68. UKW-Tagung 2023

VLAN Workshop mit virtuellem **Freifunk-Offloader**

Hardware und Kabel sparen mit Virtualisierung und VLAN-Technik

Michael Dörr
techno.turtle@gmx.net

Was sind VLANs?

- Mit Hilfe von virtuellen LANs (**VLANs**) können Netzwerke *logisch* voneinander separiert werden, obwohl *physikalisch* die gleiche Infrastruktur (Rechner, Kabel, Switches) verwendet wird.
- Dazu werden Ethernet-Pakete um die **VLAN-ID**-Bytes verlängert: die sogenannten **Tags** oder **VIDs**.
- Es gibt verschiedene VLAN-Versionen; aber nur Geräte die nach dem Standard **802.1q** arbeiten, sind miteinander interoperabel.
- Geräte, die sich in unterschiedlichen VLANs befinden, können sich per TCP/IP **nicht direkt** erreichen. Konnektivität kann nur über Router (bzw. Firewalls) erreicht werden.

Und was kann man damit machen?

- **Beispiel**: ein NUC-Rechner hat nur eine Netzwerkschnittstelle;
  für unsere Versuche werden aber mehrere, verschiedene Netzwerke benötigt:
  - separate Netzwerke für LAN und WAN (bzw. DMZ)
  - Vermeidung von Störungen durch die verschiedenen Adressbereiche der einzelnen Freifunk-Netzwerke
  - pro Netzwerk ist immer nur ein DHCP-Server möglich

Versuchsaufbau

Vor Beginn des Versuchs wird die geplante Architektur festgelegt.
Es sollen zwei verschiedene Freifunk-Offloader in einem virtualisierten Umfeld getestet werden.

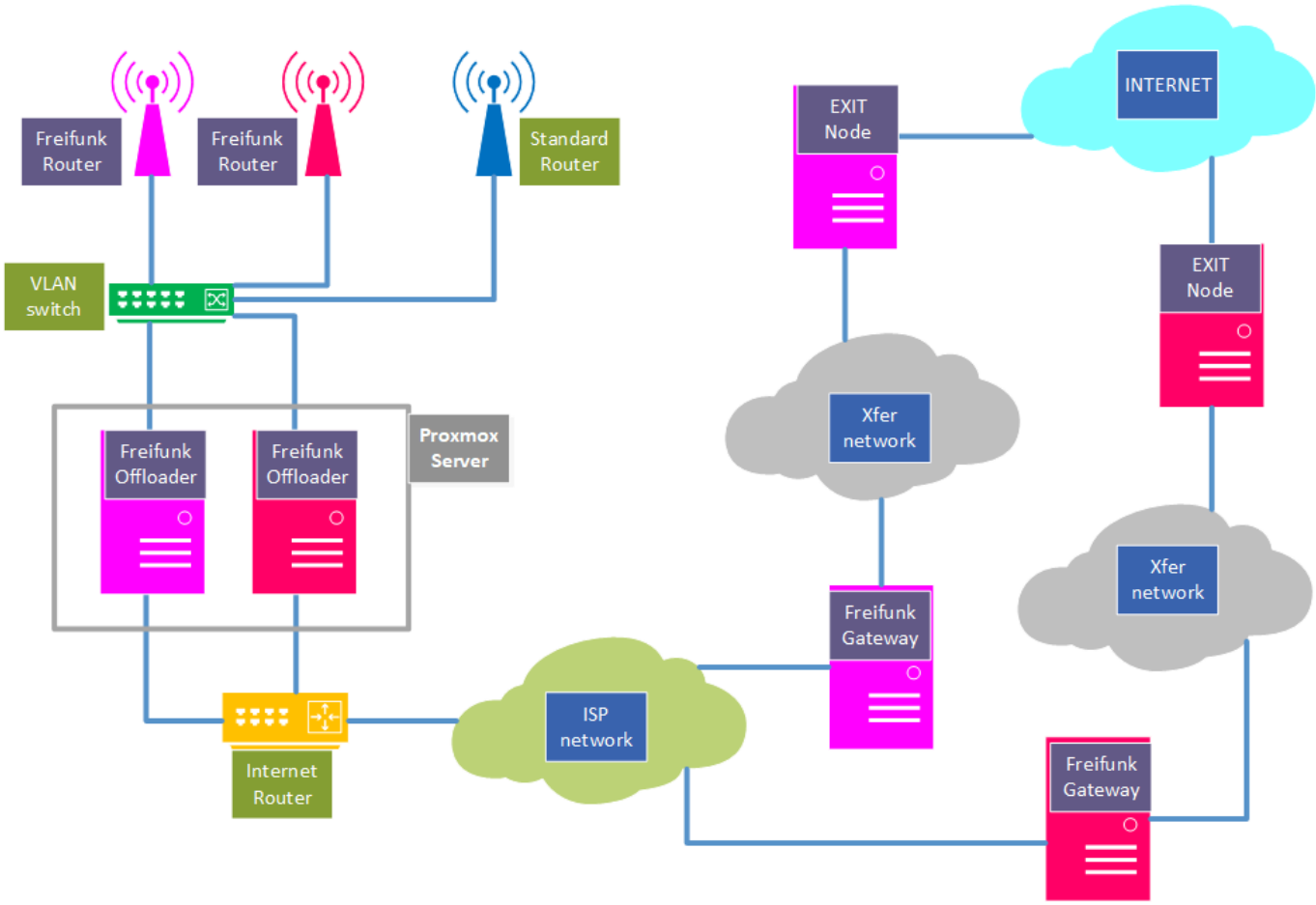
Dafür werden ein Kleinstrechner Intel-NUC, zwei VLAN-taugliche Switches und einige WLAN-Access-Points benutzt.

