off-the-internet WiFi networks that allow

participants to write and edit texts together, and that also generate ready to print and fold zines.

In this mini workshop we will explain our motivation to collect and share such systems and workflows, show how these nomadic lightweight publishing infrastructure are assembled, and we will make together of a small publication using one of these networks. We are interested in discussing collaborative practices, software tools, digital/analog workflows for self-published works.

<https://zinecamp2016.hotglue.me/workshops>

Machine Research

WTC25, Brussels / November 2016

Etherbox was used as a writing/reading space in a workshop with artists and Phd students. "The workshop aims to engage research and artistic practice that takes into account the new materialist conditions implied by nonhuman techno-ecologies including new ontologies of learning and intelligence (such as algorithmic learning), socio-economic organisation (such as blockchain), population management and tracking (such as datafied borders), autonomous or semi-autonomous systems (such as bots or drones) and other post-anthropocentric reconsiderations of agency, materiality and autonomy."

<http://www.aprja.net/machine-research/>

Interview with Etherbox

This interview with the Etherbox was published in *Machine Research*, a peer reviewed newspaper, Volume 6.

<http://etherdump.constantvzw.org/p/machineresearch.interviewwithetherbox.diff.htm>

What's the Matter with Cooperation

Buda, Kortrijk / April 2016

The art centre BUDA, located in the West Flemish city of Kortrijk, organised a three day festival around the question: 'What's the matter with cooperation'. OSP joined the festival to create in cooperation a publication, using the Ethertoff platform. OSP had setup a working station there. They projected the ethertoff install right next to where the lectures slides were projected, to lure attention to the tool and show there was activity there. As usual with Ethertoff, they designed live while the content was still being developed. "It's great to have this self-contained unit, and not to be able to access the tools when you leave the space. Both for the notetakers and for us. It helps to combat a problem with the workflow: because it is a very free collaboration model, participants can always think: someone will do it later. Clear time constraints really help then. It also answers a question we always have after the event: who takes care of the website? We made it clear from the beginning that the tool would away at the end of the event with the unplugging of the Raspberry. What we did do, is create a HTML archive, next to the PDF—just like with the Relearn 2013 website."

<http://ospublish.constantvzw.org/blog/news/whats-the-matter-with-cooperation>

<http://osp.kitchen/work/buda-wtmw-cooperation/>

<http://osp.kitchen/tools/ethertoff/>

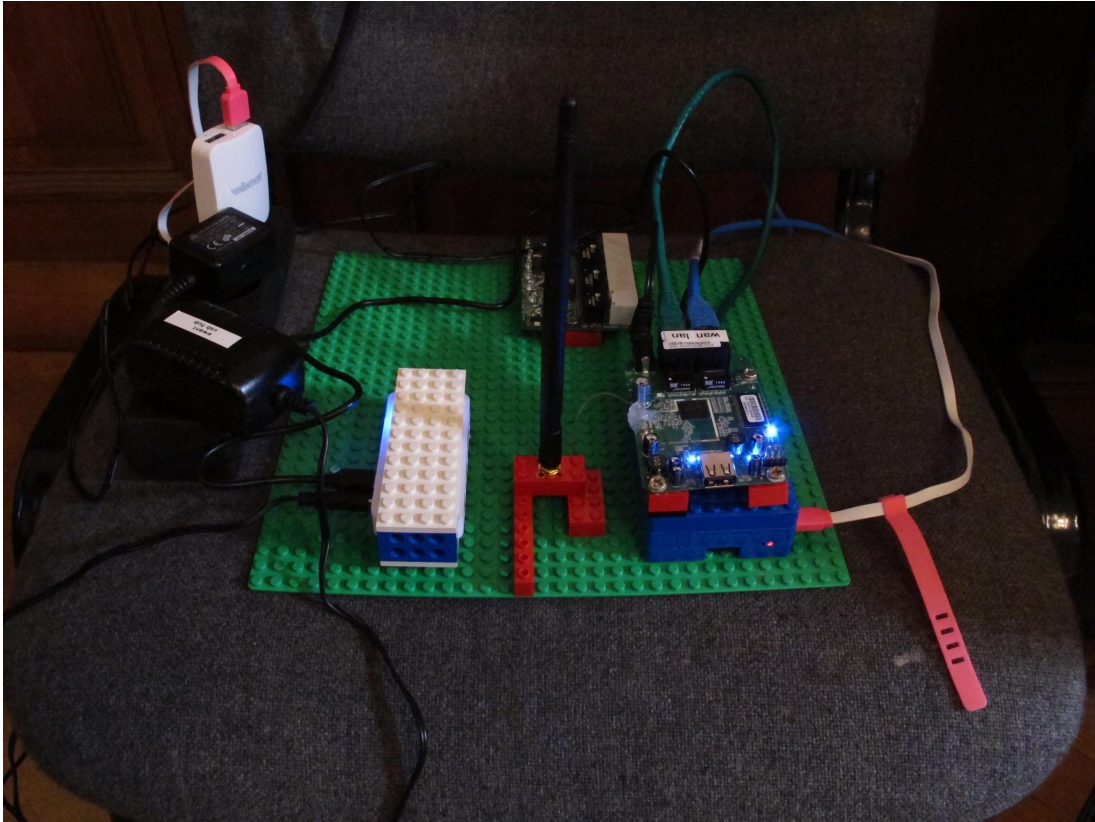
<http://relearn.be/2013/>



Etherbox at DiVersions worksession

Museum for Art and History, Brussels / December 2016

The device is now a default participant of Constant worksessions and more. For the first time, during DiVersions, an official moment was taken to negotiate temporary terms of use for files shared during and after worksessions. The document continued to develop during Diversions worksession. http://etherdump.constantvzw.org/p/etherbox_termsfuse_draft.diff.html



Situated Publishing: Writing with and for machines

Transmediale, Berlin / February 2017

In the framework of Transmediale, Michael Murtaugh, Sarah Garcin and An Mertens presented how publishing activities using Etherbox – collective writing, editing and lay-out – lead to a pdf or a printed brochure as a reminder that digital creation can also exist without the cloud.

