

Der vernetzte Camper

Erfahrungen in der WoMo-Automatisierung



Anwendungsbeispiel :)



About Me

- Promovierter Informatiker
- Security Specialist, yes.com
- Initiator Pi and More

Website: danielfett.de



TITLE

PIC1

MAIN

PIC2



Ziele und Nicht-Ziele



Herausforderungen

- Stromversorgung
- Kommunikation
- Beschränkter Platz, komplizierte Kabelwege
- Zuverlässigkeit für essentielle Funktionen



Sensoren und Aktoren

- Wasserstand Frischwassertank
- Stromverbrauch, Solar, Restkapazitäten
- Türstatus
- Alarmanlage
 - Türen
 - Fenster
 - Stromkabel
- Geschwindigkeitsanzeige
- Temperaturen
 - Außen
 - Wassersystem [Winter!]
 - Kühlschrank
- CO- und Feuersalarm
- Wasserlecksensor
- Lagesensor
- Außenlicht
- Reifendrücke
- Heizungssteuerung
- Gasreserven
- Kamera Innenraum



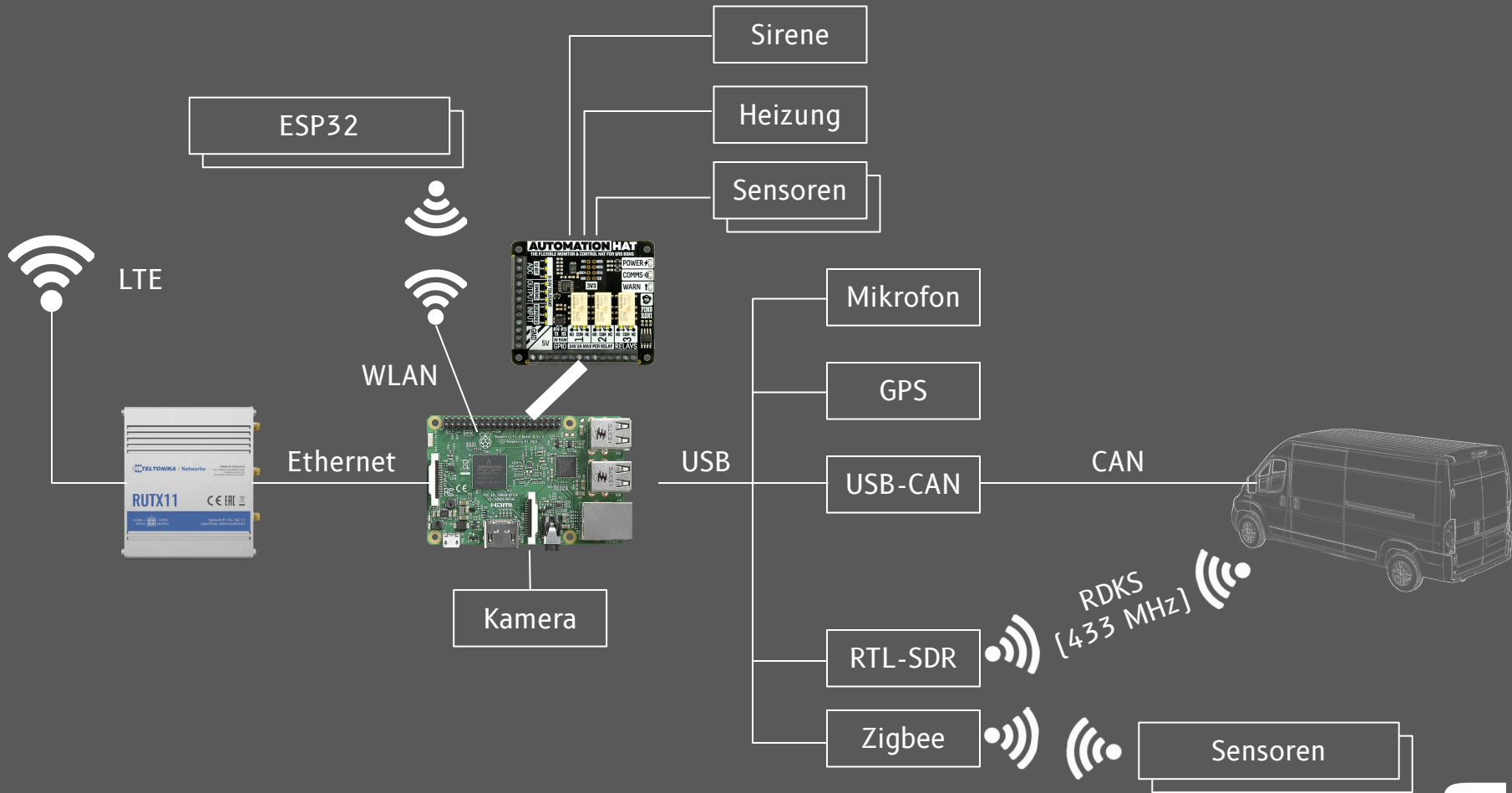
Mein Ansatz

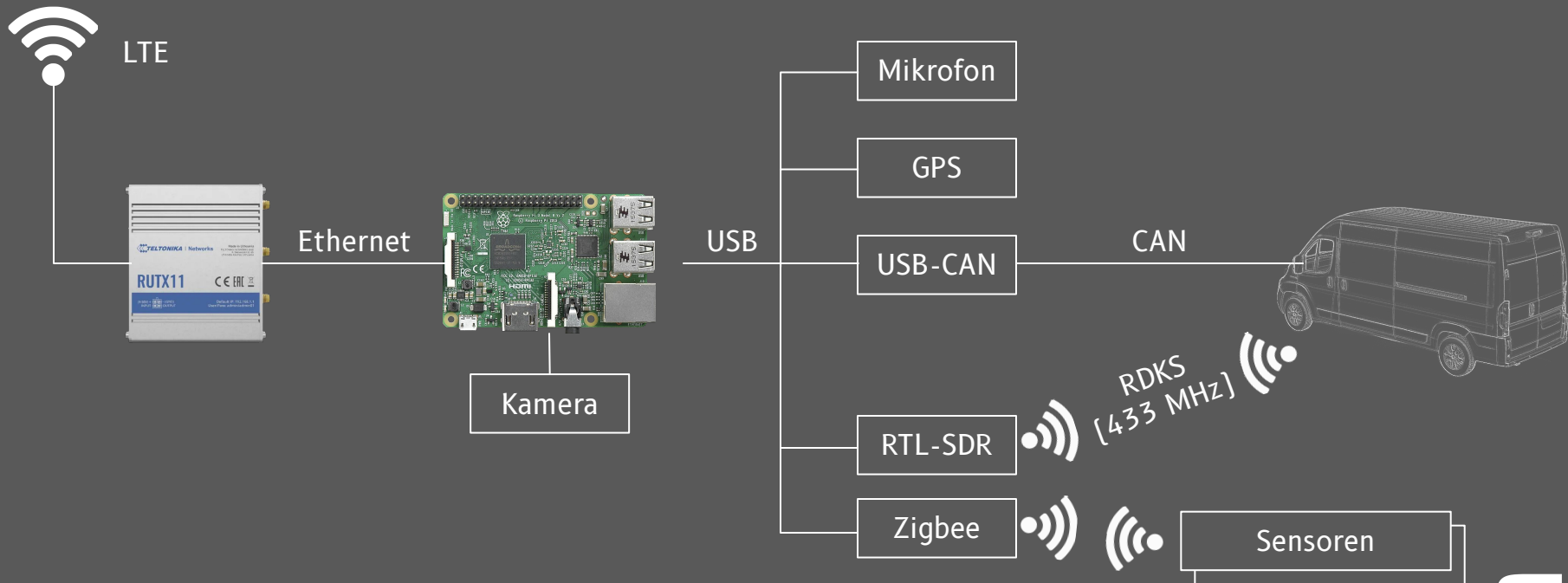
- Inkrementeller Aufbau
- Möglichst unabhängige Komponenten
- Wesentliche Funktionen ohne Automatisierung nutzbar
 - Heizung
 - Elektrik
 - Kühlschrank
 - Wasser
 - Gas
- Günstige Komponenten von der Stange
- So unsichtbar wie möglich — Raumschiff-Look vermeiden