Insgesamt werden 5 VLANs benötigt:

Das Default Client-Netzwerk erhält die Default VID=1 und als Gateway einen OPNSense-Router.
Das DMZ-Netzwerk, mit dem die Verbindung zum Internet-Router hergestellt wird, erhält die VID=2.

Die zwei Netzwerke für die beiden Freifunk-Offloader erhalten die VIDs 11 und 21.
Das zweite Client-Netzwerk erhält die VID=31 und als Gateway einen OpenWRT-Router.

Gesamtarchitektur



VLANs im Überblick

Es gibt zwei Arten von Ports an einem VLAN-fähigen Netzwerk-Switch:

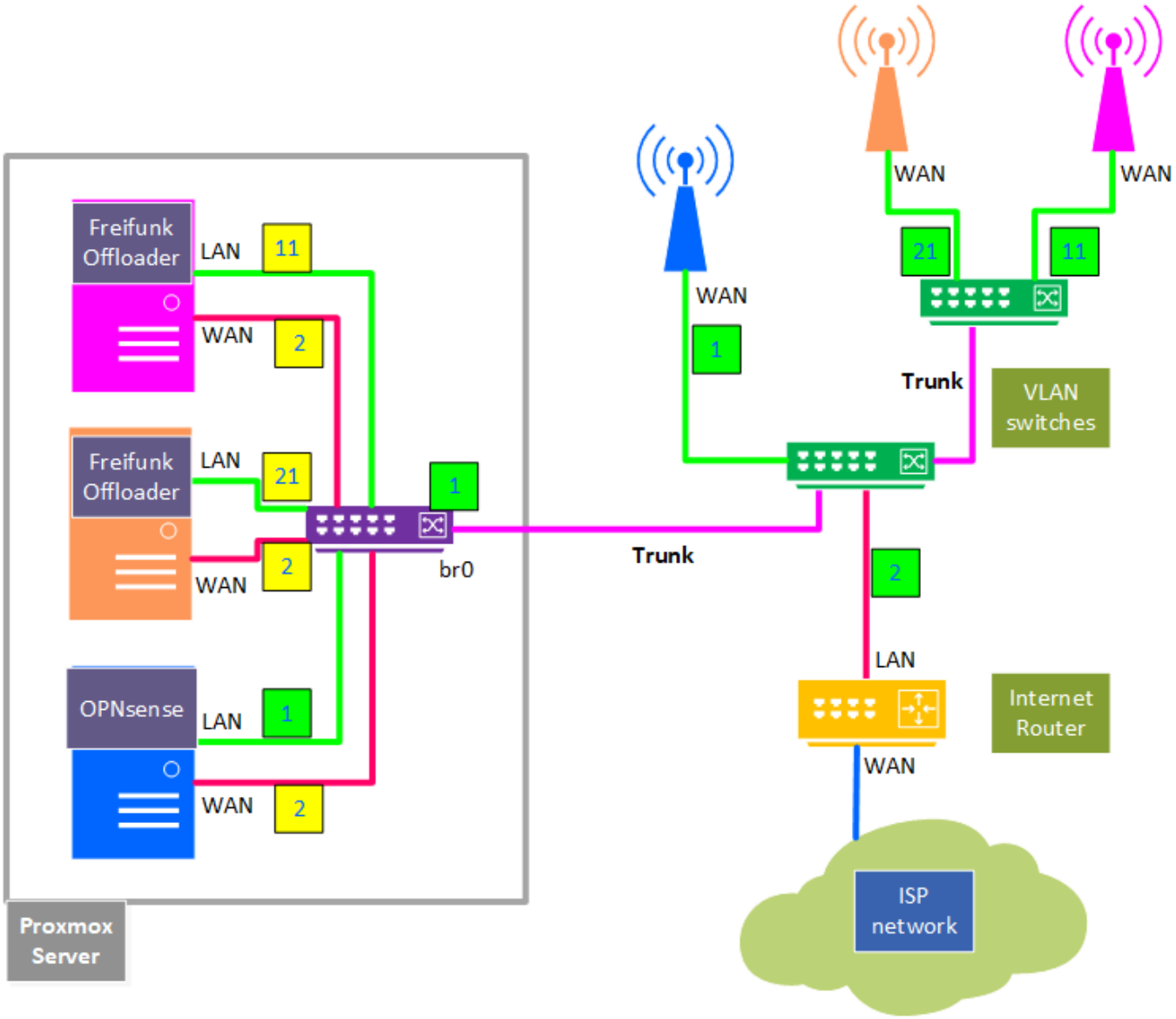
- **untagged** Ports (hier: *grün*) verpacken empfangene Pakete mit einer VID und entpacken die VID beim Senden.
- **tagged** Ports (hier: *gelb*) führen keine Änderung der VID bei ein- und ausgehenden Paketen durch.

