

Getränkeautomat mit dem Raspberry Pi

Mobile Kommunikationsnetze

NOAH BALSIGER, THOMAS ZWICKER

2026-05-12

Inhaltsverzeichnis

Was ist M2M?	2
Wie funktioniert ein Mobilfunknetz? ...	3
4G vs. 5G: technische Unterschiede	4
Was bringt 5G praktisch?	5
5G NSA vs. SA	6
Bedeutung für unser Projekt	7
Öffentliche IP & Carrier-Grade NAT (CGNAT)	8
VPN Wireguard / Tailscale	9
Dynamic DNS	10
MQTT Sicherheit & Externer Broker ..	11
GPS	12
Hardware & Aufbau	13

Probleme	14
Demo	15

Was ist M2M?



Automatische Bestellung



Wie funktioniert ein Mobilfunknetz?

Grundidee: Funkzelle statt Kabel

- Das Endgerät sendet per Funk zur nächsten Basisstation
- Die Basisstation versorgt eine bestimmte Funkzelle
- Mehrere Funkzellen zusammen ergeben die Netzabdeckung
- Beim Bewegen wechselt das Gerät automatisch die Zelle: **Handover**

Datenweg im Normalfall

UE <-> Basisstation <-> Kernnetz <-> Internet

- **UE** = Endgerät, zum Beispiel Smartphone oder Raspberry Pi mit LTE-Modul
- **Kernnetz** = zentraler Netzteil des Providers
- SIM-Karte dient zur Anmeldung und Authentifizierung

Wichtig: 4G und 5G unterscheiden sich nicht nur bei der Funkverbindung, sondern auch beim Kernnetz dahinter. Dieses Kernnetz entscheidet mit, welche Funktionen wirklich verfügbar sind.

4G vs. 5G: technische Unterschiede

Eigenschaft	4G / LTE	5G NR
Funkstandard	LTE = Long Term Evolution	NR = New Radio, neuer 5G-Funkstandard
Basisstation	eNodeB = 4G-Antenne	gNodeB = 5G-Antenne
Kernnetz	EPC = 4G-Kernnetz	5GC = 5G-Kernnetz oder weiterhin EPC
Latenz	typisch 20–50 ms	tiefer möglich, besonders mit 5G SA
Bandbreite	bis ca. 20 MHz pro Träger	bis 100 MHz unter 6 GHz, mehr bei mmWave
Antennen	MIMO = mehrere Antennen	massive MIMO und Beamforming
Ziel	mobiles Breitband	schnelles Internet, tiefe Latenz, viele IoT-Geräte

Merksatz: 5G ist nicht nur „schnelleres 4G“. 5G bringt einen neuen Funkstandard, mehr nutzbare Bandbreite, bessere Antennentechnik und optional ein neues Kernnetz.

Einfach erklärt: Latenz = Reaktionszeit, Bandbreite = Datenmenge pro Zeit, mmWave = sehr hohe Frequenzen, IoT = vernetzte Geräte, Beamforming = gezieltes Senden in Richtung des Geräts.

Was bringt 5G praktisch?

eMBB

Enhanced Mobile Broadband

- sehr hohe Datenraten
- Streaming, Downloads, Hotspots
- wichtigste Funktion für normale Smartphones

URLLC

Ultra-Reliable Low-Latency Communication

- sehr kurze Reaktionszeit
- hohe Zuverlässigkeit
- Industrie, Robotik, Fahrzeuge

mMTC

Massive Machine-Type Communication

- sehr viele kleine Geräte
- Sensoren und IoT
- wenig Daten pro Gerät, aber sehr viele Teilnehmer

Wichtig für die Praxis: Nicht jedes 5G-Netz bietet automatisch alle Vorteile. Geschwindigkeit und Latenz hängen von Frequenzbereich, Empfang, Auslastung, Endgerät und davon ab, ob NSA oder SA verwendet wird.

NSA = Non-Standalone (mit 4G)

- 5G NR wird zusätzlich zu LTE verwendet
- Steuerung läuft meist über LTE
- Kernnetz bleibt oft das 4G-Kernnetz: **EPC**
- schneller Ausbau, aber nicht alle 5G-Funktionen

UE -> LTE eNodeB + 5G gNodeB -> EPC

SA = Standalone (eigenständig)

- 5G NR arbeitet direkt mit dem 5G-Kernnetz
- Kernnetz ist das **5GC**
- niedrigere Latenz und Network Slicing möglich
- keine LTE-Ankerverbindung nötig

UE -> 5G gNodeB -> 5GC

Abkürzungen: UE = Endgerät, eNodeB = 4G-Basisstation, gNodeB = 5G-Basisstation, EPC = 4G-Kernnetz, 5GC = 5G-Kernnetz. Network Slicing = ein physisches Netz wird logisch in mehrere virtuelle Netze aufgeteilt.