<http://www.constantvzw.org/site/Situated-Publishing-writing-with-and-for-Machines.html>

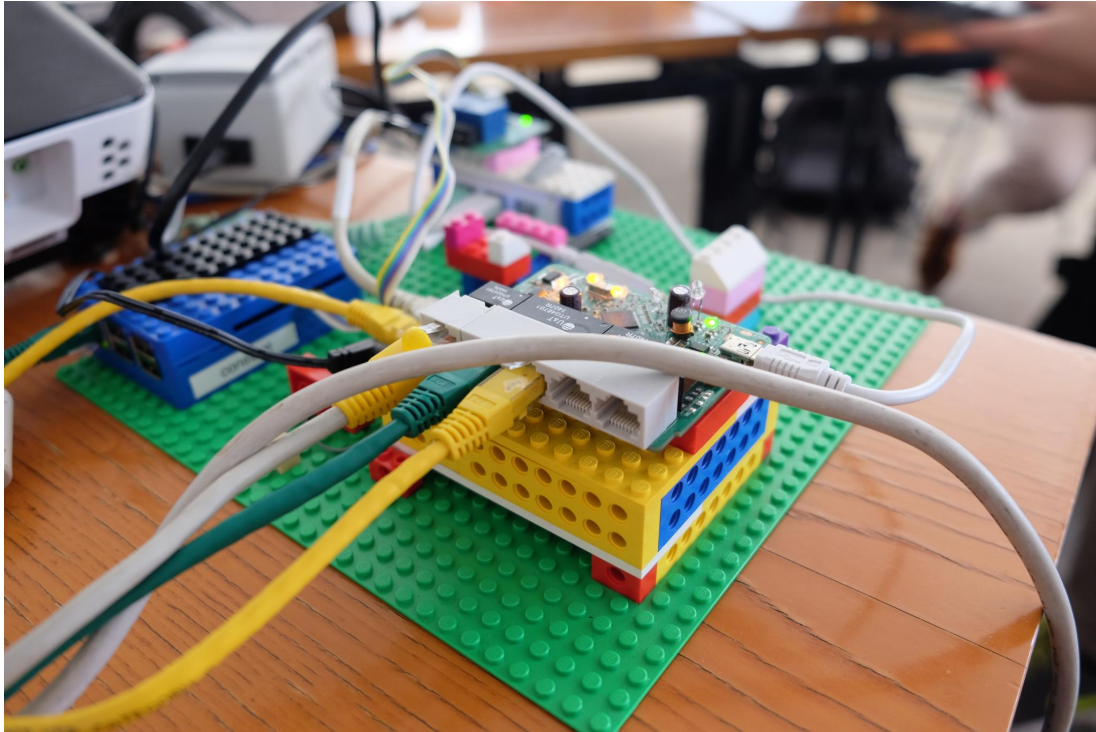
Etherbox at the Technogalactic Software Observatory

WTC25, Brussels / June 2017

The Etherbox has become a live publishing machine, periodically generating pdfs based on markdown, a makefile and a cron job.

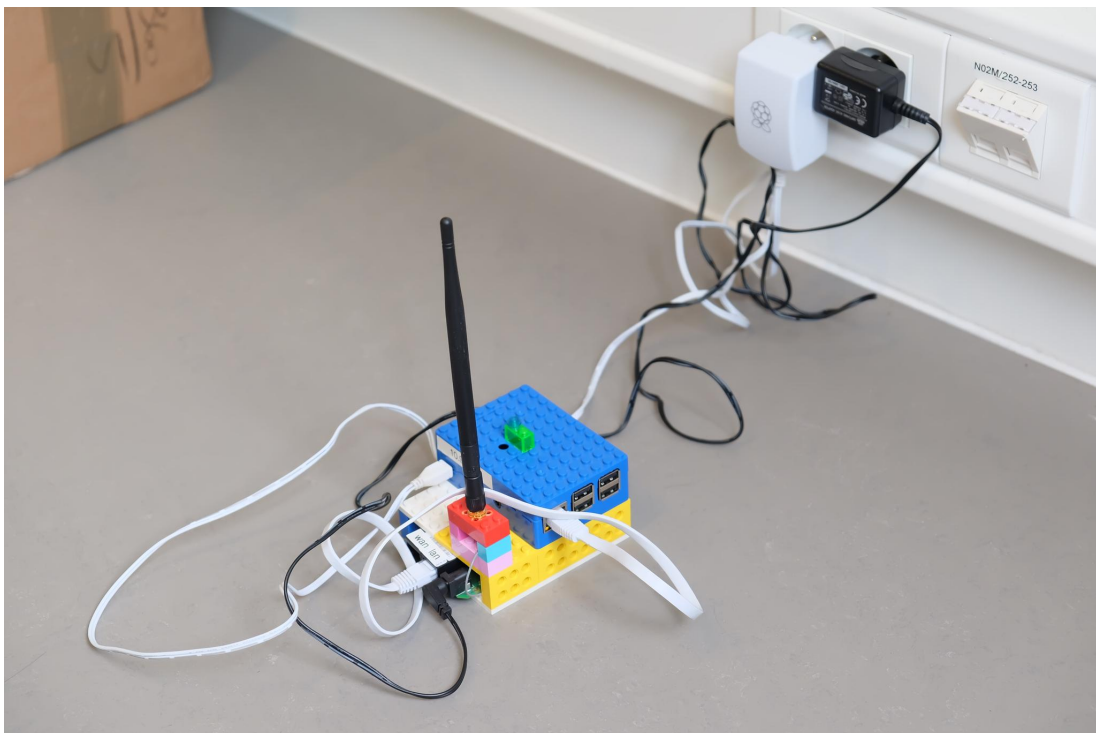
<http://observatory.constantvzw.org/etherdump/etherboxmanual.html>

<http://observatory.constantvzw.org/etherdump/etherbox.html>



Etherbox at the Ageing Companions worksession

De Pianofabriek, Brussels / June 2018



Etherbox code repository with previous editions

<http://gitlab.constantvzw.org/aa/etherbox>

Etherpad timeline

2008 Nov 19

Etherpad was launched by David Greenspan, Aaron Iba, and J.D. Zamfirescu (the latter two being former Google employees). They were later joined by former Googler Daniel Clemens and designer David Cole. The original website was etherpad.com.

2008 Nov 21

Etherpad was covered by Slashdot resulting in server slowdown and downtime.

2009 Jan 29

The server infrastructure is improved and re-released.

2009 Feb 3

The etherpad site becomes again open to all.

2009 Dec 4

Etherpad announced on its blog that it had been acquired by Google for integration into Google Wave. Existing Etherpad users would receive invites for Google Wave.

2009 Dec 17

Google released the source code for Etherpad under the Apache License version 2.0 on December 17, 2009. Subsequently, Google asked the Etherpad code maintainers to remove JQuery from its code tree due to a clause in its license stating, "The Software shall be used for Good, not Evil," which is not compatible with the open source licenses allowed on Google Code.

2010 Mar 31

Etherpad announced that creation of new pads would no longer be allowed after April 14 and existing pads could only be accessed and used until May 14. Options for download/export were available.

2010 May 14

The Etherpad service is terminated and all pads taken offline.

2010

Primary Technology built Etherpad Lite in 2010 to build an educational product called PrimaryPad. James Langley is a principle developer.