Jeder Port lässt aber nur diejenigen Pakete passieren, für die er konfiguriert ist. Alle anderen Pakete werden ignoriert.

Sogenannte **Trunks** werden benutzt, um über **ein** physisches Kabel mehrere, unabhängige Datenströme zu transportieren.

Der virtuelle Switch `vmbr0` innerhalb von Proxmox-VE transportiert das untagged LAN mit der VID=1,
die tagged LANs mit den VIDs 11 und 21 und das tagged DMZ-WAN mit der VID=2.

Die beiden externen VLAN-Switches werden benutzt, um das DMZ-Netzwerk mit der VID=2 untagged an den Internet-Router anzuschliessen.
Und die VLANs mit den VIDs (1, 11, 21) werden ebenfalls untagged an die drei Access-Points weitergeleitet.



Planung der Netzwerk-Segmente

Verwendung	**VID**	**IPv4 Adressbereich**	**Geräte**
Default Client-LAN	1	192.168.103.0/24	**nuc3**, opnsns-nuc3, laptop, *jcg-ap*
DMZ Netzwerk	2	192.168.178.0/24	router-nfh, opnsns-nuc3, opnwrt-nuc3, ffsw-nfh, ffmuc-nfh, *ffws-duew-ap*
FF-sw LAN	11	10.210.48.0/20	ffsw-nfh, *edimax-ap*
FF-muc LAN	21	10.80.200.0/21	ffmuc-nfh, *tplink-ap*
OpenWRT LAN	31	192.168.223.0/24	**nuc3**, opnwrt-nuc3, dlink-sw08, *lede-ap*
Trunk1 NUC3—SG105	1,2,11,21,31	---	nuc3-vmbr0, tl-sg105-p1

Verwendung	VID	IPv4 Adressbereich	Geräte
Trunk2 SG105—GS108	1,2,11,21,31	---	tl-sg105-p3, ng-gs108-p8