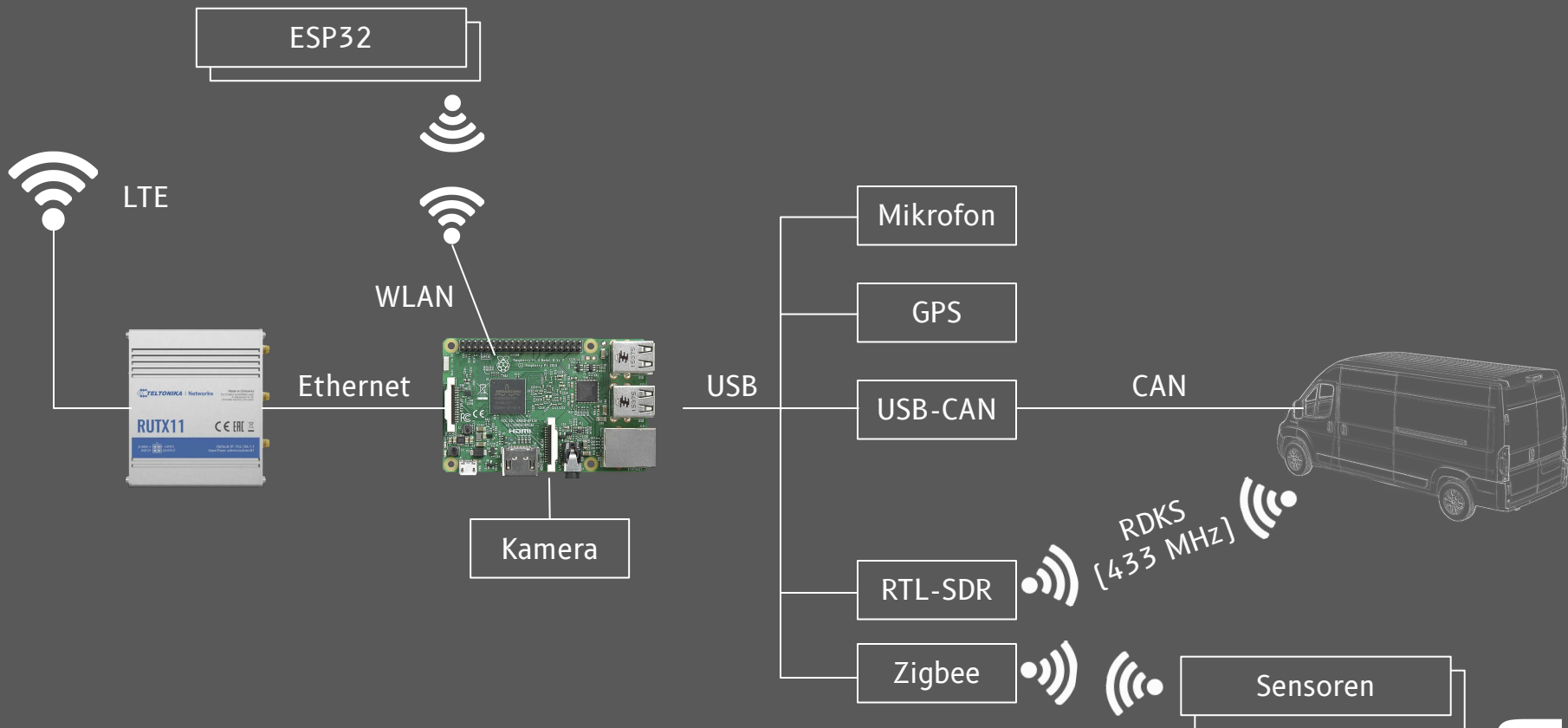


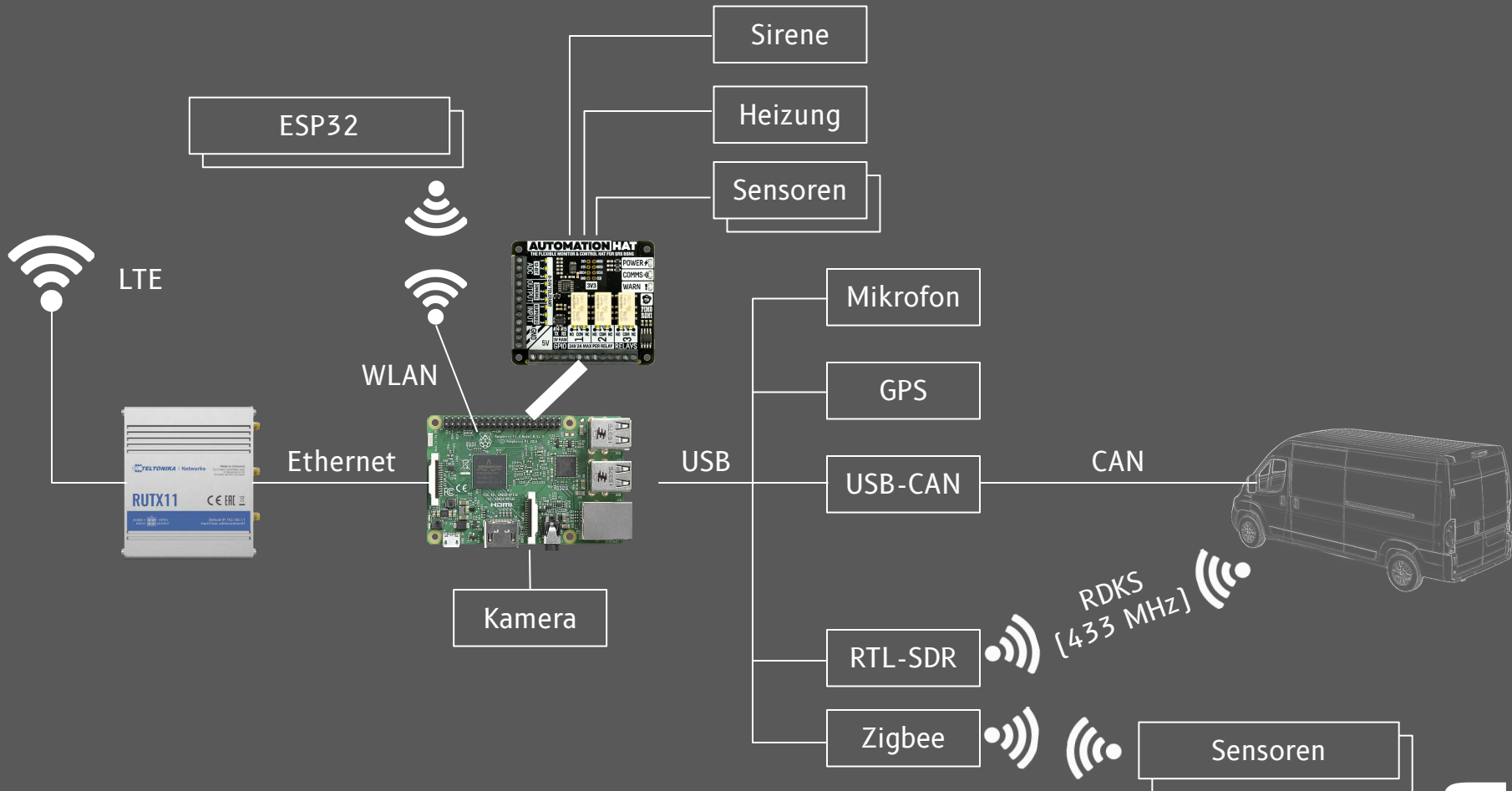
Hardware





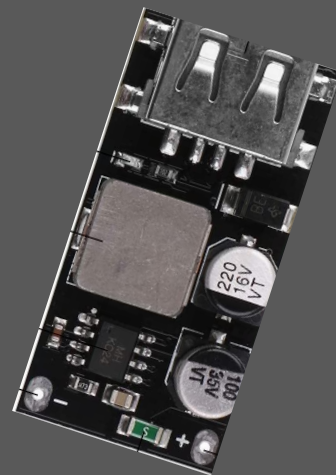






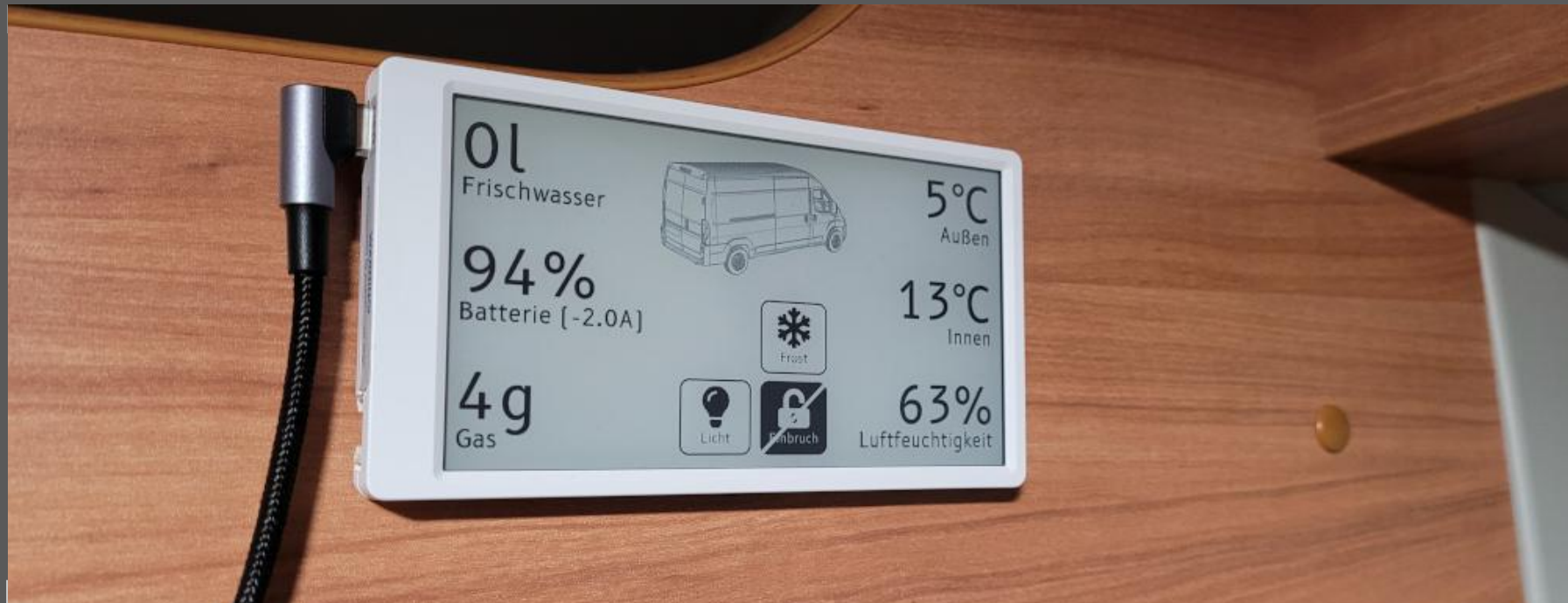
Stromversorgung

- Im Wohnmobil: 11,x - 14,8 Volt
- Geräte:
 - Raspberry Pi
 - RTL-SDR DVB-Stick
 - Zigbee-Adapter
 - GPS
 - Kamera
- DC-DC Step-Down von Amazon - sehr zuverlässig!
- Idle inkl. Router: 11-13 Watt
- Idle = ~ 36 MQTT-Messages/s $\hat{=}$ 3,5KiB/s