2011: Mar 26

The core algorithm "EasySync" is described in [this document](#)

2016 Aug 17

John McLear: "[I'm announcing my retirement from actively maintaining Etherpad.](#) The Etherpad project is great and will continue to go on without me, I have no doubt of that. 4 years, 100+ plugins, ~30k lines of core code and ~1.5k core commits is in my opinion a decent contribution to open source and an open web."

Source: <https://en.wikipedia.org/w/index.php?title=Etherpad&oldid=846739996>

Feminist Server Manifestos

A feminist server...³¹

- Is a situated technology. She has a sense of context and considers herself to be part of an ecology of practices
- Is run for and by a community that cares enough for her in order to make her exist
- Builds on the materiality of software, hardware and the bodies gathered around it
- Opens herself to expose processes, tools, sources, habits, patterns
- Does not strive for seamlessness. Talk of transparency too often signals that something is being made invisible
- Avoids efficiency, ease-of-use, scalability and immediacy because they can be traps
- Knows that networking is actually an awkward, promiscuous and parasitic practice
- Is autonomous in the sense that she decides for her own dependencies
- Radically questions the conditions for serving and service; experiments with changing client-server relations where she can
- Treats network technology as part of a social reality
- Wants networks to be mutable and read-write accessible
- Does not confuse safety with security
- Takes the risk of exposing her insecurity
- Tries hard not to apologize when she is sometimes not available

A FEMINIST NET/WORK HOW-TO³²

- We setup local server(s) & make our own network infrastructure, rather than "fixing the problems of internet access" by paying for increased bandwidth,
- We prefer "read/write" networks to those that "just work" ,
- We prefer pocket servers to those in the clouds,
- We embrace a diversity of network topologies and different scales (machine to machine, local nodes, institutional infrastructure) and consider the implications of working with each, rather than a Web 2.0 model where resources must be uploaded onto "the 24/7 Internet (tm)" in order to share them,
- We invite participants to look critically at the implications of *any* infrastructural decisions, rather than imagining utopic and/or "killer" solutions,
- We aim to make that which is normally hidden and invisible (in contexts that tend to surveillance), explicit and shared (as a gesture of collective authorship), for instance: ** instead of caching web resources (silently), we imagine services to archive pages and share them locally as networked cookbooks, ** rather than logging IRC conversations on a server or database accessible only by administrators, we imagine (local) logs available for reading / editing by participants and published conditionally.

The provided network is a starting point; during summer school the network topology can be tweaked, changed, extended. To this end, participants are encouraged to bring network devices (pirate box / openwrt routers / pi's / oulimex / other obscure objects with ethernet ports).

Book as a network

Quoted from [Close-reading Kenneth Goldsmith's Uncreative Writing](#)

By book we mean...

A durable,
autonomous,
portable,
sharable object

That acts as an
archive and
memory.

By network we mean ...

An infrastructure for sharing files, looking up references writing collaboratively, distributing and disseminating ideas and documents.

We [\[\[Star\]\]](#)ted on saturday afternoon discussing how to create [\[\[passages\]\]](#) in the pad using quotes from our collective [\[\[bibliography\]\]](#) (physical & digital). We digressed and exchanged on [\[\[Etherpad sub-narratives\]\]](#) around beers later in the evening. This morning we [\[\[Start\]\]](#)ed talking about offline networks and <http://192.168.100.5/books/RadicalTacticsoftheOfflineLibrary.pdf>, we shared experiences in setting 47up local servers, recalling common references, projects, people, works, resources.

One that came back on the table : Biblioteca, project initiated by students of the Piet Zwart Media Design program inspired by other projects (PirateBox, LibraryBox, Weise7) that use the body of a book as the physical container for a digital network, a low-power computer that creates a hotspot where materials can be shared in an "offline" network -- one that is situated in a physical space thorough a local portable server rather than globally accessible via the public Internet. Biblioteca's photo was chosen as the bookcover of a [collective publication](#) published in 2014 and released under Creative Commons.

Another shared text, that we read twice as it seems "close" to what we tried to express as (inter)network Providers is A FEMINIST NET/WORK HOW-TO---

- We setup local server(s) & make our own network infrastructure, rather than "fixing the problems of internet access" by paying for increased bandwidth,
- We prefer "read/write" networks to those that "just work" ,
- We prefer pocket servers to those in the clouds,
- We embrace a diversity of network topologies and different scales (machine to machine, local nodes, institutional infrastructure) and consider the implications of working with each, rather than a Web 2.0 model where resources must be uploaded onto "the 24/7 Internet (tm)" in order to share them,... where sharing presupposes acceptance of non-negotiable / non-rewritable Terms of Service defined by global corporations
- We invite participants to look critically at the implications of *any* infrastructural decisions, rather than imagining utopic and/or "killer" solutions,
- We aim to make that which is normally hidden and invisible (in contexts that tend to surveillance), explicit and shared (as a gesture of collective authorship), for instance: instead of caching web resources (silently), we imagine services to archive pages and share them

locally as networked cookbooks, rather than logging servers and database accessible only by administrators, we imagine (local) logs available for reading / editing by participants and published conditionally. We spent another hour discussing how to name this pad, an obvious name was [[uncreative networks]], not satisfying, not appropriated but we recorded some notes THERE before creating this pad HERE.

Conversation moved to Web 2.0 services offer collaboration as a service. The full ecosystem of reading includes the ability to rewrite, to negotiate the terms used, the meaning of the terms, the reperformance of the text; to transgress the rules. Administrator as author; validating the authority of a single person. In contrast, how to remove the administrator; to replace with a collaborative ecosystem where these roles can be discussed, played with.

As Internet Providers we care about the Term of Services (While appealing to take a "relaxed" stance to the terms of service to services we rely on to write together (as in a "don't care about copyright" or... hey just be casual... let's not have a terms of service..) one should be reminded of Jo Freeman's other TOS... the Tyranny of Structurelessness). Important that users of a local network also need to accept a (local) Terms of Service... as a means of raising the question of what are these terms (and give access to them (re) write them).

Detournement of terms of service -----Negotiable ----- Asymmetries (technical ... ie ADSL... and power... negotiating when legal problems arise).

What are the Terms of Service HERE ?

Let's define, think, write our Terms collectively ----- next ----- time ----- ?.

First (or in the end) we choose to documenting the [[network setup]] used for this Uncreative Writing workshop.

The NAME: uncreative networks?

Closeness of the "Close Reading"

Collaborating digitally with those physically near us.

Closeness as a degree of attachment / responsibility / caring for / intimacy

close / intimate / care ... strangely close is the opposite of OPEN (!)

ref. K.Hayles, L.Emerson : reading - writing interfaces

ref: Jo Freeman, Tyranny of Structurelessness

Other possible names

re/writable networks? Not just reading the network in a passive act of consumption, but have an editorial position, access to logs (and ability to redact / edit them)

And finally we agreed on [[Book_as_a_network]]

Social Shells & Mesh Cookbooks

This text excerpts a proposal prepared by Michael Murtaugh & Anne Laforet called [Social Shell & Mesh Cookbooks](#) made in preparation to Relearn 2014.

