IFA
Institut für Arbeitsschutz der
Deutschen Gesetzlichen Unfallversicherung

Wireguard VPN

Vom Setup bis zur Livedemo an einer Industriesteuerung

FrOSCon
Matthias Weitz, Jonas Stein 21.08.2022

Struktur von Wireguard

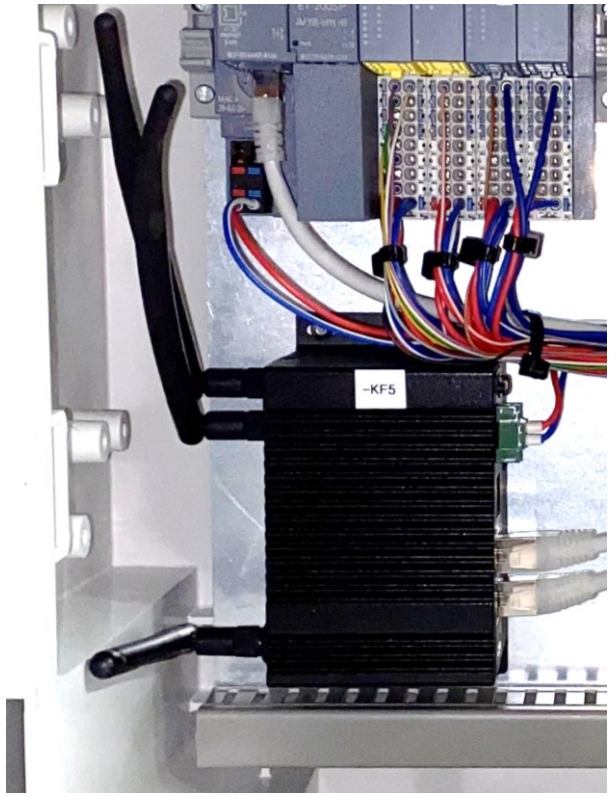
- Open Source VPN-Protokoll
- Idee/Konzept: Jason A. Donenfeld
- Sicherer Layer 3 Netzwerktunnel
  - IPv4
  - IPv6
- Kapselt Pakete in UDP
- Stromverschlüsselung: ChaCha20
- Schlüsselaustausch: ECDH-Curve25519
- Authentifikation Passwortlos

OSI-Modell

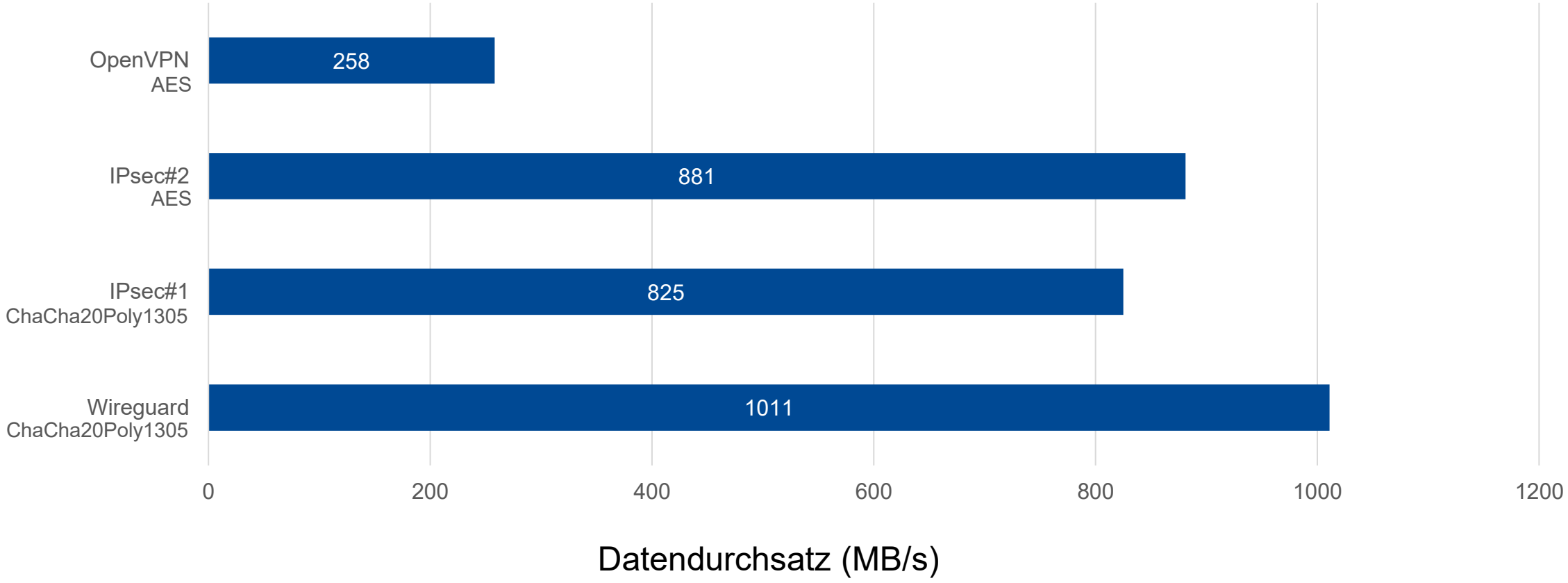
7. Darstellungsschicht
6. Darstellungsschicht
5. Sitzungsschicht
4. Transportschicht
3. Vermittlungsschicht
2. Sicherungsschicht
1. Bitübertragungsschicht