E-Paper-Display

TITLE



MAIN



DANIEL FETT

E-Paper-Display

TITLE



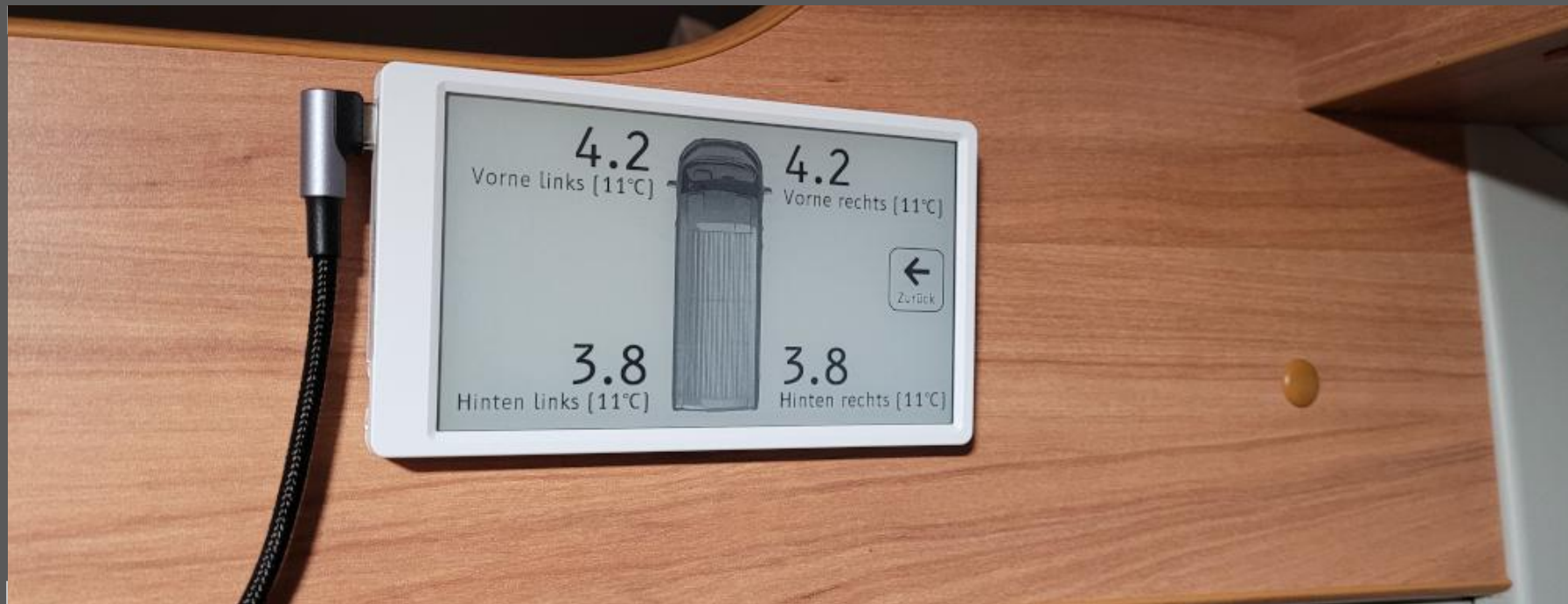
MAIN



DANIEL FETT

E-Paper-Display

TITLE



MAIN



DANIEL FETT

Software





Raspberry Pi

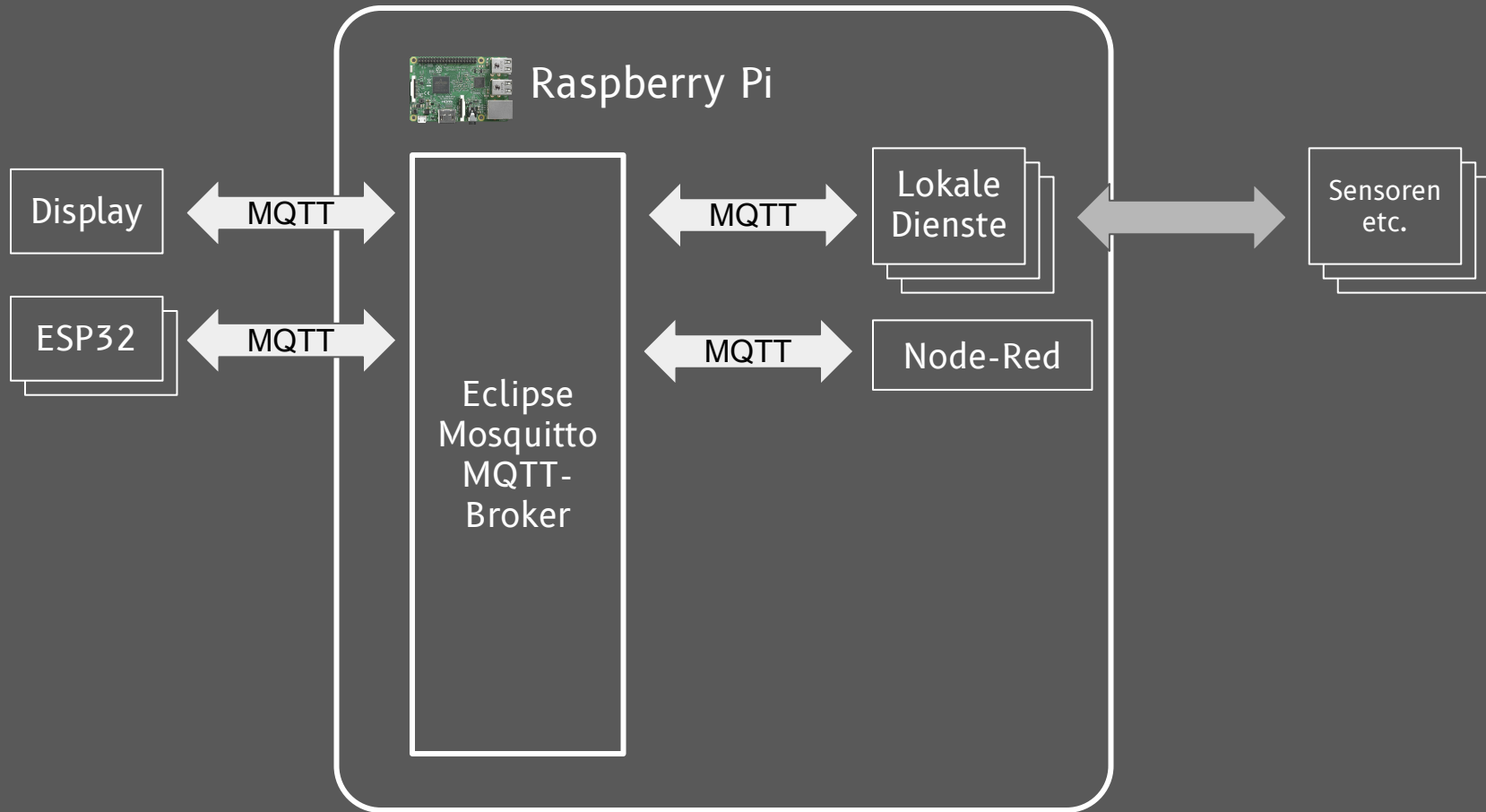
Eclipse
Mosquitto
MQTT-
Broker

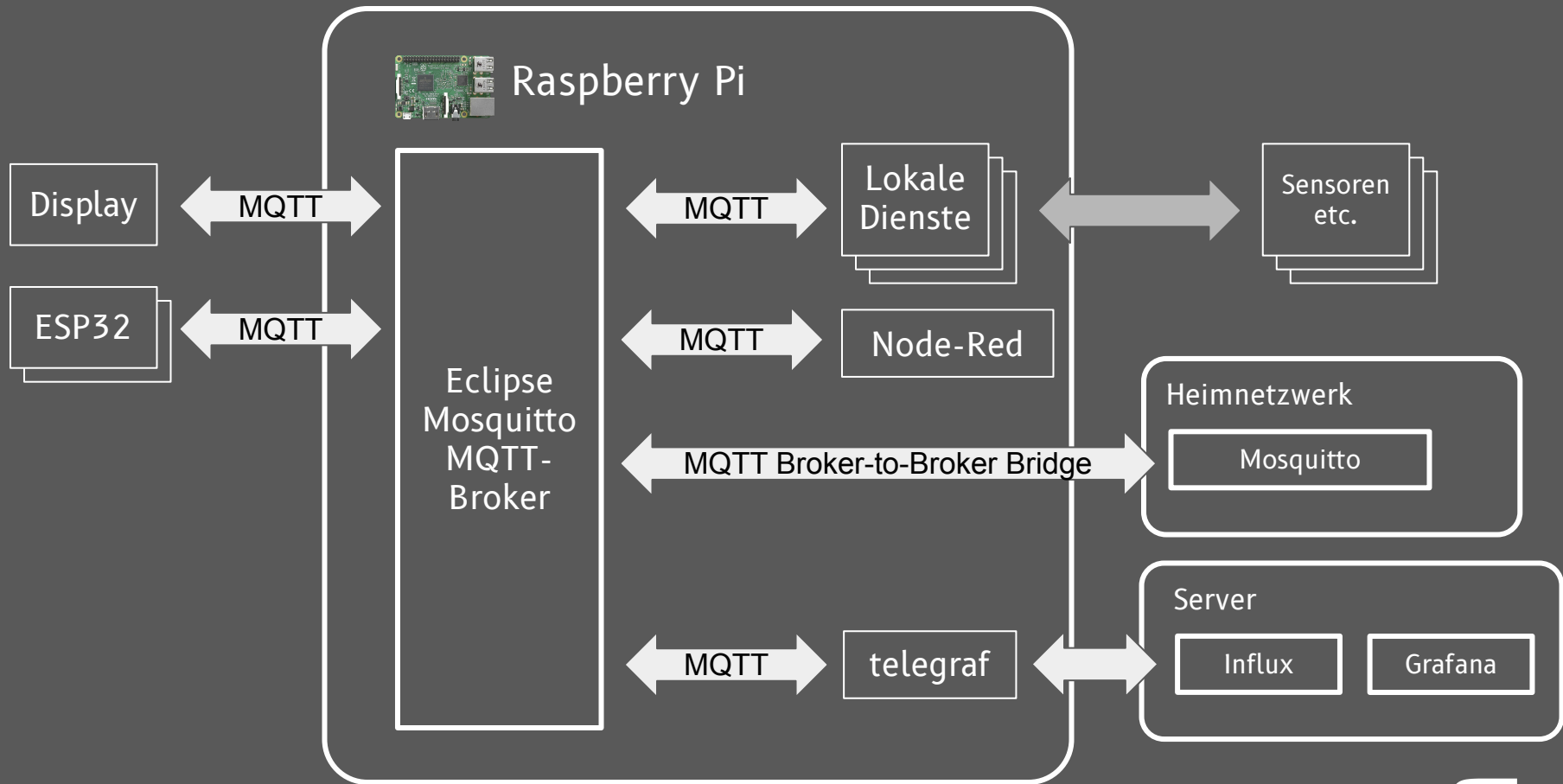
MQTT

Lokale
Dienste

Sensoren
etc.







