

- [Intro](#)
 - [Consideration](#)
 - [Goals](#)
 - [Architecture](#)
- [Requirements](#)
- [Setup](#)
 - [How to prepare the SD card with Ubuntu 22.04](#)
 - [Boot Ubuntu](#)
 - [Important notes](#)
 - [Hardware setup](#)
 - [LCD Display 4.3inch_DSI_LCD](#)
 - [Cooling solution](#)
 - [Attach the FAN](#)
 - [Installation](#)
 - [Preparation](#)
 - [Enable every user to start](#)
 - [On the new version try with the snap version](#)
 - [Install chromium](#)
 - [Install Jitsi](#)
 - [Audio Setup - To be confirmed](#)
 - [Jitsi setup](#)
 - [Jicofo Setup](#)
 - [References](#)
 - [Jitsi customization](#)
 - [Recompile jniwrapper-native-1.0-SNAPSHOT - obsolete](#)
 - [SSL Certificate](#)
 - [SSL Certificate creation](#)
 - [Nginx SSL Certificate configuration](#)
 - [Jitsi SSL Certificate configuration](#)
 - [Enable the NGINX config](#)
 - [Prosody setup](#)
 - [Pulseaudio setup](#)
 - [References](#)
 - [Audio Recording](#)
 - [Asound conf - Is this used in the last pulse version?](#)
 - [Create the output sink called recording](#)
 - [Attach the Mic to the sink](#)
 - [Test the recording with PulseAudio](#)
 - [Audio Recording References](#)
 - [Jupyter setup to record the audio with Python](#)
 - [Network setup](#)
 - [Backend setup](#)
 - [Node Js](#)
 - [Frontend setup](#)
 - [GUI](#)
 - [GUI Tuning](#)

- VNC Setup
 - VNC config
 - Stress test
- Challenge
- Best practices
- Disable the screen dimming - RO CONFIG
- RO FS
 - References
- Monitor the temperature
- Disable not useful services
 - Auto update
 - Snap
 - Remove snapd completely
 - Cleanup Snap
- Removes old revisions of snaps
- Backup
- Todo
 - Done
- References

Note: This project has been done with the help on GenAI.

Intro

This is the next version of the Translation System. With the legacy system based on Icecast and Darkice we got good the results in terms of easiness of implementation and usability but the latency (1-3 secs) is not anymore acceptable.

We are now aiming to have short latency between the original voice and the translation.

The goal is to have a translator hearing the original audio and all the other listening it without any video/comments/or other interruptions allowed.

The software choosen is Jitsi were we have very short delays.

Consideration

Running Jitsi on a Raspberry PI can looks trivial, and installing the software it is but the goal is to make the translator independent by any other device and have the meeting in control. The translator is not a technical person and therefore the interaction with the Raspberry PI must be as easy as possible like as well the user part.