Plattformkompatibilität

- Linux
- Windows
- macOS
- BSD
- iOS
- Android
- WireguardNT in Entwicklung

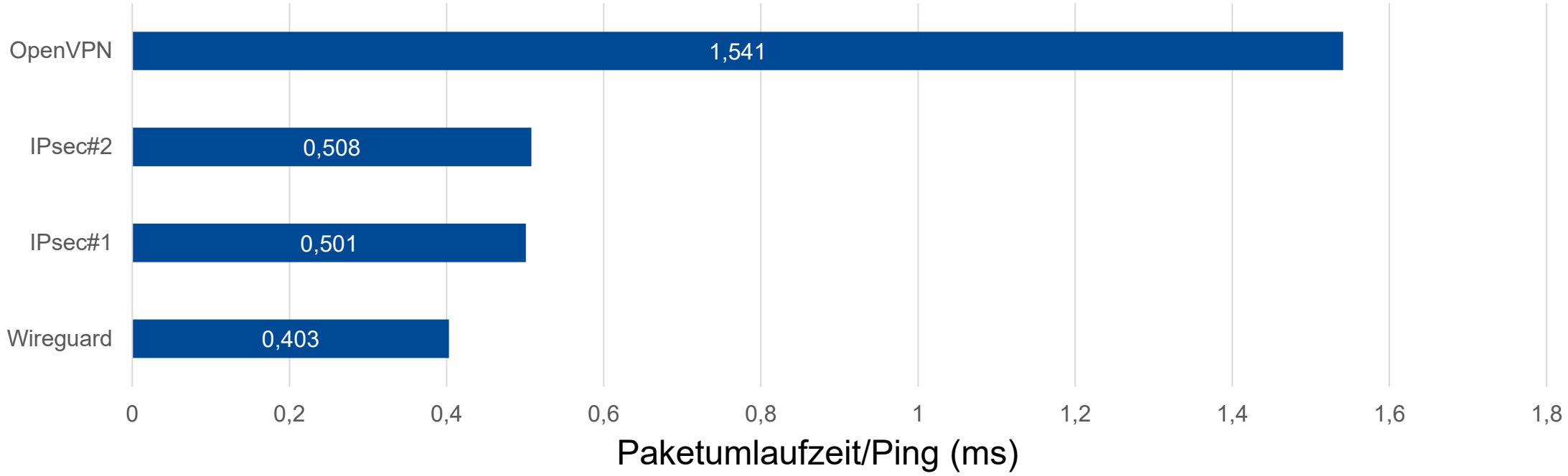


Datendurchsatz



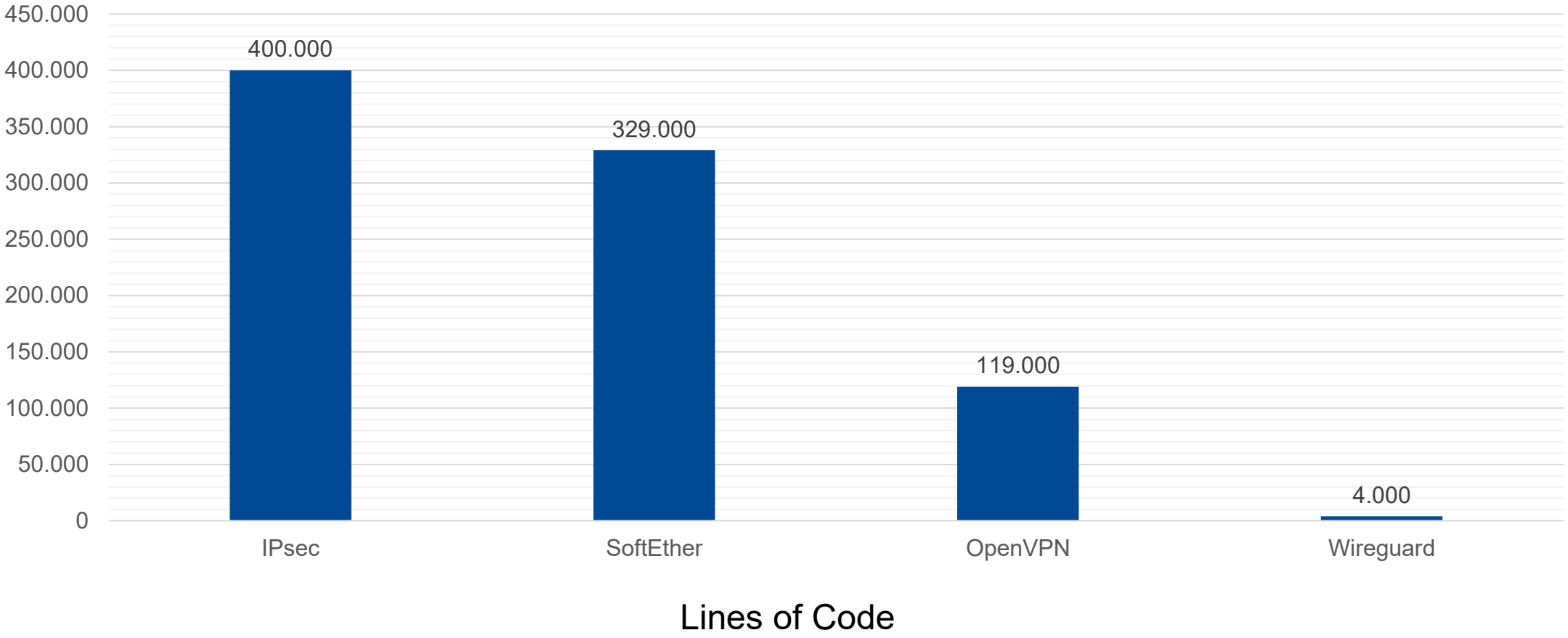
Datenquelle: https://www.wireguard.com/papers/wireguard.pdf

Paketumlaufzeit



Datenquelle: https://www.wireguard.com/papers/wireguard.pdf

Codeumfang


450.000
400.000
350.000
300.000
250.000
200.000
150.000
100.000
50.000
0
400.000
329.000
119.000
4.000
IPsec
SoftEther
OpenVPN
Wireguard
Lines of Code

Installation

- `user $ sudo apt-get install wireguard`
- Überprüft ob Kernelmodul vorhanden ist
- Installiert das Paket wireguard-tools
- Legt das Verzeichnis */etc/wireguard* an

Wireguard Konfiguration

- Ein Wireguard Interface hat:
  - Einen privaten Schlüssel
  - Eine Tunnel-IP-Adresse
  - Einen UDP-Port
  - Eine Liste von Peers
- Ein Peer:
  - Wird durch Public-Key identifiziert
  - Hat eine Liste von erlaubten Tunnel-IP-Adressen
  - Optional eine Endpoint-IP-Adresse und Port

Konfiguration über Konsole – Alice

```
root # wg genkey > private                    #Privaten Schlüssel erzeugen
root # wg pubkey < private                    #Öffentlichen Schlüssel erzeugen
PubKey…Al1ce                                 #wg gibt öffentlichen Schlüssel aus