MIQRO: MQTT Microservices

TITLE

- “Microservice”:
 - Dienst mit eng abgegrenzter Funktion
 - Gesamtanwendung aufgeteilt auf mehrere Microservices
 - Lose Kopplung der Microservices über einheitliche Schnittstelle, hier: MQTT
- MIQRO ist ein Framework für Microservices, die über MQTT kommunizieren
- Python 3

MAIN

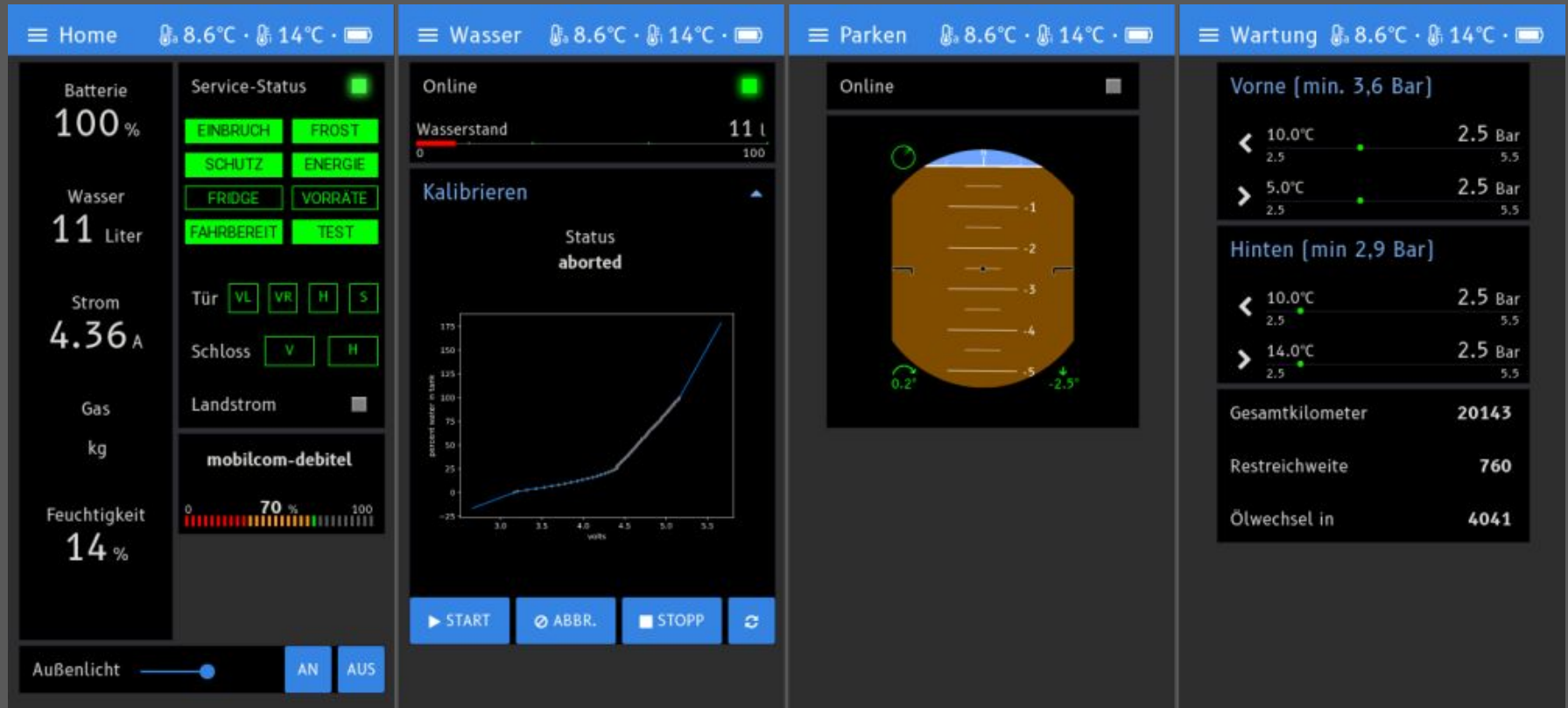


MIQRO: Features

- Typische Aufgaben mit wenigen Zeilen Code implementieren:
 - MQTT-Nachrichten in einem Service-Namespace empfangen und publizieren
 - Wiederkehrende Aufgaben, bspw. Sensor abfragen
 - Daten als JSON publizieren
 - Servicespezifische Konfigurationsdatei
 - Zustand abspeichern
 - Installation als Systemdienst

Mehr Informationen & Dokumentation: github.com/danielfett/miqro





Steuerung über Node-Red [Web-Interface]

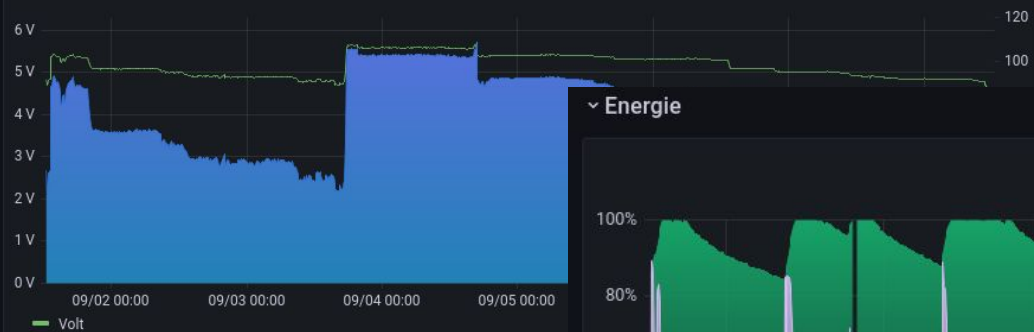
Zugriff von Überall!

- VPN-Lösung Tailscale
- Ende-zu-Ende-verschlüsseltes “Overlay”-Netzwerk
- Auf dem Smartphone nahtlos nutzbar



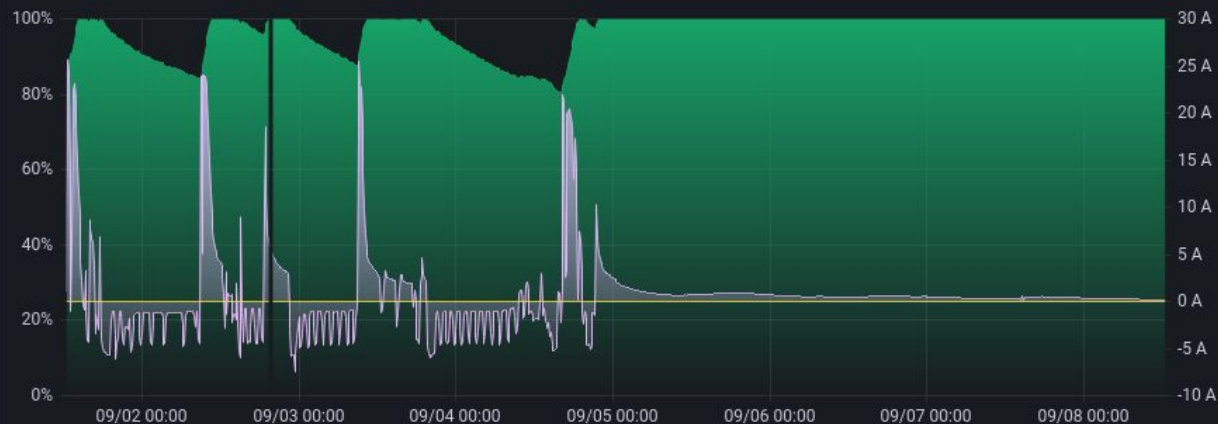
~ Wasser

Frischwasser



~ Energie

SOC & Strom



— SOC

— current (right y-axis)

