iSOCS

IoT Technologies for Distributed Sensing

ISOCS Winter Short Course 2023

16-19 January 2023, Bormio, Italy

Jan Mitrovics, JLM Innovation GmbH
jan.mitrovics@jlm-innovation.de
www.jlm-innovation.de

iSOCS
International Society for
Olfaction and Chemical Sensing

IoT: Internet of Things

Definition (Wikipedia):

„The Internet of Things (IoT) describes physical objects (or groups of such objects) with **sensors, processing ability**, software and other technologies that **connect and exchange data** with other devices and systems over the Internet or other communications networks.

Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable."

Smart devices connected to a network

Basics: Communication technologies

Distributed sensing networks

- Wired Ethernet, Fieldbus, RS485, I2C, …
- Wireless
  - Wifi / Bluetooth / GMS/UMTS/LTE/G5 / ZWave/Zigbee/Thread / 433MHz / LoRa / 6LowPAN / RFID/ IRDA / 802.15.4

Communication protocolls

- TCP/IP / Zigbee / Thread / Matter
- The Things Network

Figures of Merit for networks

- Throughput / bandwidth
- Range (max cable range, coverage of wireless network
- Max devices adressable
- Power demand (on device)
- Real time / guaranteed delivery time
- Security / Secrecy
- Fault tolerance / redundancy
- Topology (P2P / Bus / Star / Mesh)

Fieldbusses for Control systems

Broad range of fieldbusses is used!

Analog interfaces: 4-20mA, 0..10V, digital, …

Fieldbus	Bus power	Cabling redundancy	Max devices	Synchronisation	Sub millisecond cycle
AFDX	No	Yes	Almost unlimited	No	Yes
AS-Interface	Yes	No	62	No	No
CANopen	No	No	127	Yes	No
CompoNet	Yes	No	384	No	Yes
ControlNet	No	Yes	99	No	No
CC-Link	No	No	64	No	No
DeviceNet	Yes	No	64	No	No
EtherCAT	Yes	Yes	65,536	Yes	Yes
Ethernet Powerlink	No	Optional	240	Yes	Yes
EtherNet/IP	No	Optional	Almost unlimited	Yes	Yes
Interbus	No	No	511	No	No
LonWorks	No	No	32,000	No	No
Modbus	No	No	246	No	No
PROFIBUS DP	No	Optional	126	Yes	No
PROFIBUS PA	Yes	No	126	No	No
PROFINET IO	No	Optional	Almost unlimited	No	No
PROFINET IRT	No	Optional	Almost unlimited	Yes	Yes
SERCOS III	No	Yes	511	Yes	Yes
SERCOS interface	No	No	254	Yes	Yes
Foundation Fieldbus H1	Yes	No	240	Yes	No
Foundation HSE	No	Yes	Almost unlimited	Yes	No
RAPIEnet	No	Yes	256	Under Development	Conditional
Fieldbus	Bus power	Cabling redundancy	Max devices	Synchronisation	Sub millisecond cycle