Hardware für den Versuchsaufbau

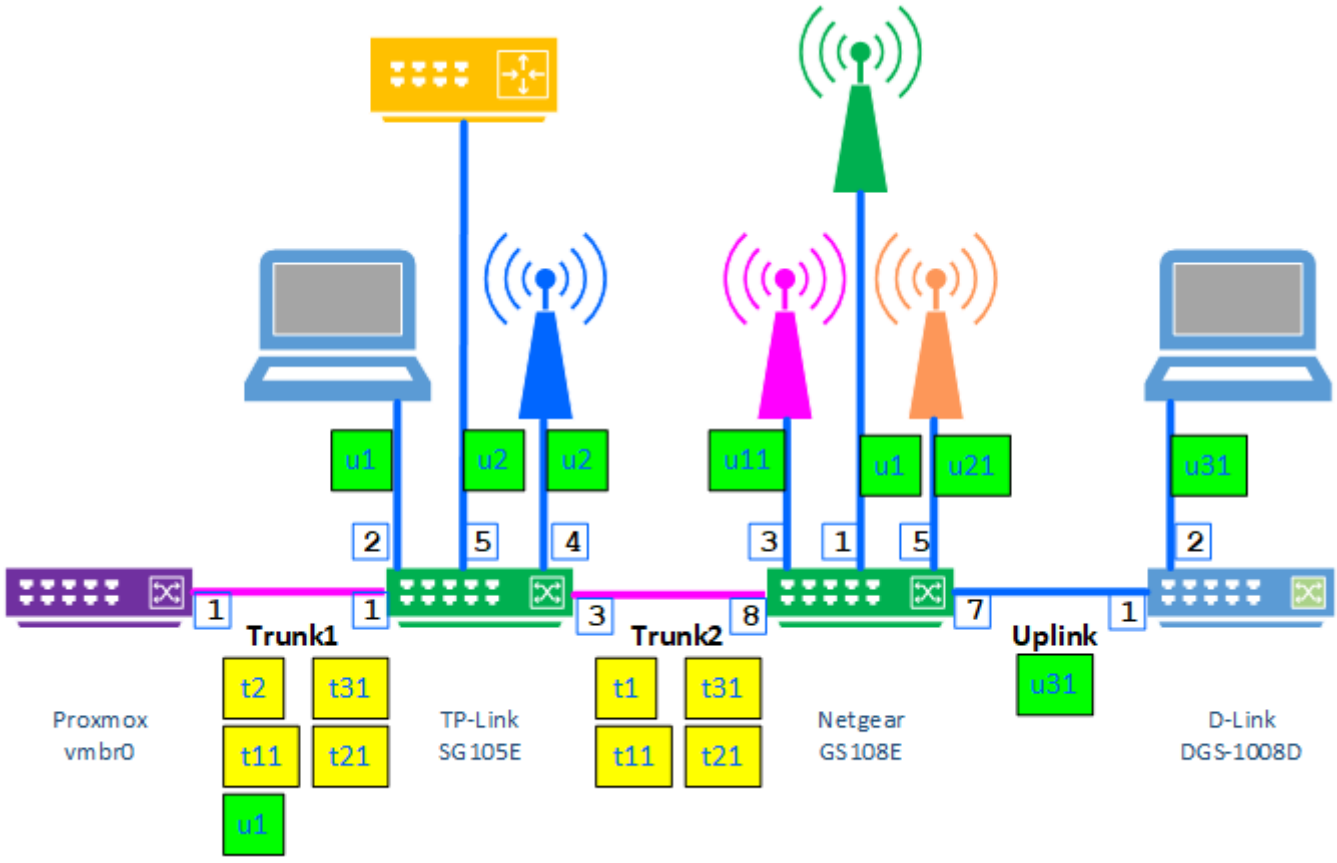
- 1x Intel NUC5i3 mit OS: Proxmox **PVE-8.0**
  Memory: 8GB
  Disk: 500GB SSD (M.2 NVMe)
  nur **1** NIC: 1 Gbit/s
- Internet-Router: Vorgabe des Providers
- 2x Web managed **802.1q** Switches
  5-port TP-Link *SG105E*
  8-port Netgear *GS108E*
- 1x unmanaged Desktop Switch
  8-port D-Link *DGS-1008D*
- 5x WLAN-Router / Access Points
  1x Edimax (BR-6428nS V5)
  1x NoName (JCG JHR-N805R)
  3x TP-Link TL-WR740N V4: Original-Firmware; OpenWRT-LEDE; Freifunk-Node

IP-Adressen und Hosts

VLAN-ID	Hostname	IPv4 Adresse
1	**nuc3**	192.168.103.99
1	opnsns-nuc3	**192.168.103.9**
1	tl-sg105-a	192.168.103.3
1	ng-gs108-b	192.168.103.4
1	*jcg-ap*	192.168.103.5
1	laptop	DHCP2 (192.168.103.x/24)
11	ffsw-nfh	DHCP4 (10.210.48.x/20)
11	*edimax-ap*	DHCP4 (10.210.48.x/20)
11	lx1-client	DHCP4 (10.210.48.x/20)
2	router-nfh	**192.168.178.1**

