



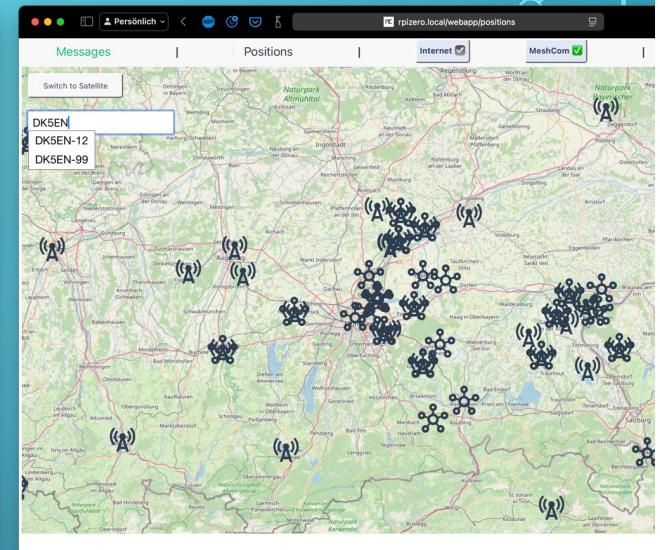
MESHCOM MCAPP INSTALLATION

SCREEN SHOTS – RESPONSIVE UI WITH DARK MODE

Two screenshots of a web-based monitoring interface for a network of LoRaWAN nodes. The interface includes tabs for 'Messages', 'Positions', 'Internet', and 'MeshCom'. Each node entry shows its ID, location, and status.

Node	ID	Location	Status
20 (113)	4B4C3287	Ziel: all	OK
222 (18)	IW1QAF-11	Il nuovo Nodo iw1qaf-11 è aperto a tutto il traffico mesh.. Vede Piemonte, parte della Lombardia e Liguria.	OK
22251 (27)	92F550FC	Il nuovo Nodo iw1qaf-11 è aperto a tutto il traffico mesh.. Vede Piemonte, parte della Lombardia e Liguria.	OK
* (428)	DL9CL-10	Happy Easter from Stuttgart SW-Germany	OK
DKSEN-99 (20)	A5E5B041	Happy Easter from Stuttgart SW-Germany	OK
OE5HWN-12 (24)	EFF85245	IW1QAF-11 buona pasqua	OK
OE9ROE-92 (2)	EFF85246	IW1QAF-11 buona pasqua	OK
DL2JA-11 (3)	EFF85246	IW1QAF-11 buona pasqua	OK
DL2JA-3 (2)	A5E5B044	DL9CL-10 mheard do9do-10	OK
DO1MH-12 (1)	Neueste unten	DL9CL-10 mheard do9do-10	OK

The interface also includes a 'cFDump' button and a text input field for sending messages to the nodes.



rpizero.local/webapp/settings				
Messages	Positions	Internet	MeshCom	Settings
X 15 (1) X	X 20 (116) ✓	X 222 (19) ✓	X 888 (3) X	
X 999 (10) X	X 7007 (2) ✓	X 8092 (1) X	X 22281 (27) ✓	
X 28275 (1) X	X 26277 (11) ✓	X 26298 (59) ✓	X 28362 (1) ✓	
X 26379 (1) X	X 26386 (1) X	X + (426) ✓	X DK5EN-99 (20) ✓	
X OE5HWY-12 (24) ✓	X OE9ROE-92 (2) ✓	X DL2JA-11 (3) ✓	X DL2JA-3 (2) ✓	
X DO1MH-12 (1) ✓	X TEST (2) ✓	X Time (10) ✓		
Callsign	DK5EN	SID	99	Scrollbar
WebSocket IP	rpizero.local	WebSocket Port	2981	LoRa IP
LoRa Port	not used	Country	EU8	TX Power
Latitude	48.123	Longitude	12.073	Altitude
APRS Name	not used	APRS Group	/	APRS Symbol

Messages	Positions	Internet	MeshCom	Settings
X 15 (1) X	X 20 (115) ✓	X 222 (19) ✓	X 888 (3) X	
X 999 (10) X	X 7007 (2) ✓	X 8092 (1) X	X 22251 (27) ✓	
X 26275 (1) X	X 26277 (1) ✓	X 26298 (59) ✓	X 26362 (1) ✓	
X 26379 (1) X	X 26386 (1) X	X *(426) ✓	X DK5EN-99 (20) ✓	
X OE5HWN-12 (24) ✓	X OE5ROE-92 (2) ✓	X DL2JA-11 (3) ✓	X DL2JA-3 (2) ✓	
X DO1MH-12 (1) ✓	X TEST (2) ✓	X Time (10) ✓		
Callsign	DK5EN	SID	99	Scrollback
WebSocket IP	rpizero.local	WebSocket Port	2981	LoRa IP
LoRa Port	not used	Country	EU8	TX Power
Latitude	48.123	Longitude	12.073	Altitude
APRS Name	not used	APRS Group	/	APRS Symbol

VORWORT

- Die Kommunikation zwischen Webbrowser am PC und Server Komponente am Raspi Zero kann nur TLS-verschlüsselt erfolgen, weil die modernen Browser dies erzwingen. Dies ist kein unnötiger Luxus, sondern wird vom Webbrowser (Chrome, Safari, ..) gefordert
- Offizielle Zertifikate von Let's Crypt werden nur für offizielle Domains ausgestellt. Wenn man mit mDNS „local“ (auf MacOS oder Windows mit iTunes installiert) oder Fritz!Box .fritz.box arbeitet, dann gibt es keine SSL-Zertifikate, die gegen ein getrustete Root-Zertifikat laufen, die im Browser und Betriebssystem vorinstalliert sind
- Jedoch ist die für den geübten Admin kein Problem, denn Caddy bringt eine PKI mit Zertifikatsrotation mit sich. x.509 Zertifikate sind trotzdem komplex
- Daher muss das self-signed Root Zertifikat der Caddy PKI import werden

- SSL-Zertifikate enthalten immer den Hostname, der mit der URL übereinstimmen muss. Es ist nicht möglich mit IP-Adressen zwischen Webbrowser und Raspi im lokalen Netz zu arbeiten. Es muss alles zwingend über DNS-Namen laufen, die auch dem cn= Eintrag im Zertifikat entsprechen
- Die Serverkomponente ist ein Python Script, das die Messages per UDP mit dem MeshCom Node austauscht und alles über einen WebSocket weiterleitet. Der WebSocket wird TLS verschlüsselt durch Caddy, unseren Reverse Proxy
- Der lighttpd Webserver wird ebenso durch Caddy TLS verschlüsselt. Die Webseite selbst ist statisch, es wird kein PHP benötigt, es ist eine Single Page App
- Wer seinen MeshCom Node nur über IP-Adresse erreicht, kann dies im `C2-mc-ws.py` Skript entsprechend anpassen.
- Zum Abschluss nicht vergessen auf dem MeshCom Knoten `--extudp on` zu konfigurieren und einzuschalten
- Stabilere Kommunikation mit viel mehr Daten läuft über Bluetooth