— solar (right y-axis)

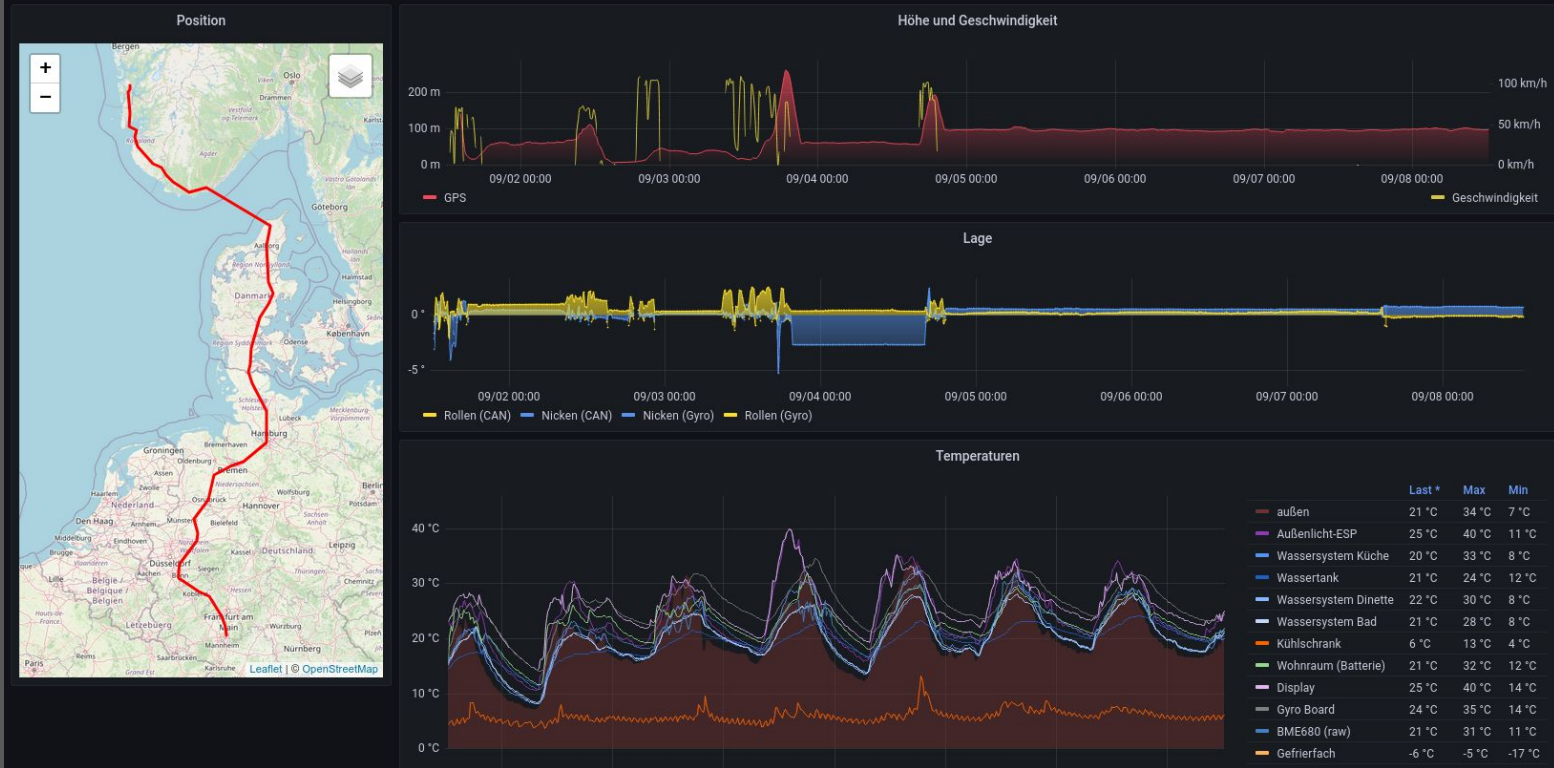
	Mean	Last *	Max	Min
SOC	96.8%	100%	100%	80.5%
current (right y-axis)	0.674 A	0.200 A	25.7 A	-7.5 A
solar (right y-axis)	0 A	0 A	0 A	0 A

Visualisierung mit Grafana



DANIEL FETT





Visualisierung mit Grafana

Komponenten



Komponenten

- Wasserstand Frischwassertank
- Stromverbrauch, Solar, Restkapazitäten
- Wasserlecksensor
- Geschwindigkeitsanzeige und Lagesensor
- Alarmanlage
- Temperatursensoren
- Außenlicht
- Reifendrucke
- Heizungssteuerung
- Gasreserven
- Kamera Innenraum

Wasserstandsensor



Wasserstand messen - nur wie?

- Kontakte?
- Schwimmer?
- Optisch?
- Ultraschall?
- Kapazitiv?
- Druck? Druck!



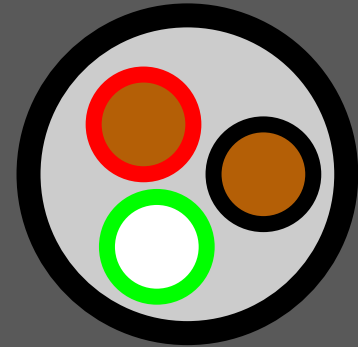
Druckbasierter Sensor

- Messbereich 0-2 Meter
- 12-24 Volt
- Ausgabe als Stromschleife 4-20mA
- Auswertung über Messwiderstand
- Zuführung Außendruck über luftführende Ader



TITLE

PIC1



MAIN

PIC2



Nichtlinearer Wassertank

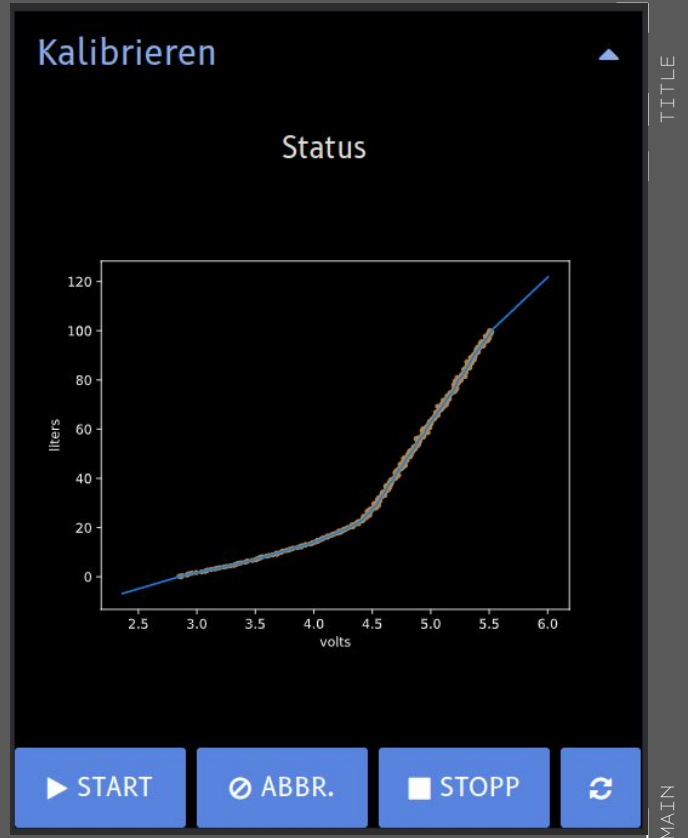
- Wassertank über Radkasten
- Lineare Auswertung nicht ausreichend!



