

2. Betriebliche Dokumentation

Die vorliegende betriebliche Dokumentation beschreibt die technische Konfiguration des im Projektbericht behandelten PV-Monitoringsystem. Sie dient Administratoren als Nachschlagewerk für den laufenden Betrieb, die Wartung und die Fehlerbehebung.

Inhaltsverzeichnis

2.1 Systeminfrastruktur und Hardware.....	1
2.2 Eingesetzte Betriebssysteme und Software.....	2
2.3 Netzwerk-Integration.....	2
2.4 Zentrale Konfigurationsdateien.....	3
2.4.1 Container-Orchestrierung (docker-compose.yml).....	3
2.4.2 Datenerfassung (Python pv_collector).....	4
2.5 Benutzer, Gruppen und Rechte.....	4
2.5.1 Betriebssystemebene.....	4
2.5.2 Applikationsebene (Grafana & InfluxDB).....	4

2.1 Systeminfrastruktur und Hardware

Die Hardware ist im zentralen Serverschrank (Rack 2, HE 14) platziert und über eine USV abgesichert. Der Zähler befindet sich im Technikraum.

Komponente	Spezifikation / Parameter	Konfiguration / Einsatz
Edge-Device	Raspberry Pi 4 Model B	4 GB RAM, 64-bit Architektur
Speichermedium	SanDisk Extreme microSDXC	32 GB, A2, Class 10 (für OS und DB)
Energiezähler	Eastron SDM120-Modbus	AC-Messung, Modbus-ID: 001, Baudrate: 9600
Schnittstelle	FTDI USB-RS485 Adapter	/dev/ttyUSB0, 8N1, Half-Duplex

Pin-Belegung (RS485 - CAT7 Leitungsverbindung):

- **A+ (Zähler):** Pin 1
- **B- (Zähler):** Pin 2

2.2 Eingesetzte Betriebssysteme und Software

Das System ist modular mittels Container-Technologie aufgebaut. Alle Container starten durch die Policy restart: always selbständig.

Software	Version	Verwendungszweck
Betriebssystem	Raspberry Pi OS Lite	Host-System (64-bit), minimaler Footprint
Container-Engine	Docker v24.x	Orchestrierung und Kapselung der Dienste
MQTT-Broker	mosquitto:latest	Interner Message Bus (pv/anlage/data)
Datenerfassung	Custom Python (pv_collector)	Modbus RTU nach MQTT Schnittstelle
Data-Connector	telegraf:latest	Subscription auf MQTT, Schreiben in InfluxDB
Datenbank	InfluxDB v2.7	Time-Series Langzeitspeicherung
Visualisierung	Grafana v11.0	Web-Dashboard und Alerting
Fernzugriff	cloudflared	Sicherer HTTPS-Tunnel ohne Portfreigabe

2.3 Netzwerk-Integration

Die Bereitstellung und Konfiguration der lokalen Netzwerkinfrastruktur obliegt dem lokalen Netzwerkadministrator des Kunden (EcoSolutions). Das PV-Monitoringsystem wird als Endgerät („Plug-and-Play“) in das bestehende Firmennetzwerk integriert.

Die physische Anbindung des Servers erfolgt über ein vom Kunden fertig konfektioniertes und bereitgestelltes RJ45-Netzkabel.

- **IP-Vergabe:** Die Adresszuweisung (IP-Adresse, Subnetzmaske, Gateway, DNS) erfolgt automatisch via DHCP durch den lokalen Router bzw. den DHCP-Server von EcoSolutions. Es ist systemseitig keine statische IP-Adresse konfiguriert.
- **Hostname:** pv-monitor
- **Intern offene Ports (Systemseitig):** 22 (SSH TCP), 1883 (MQTT TCP), 3000 (Grafana TCP), 8086 (InfluxDB TCP)

Hinweis zur Firewall und Isolation: Für die korrekte Funktion des Fernzugriffs via Cloudflared muss das System ausgehende Verbindungen über HTTPS (Port 443) aufbauen können. Darüber hinaus wird ausgehender Traffic für NTP (Port 123) zur Zeitsynchronisierung benötigt. Eingehende Verbindungen (Portweiterleitungen aus dem Internet) müssen auf der lokalen Firewall nicht konfiguriert werden. Die exakte Absicherung des Systems (z. B. durch Isolierung in einem separaten VLAN für IoT-Geräte) liegt im Ermessen der lokalen IT-Administration von EcoSolutions.

2.4 Zentrale Konfigurationsdateien

Das gesamte System wird deklarativ über Docker Compose konfiguriert.

2.4.1 Container-Orchestrierung (docker-compose.yml)

Auszug der wesentlichen Parameter für Netzwerk und Persistenz:

```
networks:
  pv_net:
    driver: bridge # Internes Bridge-Netzwerk für
Container-Kommunikation

services:
  influxdb:
    image: influxdb:2.7
    container_name: influxdb
    restart: always
    volumes:
      - ./docker/influxdb/data:/var/lib/influxdb2
      - ./docker/influxdb/config:/etc/influxdb2
    environment:
      - DOCKER_INFLUXDB_INIT_MODE=setup
      - DOCKER_INFLUXDB_INIT_USERNAME=admin
      - DOCKER_INFLUXDB_INIT_ORG=home
      - DOCKER_INFLUXDB_INIT_BUCKET=pv_data
    networks:
      - pv_net

  grafana:
    image: grafana/grafana:11.0.0
    container_name: grafana
    restart: always
    ports:
      - "3000:3000"
    volumes:
      - ./docker/grafana/data:/var/lib/grafana
    environment:
      - TZ=Europe/Berlin
    networks:
      - pv_net
```

2.4.2 Datenerfassung (Python pv_collector)

Wichtige Konfigurationsparameter für die serielle Datenkommunikation (Auszug):

```
# Modbus RTU Konfiguration
MODBUS_PORT = '/dev/ttyUSB0'
MODBUS_BAUDRATE = 9600
MODBUS_BYTESIZE = 8
MODBUS_PARITY = 'N' # None
MODBUS_STOPBITS = 1
MODBUS_TIMEOUT = 2.0 # Erhöht für Stabilität auf langen Strecken
UNIT_ID = 1

# MQTT Konfiguration
MQTT_BROKER = 'mosquitto' # Docker Service Name (DNS Resolve)
MQTT_PORT = 1883
MQTT_TOPIC = 'pv/anlage/data'
```

2.5 Benutzer, Gruppen und Rechte

2.5.1 Betriebssystemebene

- **Benutzer:** mateusz (UID 1000)
- **Gruppen:** sudo, docker, dialout (Zwingend für den physikalischen Zugriff auf /dev/ttyUSB0)
- **Rechtsschema:** Authentifizierung erfolgt sicherheitsbedingt primär über Public-Key-Verfahren (SSH-Keys).

2.5.2 Applikationsebene (Grafana & InfluxDB)

Applikation	Benutzer / Token	Rolle / Rechte
InfluxDB	grafana_reader	<i>Read-Only</i> Token speziell für das Dashboard. Beschränkt auf das Bucket pv_data.
InfluxDB	telegraf_writer	<i>Write-Only</i> Token. Berechtigt ausschließlich zum Schreiben in das Bucket pv_data.
Grafana	admin	<i>Server Admin</i> . Vollständige Konfigurationsrechte an Dashboards und dem System.
Grafana	viewer	<i>Viewer</i> . Lesezugriff zur reinen Betrachtung der Betriebsdaten ohne Änderungsrechte.