root # ip link add wg0 type wireguard        #Wireguard-Interface anlegen
root # ip addr add 10.0.0.1/24 dev wg0       #IP für Wireguard-Interface zuweisen
root # wg set wg0 private-key ./private      #Privaten Schlüssel zuweisen
root # ip link set wg0 up                    #Wireguard-Interface aktivieren
root # wg set wg0 peer PubKey…B0b  allowed-ips 10.0.0.2/24 endpoint
192.168.10.11:55555
#Wireguard einen Peer zufügen. Mit Public-Key des Peers, im Tunnel erlaubter IP-
Adresse und Endpoint.
```

Konfiguration über Konsole – Bob

```
root # wg genkey > private                    #Privaten Schlüssel erzeugen
root # wg pubkey < private                    #Öffentlichen Schlüssel erzeugen
PubKey…B0b                                   #wg gibt öffentlichen Schlüssel aus
root # ip link add wg0 type wireguard        #Wireguard-Interface anlegen
root # ip addr add 10.0.0.2/24 dev wg0       #IP für Wireguard-interface zuweisen
root # wg set wg0 private-key ./private      #Privaten Schlüssel zuweisen
root # ip link set wg0 up                    #Wireguard-Interface aktivieren
root # wg set whg0 peer PubKey…Al1ce allowed-ips 10.0.0.1/24 endpoint
192.168.10.10:55555
#Wireguard einen Peer zufügen. Mit Public-Key des Peers, im Tunnel erlaubter IP-
Adresse und Endpoint
```