Kalibrierung

- Start mit leerem Tank
- Befüllung mit konstantem Wasserdruck
- Glättung und Interpolation mit Numpy
- → Funktion Messwert→Wasserstand

Wasser
22 Liter



Strom & Co.





Batterie-Schätzeisen

Batteriecomputer

Bluebattery X200 von Kai Scheffer

- Batteriestatus [State of Charge]
- Strom
- Solarregler-Daten
- Kommunikation über Bluetooth

MQTT-Microservice:
github.com/danielfett/bluebattery.py



TITLE

PIC1

MAIN

PIC2

Aufbaubatterie



100

100



14

13

Starterbatterie



Anzeige in Node-Red



DANIEL FETT

Feuchtigkeitssensor



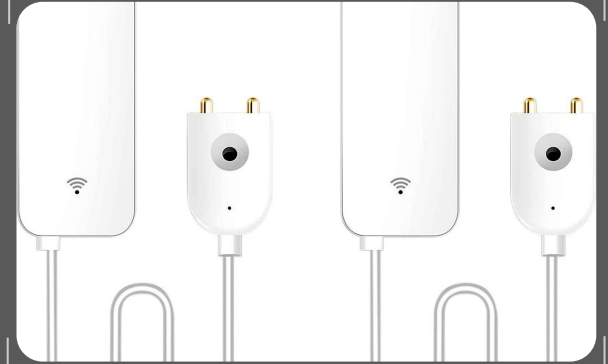
Wasserlecks entdecken

- Undichte Frischwasserleitungen
- Überlaufende/undichte Tanks
- Lecks an Dusche und Waschbecken
- Von außen eindringendes Wasser

Was tun?

Punktuelle Sensoren unzureichend:

- Räumlich verteilte Quellen
- Offene Kanten, Durchbrüche: Wasser staut sich nicht
- Ablaufrichtung unklar - Fahrzeug steht nicht immer gerade



Wassersensorkabel

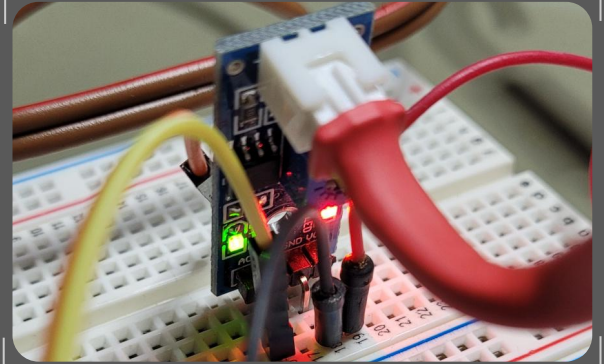


Lösung: Kabel mit durchlässigem Mantel!

Auswertung über Widerstand der Leiter

→ Wasserdetektion über die ganze Länge des Kabels

“Eve Water Guard”



Geschwindigkeitsanzeige und Lagesensor



TITLE



Der furchtbare Tacho... und die Lösung!



... und die Zweitfunktion!



... und die Zweitfunktion!



... und die Zweitfunktion!



CAN-Bus-Tacho und Lageanzeige

Geschwindigkeitsanzeige:

- Geschwindigkeitsdaten über “Entertainment”-CAN-Bus
- CAN-Interface → MQTT → [WLAN] → ESP32

Lagesensor:

- ADXL335 [GY-61]

Umschaltung Geschwindigkeitssensor - Lagesensor:

- Gurt eingesteckt oder nicht? [CAN-Bus-Signal]