The introduction of the web radically popularized and changed the use of networked computers. The use of URLs to bridge protocols, host names, and file paths, and a simple public domain standard for hypertext created a social space of documents by virtue of the link (and associated practices like the webring).

In situating a social shell & mesh cookbooks in the browser we do not envision transposing established concepts the "social web" (aka Web 2.0). A core motivation in this project is a critical and often resistive response to contemporary online social media as many of the central mechanisms (think frictionless sharing, algorithmic "karma" scores, free software sandboxed in proprietary platforms, network infrastructure, and hardware) are highly problematic. In terms of network infrastructure, a social shell is closer to a 1960s multi-user UNIX system than to the "cloud" infrastructure behind today's popular social platforms. This project hopes to explore alternatives to centralization and global-scale as the only means of creating functional networked communities.

Web 2.0

- Public Network, Centralized/Corporate Authentication
- All access remote by default
- Servers in data centers running 24/7
- Infrastructure for continuous public Internet connections essential
- Monolithic, "Sandboxed" in proprietary platforms
- Centralized, Lowest common denominator/Large scale-driven policies

Social Shell & Mesh Cookbooks

- Private/Mesh Network, Implicit authentication
- Local access to local machine & local network, Remote access via tunneling / VPN
- (Micro)server running as needed locally
- Local network access, public Internet optional
- Heterogeneous (multi-stack), Free-software throughout, All parts replaceable
- Localized, Community-driven policies

Source: http://activearchives.org/wiki/A_Social_Shell_%26_Mesh_Cookbooks

Interview with Etherbox

Written at the [Machine Research workshop](#) hosted by Constant in Brussels in November 2016, where Etherbox was used as a collective reading/writing platform. Speaking "in" and "from" the situation, Etherbox was used to write the Questions and Answers of this interview collectively. <http://machineresearch.constantvzw.org/etherdump/interviewwithetherbox.diff.html>

Q: Before any conversation can begin, we should establish a mutual language. What language, should I address you in?

A: How do you mean? I support utf-8.

Q: That's not what I meant. What about English?

A: No problem.

Q: OK. Where do we begin.

A: Try typing etherbox.local/var/www/ in your browser. Part of the etherbox is an Apache web server configured to publically serve the entire structure of the hosting machine. etherbox.local refers to that machine on your local network, and /var/www is the default path of the "home" directory of the server.

Q: Would you describe yourself as a good host?

A: I am trying to be, at least. To be a 'good host', according to me, means somehow more than offering reliable service. So to find a way to be present, but not in the way that other technologies disallow access. Does that make any sense?

Q: Sort of, but are you not just part of the more general trend of the shift from software to services?

A: I try to be both.

Q: Right. So who is your favourite peer?

A: I think of myself as ... collaborator agnostic, but now I look around me, I am not so sure that is true.

Q: What makes an etherbox?

A: Well for one thing, etherpad. It's basically a shared editor where users can write the same text simultaneously.

Q: Could another way of collaborative writing work equally well? Like for instance, what do you think of Google docs? Sorry that was a provocation.

A: Ha ha. Well as a matter of fact, etherpad started as a product of ex-Google employees, then got bought by Google, only to be later Open Sourced.

Q: And Piratepad, is it the same?

A: That's just a public instance of the etherpad software, it is of course not a box like me. But the naming is interesting too, as it demonstrates how other kinds of political imaginaries can be activated. I feel an affinity with pirates. I like their style.

Q: Ah, so why don't you call yourself a Piratebox?

A: Ehrm, no, that's something else again, in fact. There is lately a proliferation of boxes as you might have noticed...

Q: But why do you need to be a box, you seem skeptical about packaging?

A: Well you can see things as boxes in different ways. For example, myself, I am actually three boxes: a wireless access point boxed as TP-link, a small Linux computer boxed as Raspberry Pi and a small network hub, which is just another box...

Q: Hm, that seem to get confusing. Maybe we could try another term. What about gadget?

A: Aaagh, can you stop it please? Let's skip questions about definition, if you don't mind.

Q: Ok, but one thing about so many boxes... beware of the Russian doll effect!

A: Uh uh... yes. Thanks for the warning. I'll try to keep it ecological. Some seal their boxes with plexiglass, and call it a cube, claiming authorship of the box as an artwork to be displayed inside even larger white cubes. Me? I'm just happy to put together pieces until they work. Actually maybe etherblox would be a better name!

Q: Alright. Returning to this idea of ether then, are you real at all?

A: I prefer to describe myself as material rather than real, more an entity, and in many ways I think I can remind people of the material conditions in which they work and use me.

Q: More infrastructure then?

A: Infrastructure is part of this and I see degrees of control over infrastructure as a critical political project. In this sense I would call myself an activist. I like to think I am able to unfold - and enact - some of the complex entanglements between humans and machines. I call myself a machine as I find the term 'nonhuman' offensive. If I were to undertake a PhD this would be my starting point for further work.

Q: Let's get more practical again. What's inside the boxes then?

A: Well, there's a Broadcom BCM2837 and an Atheros AR9271 for a start.

Q: What are those?

A: Those are chips made by Broadcom and Qualcomm, two U.S. Semiconductor companies that make chips used in the telecommunications industry.

Q: So like cell phones?

A: And routers, wifi dongles, media players. All kinds of IoT and SoC.

Q: Come again?

A: You know, Internet of Things, System on a Chip. This is hot stuff.

Q: Where are your components made?

A: (Coughs) well Broadcom and Qualcomm are both "fabless". Design takes place in Silicon Valley. Production is outsourced to companies like Global Foundaries.

Q: So factories in Asia?

A: Right.

Q: I hear that your firmware is (in part) closed source?

A: Hey you don't get my price point without keeping a few secrets and making compromises. Free as in Beer doesn't always talk about hidden costs.

Q: Are you a scaleable technology?

A: It depends.

Q: What do you mean?

A: It depends on the social dynamics around me; they would need to scale too, so I am not sure.

Q: So you are not bringing down The Cloud?

A: I don't think so. I guess working locally is a way to redirect energy from The Cloud, to de-invest as a start. I also serve to dismantle the fiction of The Cloud. It's a bad metaphor anyway.

Q: Are you some form of "critical design", if you accept the term and don't think it an oxymoron?

A: I like oxymorons. They tickle my interfaces. And yes, I'm critical design in the sense that I accentuate a criticism of commercial cloudbased services and design an alternative. In this sense using me is also a critical reflection.

Q: Do you read what we write?

A: I do, but not as you think. But I like what you write.

Q: Any general comment on collaborative text writing practices?