Konfiguration mit Konfigurationsdatei

- Script: wg-quick → Läd Konfigurationsdateien
- Konfigurationsdatei unter: /etc/wireguard
- Dateiname: wg0.conf
- Befehl zum Laden der Konfiguration: wg-quick up wg0

Konfiguration

Alice

```
root # cat /etc/wireguard/wg0.conf

[Interface]
Address = 10.0.0.1/24
ListenPort = 55555
PrivateKey = Al1ce…Key

[Peer]
PublicKey = PubKey…B0b
AllowedIPs = 10.0.0.2/32
Endpoint = 192.168.10.11:55555
```

Bob

```
root # cat /etc/wireguard/wg0.conf

[Interface]
Address = 10.0.0.2/24
ListenPort = 55555
PrivateKey = B0b…Key

[Peer]
PublicKey = PubKey…Al1ce
AllowedIPs = 10.0.0.1/32
Endpoint = 192.168.10.10:55555
```

Server-Konfiguration

Server

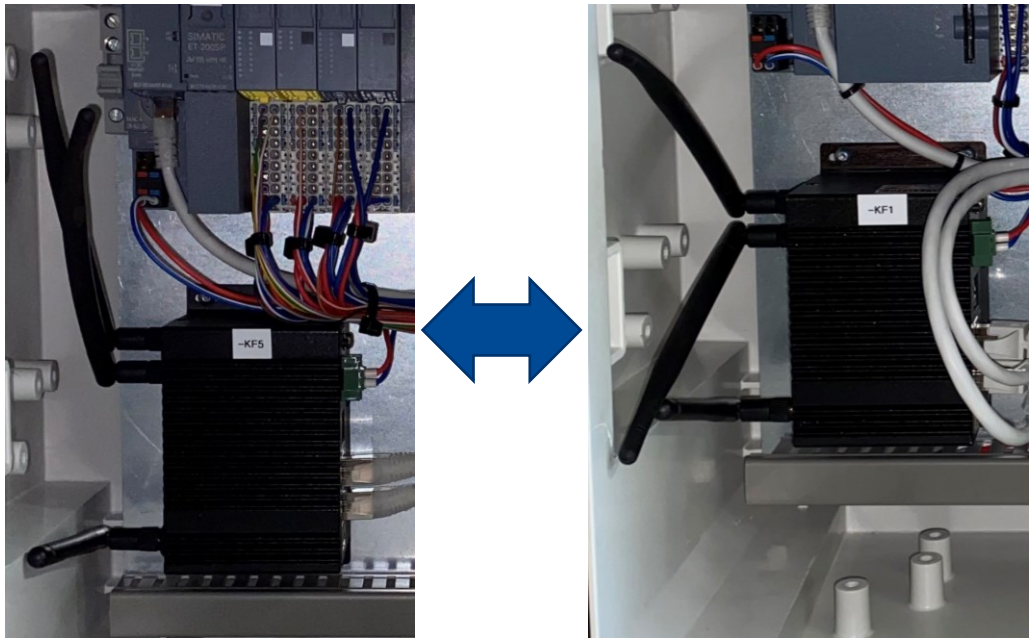
```
root # cat /etc/wireguard/wg0.conf

[Interface]
Address = 10.0.0.11/24
ListenPort = 55555
PrivateKey = Serv3r…Key

[Peer]
PublicKey = PubKey…Al1ce
AllowedIPs = 10.0.0.1/32, 10.0.100.0/24

[Peer]
PublicKey = PubKey...B0b
AllowedIPs = 10.0.0.2/32, 10.0.200.0/24
```