Was Mobilfunk dem Raspberry Pi gibt:

- Internetzugang über 4G
- Verbindung läuft über das Netz des Providers
- gut für mobile Geräte und Standorte ohne WLAN
- IP-Adresse kann wechseln

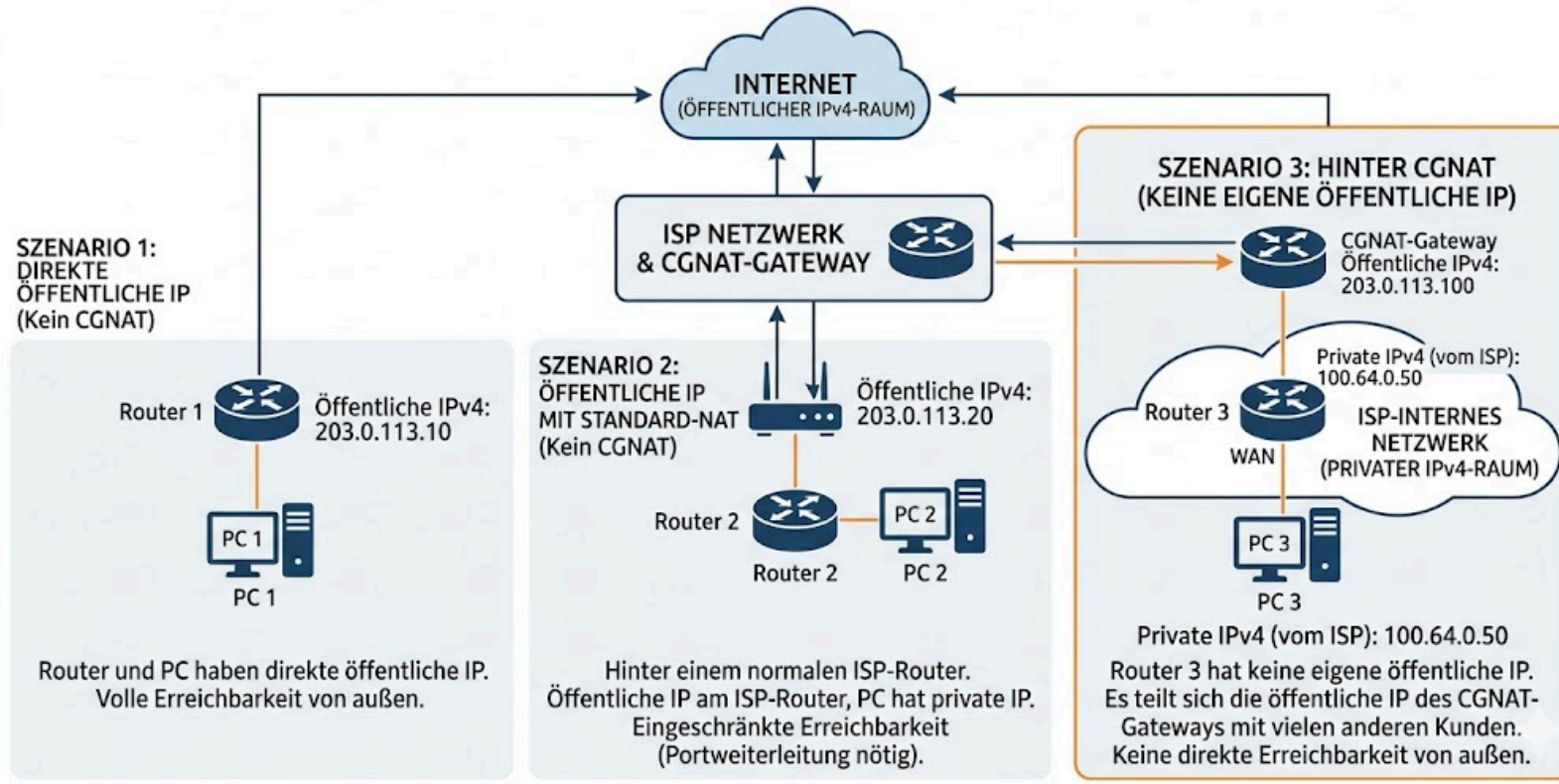
Was Mobilfunk nicht automatisch gibt:

- keine direkt erreichbare öffentliche IPv4-Adresse
- eingehende Verbindungen sind oft blockiert
- Zugriff von aussen ist dadurch schwieriger
- deshalb braucht es andere Lösungen

Folge für den weiteren Aufbau: Externer MQTT-Broker, VPN/WireGuard/Tailscale und eventuell DDNS werden wichtig, weil der Pi nicht einfach direkt aus dem Internet erreichbar ist.

Hinweis: Normales öffentliches 5G ist kein Mesh-Netz. Smartphones leiten normalerweise keine Daten anderer Smartphones zur Antenne weiter.

FUNKTIONSWEISE VON CGNAT (CARRIER GRADE NAT)

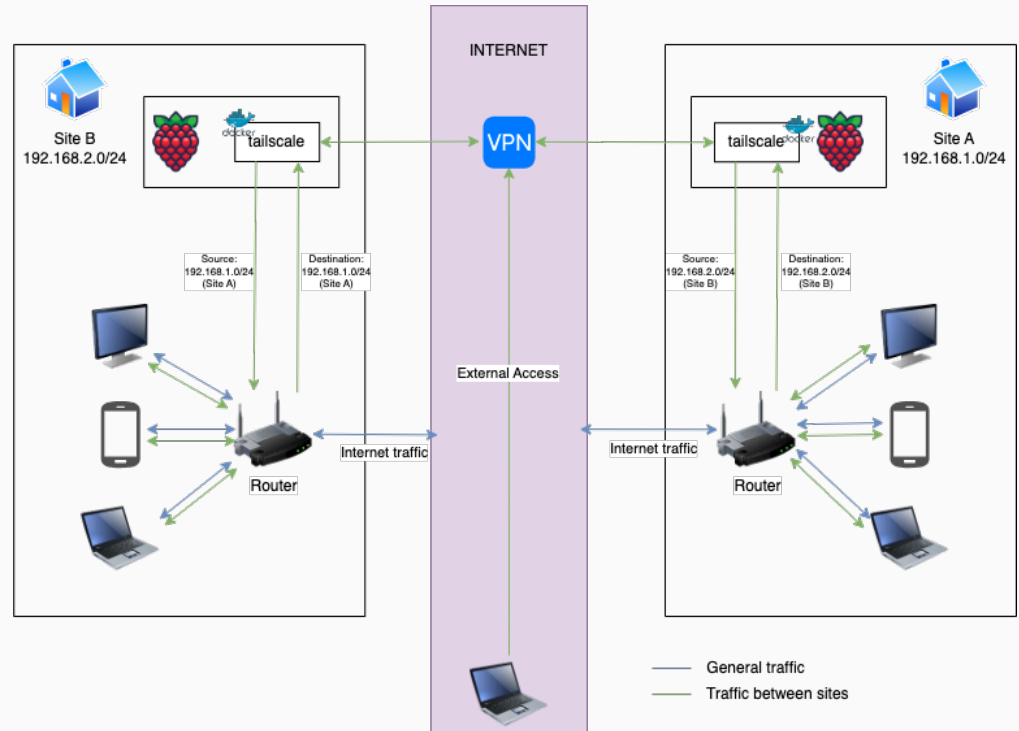


Herausforderung:

- Pi hat keine öffentliche IPv4-Adresse (CGNAT)

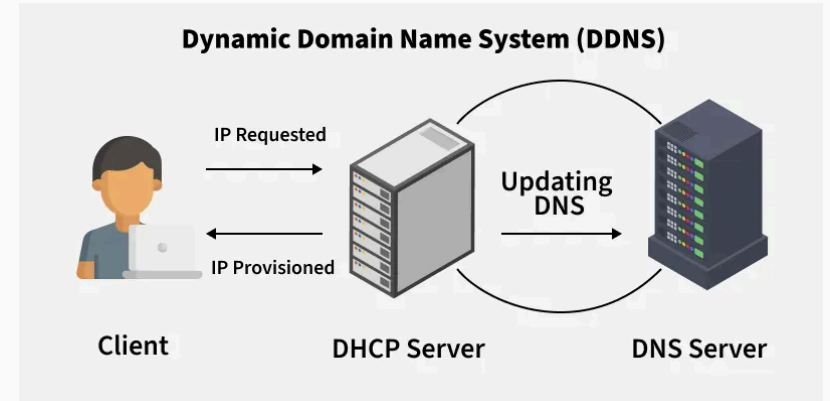
Lösung: Wireguard

- **Peer-to-Peer:** Pi baut Verbindung aktiv nach aussen auf
- **UDP-basiert:** Effizient und schnell auf mobilen Verbindungen
- **Roaming:** Verbindung bleibt stabil, auch wenn die Funkzelle wechselt