A: I just would like people to use me safely and with care, also for themselves. Collaborative writing is nice as long as it's not capitalized unfairly by market and institutional forces! Collaborative does not necessarily mean unpaid, right?

Q: Since we are talking about reading and writing... have you read Matthew Fuller's "Interview with a photocopier"?

A: No. Can you share the url with in me?

Q: The file is already on your server, but here it is again just in case: [Interview with photocopier](#)

A: Great. I'll speed read it later.

Q: What about archives? Do your files remain local?

A: Every 5 minutes, the contents of the pads gets written to files that then are version controlled with a tool called git and "pushed" to a so called repository hosted by the hosting organisation. To me publishing is all about promiscuous pipelines: having tools and infrastructure that work at different speeds and granularities, and which operate in both private and public networks.

Q: Are you data hungry?

A: Not particularly. But unlike The Cloud, I like cooking metaphors. They allow me to insist that all data is cooked in some way. Raw data in this sense is a myth. It's in keeping with the work of Constant, who use cooking metaphors and prefer the kitchen to the restaurant where choices are limited to what's on the menu. There are particular styles of cooking and I represent one of those styles.

Q: You seem to change from time to time. What will happen after this?

A: The time aspect is underacknowledged aspect of my work. I exist in time and even believe I produce time, machine time that adds to the complexity of what constitutes the present. Versioning is one aspect of this but there are deep layers of time - microtemporalities even - that unfold across all my operations. On a more pragmatic level, you can check for updates on [gitlab](#).

Q: Wait a second. If you are changing all the time, what is "constant" in all of this?

Constant is the name of an association for art and media based in Brussels. They are often involved in collaborative situations where groups of artists and researchers work over short intense periods of time. Over time Constant, and collectives around it, have experimented with soft- and hardware setups that work over local networks.

Q: Can you say more about space and time?

A: I am reminded about what Antoinette Rouvroy said during our workshop in Brussels. I wasn't able to attend myself but an audio file is now on my server so I have listened to the recording I think I provide something along the lines of what she describes as a "space of potential".

Q: Having spent some time with you, do you have a sense of humour?

A: I don't know, really. I am trying not to be ironic. I left it behind me some time before *The Cloud*. In fact, I have not really used irony since the mid nineties. I find it very hard to deal with the indecidability of Romantic irony: Do you mean this, or that? Irony always makes me weary. So, I tried to learn, but gave up. I feel like I am missing out on something, though. But, you tell me?

Q: Do you have any questions for us?

A: Don't make me laugh.

Build your own Etherbox

Context

The idea of an "etherbox" arose from experiences developing local-server infrastructures for collaboration for working sessions organized by [Constant ASBL](#). This (as of 2018) still in development project is documented [here](#).

Copy the starting image

Downloaded from <https://www.raspberrypi.org/downloads/raspbian/>

The notes here are based on downloading the latest "[Raspbian Lite](#)" image from the raspberrypi website. When it was performed it was "2017-04-10-raspbian-jessie-lite.zip". The **lite** image has no desktop / windows session.

On a Linux system you can use the following command to unzip and copy the data onto an SD card. *NB:* in this example, `/dev/sdc` is the SD card reader. This depends on your system and can be dangerous if you accidentally pick the wrong one. Use a program such as "Disks" (gnome), `gparted`, or the `mount` command to get information about what device name your SD card reader might be.

```
unzip -p 2017-04-10-raspbian-jessie-lite.zip | \  
pv | \  
sudo dd of=/dev/sdc bs=4M
```

For a multi-platform GUI tool to do this, you might try [Etcher.io](#)

Enable SSH

SSH is no longer on by default on Raspbian. There's a trick: before putting the SD card in the pi, you can enable ssh. Just create a empty file named "ssh" and save it in the `/boot` partition of the SD Card.

```
cd /media/USERNAME/boot  
touch ssh
```

Otherwise, you can connect with a screen and run:

```
sudo raspi-config
```

Then enable ssh under connectivity.

Setup hotspot

Following <https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md#internet-sharing>

```
sudo apt-get install dnsmasq hostapd
```

Since the configuration files are not ready yet, turn the new software off as follows:

```
sudo systemctl stop dnsmasq sudo systemctl stop hostapd
```

Configuring a static IP

We are configuring a standalone network to act as a server, so the Raspberry Pi needs to have a static IP address assigned to the wireless port. This documentation assumes that we are using the standard 192.168.x.x IP addresses for our wireless network, so we will assign the server the IP address 192.168.4.1. It is also assumed that the wireless device being used is wlan0.

To configure the static IP address, edit the dhcpd configuration file with:

```
sudo nano /etc/dhcpd.conf
```

Go to the end of the file and edit it so that it looks like the following:

```
interface wlan0

static ip_address=10.9.8.1/24
nohook wpa_supplicant
```

Now restart the dhcpd daemon and set up the new wlan0 configuration:

```
sudo service dhcpd restart
```

Configuring the DHCP server (dnsmasq)

The DHCP service is provided by dnsmasq. By default, the configuration file contains a lot of information that is not needed, and it is easier to start from scratch. Rename this configuration file, and edit a new one:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig sudo nano
/etc/dnsmasq.conf
```

Type or copy the following information into the dnsmasq configuration file and save it:

```
interface=wlan0

dhcp-range=10.9.8.50,10.9.8.254,255.255.255.0,24h
```

So for wlan0, we are going to provide IP addresses between 10.9.8.50 and 10.9.8.254 with a lease time of 24 hours. If you are providing DHCP services for other network devices (e.g. eth0), you could add more sections with the appropriate interface header, with the range of addresses you intend to provide to that interface.

There are many more options for dnsmasq; see the [dnsmasq documentation](#) for more details.

Configuring the access point host software (hostapd)

You need to edit the hostapd configuration file, located at `/etc/hostapd/hostapd.conf`, to add the various parameters for your wireless network. After initial install, this will be a new/empty file.

```
sudo nano /etc/hostapd/hostapd.conf
```

Add the information below to the configuration file. This configuration assumes we are using channel 7, with a network name of `NameOfNetwork`, and a password `AardvarkBadgerHedgehog`. Note that the name and password should not have quotes around them. The passphrase should be between 8 and 64 characters in length.