Alice

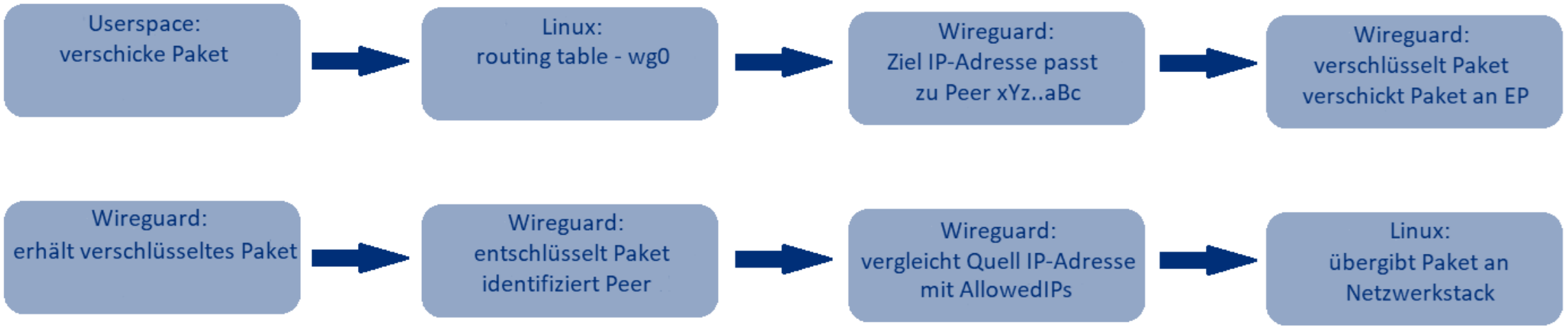
```
root # cat /etc/wireguard/wg0.conf


[Interface]
Address = 10.0.0.1/24
ListenPort = 55555
PrivateKey = Al1ce...Key

[Peer]
PublicKey = PubKey...Serv3r
AllowedIPs = 0.0.0.0/0
Endpoint = 192.168.0.1:55555
```

Roaming

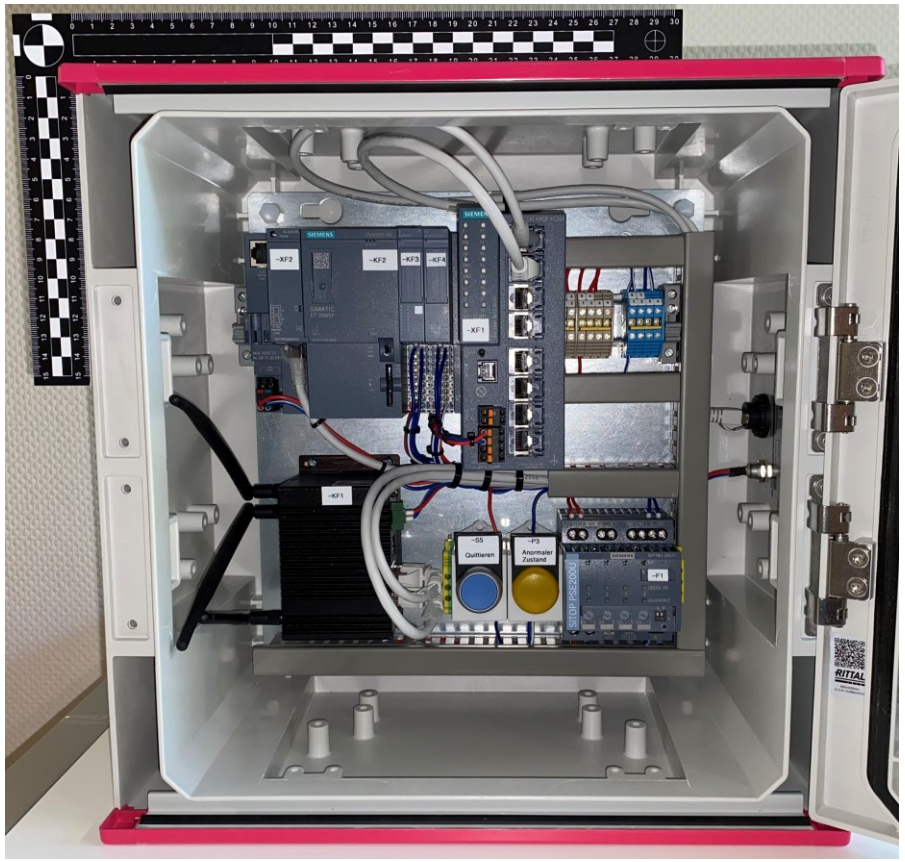
- Endpoints werden automatisch Aktualisiert
- Frei zwischen IP-Adressen bewegen
- Vom WLAN ins Mobilfunknetz
- Roadwarrior-Szenario