Dynamic DNS

- Mobilfunk-IPs sind meist dynamisch und wechseln oft
- **Dynamic DNS (DDNS):** Verknüpft wechselnde IP mit festem Hostnamen
- **Vorteil:** Einfacher Zugriff über `mobkom.xxx.de` statt Kryptischer IP-Adressen



Transport-Ebene (TLS/SSL)

- Verschlüsselung des gesamten Kanals
- Schützt auch Metadaten (Topics)
- Zertifikatsmanagement erforderlich
- Sicher gegen Man-in-the-Middle

Daten-Ebene (Payload)

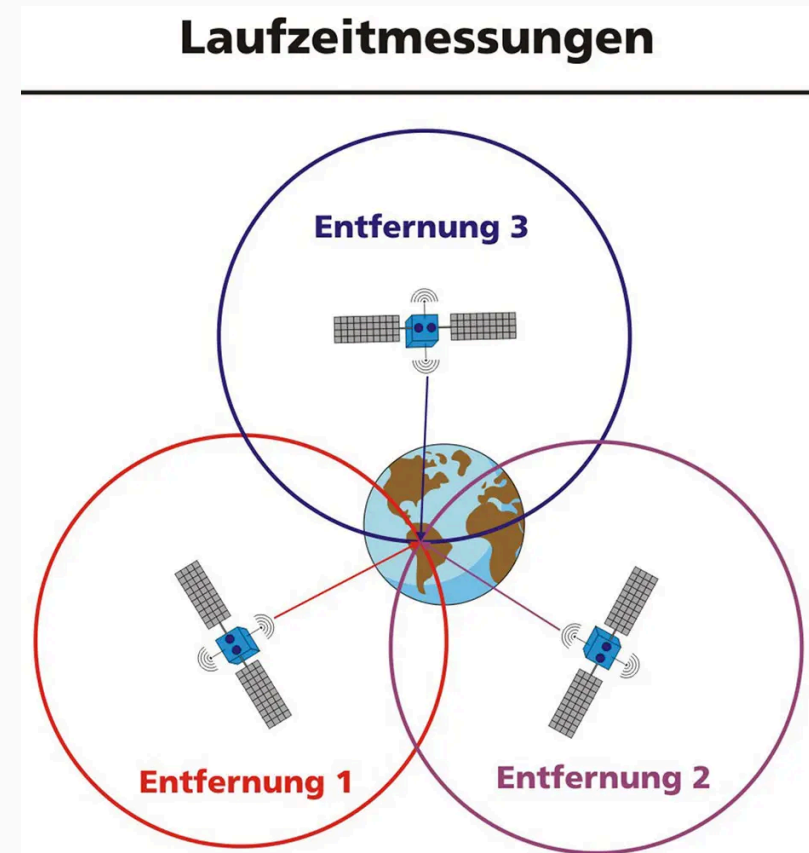
- Verschlüsselung direkt in Python (AES)
- Schutz vor kompromittierten Brokern
- „Zero Trust“: Broker sieht nur Chiffre
- Sehr geringer Overhead

Sicherung des externen Brokers:

- **Authentifizierung:** Kein Zugriff ohne gültige Credentials
- **ACLs (Access Control Lists):** Pi darf nur auf /getraenke/status schreiben
- **Vorteil Extern:** Erreichbarkeit ohne Port-Forwarding/VPN-Zwang für Clients

Ortung via LTE-Modul

- Integriertes GNSS-Modul auf dem Waveshare HAT
- **Anwendungsfall:** Tracking des Automaten oder Diebstahlschutz
- **Herausforderung:** GPS-Antenne benötigt meist Sichtkontakt zum Himmel



Die technischen Tücken

Anbindung via GPIO (UART)	Anbindung via USB
Einfaches Aufstecken	Zusätzliches Kabel nötig
Max. 115'200 Baud (0.1 Mbit/s)	WWAN-Interface (bis 150 Mbit/s)
GPS/LTE teilen sich serielle Ressource	Parallele Datenströme möglich
Nur für kleine Sensordaten (MQTT)	Massentauglich (Streaming/Updates)

Wichtig: Ohne USB-Kabel können GPS und LTE nicht simultan mit voller Performance genutzt werden.