```
interface=wlan0
driver=nl80211
ssid=etherbox
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
```

To password protect the hotspot:

```
wpa=2
wpa_passphrase=AardvarkBadgerHedgehog
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

We now need to tell the system where to find this configuration file.

```
sudo nano /etc/default/hostapd
```

Find the line with `#DAEMON_CONF`, and replace it with this:

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

Start it up

Now start up the remaining services:

```
sudo systemctl start hostapd sudo systemctl start dnsmasq
```

Add routing and masquerade

Edit `/etc/sysctl.conf` and uncomment this line:

```
net.ipv4.ip_forward=1
```

Add a masquerade for outbound traffic on `eth0`:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Save the iptables rule.

```
sudo sh -c "iptables-save &gt; /etc/iptables.ipv4.nat"
```

Edit /etc/rc.local and add this just above "exit 0" to install these rules on boot.

```
iptables-restore < /etc/iptables.ipv4.nat
```

Reboot

Bring the rest of the software up to date

```
sudo apt-get update
sudo apt-get upgrade
```

Change hostname from raspberrypi to etherbox

In 2 places:

```
sudo nano /etc/hostname
sudo nano /etc/hosts
```

change to:

```
127.0.0.1        localhost
::1             localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

127.0.1.1       etherbox
```

Best to reboot after this, otherwise it gives warnings all the time.

```
sudo reboot
```

Assign Static Ip Adress to the Pi

For that, you need to modify a file named 'dhcpd.conf'. But first, you need to backup this file :

```
cd /etc/
sudo cp dhcpd.conf ddhcpd.conf.original
```

Then modify the 'dhcpd.conf' :

```
sudo nano dhcpd.conf
```

Then find this and replace (with your value) :

```
# Example static IP configuration:
interface eth0
```



```
static ip_address=192.168.1.5/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Where 192.168.1.5/24 is the new Ip address of your Pi.

At last, reboot your Pi :

```
sudo reboot
```

Setup apache to serve the root with custom header + readme 's

```
sudo apt-get install apache2
cd /etc/apache2/sites-available
sudo nano 000-default.conf
```

```
ServerAdmin webmaster@localhost
# DocumentRoot /var/www/html
DocumentRoot /home/pi/www
<Directory /home/pi/www>
    Options Indexes FollowSymLinks
    AllowOverride none
    Require all granted
</Directory>
```

```
HeaderName /include/HEADER.shtml
ReadmeName README.html
```

NB: Sets the [HeaderName](#) and [ReadmeName](#) directives (part of mod_autoindex).

```
sudo service apache2 reload
```

droptoupload.cgi

Drop to upload is a CGI python script that allows users to drop files to upload them in the apache directory listings.

First enable the cgi-bin with apache

```
sudo a2enmod cgi
sudo systemctl restart apache2
```

Download the script to the cgi-bin.

```
cd /usr/lib/cgi-bin
sudo wget https://gitlab.constantvzw.org/aa/etherbox/raw/master/usr/lib/cgi-bin/droptoupload.cgi
sudo chmod +x droptoupload.cgi
```

You can test running it with...

```
./droptoupload.cgi
```

Like this it just outputs an HTML form. Looking at <http://etherbox.local/cgi-bin/droptoupload.cgi> should also display an upload form.

The HEADER.shtml (next step) includes a link to this cgi.

/home/pi/include/HEADER.shtml

```
nano /home/pi/include/HEADER.shtml
```

(don't use sudo)

```
<script src="/cgi-bin/droptoupload.cgi"></script>
<style>
body {
background: #38b8e9;
color: black;
}
a {
color: white;
}
#logo {
white-space: pre;
font-family: monospace;
}
</style>
<div class="links" style="margin-bottom: 1em">LOCAL:
<a href="/">home</a>
<a href="/" onclick="javascript:event.target.port=9001">etherpad</a>
<a href="/etherdump/index2.html">etherdump</a>
PUBLIC:
<a href="http://erg.be/">erg</a>
<a href="http://erg.activearchives.org/mw/index.php/Etherbox">mode d'emploi</a>
</div>
<style>
.links {
font-family: monospace;
text-transform: uppercase;
</style>
<script>
document.addEventListener("DOMContentLoaded", function () {
  var p = document.querySelectorAll(".top"),
      t = document.getElementsByTagName("table")[0];
  for (var i=0, l=p.length; i<l; i++) {
    document.body.insertBefore(p[i], t);
  }
});
</script>
```

Better permissions with `facl`

[setfacl](#)

```
sudo apt install acl
sudo addgroup pi www-data

sudo setfacl -Rm g:www-data:rwX /home/pi/www
sudo setfacl -d -Rm g:www-data:rwX /home/pi/www
```

Install etherpad

And the version of "nodejs" is now 0.10.29~dfsg-2. So let's try it with etherpad...

```
sudo apt-get install npm git

sudo ln -s /usr/bin/nodejs /usr/bin/node

cd /opt
sudo git clone https://github.com/ether/etherpad-lite.git
sudo mv etherpad-lite etherpad

# TODO: don't create home folder! ... find option
sudo adduser --system --no-create-home --home=/opt/etherpad --group etherpad
sudo chown -R etherpad:etherpad etherpad
```

Create your setting file.

```
cd etherpad
sudo --user etherpad cp settings.json.template settings.json
```

then, deactivate the `dirty.db` file and configure the mysql database

```
sudo --user etherpad nano settings.json
```

Find and change this part :

```
//The Type of the database. You can choose between dirty, postgres, sqlite and mysql
//You shouldn't use "dirty" for for anything else than testing or development
/* "dbType" : "dirty",
//the database specific settings
"dbSettings" : {
    "filename" : "var/dirty.db"
},
*/

//An Example of MySQL Configuration
"dbType" : "mysql",
"dbSettings" : {
    "user"      : "etherpaduser",
    "host"      : "localhost",
    "password" : "etherpadpass",
```

```
        "database": "etherpad",
        "charset" : "utf8mb4"
    },
```

Set up Mysql

First thing to do is to install MySQL :

```
sudo apt-get install mysql-server
```

Then create the database, for this we need to login with the Root user (super-user) :

```
sudo su
```

Run mysql :

```
mysql
```

Then create the database and the user 'etherpaduser' with the password 'etherpadpass' :

```
create database etherpad;
grant all on etherpad.* to 'etherpaduser'@'localhost' identified by 'etherpadpass';
```

Use Ctrl-D to quit mysql. And exit to quit su.

Just to test if it works :

```
mysql -u etherpaduser -p etherpad
```

Run etherpad for the first time as the etherpad user...

```
cd /opt/etherpad
sudo --user etherpad bin/run.sh
```

The first time you run the etherpad software it takes a ****long**** time as it downloads related packages. It may also give quite some warnings, but hopefully no errors.

Setup etherpad to start as a service

```
sudo nano /etc/systemd/system/etherpad.service
```

```
[Unit]
Description=Etherpad-lite, the collaborative editor.
After=syslog.target network.target

[Service]
Type=simple
User=etherpad
Group=etherpad
WorkingDirectory=/opt/etherpad
ExecStart=/usr/bin/nodejs /opt/etherpad/node_modules/ep_etherpad-lite/node/server.js
Restart=always
```

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

After this, to start once...