Source Wikipedia

Fieldbus vs IoT

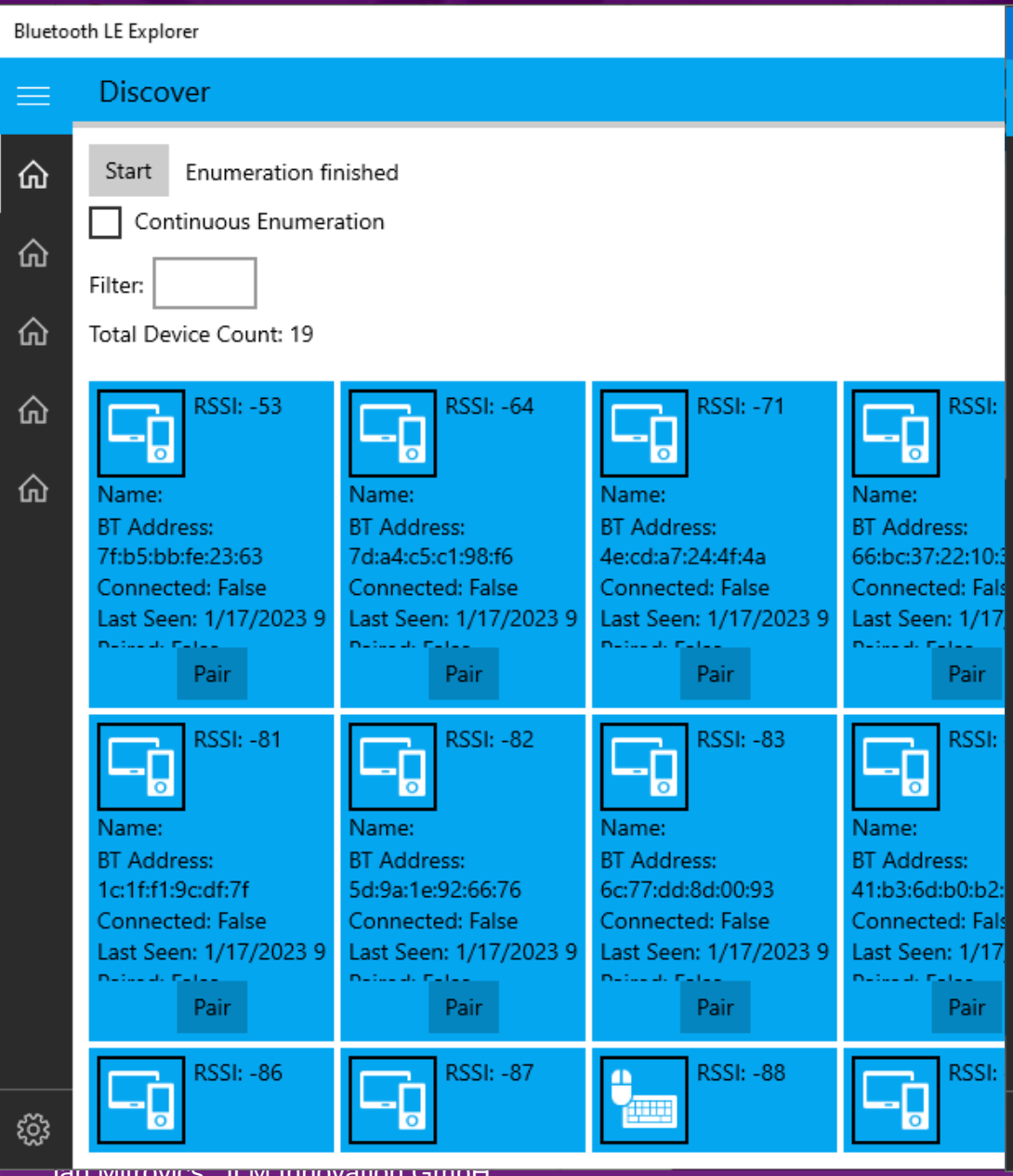
Fieldbus	IoT
Typical Applications	
Industrial automation Motor control / Automotive	Distibuted sensing Monitoring Home automation
Typical Requirements	
Reliability Deterministic (fast) timing	Scalability Interoperability Low Cost