Happy Meshing in MeshCom de DK5EN

INSTALL PROCEDURE

- Flash SD card, insert, boot up, wait for FS expansion (takes 2 minutes)
- Login via ssh and your user, execute statements one after another

```
■ curl -fsSL https://raw.githubusercontent.com/DK5EN/McAdvChat/main/install_caddy.sh | bash  
■ curl -fsSL https://raw.githubusercontent.com/DK5EN/McAdvChat/main/mc-install.sh | sudo bash  
■ curl -fsSL https://raw.githubusercontent.com/DK5EN/McAdvChat/main/install_mcproxy.sh | bash  
■ sudo vi /etc/mcadvchat/config.json  
■ curl -fsSL https://raw.githubusercontent.com/DK5EN/McAdvChat/main/install_mcproxy.sh | bash
```

- Got to web browser,
 - On a Mac Computer with mDNS responder: import root certificate <https://mcapp.local/root.crt>
 - On a Windows PC, but with a Fritz!Box as DSL Router: <https://mcapp.fritz.box/root.crt>
- Go to web browser,
 - On a Mac: <https://mcapp.local/webapp>
 - On a PC with Fritz!Box: <https://mcapp.fritz.box/root.crt>
- Click connect MC
- Enjoy

UPDATING TO THE LATEST VERSION

- At a later time, update everything, just use the script
- mc-install.sh is to keep all installed files up to date
- install_mcproxy.sh does maintain config files

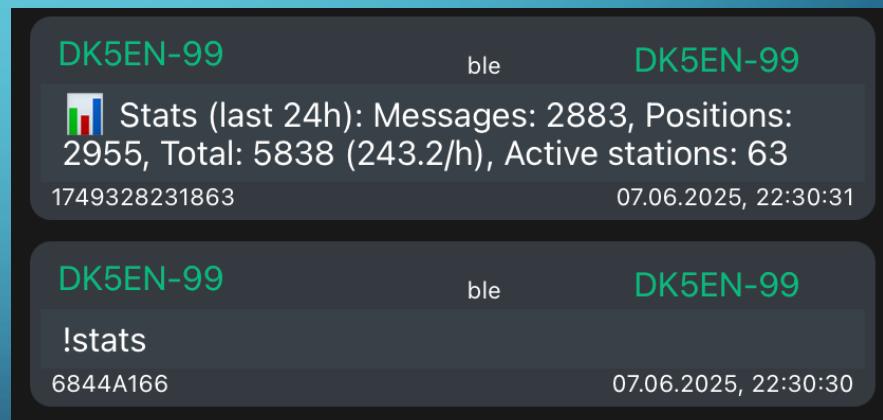
```
martin@McApp:~ $ curl -fsSL https://raw.githubusercontent.com/DK5EN/McAdvChat/main/mc-install.sh | sudo bash

[INFO] Skript läuft unter Benutzer: martin
[INFO] Lokale WebApp-Version: v0.1.0
[INFO] Lokale Python-Skript-Version: v0.2.0
[INFO] Lokale Shell-Skript-Version: v0.2.0
[INFO] Install-Skript-Version: v0.1.0
[INFO] Remote WebApp-Version: v0.1.0
[INFO] Remote Python-Skript-Version: v0.2.0
[INFO] Remote Shell-Skript-Version: v0.2.0
[INFO] Reloade Webserver ...
[INFO] Prüfe WebApp unter https://rpiZero.local/webapp/version.txt
[INFO] WebApp erfolgreich aktualisiert auf Version v0.1.0
[INFO] Installations-Skript erfolgreich abgeschlossen.

martin@McApp:~ $ curl -fsSL https://raw.githubusercontent.com/DK5EN/McAdvChat/main/install_mcproxy.sh | bash
```