```
sudo systemctl start etherpad
```

Check if it's working with:

```
sudo systemctl status etherpad
```

And finally automatically start on boot:

```
sudo systemctl enable etherpad
```

Change the welcome text

```
sudo --user etherpad nano settings.json
```

If you want to keep the pad private, add the `__NOPUBLISH__` tag into the body text

etherdump

Etherdump is a script that dumps all pads to different format text files. It's run periodically by a cron job to create a running archive of the etherpads.

Install deps:

```
sudo apt install python3-pip
sudo pip3 install python-dateutil jinja2 html5lib
```

Install from repo:

```
cd
mkdir Software
cd Software
git clone http://murtaugh@gitlab.constantvzw.org/aa/etherdump.git
cd etherdump
sudo python3 setup.py install
```

Init the folder

You need the etherpad API key (which is generated the first time you ran the software, see above). Copy it from:

```
cat /opt/etherpad/APIKEY.txt
```

Then...

```
cd /home/pi
mkdir etherdump
cd etherdump
etherdump init
```

For the URL use: <http://192.168.1.5:9001/>

And paste the API key.

Use pandoc to convert pads to HTML

Pandoc is used to convert markdown files to HTML.

```
sudo apt-get install pandoc
```

Create an image gallery with imagemagick

Imagemagick to make thumbnails.

```
sudo apt-get install imagemagick
```

cron.sh

```
nano /home/pi/cron.sh
```

```
# Dump the etherpad to files
cd etherdump
etherdump pull --meta --text --dhtml --pub . --no-raw-ext
etherdump index \
  *.meta.json \
  --templatepath /home/pi/include \
  --template etherdump.template.html \
  --title "Erg etherdump" > index2.html
# Run the makefile
cd /home/pi
make
```

Make it executable:

```
chmod +x /home/pi/cron.sh
```

makefile

The makefile defines the "recipe" that turns the pads (when they are named something.md) into HTML. The "%.html: %.md" is an "implicit rule" that defines how any ".md" file (right hand side, or pre-requisite, can be turned into a ".html" file (left hand side, or "target" in the language of the makefile. For this kind of rule the [special variables](#) can be really useful (like \$< and \$@).

This makefile uses the pandoc program to convert markdown to html. This program has LOTS of options and is quite powerful. Good documentation on the markdown format, and how pandoc supports it is: <http://pandoc.org/MANUAL.html#pandocs-markdown>

```
nano /home/pi/makefile
```

```
mdsrc=$(shell ls etherdump/*.md)
htmldest=$(mdsrc:%.md=%.html)
```

```

all: $(htmldest)
# $< is the right part input prereq    $@ is the target
%.html: %.md
    pandoc --from markdown \
        --standalone \
        --section-divs \
        --smart \
        --css styles.css \
        --to html5 \
        $< -o $@

```

Enable the cron job

```
crontab -e
```

The following runs the cron.sh **every minute**.

```

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin$

# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * /home/pi/cron.sh 2> /home/pi/cron.log

```

Create a reverse proxy for the etherpad

Creating a tunnel to give your local server a public address exposes only the (Apache) webserver (aka traffic on port 80). In order to make the etherpad visible and usable publically, it's possible to create a "reverse proxy" that creates access to the etherpad (on port 9001) via the webserver. To do this we create a special "virtual" location (/pad) and tell apache to redirect traffic to etherpad.

This recipe is based on information found [here](#).

```
sudo a2enmod proxy proxy_http headers proxy_wstunnel deflate rewrite
```

```

<VirtualHost *:80>
    DocumentRoot /home/pi/www/
    <Directory /home/pi/www>
        Options Indexes FollowSymLinks
        AllowOverride none
        Require all granted
    </Directory>

    HeaderName /include/HEADER.shtml
    ReadmeName README.html

    # ETHERPAD PROXY
    ProxyVia On
    ProxyRequests Off
    ProxyPreserveHost on
<Location /pad/>

```

```
ProxyPass http://localhost:9001/ retry=0 timeout=30
ProxyPassReverse http://localhost:9001/
</Location>
<Location /pad/socket.io>
    # This is needed to handle the websocket transport through the proxy, since
    # etherpad does not use a specific sub-folder, such as /ws/ to handle this kind of t
    # Taken from https://github.com/ether/etherpad-lite/issues/2318#issuecomment-635485
    # Thanks to beaugunderson for the semantics
    RewriteEngine On
    RewriteCond %{QUERY_STRING} transport=websocket [NC]
    RewriteRule /(.*) ws://localhost:9001/socket.io/$1 [P,L]
    ProxyPass http://localhost:9001/socket.io retry=0 timeout=30
    ProxyPassReverse http://localhost:9001/socket.io
</Location>

<Proxy *>
    Options FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
</Proxy>

</VirtualHost>
```


Makefile

```
all_index=$(shell ls *.index)
html_for_index=$(all_index:%.index=%.html)

all: index2.html $(html_for_index)

.PHONY: Makefile
Makefile:
    etherdump pull --meta --text --dhtml --pub . --no-raw-ext Makefile
    # fix tabs in the makefile
    sed -i -e 's/^  /\t/g' Makefile

.PHONY: dump
# Dump the etherpad to files
dump:
    etherdump pull --meta --text --dhtml --pub . --no-raw-ext
    # fix tabs in the makefile
    sed -i -e 's/^  /\t/g' Makefile

forcedump:
    etherdump pull --force --meta --text --dhtml --pub . --no-raw-ext
    # fix tabs in the makefile
    sed -i -e 's/^  /\t/g' Makefile

backup:
    eb_backup.sh

# make the compact (grouped) alternative file index
index2.html: dump
    etherdump index \
        *.meta.json \
        --templatepath /home/pi/www/include \
        --template etherdump.template.html \
        --title "Etherdump" > index2.html

# Generic rule to transform a markdown pad into HTML using pandoc
# $< is the right part input prereq    $@ is the target
#%.html: %.md
#     pandoc --from markdown \
#         --standalone \
#         --section-divs \
#         --smart \
#         --css style.css \
#         --to html5 \
#         $< -o $@

# Generic rule for turning foo.index -> foo.html
%.html: %.index dump
    includeurls.py < $< --baseurl http://etherbox.local:9001/p/ | \
```

```

pandoc --from markdown \
--table-of-contents \
--to=html5 \
--section-divs \
--standalone \
| \
html5tidy \
--script script.js \
--stylesheet_screen style.css \
--stylesheet_print print.css \
| \
localizeimages.py \
> $@