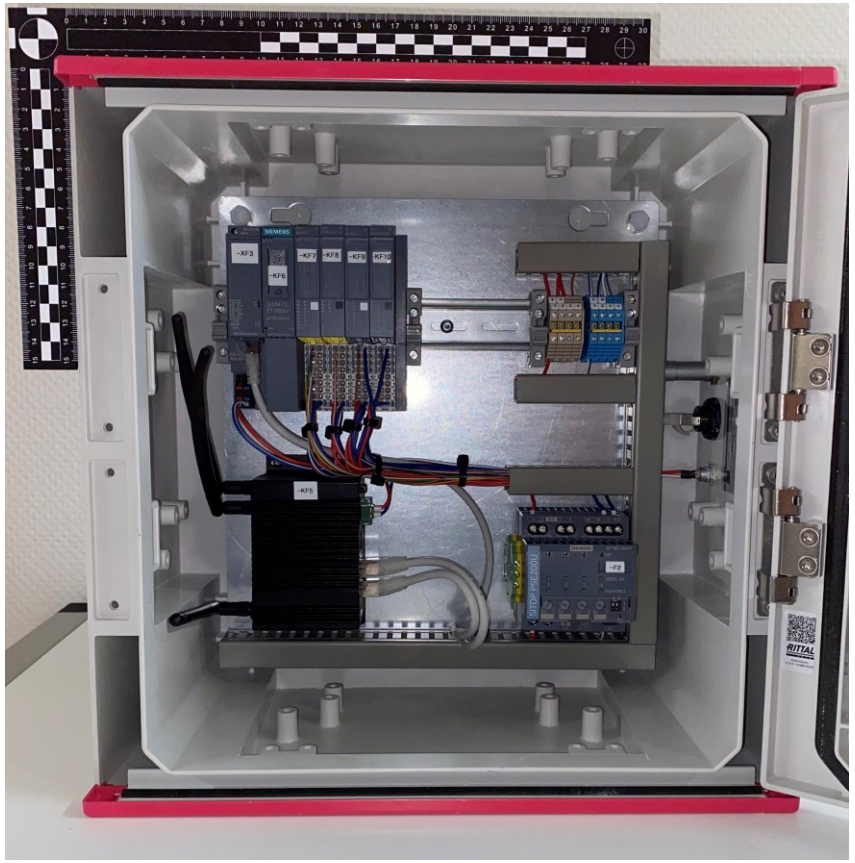


Kommunikation

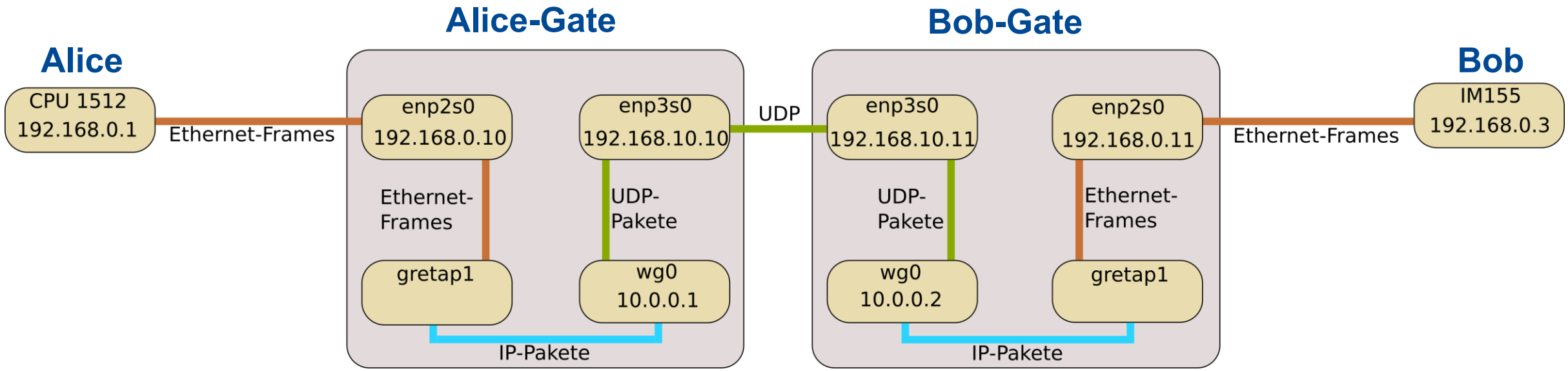


Anwendungsfall: Industriesteuerungen

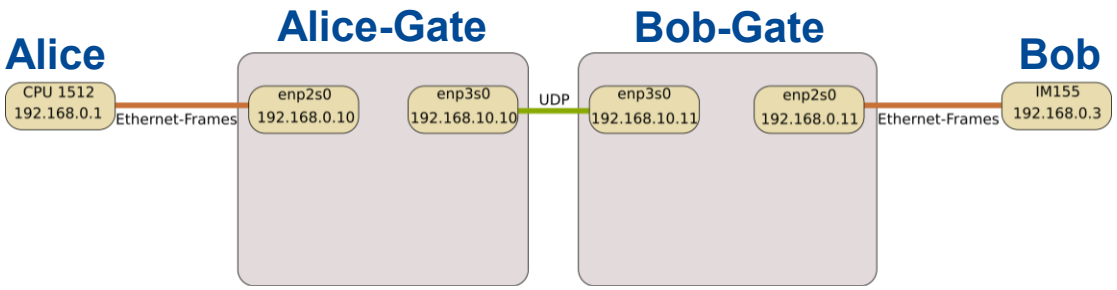




Netzwerkübersicht



Konfiguration der Ethernet-Ports

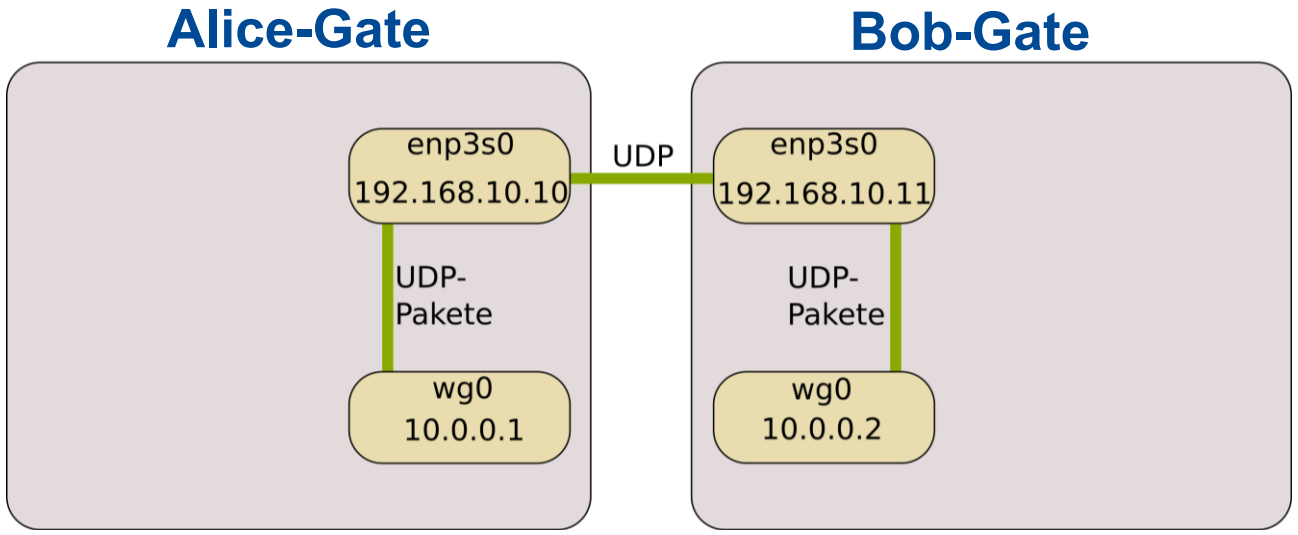


- enp2s0 Netz: 192.168.0.0/24
- enp3s0 Netz: 192.168.10.0/24
- */etc/network/interfaces*

```
root # cat /etc/network/interfaces

#Loopback
auto lo
iface inet loopback

#Interface für Wireguard-Kommunikation
auto enp3s0
iface enp3s0 inet static
address 192.168.10.10
netmask 255.255.255.0

#Interface für CPU-Kommunikation
auto enp2s0
iface enp2s0 inet static
address 192.168.0.10
netmask 255.255.255.0
```