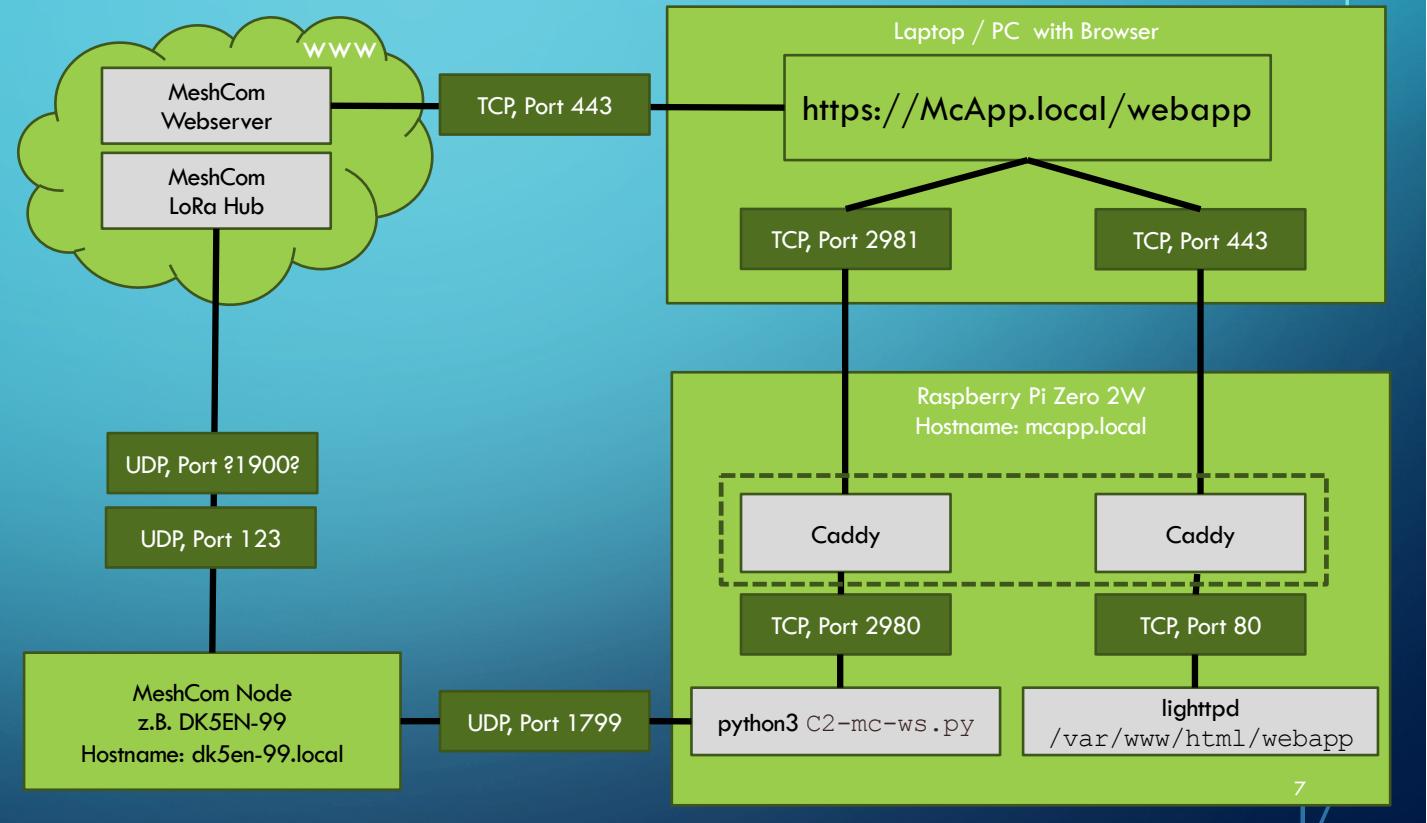
MCAPP COMMANDS

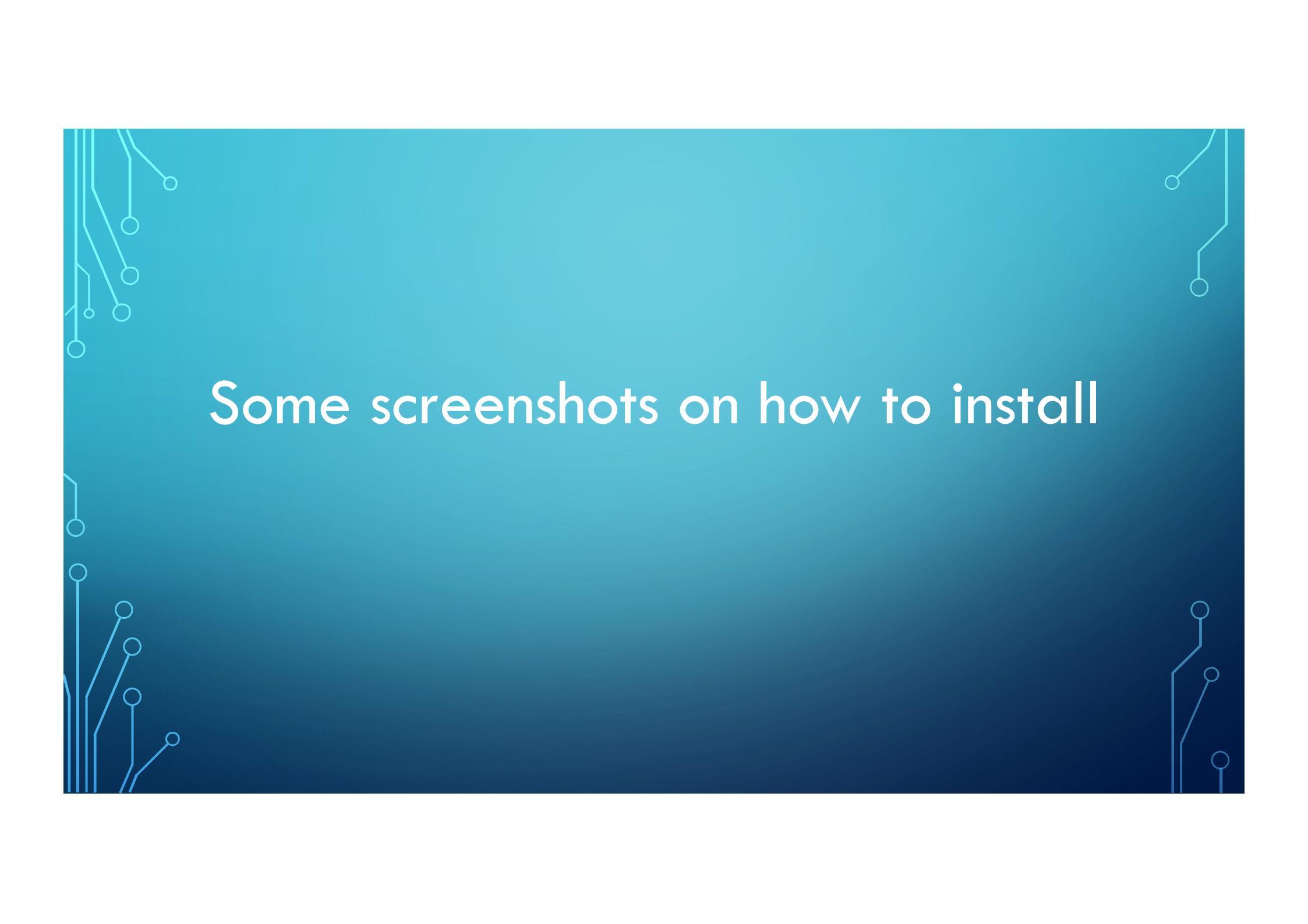
- If you configure your Callsign with SID in the config file, you can send commands to the server process
- !stats
- !stats 96
- !help
- !search db0ed
- !search d12ja-1
- !mheard oder !mh
- !mh 25



HIGH LEVEL ARCHITECTURE

- High level overview of how everything is tied together





Some screenshots on how to install

SD CARD & RPI ZERO

- Insert your at least 32GB SD Card into your card reader
- Please only use SD Cards with 100MBit/s like SanDisk
- Have your Raspberry Pi Zero 2W at Hand