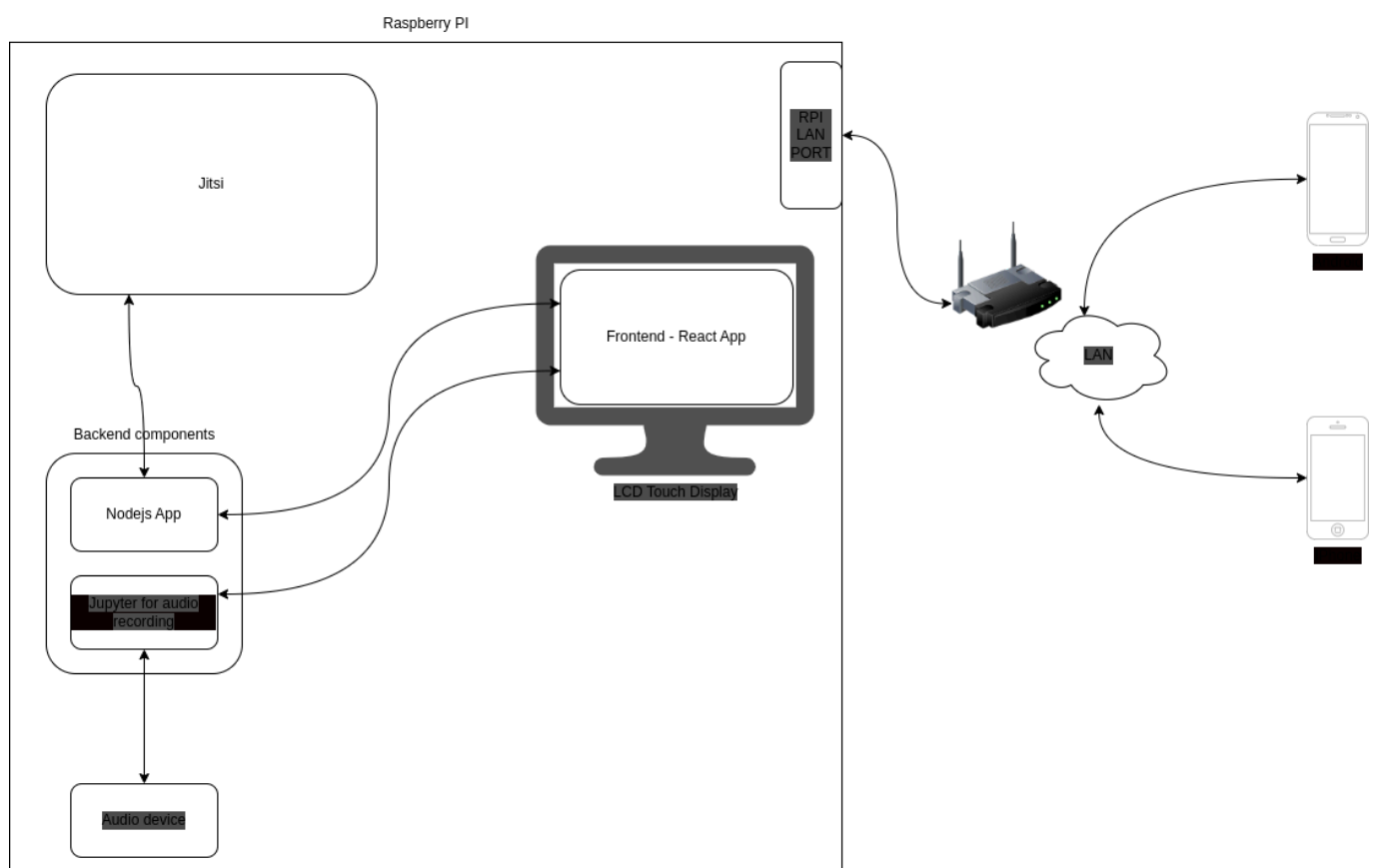
Jitsi has per se some hard requirements to work properly. The first one is the SSL certificate. The system is not working without a valid SSL certificate. The second one is the audio. The audio must be properly configured to work properly.

This system can be used for very high confidential meetings because the traffic is not leaving the LAN.

Goals

- close to zero latency
- the system can work also in a not internet connected network
- the traffic is every case not leaving the LAN
- the translator is autonomous in starting the meeting
- he can control the meeting
 - rejoin
 - restart
 - mute all
 - check the participants counts
 - poweroff
- *the translated audio can be recorded - optional*

Architecture



The chosen OS is Ubuntu for Raspberry PI 4. The reason is that the Raspberry PI OS is not supporting the latest version of Jitsi and the Ubuntu is the only one that is supporting it.

Requirements

- Raspberry PI 4 (2GB) or higher - With the 4GB version you would need less tuning for Jitsi
- An USB microphone adapter - Write the model here
- Microphone
- A valid SSL certificate

Setup

How to prepare the SD card with Ubuntu 22.04

Download the Ubuntu 22.04 Server image from [Ubuntu](#). Identify the disk with the SD card using lsblk or df to see the mounted device. Be sure to umount before flash it.

Flash it on the SD card with the following commands: SD_DEVICE="PUT YOUR VALUE HERE" # I.E: /dev/sde

```
xz -d ubuntu-22.04.2-preinstalled-server-arm64+raspi.img.xz
sudo dd bs=4M if=./ubuntu-22.04.2-preinstalled-server-arm64+raspi.img
of=$SD_DEVICE conv=fdatasync status=progress
```

Or use the Raspberry PI Imager*

```
sudo apt install rpi-imager
rpi-imager
```

Boot Ubuntu

We still need some manual operation to configure the system. The first step is to boot the Raspberry PI and configure the network attaching the RP to a monitor and a keyboard.

- Login with ubuntu/ubuntu
- Configure the network with netplan
- Enable and start the ssh service

Can we automate this step?

Important notes

PORT 8888 for jicofo

If the port is busy everything seems to start but the participants cannot see each others! [issue-6449](#)

Hardware setup

LCD Display 4.3inch_DSI_LCD

https://www.waveshare.com/wiki/4.3inch_DSI_LCD

Cooling solution

<https://www.arrow.com/en/research-and-events/articles/raspberry-pi-4-cooling-solutions-comparison>

<https://all3dp.com/2/raspberry-pi-4-heatsink-placement/>

With 2 small heatsync and a small fan the temperature is always below 50C.

Attach the FAN

PIN 4 - 5V - Red cable PIN 6 - GND - Black cable

Pin1 is in the opposite from the USB ports on the internal side of the board. For more details see <https://www.raspberrypi.org/documentation/usage/gpio/>

see <https://linuxhint.com/gpio-pinout-raspberry-pi/>

Installation

The full installation and configuration process is now done via Ansible. The Ansible playbook is in the folder IaC.