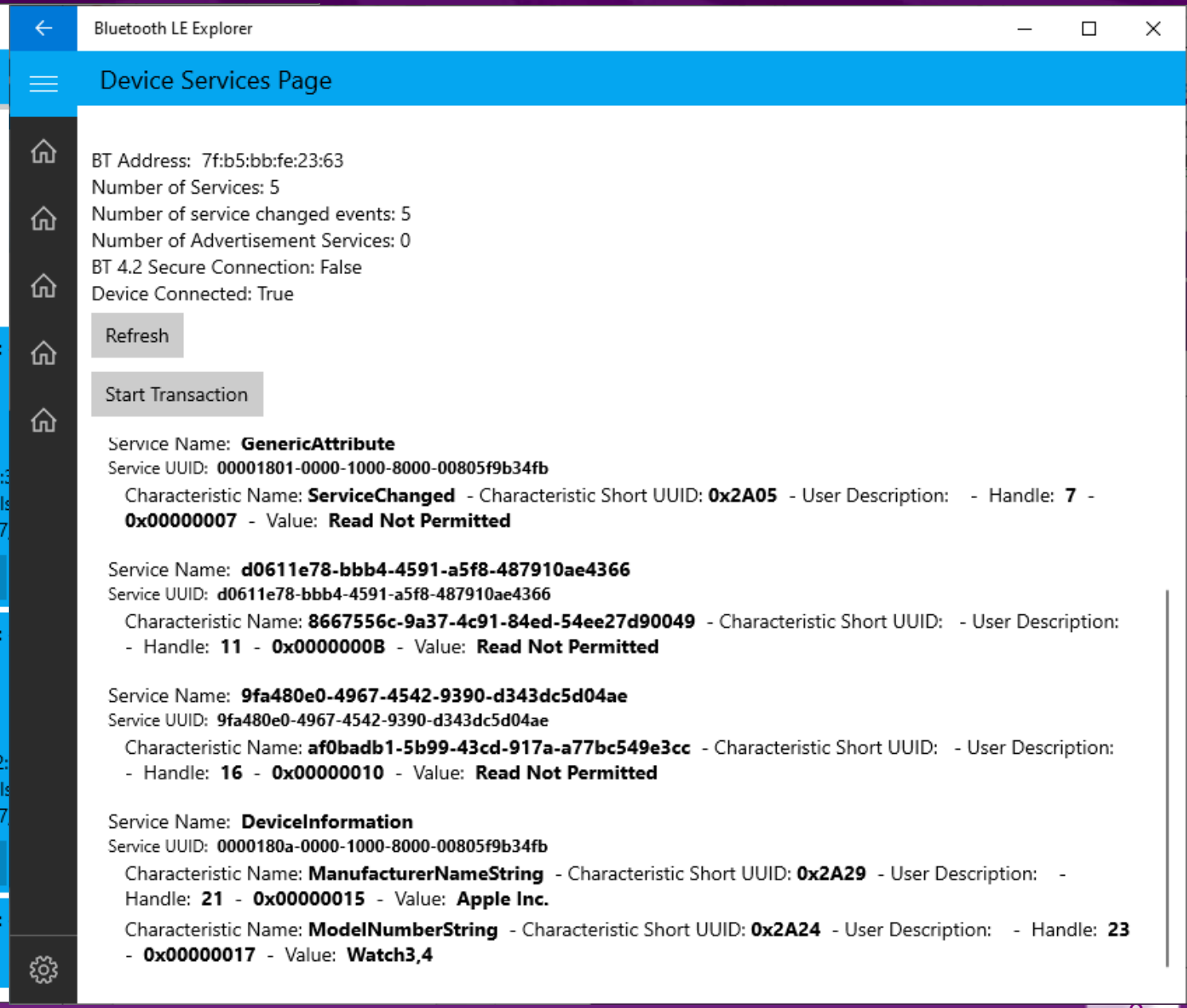
Communicaton: Wireless Networks

Type	Range	Power (on node)	Bandwidth	Topology	Cost
UMTS/LTE/5G	km / wide coverage	high	Very high	Star	high
Wifi	10-100m	high	Very high	Star (mesh)	low
Bluetooth LE	10-100m	Very low	Low	Star, p2p, mesh	low
LoraWAN	2km / wide coverage	Very low	Very low	Star	low
802.15.4 e.g. Zigbee	10-100m	Very low	Low	Mesh	low
RFID / NFC	1m	Very low / passive	Bytes on demand	p2p	Very low