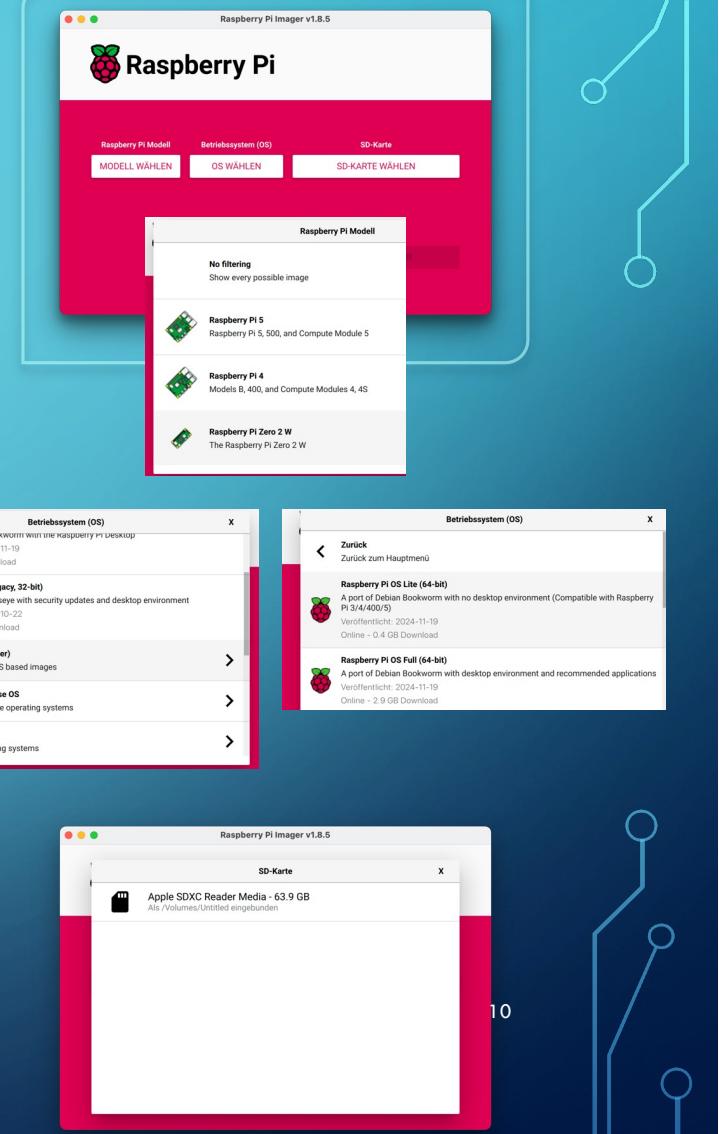


CUSTOM OS INSTALLATION

We want to install a headless, 64Bit Debian Bookwork

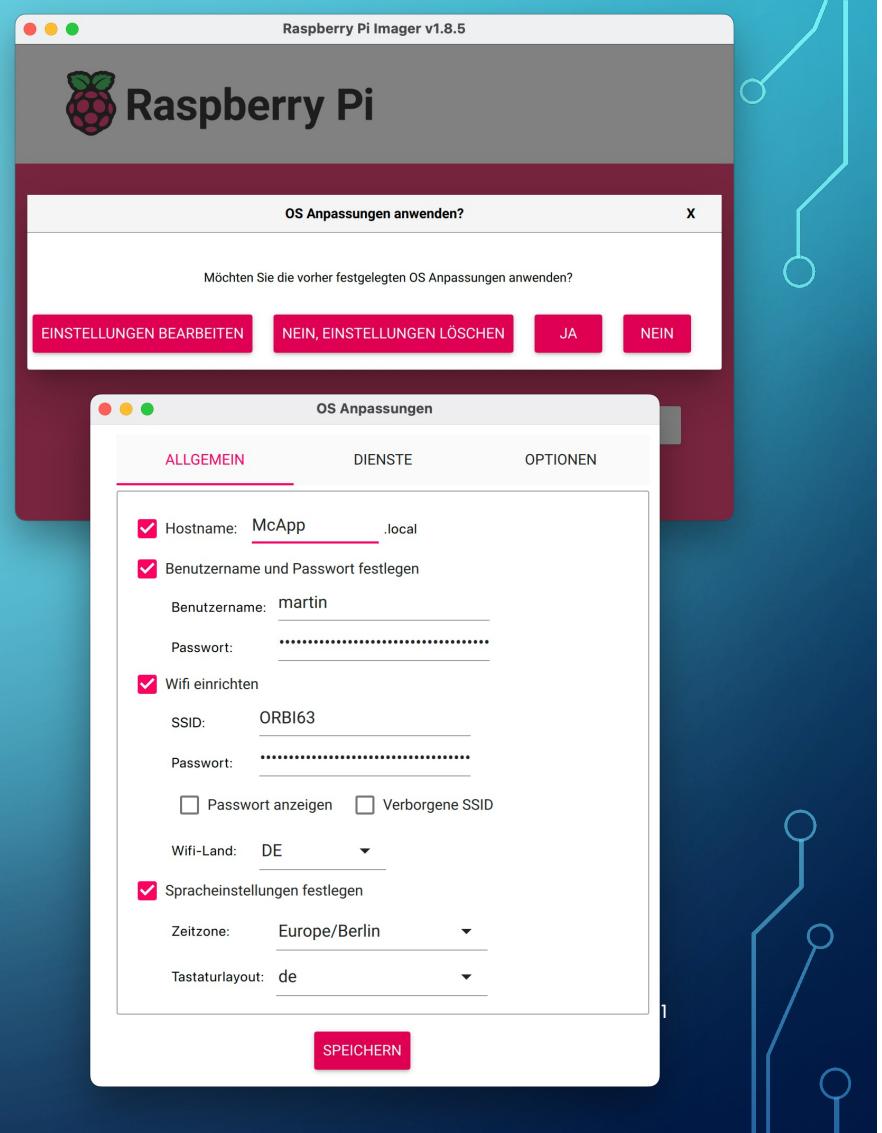
- I do recommend Raspberry PI Imager, which can be found here
<https://www.raspberrypi.org/downloads/>
- Select your Model: Raspberry Pi Zero 2 W
- For OS, select “other“ – „Raspberry Pi OS Lite (64-bit), with no desktop, approx. 0,4GB
- Select your SD Card

.. And click next



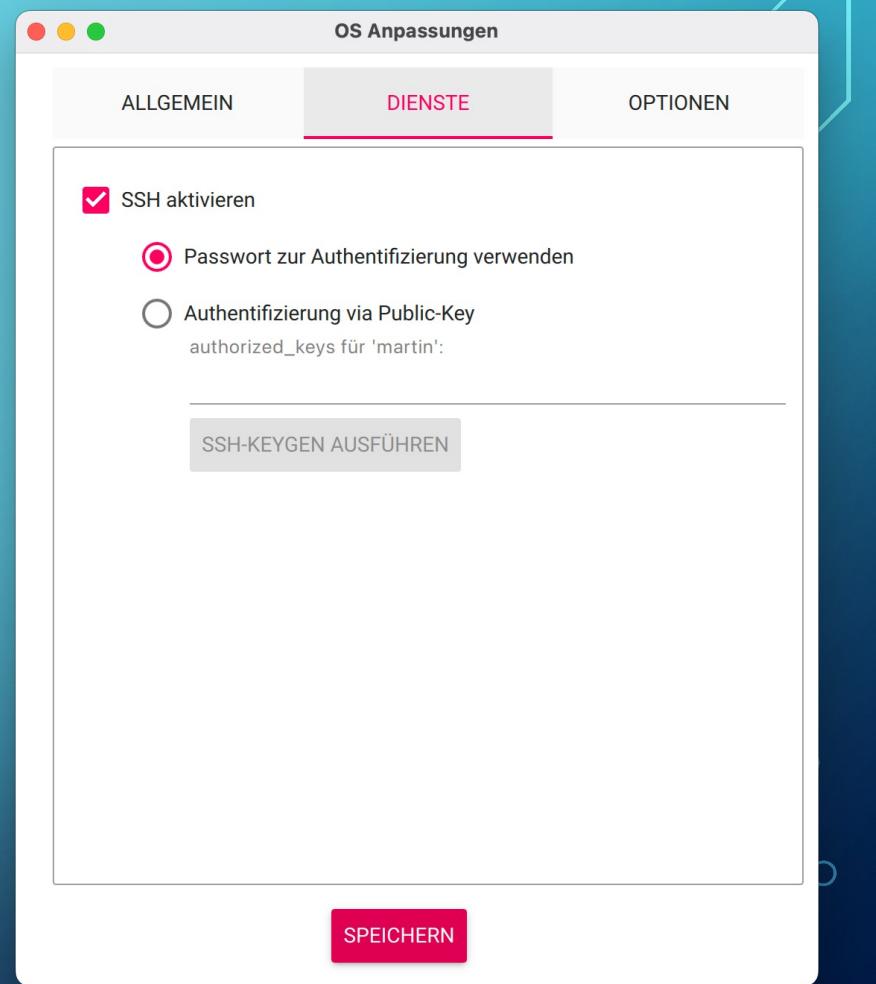
IMPORTANT CUSTOMIZATIONS

- Choose your hostname: McApp in our example
- Choose what ever username, you want. I do not recommend to setup a standard „pi“ user, as this is a security risk
- Choose your login password, which should later be changed to a pre-shared ssh key
- Make sure you have your WiFi Settings correct, because otherwise you will not be able to access your headless system



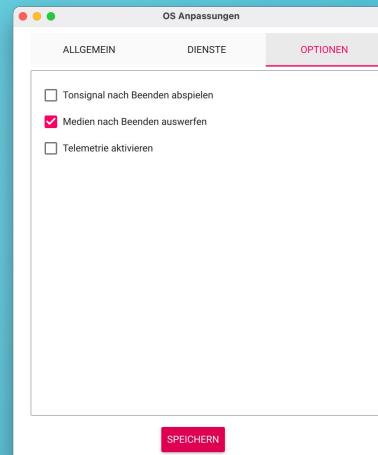
ACTIVATE SSH

- Make sure that ssh is activated.
- For the initial setup, we start with password Authentication.
- If you are experienced, you can also set a pre-shared key.



OPTIONS

- Nothing to change here, everything standard.
- Now click save
- Then click yes to apply custom settings
- Now agree to erase everything on the SD card.