The playbook is divided in 3 different roles:

Role Name	Included Tasks	Goal
base_infra	required_variables.yml	Check that all required variable are set
	netplan_config.yml	Configure the network
	default_gw.yml	Setup the needed Iptables rules
	install_and_configure_software.yml	Install and configure all the required base software
	dnsmasq.yml	Setup dnsmasq and DNS servers
	configure_jitsi.yml	Configure Jitsi
	enable_jitsi_services.yml	Enable all the Jitsi server
	gemeinde_backend	Start the npm backend server
translator_user		Configure the translator user where the React app will run

Now the installation is done via Ansible. The Ansible playbook is in the folder IaC.

Preparation

1. Copy the public key of the user that will run the playbook in the Raspberry PI enduser home folder
`~/.ssh/authorized_keys`
2. Copy the file secrets.env.sample in secrets.env and fill it with the correct values
3. Source the secrets.env file
 - `source secrets.env`

```
. secrets.env
ansible-playbook -i hosts full_setup.yml
```

The default configuration is setting by default 2 different translation users for 2 different languages.

You can customize the configuration by changing the file `IaC/full_setup.yml`.

To test the single roles specific yml files have been created:

- base_infra.yml
- backend_toile.yml
- frontend_toile.yml

► To see all the configurations done check the details here of look into the IaC folder

```
sudo apt install openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless openjdk-8-jdk tightvncserver xinit xserver-xorg gnome-tweaks pavucontrol xserver-xorg-legacy
sudo apt-get install default-jre-headless ffmpeg curl alsa-utils icewm xdotool xserver-xorg-video-dummy ruby-hocon build-essention libtool maven git
```

Enable every user to start

```
sudo vi /etc/X11/Xwrapper.config
allowed_users=anybody
```

Select Java 8

```
update-alternatives --config java # Select jdk 11 # openjdk-8 doesn't work anymore.
```

Chrome installation without snap

Chrome is not compiled anymore for RaspPI therefore we are using Chromium. See [3], [4]

[3]: <https://askubuntu.com/questions/1204571/how-to-install-chromium-without-snap>

[4]: <https://packages.debian.org/bullseye/arm64/apt/download>

The default Chromium version is a snap package and it doesn't suit my needs because I need to customize some aspects of the browser.

write the file /etc/apt/sources.list.d/debian.list

```
```bash
```

```
cat <<EOF > /etc/apt/sources.list.d/debian.list
deb [arch=arm64] http://ftp.ch.debian.org/debian bullseye main
deb [arch=arm64] http://ftp.ch.debian.org/debian bullseye-updates main
EOF
```

## On the new version try with the snap version

apt install chromium-browser/jammy-updates chromium-chromedriver/jammy-updates

On ubuntu 22.04

```
```bash
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
0E98404D386FA1D9 6ED0E7B82643E131 648ACFD622F3D138 605C66F00D6C9793
```

```
cat <<EOF >/etc/apt/preferences.d/chromium.pref
# Note: 2 blank lines are required between entries
Package: *
Pin: release a=eoan
Pin-Priority: 500

Package: *
Pin: origin "deb.debian.org"
Pin-Priority: 300

# Pattern includes 'chromium', 'chromium-browser' and similarly
# named dependencies:
Package: chromium*
Pin: origin "deb.debian.org"
Pin-Priority: 700
EOF
```

Install chromium

```
apt update && apt install chromium/oldstable chromium-driver/oldstable
```

To avoid the warning message about the security of the browser we need to disable the security warning.

```
mkdir -p /etc/opt/chrome/policies/managed
echo '{ "CommandLineFlagSecurityWarningsEnabled": false }'
>>/etc/opt/chrome/policies/managed/managed_policies.json
```

<https://www.linuxcapable.com/how-to-install-chromium-browser-on-ubuntu-22-04-lts/>

I need a plain installation to customize it. True?

```
wget https://launchpad.net/~xtradeb/+archive/ubuntu/apps/+files/xtradeb-apps-apt-source_0.3_all.deb
apt install ./xtradeb-apps-apt-source_0.3_all.deb
```

Install Jitsi

```
curl -sL https://download.jitsi.org/jitsi-key.gpg.key | sudo sh -c 'gpg --dearmor > /usr/share/keyrings/jitsi-keyring.gpg'
echo "deb [signed-by=/usr/share/keyrings/jitsi-keyring.gpg] https://download.jitsi.org stable/" | sudo tee /etc/apt/sources.list.d/jitsi-stable.list

sudo apt install jitsi-meet jitsi-meet-prosody jitsi-videobridge2 jitsi-meet-web
```

