Running OpenEMS Edge on a Raspberry PI

This guide explains how to:

1. Install and run **OpenEMS Edge on a Raspberry Pi**
2. Configure WebSocket + simulation
3. Run **OpenEMS UI on a macOS machine (not on the Pi)**
4. Connect the UI to the Edge over the internet

The structure is intentionally split into two clear parts:

- **PART A — Raspberry Pi (Edge)**
- **PART B — macOS Machine (OpenEMS UI)**

==========

===

PART A — Raspberry Pi (Edge)

==========

===

1. Raspberry Pi OS Setup (Headless)

Use Raspberry Pi Imager:

- Device: Raspberry Pi 4
- OS: Raspberry Pi OS Lite (64-bit)
- Configure:

    - Hostname
    - Username & password
    - Enable SSH
    - Configure Wi-Fi
    - Enable Raspberry Pi Connect

Boot the Pi and connect via:

- SSH
- OR Raspberry Pi Connect

---

2. Install Java 21 (Temurin ARM64)

On the Pi:

```
sudo apt update
sudo apt install -y wget apt-transport-https gpg

wget -qO - https://packages.adoptium.net/artifactory/api/gpg/key/public \
 | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/adoptium.gpg > /dev/null

echo "deb https://packages.adoptium.net/artifactory/deb \
$(awk -F= '/^VERSION_CODENAME/{print $2}' /etc/os-release) main" \
 | sudo tee /etc/apt/sources.list.d/adoptium.list

sudo apt update
sudo apt install -y temurin-21-jdk
```

Verify:

```
java -version
```

---

3. Install OpenEMS Edge

```
mkdir -p ~/downloads
cd ~/downloads
wget https://github.com/OpenEMS/openems/releases/download/2025.11.0/openems-edge.jar
sudo chmod +x openems-edge.jar
```

```
sudo mkdir -p /usr/lib/openems
sudo mv openems-edge.jar /usr/lib/openems/
sudo mkdir -p /etc/openems.d
```

---

4. Configure systemd Service

Create:

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

Paste:

```
[Unit]
Description=OpenEMS Edge
After=network.target

[Service]
User=root
Group=root
Type=notify
WorkingDirectory=/usr/lib/openems
ExecStart=/usr/bin/java -Dfelix.cm.dir=/etc/openems.d/ \
  -jar /usr/lib/openems/openems-edge.jar
SuccessExitStatus=143
Restart=always
RestartSec=10
WatchdogSec=60

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

Enable & start:

```
sudo systemctl daemon-reload
sudo systemctl enable openems
sudo systemctl start openems
```

Check:

```
systemctl status openems
```

---

5. Access OpenEMS Config Manager

On the Pi, open:

```
http://localhost:8080/system/console/configMgr
```

---

6. Add Required Components

In Config Manager:

1. Add a **Scheduler** (any default scheduler)
2. Add **Controller.Api.Websocket**
   - Port: 8085
   - Enabled: true

Restart Edge:

```
sudo systemctl restart openems
```

Verify WebSocket is listening:

```
sudo ss -lntp | grep 8085
```

You should see Java listening on port 8085.

---

7. (Optional) Add Simulation Components

In Config Manager, add:

- Simulator ESS
- Simulator Grid Meter
- Simulator PV

Save and restart OpenEMS.

==========

==========

PART B — macOS Machine (OpenEMS UI)

==========

==========

1. Install Docker Desktop

```
brew install --cask docker
open -a Docker
```

Wait until Docker reports "Docker is running".

Verify:

```
docker version
```

You must see both Client and Server.

---

2. Clone OpenEMS Source

```
git clone -b 2025.11.0 https://github.com/OpenEMS/openems
cd openems
```

3. Build OpenEMS UI Image

```
docker build . \
  -t openems_ui \
  -f tools/docker/ui/Dockerfile.edge
```

4. Run OpenEMS UI

Replace YOUR_PI_IP with:

- Public IP
- OR VPN IP
- OR Public DNS

Example:

```
docker container run \
  -e WEBSOCKET_HOST=YOUR_PI_IP \
  -p 80:80 \
  -p 443:443 \
  --restart unless-stopped \
  --name openems_ui_container \
  openems_ui
```

5. Open the UI

On your Mac:

```
http://localhost/login
```

Default login:

- Username: admin
- Password: admin

If UI shows "disconnected":

- Confirm port 8085 is listening on the Pi
- Confirm WebSocket controller exists
- Confirm WEBSOCKET_HOST is correct

Why UI Runs on macOS and Edge Runs on Pi

- Edge interacts with hardware and benefits from native systemd management
- UI behaves like a stateless web application and is ideal for containerization
- Separating them improves stability and flexibility

Troubleshooting

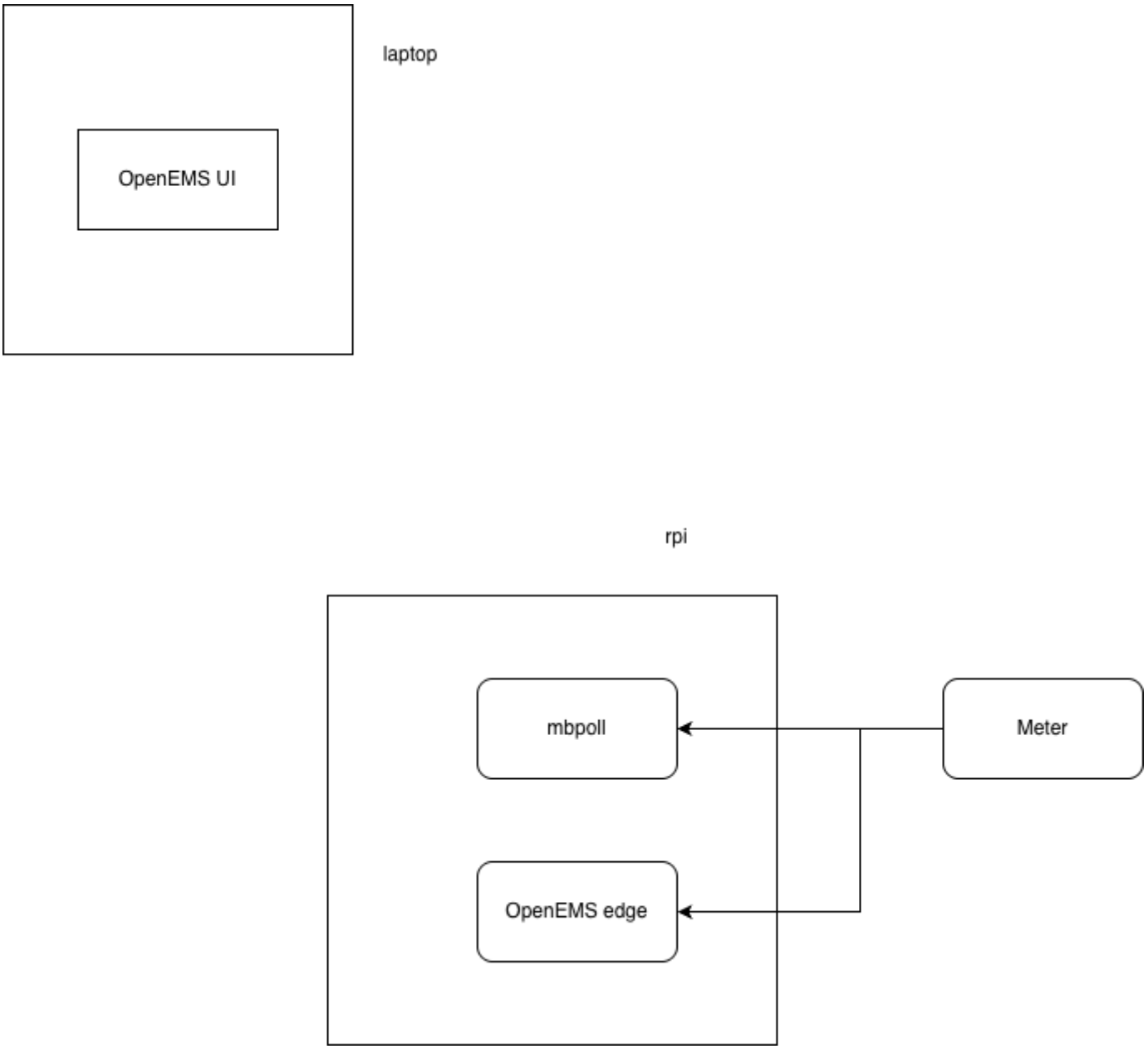
- UI loads but no connection → check port 8085
- Nothing listening on 8085 → WebSocket not configured
- Docker errors on Mac → ensure Docker Desktop is running

Security Notes

- Prefer VPN/tunnel (Tailscale/WireGuard) over port forwarding
- Use SSH key-based authentication
- Do not expose ConfigMgr (8080) publicly

You now have:

- OpenEMS Edge running on Raspberry Pi
- WebSocket enabled
- Optional simulation components
- OpenEMS UI running on macOS
- UI connected over internet



Revision #6
Created 2026-02-06 07:46:28 UTC by aaron.tushabe
Updated 2026-02-12 05:56:05 UTC by aaron.tushabe