FLASHING THE SD CARD

- Now wait for the flashing to be finished
- On MacOS you get asked about your Admin password, as this is a low level write, that needs more privileges
- After a short while you should see the success message
- Close Raspberry Pi Imager, eject your SD card

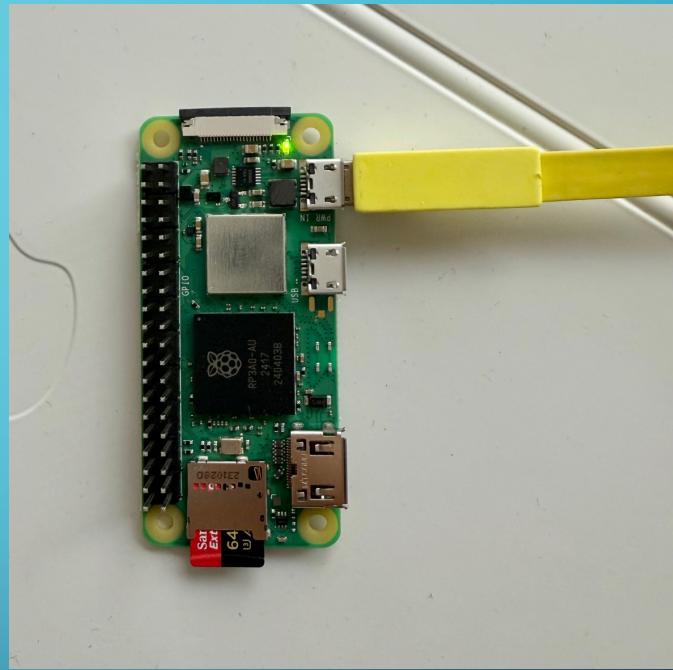


FIRST TIME BOOT UP

- Insert your SD card into Raspberry Pi 2 Zero
- Attach 5V via Mini USB Jack
- The greenlight starts flashing

Raspberry Pi is booting up and expanding the filesystem. Depending on your SD card, this takes at least 2 Minutes. Grab yourself a coffee and wait

- If you have mDNS, then you can try to ping your Raspberry Pi
- Otherwise check your WiFi Router for the IP of the new device

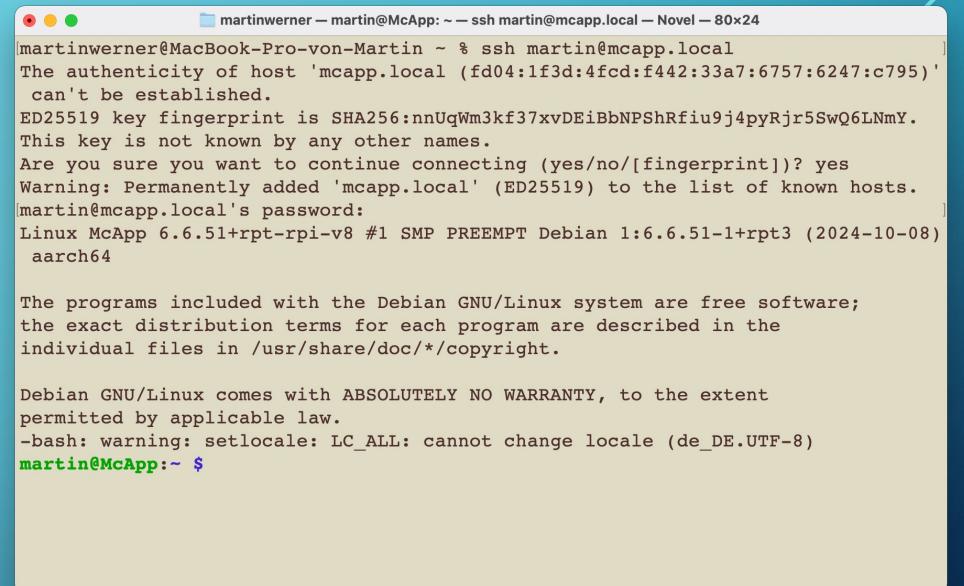


```
martinwerner@MacBook-Pro-von-Martin .ssh % ping mcapp.local
ping: cannot resolve mcapp.local: Unknown host
martinwerner@MacBook-Pro-von-Martin .ssh % ping mcapp.local
PING mcapp.local (192.168.68.70): 56 data bytes
64 bytes from 192.168.68.70: icmp_seq=0 ttl=64 time=121.193 ms
64 bytes from 192.168.68.70: icmp_seq=1 ttl=64 time=11.150 ms
64 bytes from 192.168.68.70: icmp_seq=2 ttl=64 time=8.156 ms
64 bytes from 192.168.68.70: icmp_seq=3 ttl=64 time=3.976 ms
64 bytes from 192.168.68.70: icmp_seq=4 ttl=64 time=8.038 ms
64 bytes from 192.168.68.70: icmp_seq=5 ttl=64 time=15.590 ms
```