Set a domain where you have a valid SSL certificate. In our case translation.sennsolutions.com

Update SSL Cert. See file `update_cert.sh`

Audio Setup - To be confirmed

To work properly Jitsi need the aloop module. It can be found in the package [linux-modules-extra-5.15.0-1015-raspi](#)

```
apt install linux-modules-extra-5.15.0-1015-raspi
rm /lib/systemd/system/alsa-utils.service ; systemctl daemon-reload
apt install unzip ffmpeg curl alsa-utils icewm xdotool xserver-xorg-video-dummy # Why this?
```

Jitsi setup

Limit the memory usage Edit the file `/etc/jitsi/videobridge/config` and add the following line:

```
VIDEOBRIDGE_MAX_MEMORY=1024m
```

And restart the videobridge.

Edit the file `/etc/jitsi/jicofo/config` , add the following line:

```
JICOFO_MAX_MEMORY=1024m
```


And restart Jicofo.

Jicofo Setup

During the development I experimented a very weird behaviour. Jitsi was working properly at home but not in its destination place. The error message that I got was "No bridge available". I do not yet understand why this happened but I found a workaround. I've added the following line in the file

`/etc/jitsi/jicofo/sip-communicator.properties`

```
org.jitsi.jicofo.HEALTH_CHECK_INTERVAL=-1
```

This actually disables the health check and the system is working properly. I need to investigate more on this.

References

<https://community.jitsi.org/t/can-not-invite-participant-no-bridge-available-since-last-update-debian-10/63046>

Jitsi customization

Edit the file `/etc/jitsi/meet/translation.sennsolutions.com-config.js`

Recompile jniwrapper-native-1.0-SNAPSHOT - obsolete

Not anymore needed with the latest version of Jitsi. The `jniwrapper-native-1.0-SNAPSHOT.jar` is already compiled for ARM64.

► Click to expand!

```
git clone https://github.com/sctplab/usrctp.git
git clone https://github.com/jitsi/jitsi-sctp
mv ./usrctp ./jitsi-sctp/usrctp/
cd ./jitsi-sctp
mvn package -DbuildSctp -DbuildNativeWrapper -DdeployNewJnilib -DskipTests
```

Copy libjnisctp.so

```
cp ./jniwrapper/native/target/libjnisctp-linux-aarch64.so \
./jniwrapper/native/src/main/resources/lib/linux/libjnisctp.so
```

Re-package and Copy jniwrapper-native-1.0-SNAPSHOT.jar into Jitsi VideoBridge2

When running `mvn package` ensure all unit tests are successful. You will see some warnings about "Using platform encoding", but that's fine because we're building on the platform that we intend to run this on anyway.

```
mvn package
sudo cp ./jniwrapper/native/target/jniwrapper-native-1.0-SNAPSHOT.jar \
/usr/share/jitsi-videobridge/lib/jniwrapper-native-1.0-SNAPSHOT.jar
```

```
sudo usermod -aG adm,audio,video,plugdev jibri
```

SSL Certificate

SSL Certificate creation

See the document [letsencrypt.md](#)

Nginx SSL Certificate configuration

New file /etc/nginx/sites-available/translation.sennsolution.com.conf with the new domain

Jitsi SSL Certificate configuration

Cert copied in Jitsi folder following the one configured in the nginx config file

```
scp ~/letsencrypt/archive/translation.sennsolutions.com/fullchain1.pem
pi@translation.home.local:/tmp/ # from bigone
mv /tmp/fullchain1.pem /etc/ssl/translation.sennsolutions.com.crt
# on translation

scp ~/letsencrypt/archive/translation.sennsolutions.com/privkey1.pem
pi@translation.home.local:/tmp/ # from bigone
mv /tmp/privkey1.pem /etc/ssl/translation.sennsolutions.com.key
# on translation

# Create a new Jitsi config with all the domain changed
vi /etc/jitsi/meet/translation.sennsolutions.com-config.js

#Update all Jitsi config
cd /etc/jitsi
sed -i=bck 's/home\.local/sennsolutions\.com/' jibri/jibri.conf
jicofo/config jicofo/jicofo.conf videobridge/sip-communicator.properties
videobridge/config videobridge/jvb.conf

#Update Prosodi
cd /etc/prosody
sed': sed -i.bck
's/translation\.home\.local/translation\.sennsolutions\.com/g'
prosody.cfg.lua
cd conf.avail
create a file /etc/prosody/conf.avail/translation.sennsolutions.com.cfg.lua
with all the domain changed
sed': sed -i.bck
's/translation\.home\.local/translation\.sennsolutions\.com/g'
```

```
translation.sensolutions.com.cfg.lua
cd ../conf.d
ln -s /etc/prosody/conf.avail/translation.sensolutions.com.cfg.lua

## create the prosody cert
# This certificate is for Prosody.
# It expires on Jan 14 2024 - Can We make it longer?
# Whats happens if it expires?
cd /etc/prosody/certs
make translation.sensolutions.com.cnf
make translation.sensolutions.com.key
make translation.sensolutions.com.cnf.csr
make translation.sensolutions.com.cnf.crt
mv translation.sensolutions.com.cnf.crt translation.sensolutions.com.crt
mv translation.sensolutions.com.cnf.key translation.sensolutions.com.key
mv translation.sensolutions.com.cnf.csr translation.sensolutions.com.csr

# Restart the Jitsi services to apply the new config
```