VLAN-ID	Hostname	IPv4 Adresse
2	opnsns-dmz	DHCP1 (192.168.178.x/24)
2	opnwrt-dmz	DHCP1 (192.168.178.x/24)
2	ffsw-nfh-dmz	DHCP1 (192.168.178.x/24)
2	ffmuc-nfh-dmz	DHCP1 (192.168.178.x/24)
2	*ffws-duew-ap*	DHCP1 (192.168.178.x/24)
21	ffmuc-nfh	DHCP5 (10.80.200.x/21)
21	*tplink-ap*	DHCP5 (10.80.200.x/21)
21	lx2-client	DHCP5 (10.80.200.x/21)
31	opnwrt-nuc3	**192.168.223.9**
31	**nuc3**	192.168.223.99
31	*lede-ap*	DHCP3 (192.168.223.x/24)
31	lx3-client	DHCP3 (192.168.223.x/24)

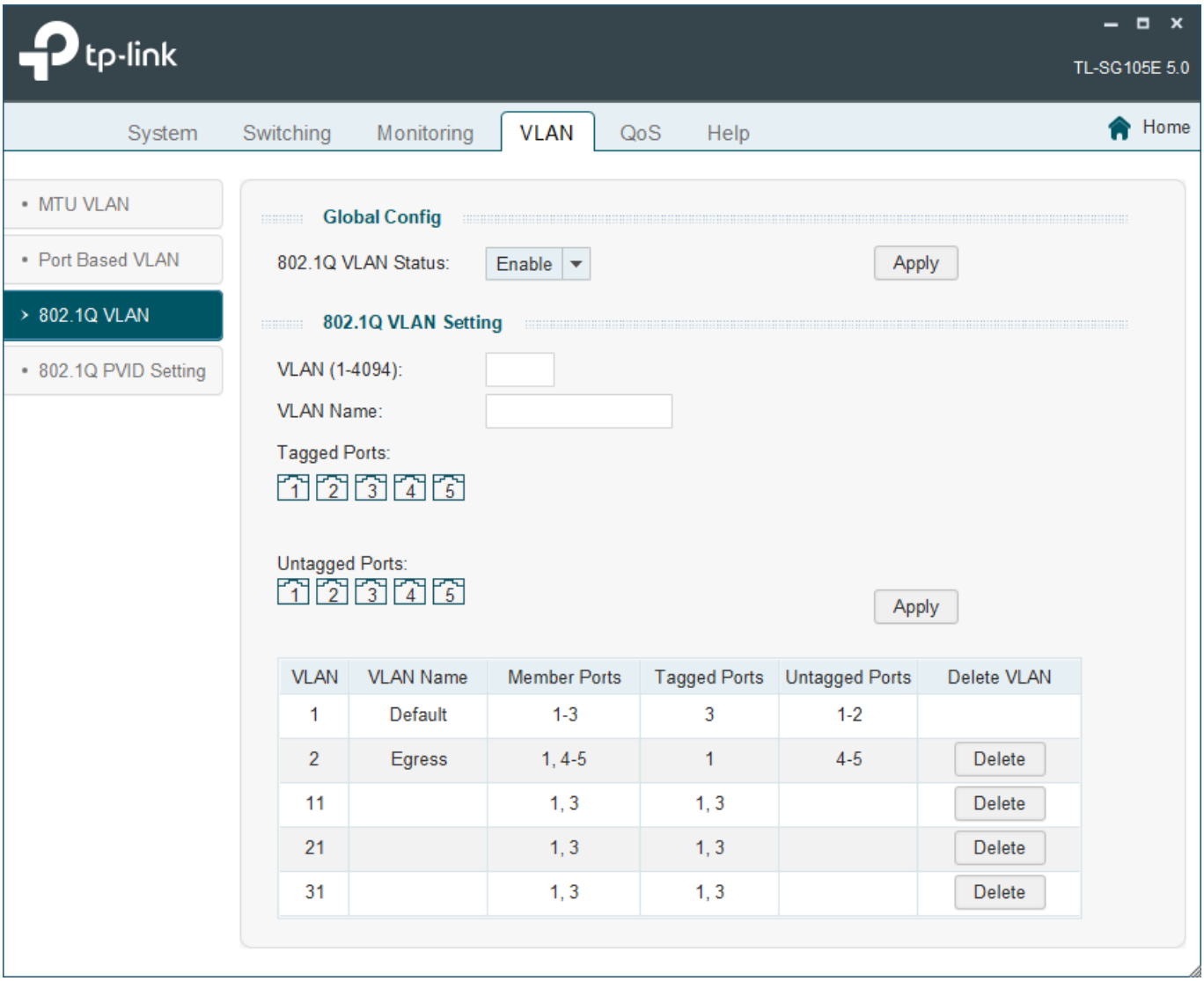
Details der Netzwerk-Verkabelung



Konfiguration TP-Link **SG105E**

via Web-Browser und interner Web-App

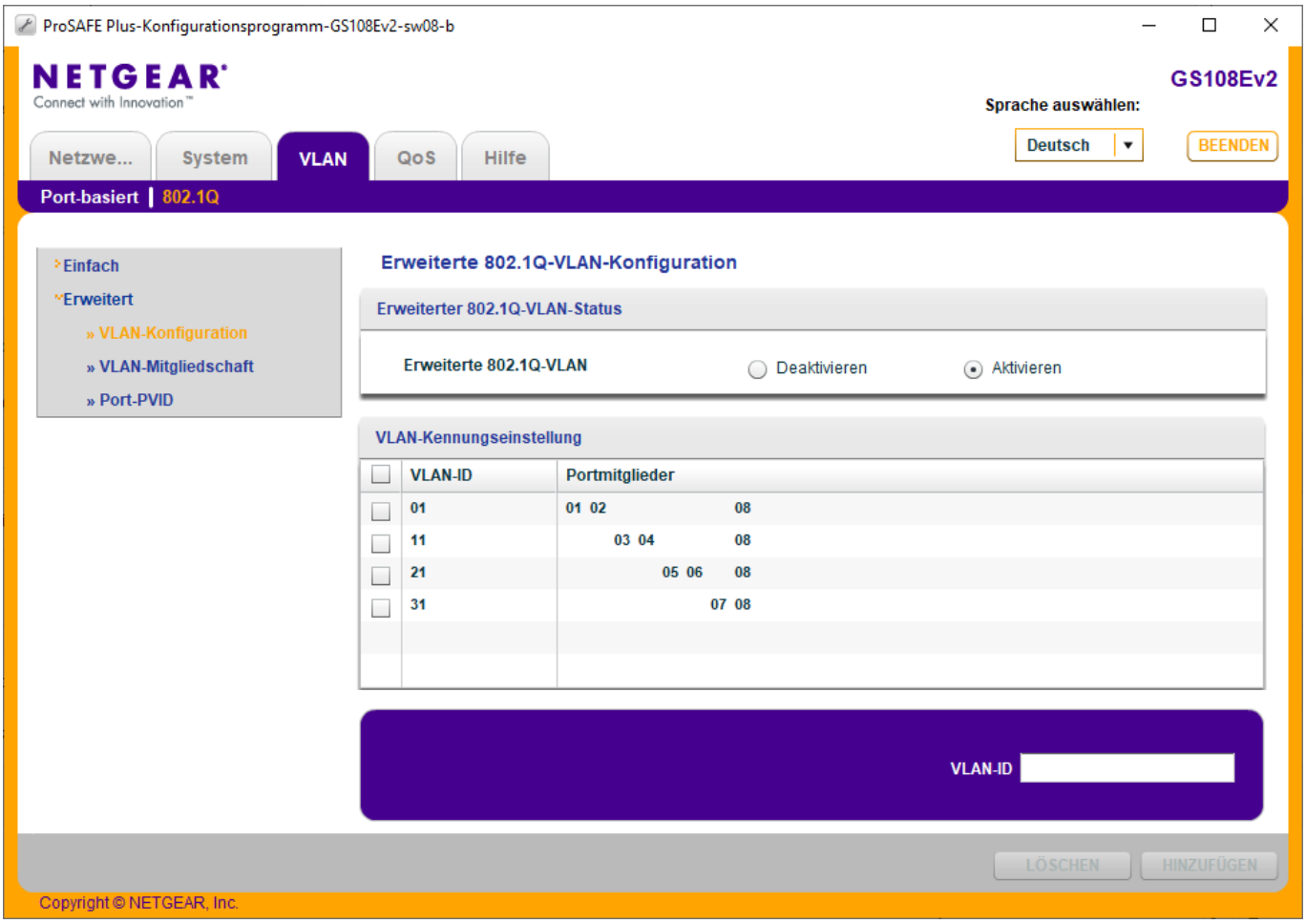
Ports	Untagged	Tagged	PVID	Device
1	1	2,11,21,31	1	**nuc3-vmbr0**
2	1		1	laptop
3		1,11,21,31	1	**ng-gs108-p8**
4	2		2	*ffws-ap*
5	2		2	rtr-nfh