TIME TO ACCESS YOUR RASPI

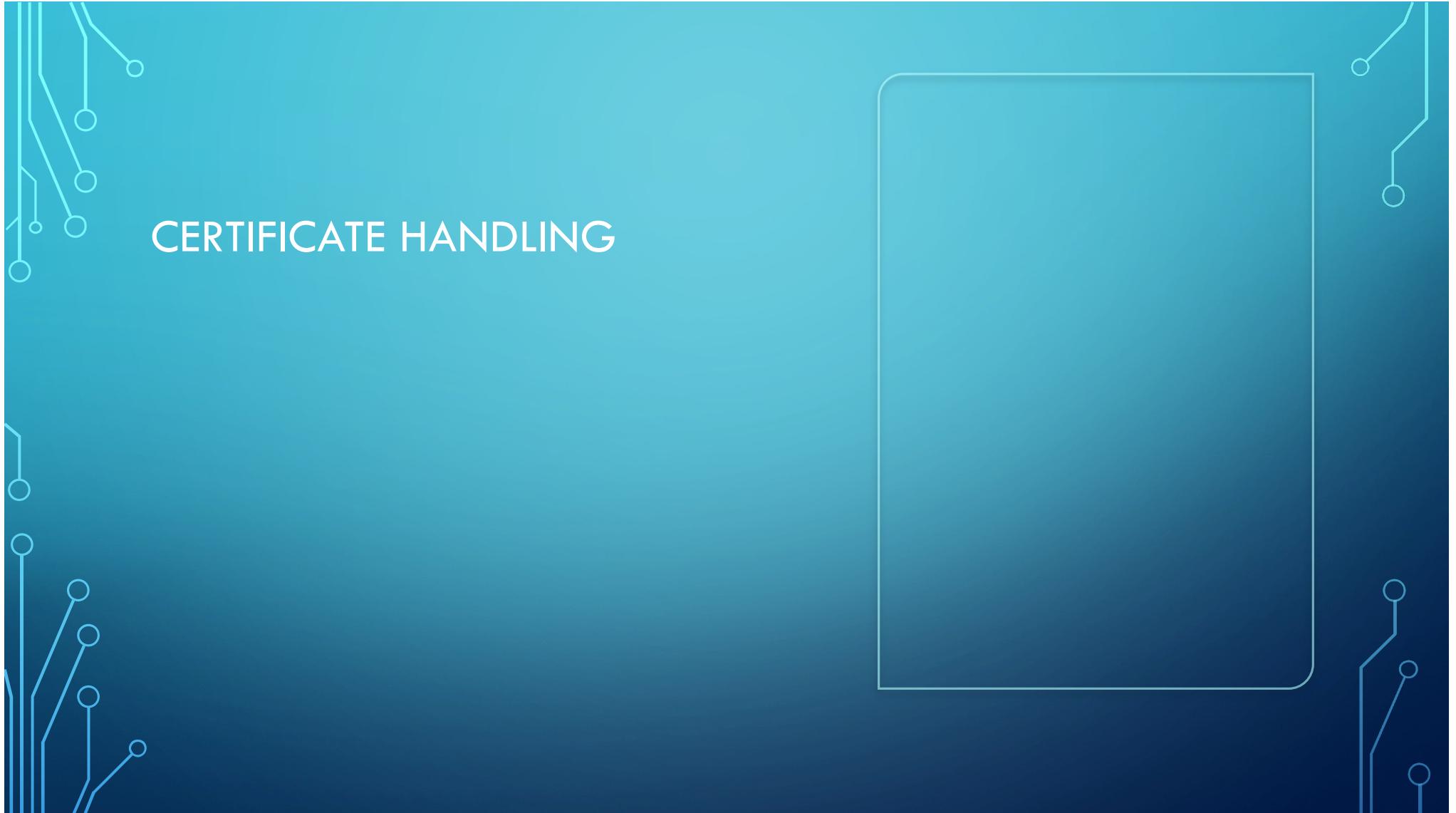
Use putty on Windows or term on MacOS

- Make sure to use the correct username for your Pi Zero
 - Accept the new ssh fingerprint
 - Enter your password
 - You should now have ssh access to your Raspi
-
- Now run the 4 commands shown on Page 4- “New install procedure”



The screenshot shows a terminal window titled "martinwerner — martin@McApp: ~ — ssh martin@mcapp.local — Novel — 80x24". The session is connecting to "martinwerner@MacBook-Pro-von-Martin ~ % ssh martin@mcapp.local". It displays the SSH key fingerprint and asks if the user wants to continue connecting. The user responds with "yes". It then prompts for the password, which is entered as "aarch64". The terminal then shows the Debian system information, including the kernel version "Linux McApp 6.6.51+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.51-1+rpt3 (2024-10-08)". Finally, it shows the message "The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright".

CERTIFICATE HANDLING



CHECK THE WEB SERVER DOWNLOAD SSL CERTIFICATE

- First we access our lighttpd via http – the unencrypted version.

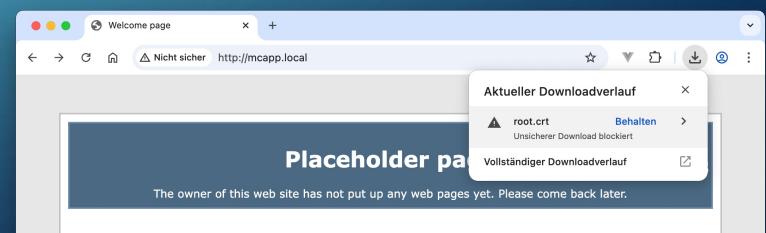
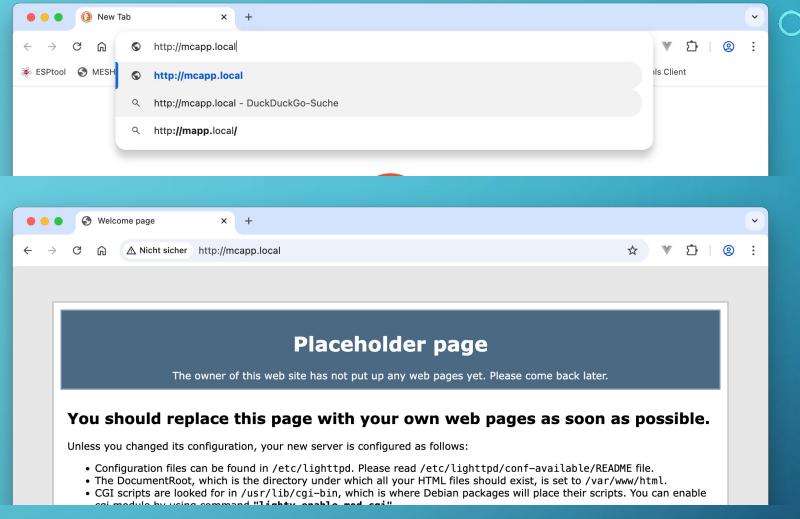
<http://mcapp.local>

- You should see the placeholder page

- Now we download the self-signed root certificate

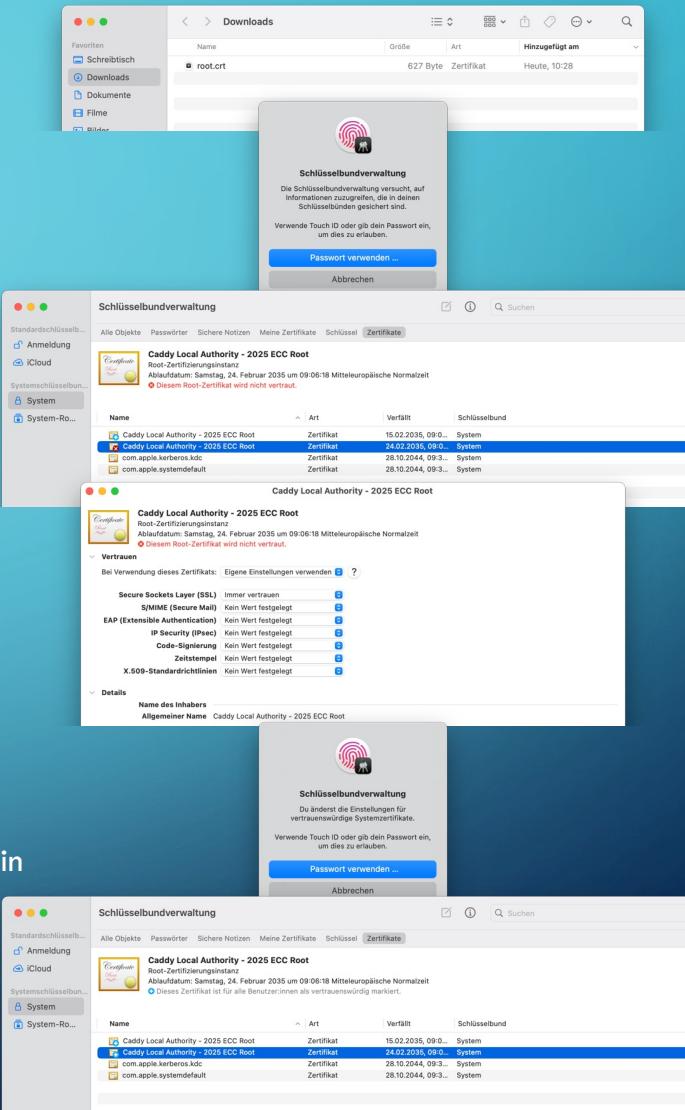
<http://mcapp.local/root.crt>

- Make sure to accept the blocked download



INSTALL THE SSL CERTIFICATE

- Locate your root.crt in your download folder
 - Double click root.crt
 - Enter your Admin password
 - Now locate the newly installed certificate
 - Trust the certificate for TLS encryption
-
- Same is true for the iPhone. After import, you have to accept it in General – Profiles AND then you need to trust it, which is hidden in the settings menu