Enable the NGINX config

```
cd /etc/nginx/sites-enabled
ln -s /etc/nginx/sites-available/translation.sensolution.com.conf
rm translation.home.local.conf
systemctl restart nginx
```

Prosody setup

```
prosodyctl register jibri auth.translation.home.local jibriauthpass
prosodyctl register recorder recorder.translation.home.local
jibrirecorderpass

ln -s /etc/opt/chrome/policies/ /etc/chromium/ # config for chrome and
Chromium are stored in different places
```

Pulseaudio setup

The current setup is using PulseAudio to manage the audio. The unique Pulseaudio server is running within the main (lang1) user. The other users are using the Pulseaudio server of the main user. This is done to avoid the problem of the audio device that can be used only by one user at time.

The Pulseaudio server is started by the user lang1 and it is configured to start automatically at boot. The lang2 point to the network Pulseaudio server via the config `default-server = unix:/tmp/pulse-socket` in the file `/home/lang2/.config/pulse/client.conf`

The default device is set in the `.xinitrc` file with the command `pacmd set-default-source pacmd set-default-source alsa_input.usb-C-Media_Electronics_Inc._USB_Audio_Device-`

00.mono-fallback.2

While chromium on the lang2, is started with: `PULSE_SOURCE="alsa_input.usb-C-Media_Electronics_Inc._USB_Audio_Device-00.mono-fallback" /usr/bin/chromium --app=http://localhost:3001`

References

<https://askubuntu.com/questions/14077/how-can-i-change-the-default-audio-device-from-command-line>

https://wiki.archlinux.org/title/PulseAudio/Examples#Set_default_input_sources

<https://askubuntu.com/questions/71863/how-to-change-pulseaudio-sink-with-pacmd-set-default-sink-during-playback/72076#72076>

<https://shallowsky.com/linux/pulseaudio-command-line.html>

<https://copyprogramming.com/howto/change-pulseaudio-input-output-from-shell>

Audio Recording

To record the audio Jibri should be used but I couldn't manage to make it working. So I am recording the audio with Python intercepting the Audio device.

Asound conf - Is this used in the last pulse version?

Edit the file `/etc/asound.conf` `pcm.trans { type dsnoop ipc_key 5678293 ipc_perm 0666 ipc_gid audio slave { pcm "hw:2,0" channels 2 #period_size 1024 #buffer_size 4096 #rate 48000 #periods 0 #period_time 0 } }`

Create the output sink called recording

```
pacmd load-module module-null-sink sink_name=recording
sink_properties=device.description=recording

pacmd list-sinks | egrep '^s+name: .*'
      name: <alsa_output.usb-
Creative_Technology_Ltd_Sound_Blaster_Play__3_00252407-00.analog-stereo>
      name: <alsa_output.platform-bcm2835_audio.stereo-fallback>
      name: <alsa_output.platform-bcm2835_audio.stereo-fallback.2>
```

The first one is the USB Mic

```
pacmd load-module module-combine-sink sink_name=combined
sink_properties=device.description=combined
slaves=recording,alsa_output.usb-
Creative_Technology_Ltd_Sound_Blaster_Play__3_00252407-00.analog-stereo
```

Attach the Mic to the sink

```
pacmd list-sources | egrep '^s+name: .*' | grep input
      name: <alsa_input.usb-
Creative_Technology_Ltd_Sound_Blaster_Play__3_00252407-00.analog-stereo>

pacmd load-module module-loopback source=alsa_input.usb-
Creative_Technology_Ltd_Sound_Blaster_Play__3_00252407-00.analog-stereo
sink=recording latency_msec=1
```

Test the recording with PulseAudio

```
parecord --channels=2 -d recording.monitor output.wav
```

Audio Recording References

- <http://mocha.freeshell.org/audio.html>
- <https://raw.githubusercontent.com/larsimmisch/pyalsaaudio/main/recordtest.py>
- <https://unix.stackexchange.com/questions/361594/device-or-resource-busy-error-thrown-when-trying-to-record-audio-using-arecord>
- <https://bbs.archlinux.org/viewtopic.php?id=187258>
- <https://alsa.opensrc.org/DmixPlugin>
- <https://alsa.opensrc.org/Dsnoop>
- <http://larsimmisch.github.io/pyalsaaudio/libalsaaudio.html#examples>

Jupyter setup to record the audio with Python

To start the python recording script I've used Jupyter Notebook with a different port:

```
jupyter notebook --ip 0.0.0.0 --port 8889
```

A problem that I found, is the difficulty to record the audio while the resource is used by another software. See [gist](#) for more details.