Konfiguration Netgear ProSafe+ **GS108E**

via Windows-Application

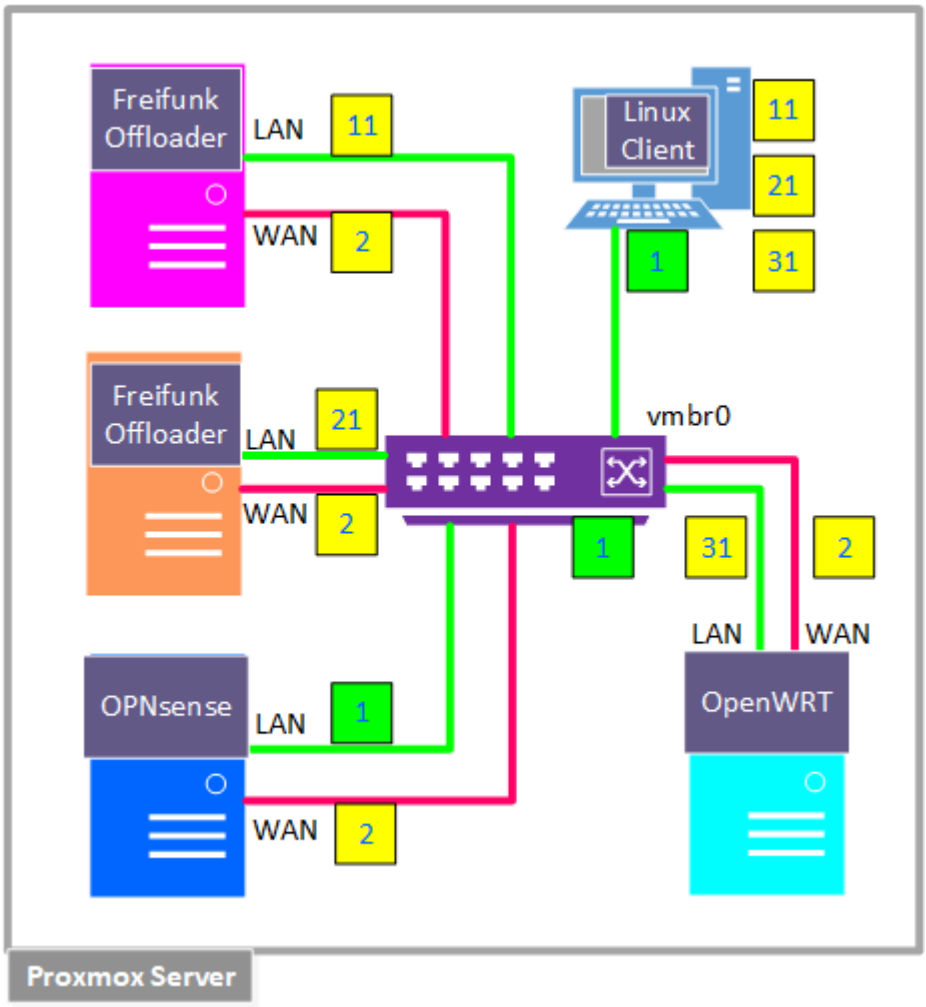
Ports	Untagged	Tagged	PVID	Device
1, 2	1		1	*jcg-ap*
3, 4	11		11	*edimax-ap*
5, 6	21		21	*tplink-ap*
7	31		31	dlink-sw08
8		1,11,21,31	1	**tl-sg105-p3**



Proxmox-Server im Überblick:

- 1 Linux Bridge `vmbr0`
- 5 Linux VLANs: u1, t2, t11, t21, t31
- 6 VMs:
  2x Firewall