MQTT

```
/etc/mosquitto/conf.d/websockets.conf
```

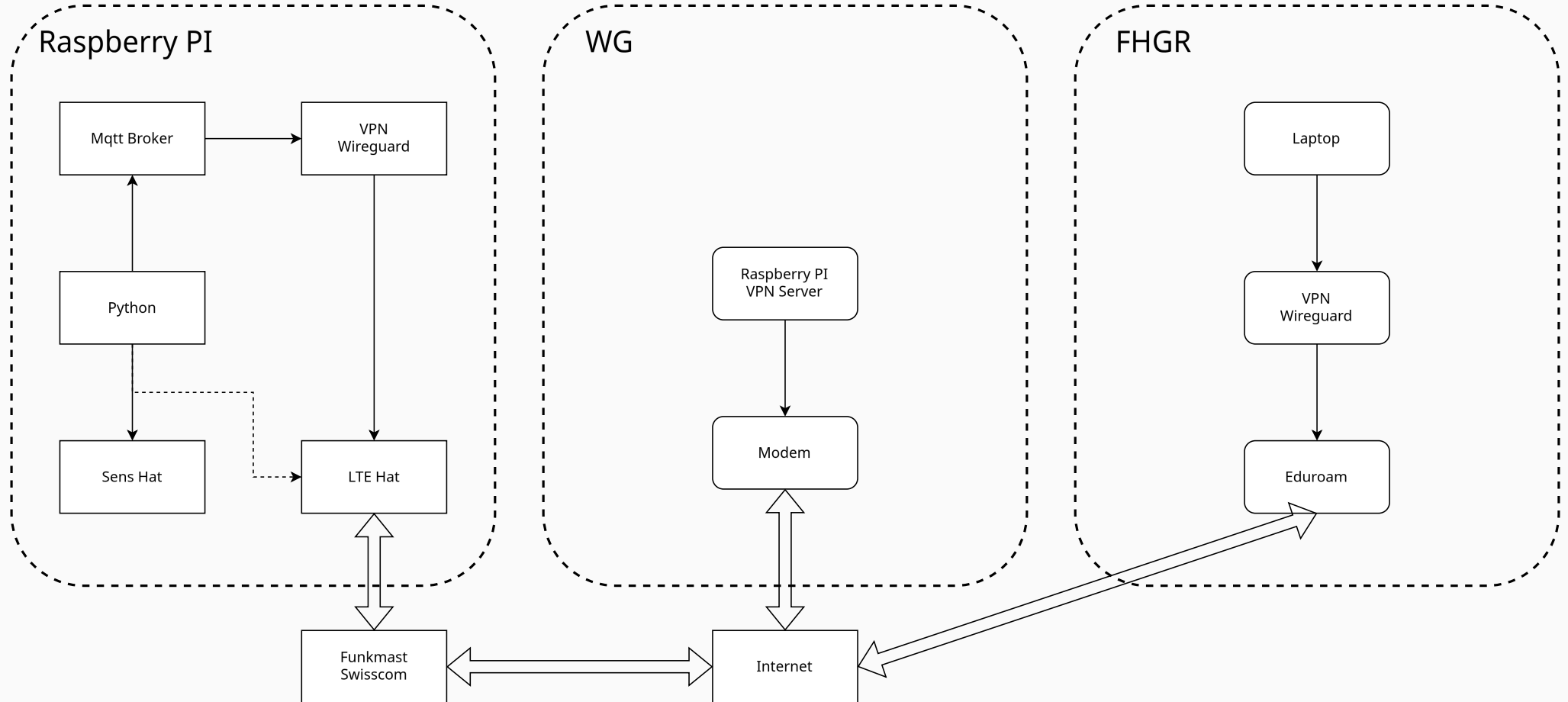
```
listener 9001 0.0.0.0  
protocol websockets  
allow_anonymous true
```

Ohne 0.0.0.0 nur local ab ver. 2.0.

IMU

- Sehr Träge
- Eigene berechnung über x, y, z

Demo



Danke fürs Zuhören
Fragen?