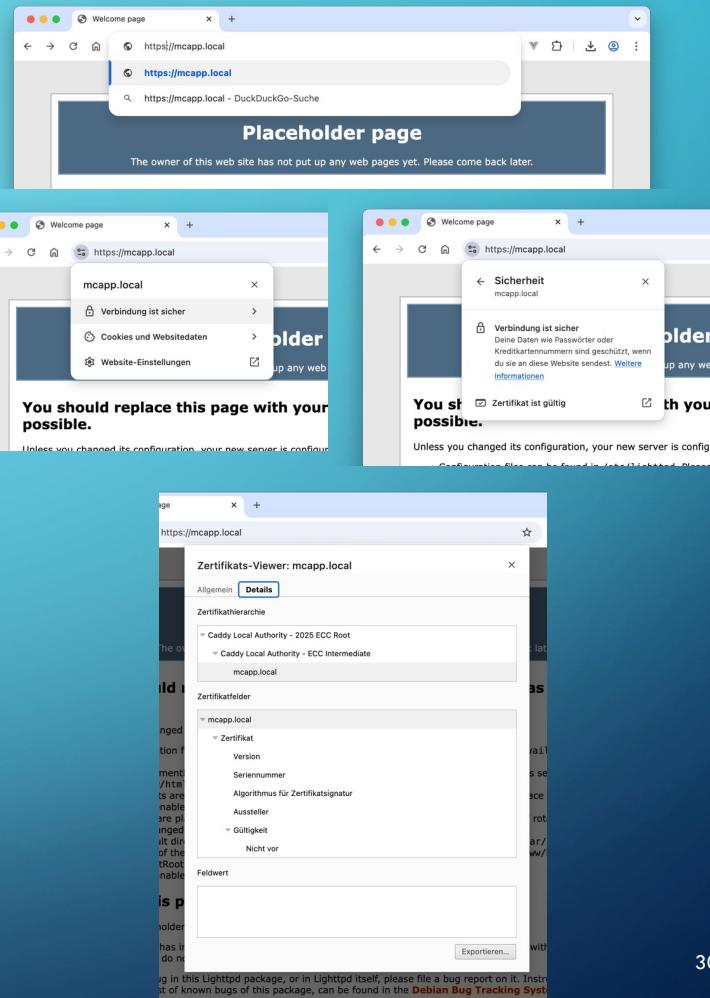


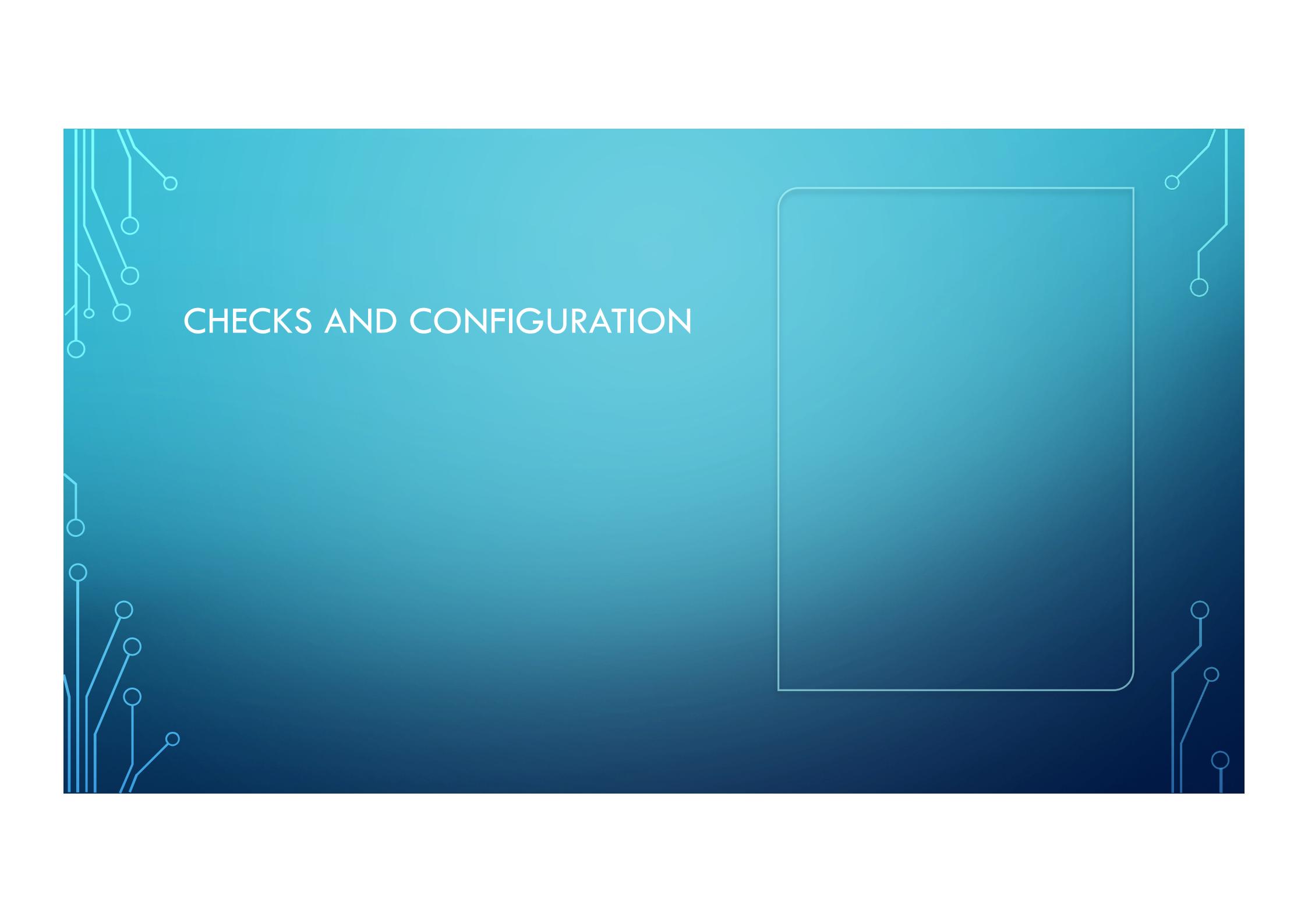
TESTING SSL ACCESS

- Go to your web browser, now we access https

<https://mcapp.local>

- If everything worked out, as expected, you should see a fully trusted root chain.

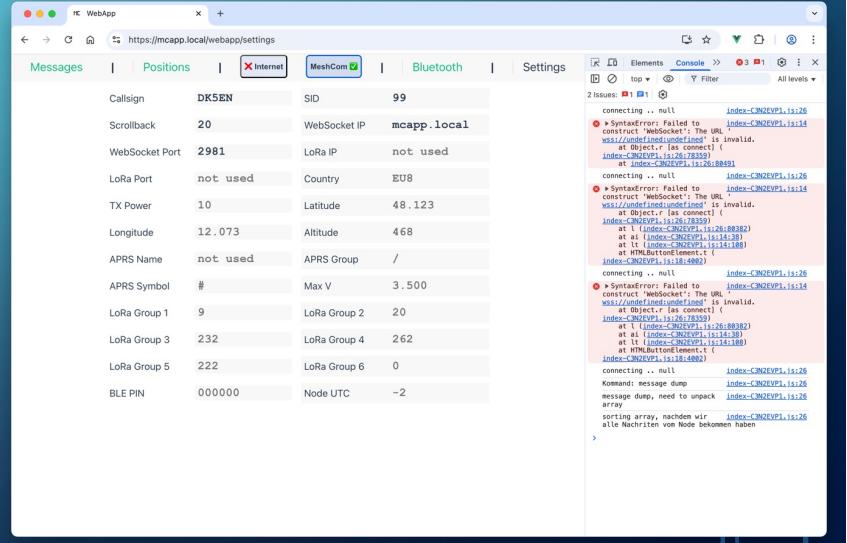
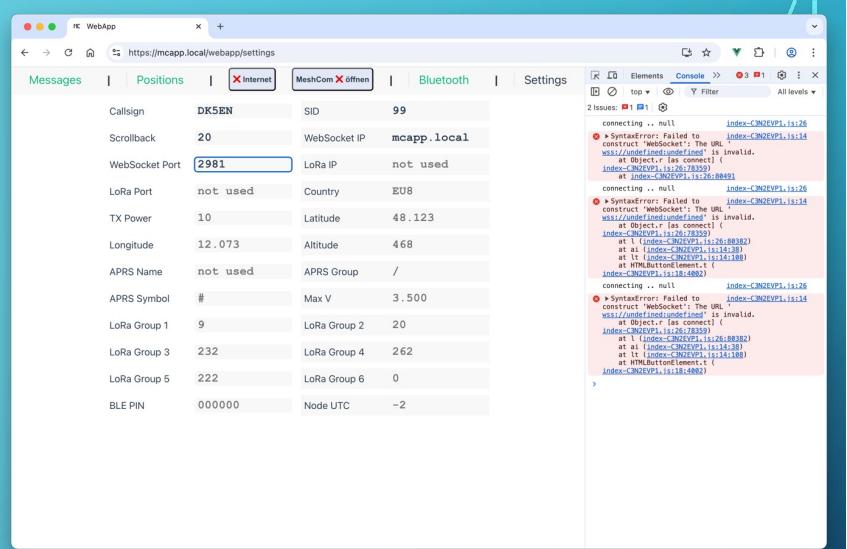