TOFIX: Add the script to start the recording and or attach the Jupyter notebook

Network setup

Backend setup

Node Js

Frontend setup

GUI

The small LCD display doesn't allow to have a full GUI. The idea is to have a small display with some buttons to control the meeting. To be able to control the meeting we need to have a way to interact with the Jitsi

server. The idea is to use the Jitsi API to control the meeting. However a GUI is needed and we need to start XWindows to have it with a browser.

Chrome is not anymore available and therefore we are using Chromium with a React App. Chromium is started in a VNC session so that the system can be easily debugged/controlled remotely.

To start Jitsi as moderator you need to be the first to join the meeting. The first user is the moderator and can control the meeting. The moderator can mute all the users and can also unmute a single user.

The preferred way to start Jitsi is with

```
https://translation.home.local/english#config.startWithAudioMuted=false&config.startWithVideoMuted=true
```

GUI Tuning

To enhance the audio recording level I've started pavucontrol and set the Recording device level to 130%.

VNC Setup

Vncserver is started via systemd with the follow config:

```
/etc/systemd/system/vncserver.service
[Unit]
Description=TightVNC server
After=syslog.target network.target

[Service]
Type=forking
User=pi
#PAMName=login
PIDFile=/home/pi/.vnc/%H:1.pid
ExecStartPre=-/usr/bin/vncserver -kill :1 > /dev/null 2>&1
ExecStart=/usr/bin/vncserver
ExecStop=/usr/bin/vncserver -kill :1 ; pkill parecord ; pkill pulseaudio

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

VNC config

TOFIX: Is the file /home/pi/.xsessionrc relevant?

```
.xsessionrc
# Works with vncserver
set -x
exec pulseaudio -v --start -D &
exec icewm --replace &
```

```
exec pacmd load-module module-null-sink sink_name=recording
sink_properties=device.description=recording && \
pacmd load-module module-combine-sink sink_name=combined
sink_properties=device.description=combined
slaves=recording,alsa_output.usb-
Creative_Technology_Ltd_Sound_Blaster_Play__3_00252407-00.analog-stereo && \
pacmd load-module module-loopback source=alsa_input.usb-
Creative_Technology_Ltd_Sound_Blaster_Play__3_00252407-00.analog-stereo
sink=recording &
exec chromium https://translation.home.local/english &
exec parecord --channels=2 -d recording.monitor /home/pi/output.wav &
```

Stress test

To verify the Jitsi setup I wrote 2 small script to verify the system underload. Create a venv and install the requirements.txt

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

Run it it with

```
python start_selenium_chrome.py
```

The Chrome version is using an Xvfb session to run the browser while the firefox version is opening real windows in the running XWindows session.

The default test will test the jitsi server <https://translation.sennsolutions.com/ukr> and will try to join the meeting with 30 different users.

Check the script to customize the test.

Challenge

Best practices

- Best recording with the Mic at 5-10cm far from the mouth

Disable the screen dimming - RO CONFIG

```
cat > /home/pi/.xprofile <<EOF
xset s off
xset s noblank
EOF

#Check with xset -q
Screen Saver:
  prefer blanking:  no    allow exposures:  yes
  timeout:  0    cycle:  600
```

<https://bbs.archlinux.org/viewtopic.php?id=151590>

RO FS

Before

Filesystem	1K-blocks	Used	Available	Use%	Mounted on	
tmpfs	188908	3248	185660	2%	/run	
/dev/mmcblk0p2	14975908	9055096	5259600	64%	/	
tmpfs	944532	25292	919240	3%	/dev/shm	
tmpfs	5120	4	5116	1%	/run/lock	
/dev/mmcblk0p1	258095	151512	106583	59%	/boot/firmware	<---
THIS MUST STAY AS IS						
tmpfs	188904	4	188900	1%	/run/user/1000	

vi /etc/overlayroot.conf overlayroot="tmpfs"

Be sure that the /boot/firmware/cmdline.txt is not using the root=overlay **reboot**

After

Filesystem	1K-blocks	Used	Available	Use%	Mounted on	
tmpfs	188908	3248	185660	2%	/run	
/dev/mmcblk0p2	14975908	9048440	5266256	64%	/media/root-ro	<-- THIS
IS THE REAL RO ROOT						
tmpfs-root	944532	112092	832440	12%	/media/root-rw	
overlayroot	944532	112092	832440	12%	/	<--- THIS
IS THE RW OVERLAY ROOT						
tmpfs	944532	6868	937664	1%	/dev/shm	
tmpfs	5120	4	5116	1%	/run/lock	
/dev/mmcblk0p1	258095	151512	106583	59%	/boot/firmware	<--- THIS
IS LIKE BEFORE						
tmpfs	188904	4	188900	1%	/run/user/1000	

sudo overlayroot-chroot to go in a RW view

OR

To disable add `overlayroot=disabled` to the file `/boot/firmware/cmdline.txt` and reboot

See <https://spin.atomicobject.com/2015/03/10/protecting-ubuntu-root-filesystem/>

It looks like that the memory isn't limited to half of the available memory. Therefore if anything is writing too much to the FS the system will fail because of the lack of memory.

To limit the memory you can do on runtime

Added in the `/home/pi/.vnc/xstartup`