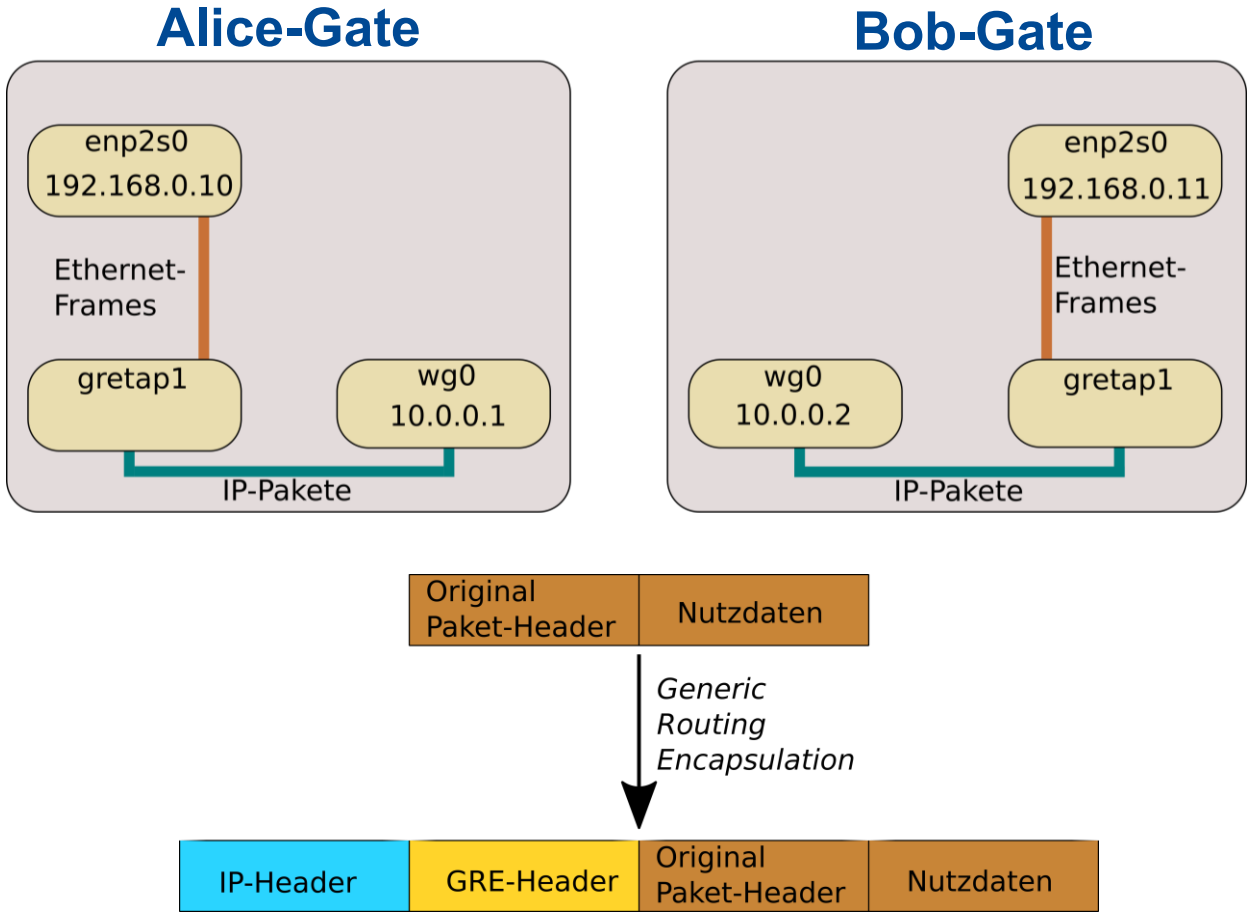
Sicherer VPN Tunnel

```
root # cat /etc/wireguard/wg0.conf

[Interface]
Address = 10.0.0.1/24
Listenport = 55555
PrivateKey = Al1ce...Key

[Peer]
PublicKey = PubKey..B0b
AllowedIPs = 10.0.0.2/32
Endpoint = 192.168.10.11:55555
```



Layer 2 Forwarding

- Ethernet-Frame → IP-Paket
- Generic Routing Encapsulation
- Ethernet-Bridge



Layer 2 Forwarding

- PreUp, PreDown, PostUp, PostDown
- ip Befehle aus dem Paket *iproute2*
- brctl Befehle aus dem Paket *bridge-utilitys*

```
root # cat /etc/wireguard/wg0.conf

[Interface]
Address = 10.0.0.1/24
Listenport = 55555
PrivateKey = Al1ce...Key

#Gretap Konfiguration
PreUp = ip link add gretap1 type gretap local 10.0.0.1 remote 10.0.0.2
PreUp = ip link set gretap1 up

#Ethernet-Bridge Konfiguration
PreUp = brctl addbr br0
PreUp = brctl addif br0 enp2s0
PreUp = brctl addif br0 gretap1
PreUp = ip link set br0 up
PreUp = echo 16384>/sys/class/net/bridge/group_fwd_mask

[Peer]
PublicKey = PubKey...Bob
AllowedIPs = 10.0.0.2/32
Endpoint = 192.168.10.11:55555
```

Zusammenfassung

- Sicherer Tunnel
- Einfache Bedienung
- Hoch performant
- Überprüfbarer Code
- Eingebautes Roaming
- Zukünftig mehr Kernel Portierungen

LLDP-Frames
(Link Layer Discovery Protokoll)

- Kommunikationsaufbau der Siemensgeräte
- Ziel-MAC-Adresse: 01:80:C2:00:00:0E
- Von der Ethernet-Bridge gefiltert
- Bitmaske verhindert Filtern

MAC	0F	0E	0D	0C	0B	0A	09	08	07	06	05	04	03	02	01	00
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bitmaske: 0100000000000000 → 2^{14} = 16384