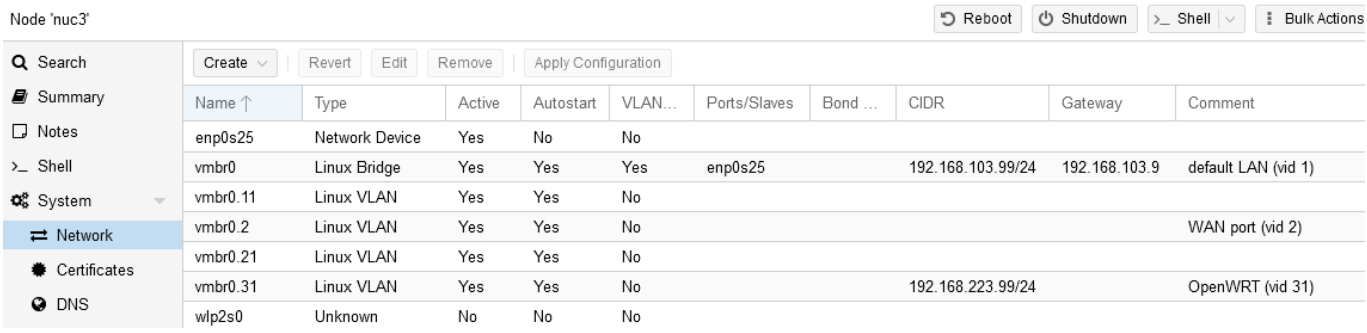
2x Linux-Client
2x FF-Offloader



Konfiguration **NUC3**: Bridge vmbr0

via Web-Browser und PVE-GUI

Ports	**Untagged**	**Tagged**	**Device**	**VM**
enp0s25	1	2,11,21,31	**tl-sg105-p1**	---
vmbr0	1		---	**nuc3**, opnsns
vmbr0.2		2	---	opnsns, opnwrt, ffsw-nfh, ffmuc-nfh
vmbr0.11		11	---	ffsw-nfh, lx1-client
vmbr0.21		21	---	ffmuc-nfh, lx2-client
vmbr0.31		31	---	**nuc3**, opnwrt, lx3-client



VLANs unter Linux: Details der Netzwerk-Konfiguration

Proxmox basiert auf Debian: das PVE-GUI erstellt die Datei `/etc/network/interfaces`

https://192.168.103.99:8006/?console=shell&xtermjs=1&vmid=0&vmname=&node=nuc3&cmd=

```
root@nuc3:~# cat /etc/network/interfaces
# network interface settings; autogenerated
# Please do NOT modify this file directly, unless you know what
# you're doing.
#
# If you want to manage parts of the network configuration manually,
# please utilize the 'source' or 'source-directory' directives to do
# so.
# PVE will preserve these directives, but will NOT read its network
# configuration from sourced files, so do not attempt to move any of
# the PVE managed interfaces into external files!

auto lo
iface lo inet loopback

iface enp0s25 inet manual

auto vmbr0
iface vmbr0 inet static
        address 192.168.103.99/24
        gateway 192.168.103.9
        bridge-ports enp0s25
        bridge-stp off
        bridge-fd 0
        bridge-vlan-aware yes
        bridge-vids 2-4094
#default LAN (vid 1)

iface wlp2s0 inet manual

auto vmbr0.2
iface vmbr0.2 inet manual
#WAN port (vid 2)

auto vmbr0.31
iface vmbr0.31 inet static
        address 192.168.223.99/24
#OpenWRT (vid 31)

auto vmbr0.11
iface vmbr0.11 inet manual

auto vmbr0.21
iface vmbr0.21 inet manual

root@nuc3:~#
```

Vorbereitung des Workshops

Hardware: Was wurde vorab gemacht?

- Installation **NUC5i3** mit Virtualisierer Proxmox PVE-7.3 (März 2023):
  https://pve.proxmox.com/wiki/Installation
    - Upload von Freifunk-Gluon- und OpenWRT-Images auf *NUC3*
    - Upgrade Proxmox auf neueste Version **PVE-8.0** (August 2023)
- Konfiguration der beiden VLAN-Switches:
  5-port TP-Link *SG105E*
  8-port Netgear ProSafe+ *GS108E*
- Einrichtung der VLANs **11** & **21** & **31**
- Konfiguration der 5 WLAN-Router als Access Points (APs)

Software: Was ist bereits erledigt?

- Installation VM OPNsense-23.1 (März 2023):
  https://www.sunnyvalley.io/docs/network-security-tutorials/opnsense-installation
  https://schulnetzkonzept.de/opnsense
    - Einrichtung DHCP und DNS auf LAN
      Einrichtung IPv6 auf WAN und LAN
    - Upgrade auf **OPNsense-23.7** (August 2023)
- Installation VM **OpenWRT 22.03.3**:
  siehe OpenWRT Tutorials von "Hoerli":
  Youtube-Playlist und Blog https://hoerli.net/category/openwrt/
- Installation von Freifunk-**Offloader** VMs:
  Freifunk-Weinstrasse (**ffsw-nfh**)
  Freifunk-München (**ffmuc-nfh**)
- Installation mehrerer Linux-Client-VMs
  Knoppix 9.1
  Porteus 5.5

Durchführung des Workshops

Was ist noch zu tun?

- **Tests** der Freifunk-Netzwerke mit Linux-, Windows-, MacOS-Clients
- Freifunk-**Offloader** VMs aktualisieren
- weitere Freifunk-Communities testen
- ? ? ?

DANKE für Euer Interesse!

Dieser Vortrag kann unter https://tech.dortoka.ipv64.de/talks/ nachgelesen werden.