```
```bash
mount -o remount,size=400M /media/root-rw/
```

Unfortunately the tmpfs doesn't solve the problem because when the tmpfs got filled Chrome dies.

Taking a different path disabling all the useless logging and writing to the FS.

- disable rsyslogd `systemctl stop rsyslog.service`
- disable dnsmasq logging in the file `/etc/dnsmasq.d/01-pihole.conf` comment the lines: `#log-queries`  
`#log-facility=/var/log/pihole/pihole.log`
- Auto `/var/log` clean up `/usr/local/sbin/cleanup_fs.sh`

## References

<https://www.golinuxcloud.com/change-tmpfs-partition-size-redhat-linux/> <https://www.unix.com/shell-programming-and-scripting/260398-unix-script-cleanup.html>

## Monitor the temperature

---

```
#!/usr/bin/env bash

DELAY=${1:-5}
while true; do
 temp=$(vcgencmd measure_temp)
 echo "$(date): $temp"
 sleep $DELAY
done
```

Configured in `/usr/local/bin/check_temp.sh`

# Disable not useful services

## Auto update

Edit the file `/etc/apt/apt.conf.d/20auto-upgrades` like this:

```
APT::Periodic::Update-Package-Lists "0";
APT::Periodic::Download-Upgradeable-Packages "0";
APT::Periodic::AutocleanInterval "0";
APT::Periodic::Unattended-Upgrade "0";
```

## Snap

```
systemctl disable snapd.service
systemctl disable snapd.socket
systemctl disable snapd.seeded.service
```

<https://www.brsmedia.in/how-to-disable-snap-on-ubuntu-22-04/>

## Remove snapd completely

```
Name Version Rev Tracking Publisher Notes bare 1.0 5 latest/stable canonical✓ base chromium
116.0.5845.110 2603 latest/stable canonical✓ - core20 20230622 1977 latest/stable canonical✓ base
core22 20230801 867 latest/stable canonical✓ base cups 2.4.6-4 981 latest/stable openprinting✓ - gnome-
3-38-2004 0+git.efb213a 145 latest/stable canonical✓ - gnome-42-2204 0+git.ff35a85 127 latest/stable
canonical✓ - gtk-common-themes 0.1-81-g442e511 1535 latest/stable canonical✓ - lxd 5.0.2-838e1b2
24326 5.0/stable/... canonical✓ - snapd 2.59.5 19459 latest/stable canonical✓ snapd
```

```
sudo apt purge snapd
```

```
apport-symptoms avahi-utils blt cups-pk-helper distro-info geoclue-2.0
gir1.2-atk-1.0 gir1.2-freedesktop gir1.2-gdesktopenums-3.0 gir1.2-
gdkpixbuf-2.0
gir1.2-gnomedesktop-3.0 gir1.2-gtk-3.0 gir1.2-handy-1 gir1.2-harfbuzz-0.0
gir1.2-notify-0.7 gir1.2-pango-1.0 gir1.2-polkit-1.0 gir1.2-soup-2.4
gnome-desktop3-data gnome-settings-daemon gnome-settings-daemon-common
gnome-shell-common iio-sensor-proxy libaccountsservice0 libavahi-glib1
libflashrom1 libftdi1-2 libgeoclue-2.0 libgeocode-glib0 libglu1-mesa
libgnome-desktop-3-19 libgweather-3-16 libgweather-common libhandy-1-0
libimagequant0 liblua5.1-0-dev libncurses-dev libnginx-mod-http-geoip2
libnginx-mod-stream-geoip2 libnm0 libopengl0 libpangoft1-1.0-0
libpython3.9-minimal libpython3.9-stdlib libraqm0 libreadline-dev
libsoup-3.0-0 libsoup-3.0-common libsoup-gnome2.4-1 libtk8.6 libtool-bin
libunbound8
libxkbregistry0 linux-image-5.15.0-52-generic linux-modules-5.15.0-52-
generic linux-modules-extra-5.15.0-52-generic lua-any lua-bit32 lua-posix
```

```
lua-readline lua-unbound luarocks mutter-common pastebinit python3-
alabaster python3-automat python3-click python3-colorama python3-configobj
python3-constantly python3-debconf python3-debian python3-distro-info
python3-hamcrest python3-hyperlink python3-imagesize python3-incremental
python3-magic python3-olefile python3-problem-report python3-pyasn1
python3-pyasn1-modules python3-snowballstemmer python3.9 python3.9-minimal
run-one
sphinx-common sphinx-rtd-theme-common squashfs-tools tk8.6-blt2.5 x11-
apps x11-session-utils xinput xserver-xorg-input-all xserver-xorg-input-
libinput
xserver-xorg-input-wacom xserver-xorg-legacy xserver-xorg-video-all
xserver-xorg-video-amdgpu xserver-xorg-video-ati xserver-xorg-video-fbdev
xserver-xorg-video-nouveau xserver-xorg-video-radeon xserver-xorg-video-
vesa yaru-theme-gnome-shell

sudo apt autoremove
```