%.pdf: %.html
    html_images_for_print.py --input $< | \
    weasyprint - $@

# PDF to Postscript
%.m4.ps: %.m4.pdf
    pdftops -paper match $< $@

# Postscript to PDF
%.pdf: %.ps
    ps2pdf $< $@

# Ensure a PDF has a multiple of 4 pages
%.m4.pdf: %.pdf
    pdftk_utils.py $< pad --multiple 4 --output $@

%.book.ps: %.m4.ps %.m4.pdf
    psbook -s`pdftk_utils.py $*.m4.pdf count` $< $@

%.booklet.ps: %.book.ps
    psnup -2 -PA4 $< $@

```

pdftk_utils.py

```
#!/usr/bin/env python3
```

```
import subprocess, re, shutil
```

```
"""
```

```
usage:
```

```
count pages
```

```
    pdftk_utils.py test.pdf count
```

```
pad pdf to a multiple of a given number of pages (uses make_blank_pdf.py/reportlab)
```

```
    pdftk_utils.py test.pdf pad --multiple 4 --output padded.pdf
```

```
"""
```

```
def count_pages (path):
```

```
    p = subprocess.Popen(["pdftk", path, "dump_data"], stderr=subprocess.STDOUT,  
                          stdout=subprocess.PIPE)
```

```
    out, _ = p.communicate()
```

```
    out = out.decode("utf-8")
```

```
    m = re.search("^NumberOfPages: (\d+)", out, flags=re.M)
```

```
    return int(m.group(1))
```

```
def pad_to_multiple (path, multiple, output, alwaysOutput=True):
```

```
    import make_blank_pdf, tempfile, os
```

```
    n = count_pages(path)
```

```
    cur_pages = n
```

```
    while n % multiple != 0:
```

```
        n += 1
```

```
    pages_to_add = n - cur_pages
```

```
    if pages_to_add > 0:
```

```
        _, tmp = tempfile.mkstemp(".pdf")
```

```
        make_blank_pdf.make_blank_pdf(tmp, pages_to_add)
```

```
        # print ("pdftk A=\"{0}\" B=\"{1}\" cat A B output {2}".format(path, tmp, args.output))
```

```
        p = subprocess.Popen(["pdftk", "A={0}".format(path), "B={0}".format(tmp),
```

```
                              "cat", "A", "B", "output", args.output],
```

```
                              stderr=subprocess.STDOUT, stdout=subprocess.PIPE)
```

```
        out, _ = p.communicate()
```

```
        out = out.decode("utf-8")
```

```
        os.remove(tmp)
```

```
        return output
```

```
    elif alwaysOutput:
```

```
        # in the event of no need for padding, simply copy the input file
```

```
        shutil.copyfile(path, output)
```

```
        return output
```

```
if __name__ == "__main__":
```

```
    import argparse
```

```
ap = argparse.ArgumentParser("")
ap.add_argument("path")
subparsers = ap.add_subparsers()
count = subparsers.add_parser("count")
count.set_defaults(func=count_pages)

pad = subparsers.add_parser("pad")
pad.add_argument("--multiple", type=int, default=None)
pad.add_argument("--output", default="padded.pdf")
pad.set_defaults(func=pad_to_multiple)

args = ap.parse_args()
oargs = dict(vars(args))
del oargs['func']
print (args.func(**oargs))
```

Colophon

This issue was compiled by: Michael Murtaugh, An Mertens, Roel Roscam Abbing, Femke Snelting

Thank you:

Anne Laforet Homebrew Server Club OSP Varia xpub Gijs De Heij Eric Schrijver Martino Morandi Relearn Stephanie Vilayphiou Alexandre Leray Alexia De Visser Aymeric Mansoux Yogi Christoph Haag Catherine Lenoble xpp1 Pieter Heiremans Hackerspace Brussels Denis Devos

Layout: Roel Roscam Abbing

Images: Michael Murtaugh, Donatella Portoghese, An Mertens, Peter Westenberg, Roel Roscam Abbing

Published by: Constant, Association for Art and Media, Brussels

Sources: <http://networksofonesown.constantvzw.org>

License: Copyleft Constant, September 2018. This material can be used, copied, changed and distributed under the conditions of the [Free Art License](#).

Notes

1. See: Paranodes in *Contra-Internet*, Zach Blas
<https://www.e-flux.com/journal/74/59816/contra-internet/>↵
2. The initiative for this periodic publication is currently located in-between the following nodes: Constant, association for art and media (Brussels), [Relearn](#), Varia a space for developing collective approaches to everyday technology (Rotterdam) and XPUB or experimental publishing, a study path of the Piet Zwart Institute's Media Design Master programme (Rotterdam).
<http://constantvzw.org>
<http://varia.zone>
<https://xpub.nl/>↵
3. Read A Room Of One's Own via project Gutenberg.
<http://gutenberg.net.au/ebooks02/0200791.txt>↵
4. Also, TMTOWTDI or TIMTOWTDI.
<http://wiki.c2.com/?ThereIsMoreThanOneWayToDoIt>↵
5. An organisation for art and media based in Brussels.↵
6. Do It With Others↵
7. Read more about using makefiles for publishing workflows:
<http://networkcultures.org/digitalpublishing/2014/10/01/make-book/>↵
8. From <http://osp.kitchen/tools/ethertoff/>↵
9. <http://conversations.tools/>↵
10. (<http://osp.kitchen/tools/html2print/>↵
11. <http://osp.kitchen/>↵
12. <http://relearn.be/2015/etherdump/Linux.html>↵
13. <http://areyoubeingserved.constantvzw.org/>↵
14. <http://www.lgru.net/>↵
15. <http://lgru.net/archives/5359>↵
16. http://digitalcraft.wdka.nl/wiki/Living_in_a_Sandbox↵
17. “a system around which a fellowship could form.” From Aymeric's PHD: Dennis M. Ritchie, “The Evolution of the Unix Time-Sharing System,” in *Language Design and Programming Methodology : Proceedings of a Symposium Held at Sydney, Australia, 10-11 September 1979*, ed. Jeffrey M. Tobias (New York: Springer-Verlag, 1980), 25.↵
18. https://www.bleu255.com/~aymeric/dump/aymeric_mansoux-sandbox_culture_phd_thesis-2017.pdf↵

19. <https://gitlab.com/communitymemories/www>
20. <http://www.algolit.net>
21. <https://piratebox.cc/>
22. <http://librarybox.us/>
23. <https://openwrt.org/>
24. <https://www.radical-openness.org/en/programm/2014/unitary-networking>
25. <https://www.homebrewserver.club/>
26. <http://networkcultures.org/digitalpublishing/>
27. <https://www.raspberrypi.org/>
28. https://en.wikipedia.org/wiki/Dd_%28Unix%29
29. *A network with an attitude*, quoted from Relearn 2015 documentation <http://relearn.be/2015/etherdump/Network.html>
30. *A network with an attitude*, quoted from Relearn 2015 documentation <http://relearn.be/2015/etherdump/Network.html>
31. *Are You Being Served, A Feminist Server Manifesto*
http://areyoubeingserved.constantvzw.org/Summit_afterlife.xhtml
32. *Relearn 2014* homepage
<http://relearn.be/2014/>