CHECKS AND CONFIGURATION

OPEN WEBBROWSER

- Click option command J for Debug Output
- Go to url: <https://mcapp.local/webapp/>
- Click on Settings
- Enter your Callsign
- Your SID
- 20 for scroll back buffer
- mcapp.local for your UDP Proxy server
- 2981 for the secure socket of the UDP Proxy that runs in python
- Click Connect MeshCom



TIME TO TEST EVERYTHING

- Go to <https://meshcom.oevsv.at/#>
- Click on Test
- Now enter a test Message
- Check on MeshCom Test page

Messages | Positions | Internet | MeshCom | Bluetooth | Settings

Ziel: all

Node	Message	Time
20 (6)	0.0 mm Wind: 17 Km/h Pressure MSL: 1003.9 hPa	276CE0C1 22291 17.04.2025, 12:01:20
D9KMS-12	Guten Mittag aus Fulda, Osthessen, 9 Grad, bedeckt, es hat geregnet, 73 Markus	DK9MS-12 20 17.04.2025, 12:28:19
OE5HWN-6	Mittagspause in der Kalten Kucht	BOT GATE 8C590CA4 17.04.2025, 12:30:22
999 (1)		
1211 (1)	guten Hunger Helmut	DO5DHA-12 3DCB0FAA 17.04.2025, 12:30:55
8092 (8)		
DK8GO-12	hier liegt schnee	DK8GO-12 75D311E4 17.04.2025, 12:32:10
8421 (1)		
20857 (3)		
22201 (2)	DB0SEP BBS online https://qrz.com/db/db0sep	DB0SEP-12 A2659230 17.04.2025, 13:00:01
22251 (12)		

cFDump

Gesendet: Eine kleine Testnachricht, ob was raus geht

TEST

all DK5EN-99 | WsProxy:mcapp.local:2981 | remaining:124

Eine kleine Testnachricht

cMsgRead

Console output (partial):

```

connecting .. null
index-C3N2EVPl.js:26
SyntaxError: Failed to construct 'WebSocket': The URL 'ws://undefined' is invalid.
at Object.r [as connect] (index-C3N2EVPl.js:26:78359)
at a1 (index-C3N2EVPl.js:12:26778359)
at index-C3N2EVPl.js:12:26788491
connecting .. null
index-C3N2EVPl.js:26
SyntaxError: Failed to construct 'WebSocket': The URL 'ws://undefined' is invalid.
at Object.r [as connect] (index-C3N2EVPl.js:26:8832)
at a1 (index-C3N2EVPl.js:12:1438)
at index-C3N2EVPl.js:12:14108
at HTMLButtonElement.t (index-C3N2EVPl.js:12:4882)
connecting .. null
index-C3N2EVPl.js:26
SyntaxError: Failed to construct 'WebSocket': The URL 'ws://undefined' is invalid.
at Object.r [as connect] (index-C3N2EVPl.js:26:8832)
at a1 (index-C3N2EVPl.js:12:1438)
at index-C3N2EVPl.js:12:14108
at HTMLButtonElement.t (index-C3N2EVPl.js:12:4882)
connecting .. null
index-C3N2EVPl.js:26
Kommand: message dump
index-C3N2EVPl.js:26
message dump, need to unpack
index-C3N2EVPl.js:26
array
sorting array, nachdem wir
index-C3N2EVPl.js:26
alle Nachrichten vom Node bekommen haben
WE Connect ausgelöst
index-C3N2EVPl.js:26
sorting array, nachdem wir
index-C3N2EVPl.js:26
alle Nachrichten aus dem internet bekommen haben

```

Privat < meshcom.oevsv.at/#

Index	Date	From	To	Type	Content	Flags	Details
34	12:22:45	A2659224	DB0SEP-12	DB0SEP-12	D9KMS-12	4 0 0	Ping received, BBS online
35	2025-04-17 12:24:55	DA70E2F6	DK9MS-12	DK9MS-12	DB0SEP-12	4 0 0	db0sep h
36	2025-04-17 12:24:58	A2659226	DB0SEP-12	DB0SEP-12	DK9MS-12	4 0 0	Commands-> db0sep br,bs,mh,r,l,e,s,u,h,p,t https://www.qrz.com/db/db0sep - done..
37	2025-04-17 12:26:37	DA70E2F9	DK9MS-12	DK9MS-12	DB0SEP-12	4 0 0	db0sep p
38	2025-04-17 12:26:42	A2659228	DB0SEP-12	DB0SEP-12	DK9MS-12	4 0 0	Ping received, BBS online
39	2025-04-17 12:56:05	DA70E2FE	DK9MS-12	DK9MS-12	DB0SEP-12	4 0 0	db0sep r 1
40	2025-04-17 12:56:08	A265922D	DB0SEP-12	DB0SEP-12	DK9MS-12	4 0 0	mal sehen ob das bei dir auch ankommt. 73 de Helmut - done..
41	2025-04-17 12:56:44	DA70E300	DK9MS-12	DK9MS-12	DB0SEP-12	4 0 0	db0sep e 1
42	2025-04-17 12:56:50	A265922F	DB0SEP-12	DB0SEP-12	DK9MS-12	4 0 0	Delete ok - done..
43	2025-04-17 13:00:01						
44	2025-04-17 13:00:39	EA0EB280	DK5EN-99	DK5EN-99	TEST	4 0 0	Eine kleine Testnachricht, ob was raus geht

Starttime:2025-04-15 20:50:37 ID:338658 MAC:ff9f9fb

UPDATE LOCALES AND KEEP SYSTEM UPDATE

```
martin@McApp:~ $ sudo raspi-config
```

- 5 Localization -> L1 locale -> select de_AT.UTF-8 -> OK -> C.UTF-8 -> OK
- Finish

```
martin@McApp:~ $ sudo apt-get update
```

```
martin@McApp:~ $ sudo apt-get dist-upgrade
```

```
martin@McApp:~ $ sudo reboot
```