## Cleanup Snap

```
#!/bin/bash
```

## Removes old revisions of snaps

---

```
CLOSE ALL SNAPS BEFORE RUNNING THIS
cat > /usr/local/sbin/clean-snap.sh <<EOF
set -eu
snap list --all | awk '/disabled/{print $1, $3}' |
 while read snapname revision; do
 snap remove "$snapname" --revision="$revision"
 done
EOF

chmod +x /usr/local/sbin/clean-snap.sh && /usr/local/sbin/clean-snap.sh
```

## Backup

---

Copy the root public key to the translation server

```
cd /mnt/bigint/Jitsi_Translation_Raspberrypi
rsync -axP root@translation.sennsolutions.com:/boot/ boot_fs/
rsync -axP root@translation.sennsolutions.com:/ root_fs/
```

# Todo

---

- SSL certificate automated update procedure using getssl
- Show a message when the shutdown button is pressed asking for confirmation
- Show when the translator voice is really transmitted
- Make it completely independent from the translation.sennsolutions.com domain
- Automated Raspberry Pi installation and configuration
- Use the Raspberry Pi as a Wifi Access Point
- RO FS to avoid SD card corruption

# Done

- Display how many users are connected - Done
- Show something when the botton Mute All is pressed - Done

# References

---

<https://doganbros.medium.com/jitsi-iframe-in-use-a-shody-guide-to-embedding-jitsi-into-your-website-b13bf8d1c4f6>

<https://jitsi.org/blog/introducing-the-jitsi-meet-react-sdk/>

<https://github.com/jitsi/jitsi-meet-react-sdk>

<https://jitsi.github.io/handbook/docs/dev-guide/dev-guide-iframe/>

<https://ubunlog.com/en/create-react-app-crea-tu-primera-aplicacion-con-react/>

<https://stackoverflow.com/questions/55763428/react-native-error-enospc-system-limit-for-number-of-file-watchers-reached>

<https://stackabuse.com/building-a-rest-api-with-node-and-express/>

<https://forums.raspberrypi.com/viewtopic.php?t=11444> - How to change the keyboard on Raspbian

<https://support.google.com/pixelphone/thread/139593141/local-dns-resolution-suddenly-stopped-working?hl=en-GB>

<https://android.stackexchange.com/questions/151880/why-is-android-refusing-to-resolve-dns-records-pointing-to-internal-ip-addresses>

<https://www.androidpolice.com/use-preferred-dns-server-android-tutorial/>

<https://support.thepihut.com/hc/en-us/articles/360016173477-Pi-4-Pi-400-WiFi-problems>

<https://reelyactive.github.io/diy/pi-kiosk/>

<https://blog.logrocket.com/optimizing-performance-react-app/>

<https://radixweb.com/blog/steps-to-build-react-project-with-create-react-app>

<https://www.freecodecamp.org/news/how-to-build-a-react-app-different-ways/>

<https://blog.bitsrc.io/6-best-ways-to-create-a-new-react-application-57b17e5d331a?gi=a22086bee9cb>

<https://www.copypcat.dev/blog/reactjs-build-production/>

<https://shelcia.medium.com/create-a-video-call-web-app-in-10-minutes-with-jitsi-and-react-5453032f2173>

<https://jitsi.github.io/handbook/docs/dev-guide/dev-guide-ljm-api/#jitsimeetjs>

<https://jitsi.github.io/handbook/docs/dev-guide/dev-guide-ljm-api/>

<https://www.digitalocean.com/community/tutorials/how-to-deploy-a-react-application-with-nginx-on-ubuntu-20-04>