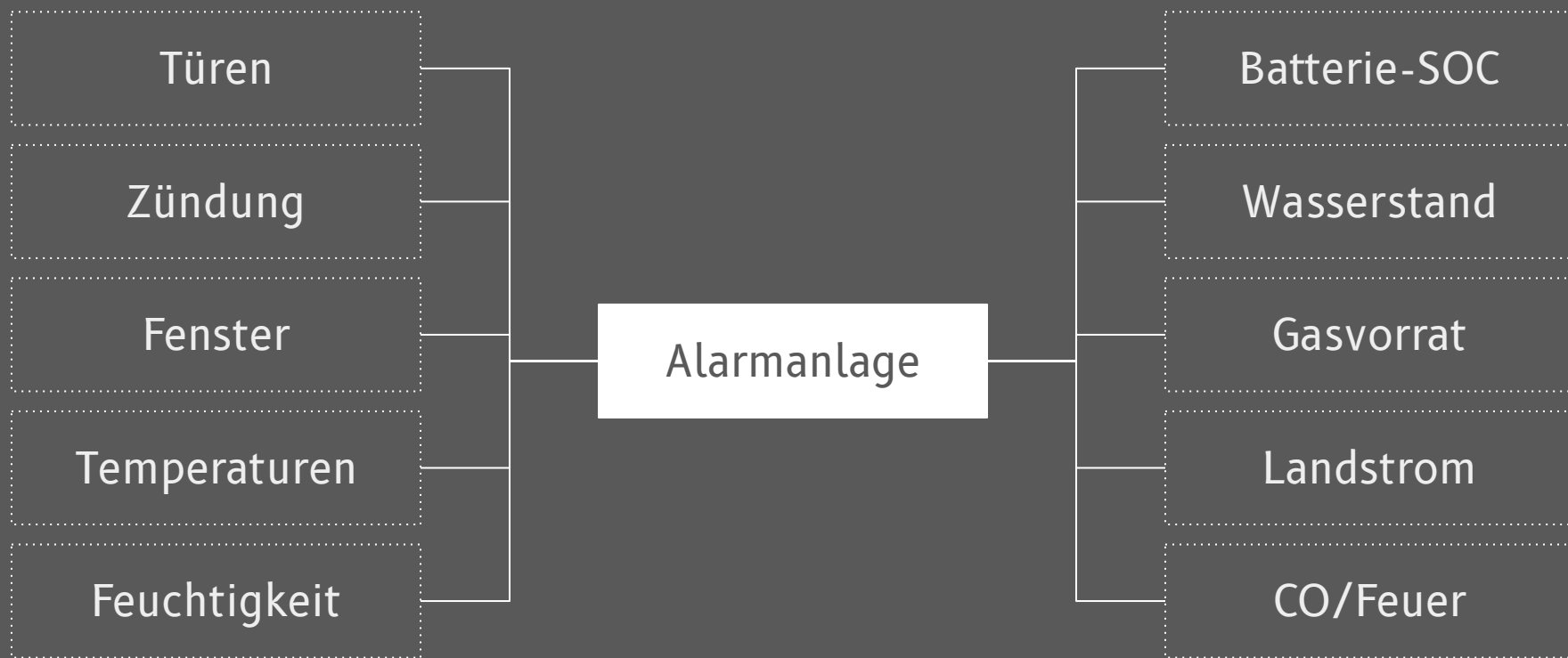
Alarmanlage



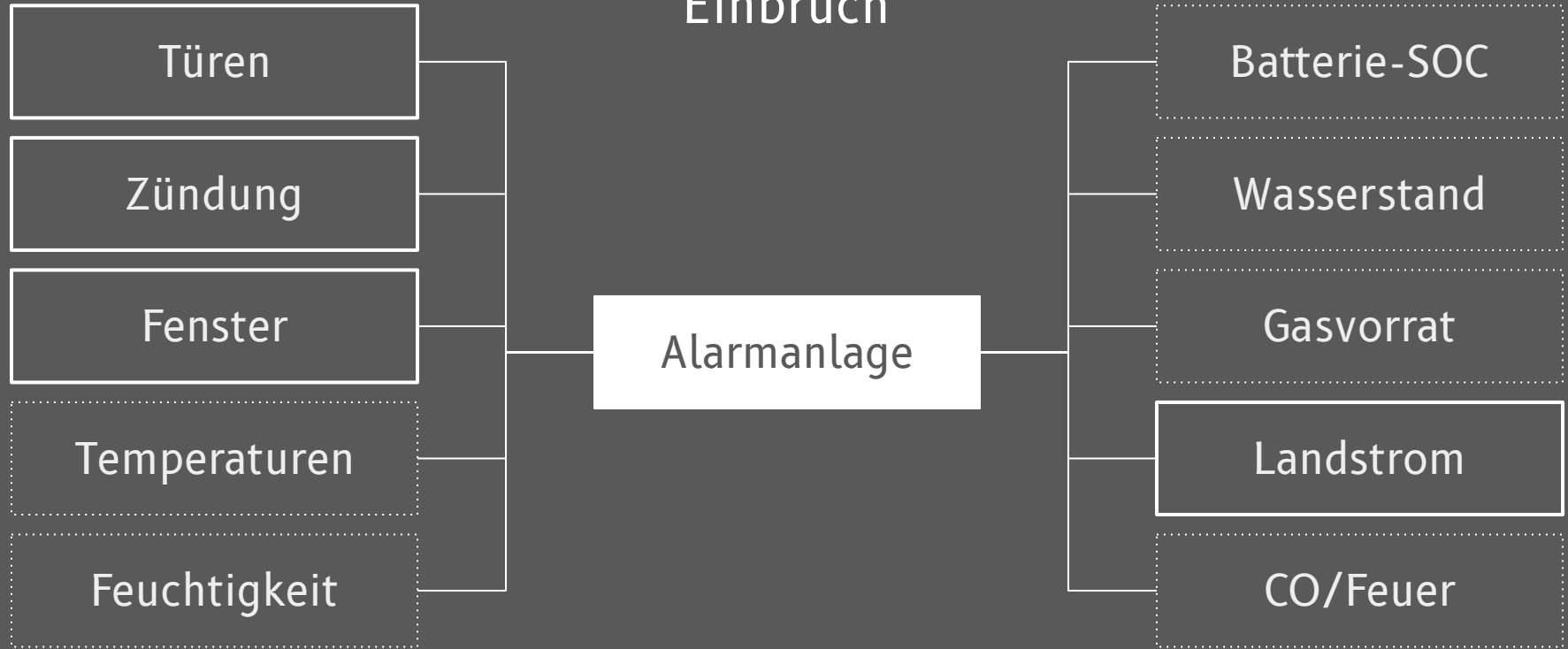
Alarmanlage

- Schutz [Safety] und Sicherheit [Security]
- MQTT-Microservice [natürlich...]
- Features:
 - Beliebig viele Alarmgruppen
 - Totmannschaltung für Sensoren und Service
 - Voralarm
 - Alarmrückstellung
 - Alarmunterdrückung
- Prioritäten für Alarmer (Einbruch vs. niedriger Batteriestand)
- Frei konfigurierbar

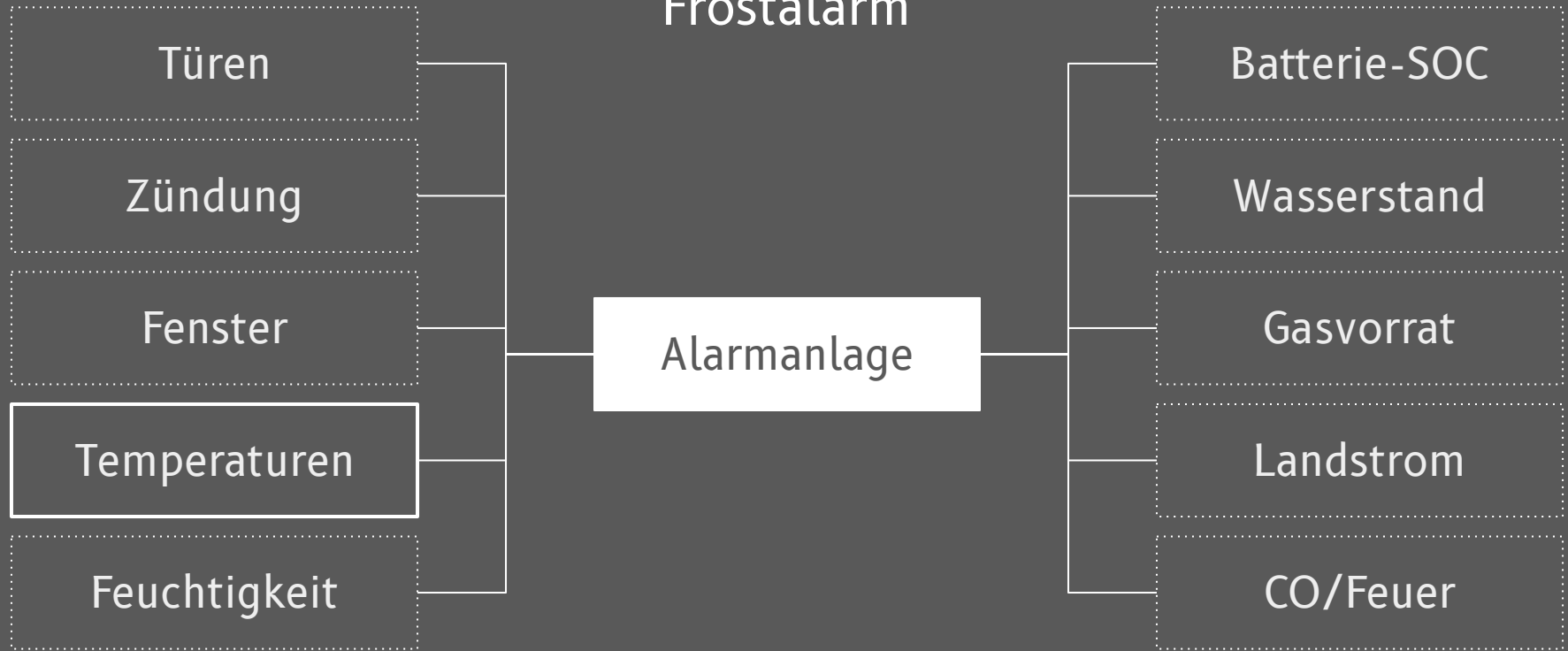




Einbruch



Frostalarm



Schutz



Energie



Kühlschrank



Vorräte



Fahrbereitschaft



Ausgabe/Benachrichtigungen

- SMS
- Pushover-Nachricht - auch am Nicht-Stören-Modus vorbei
- Sirene



Was gibt's noch?





womolin.de



Vielen Dank!

Mehr Informationen:

danielfett.de

mail@danielfett.de

[@dfett42](#)