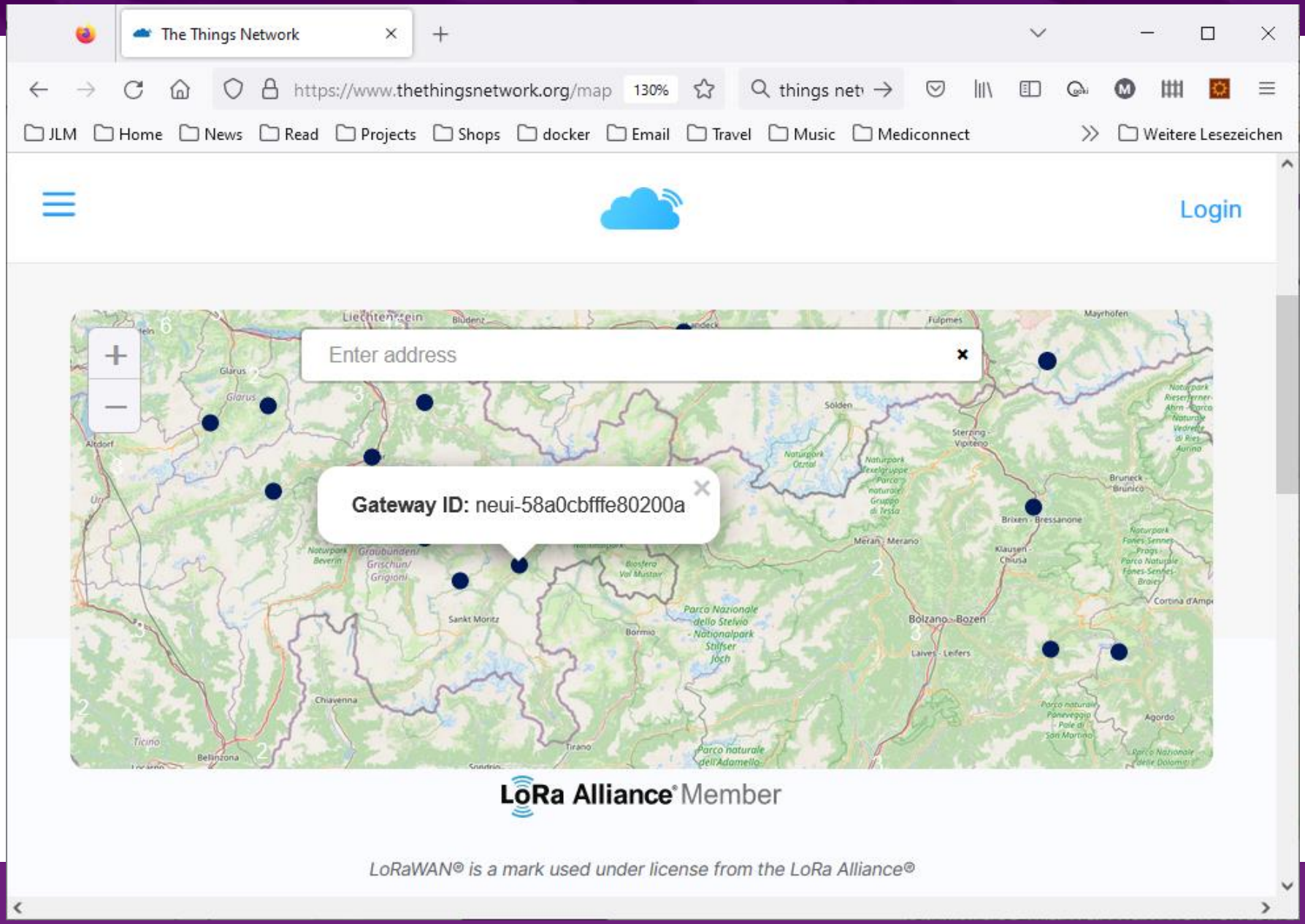
matter

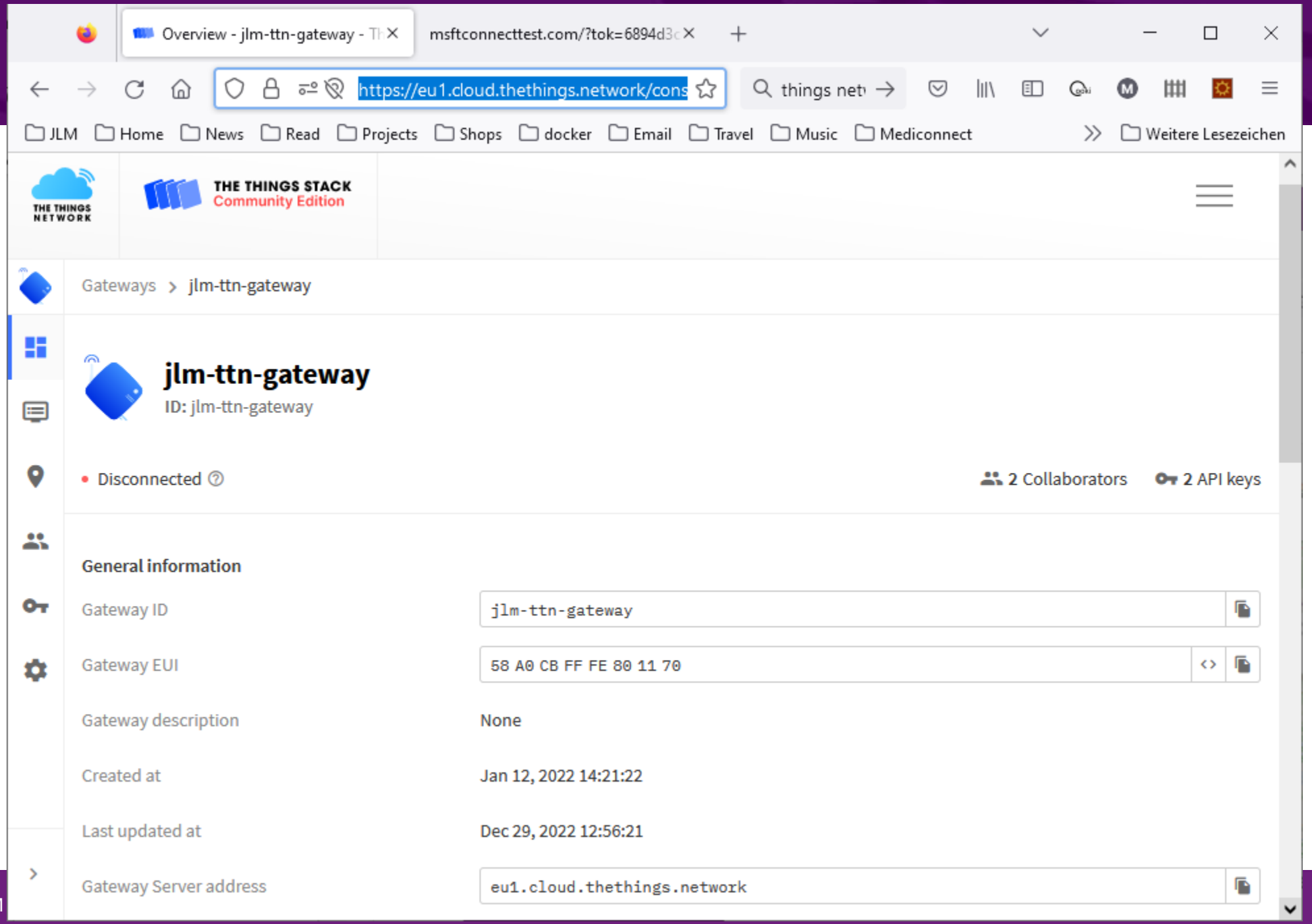
Communication Bluetooth BLE





Communication LoraWAN

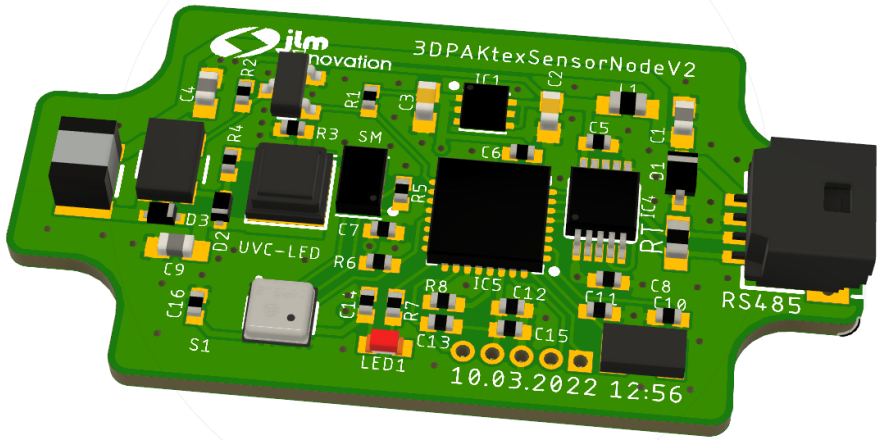


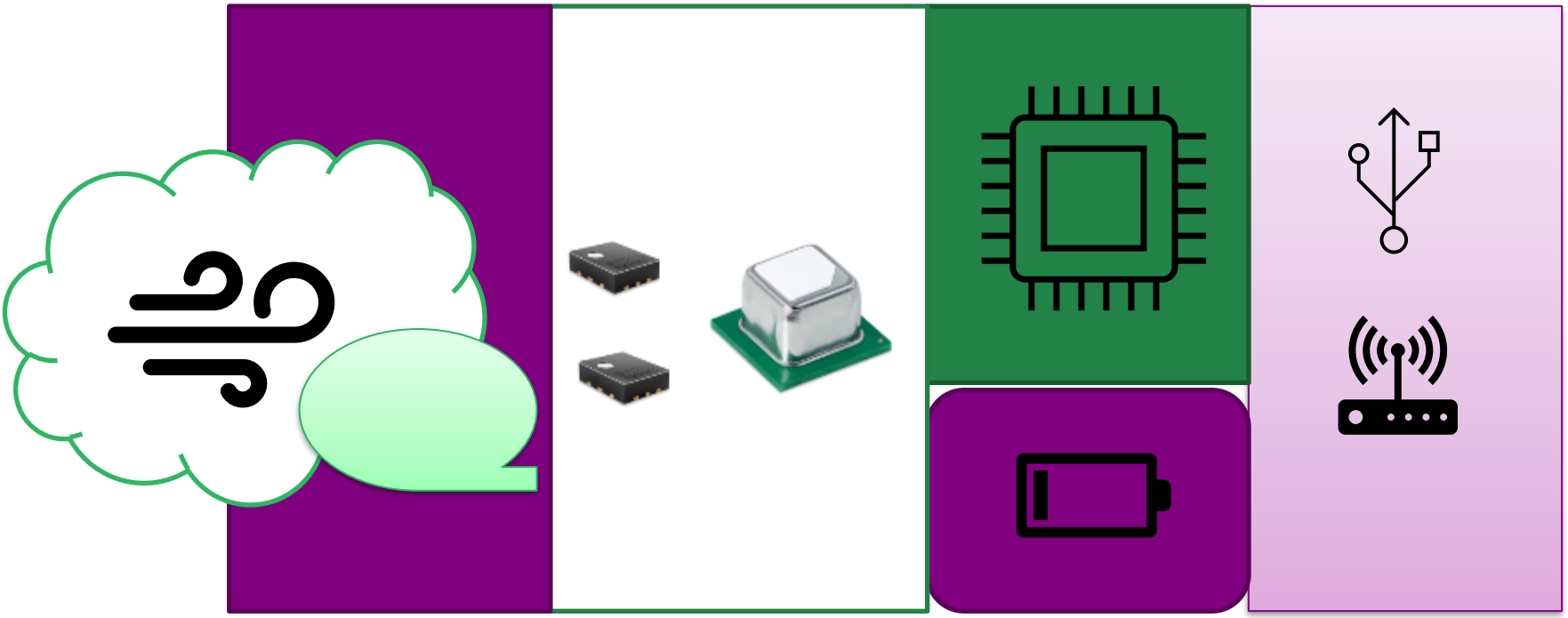

Overview - jlm-ttn-gateway - Th
msftconnecttest.com/?tok=6894d3c
https://eu1.cloud.thethings.network/cons
things net
JLM
Home
News
Read
Projects
Shops
docker
Email
Travel
Music
Mediconnect
Weitere Lesezeichen
THE THINGS NETWORK
THE THINGS STACK
Community Edition
Gateways
jlm-ttn-gateway
jlm-ttn-gateway
ID: jlm-ttn-gateway
Disconnected
2 Collaborators
2 API keys
General information
Gateway ID
jlm-ttn-gateway
Gateway EUI
58 A0 CB FF FE 80 11 70
Gateway description
None
Created at
Jan 12, 2022 14:21:22
Last updated at
Dec 29, 2022 12:56:21
Gateway Server address
eu1.cloud.thethings.network

Smart Sensor Trends

- Miniaturization
  - Lower Size, Lower Power Consumption, Lower Cost
- Integration
  - Combination of different sensors in one package (e.g. BME-680, SCD-41, SEN-5x)
  - New sensor types through combination of technologies (e.g. photoacoustic sensors with MEMS microphones)
- Software
  - More complex algorithms
  - Product diversification via firmware / operation modes

Sensor Nodes

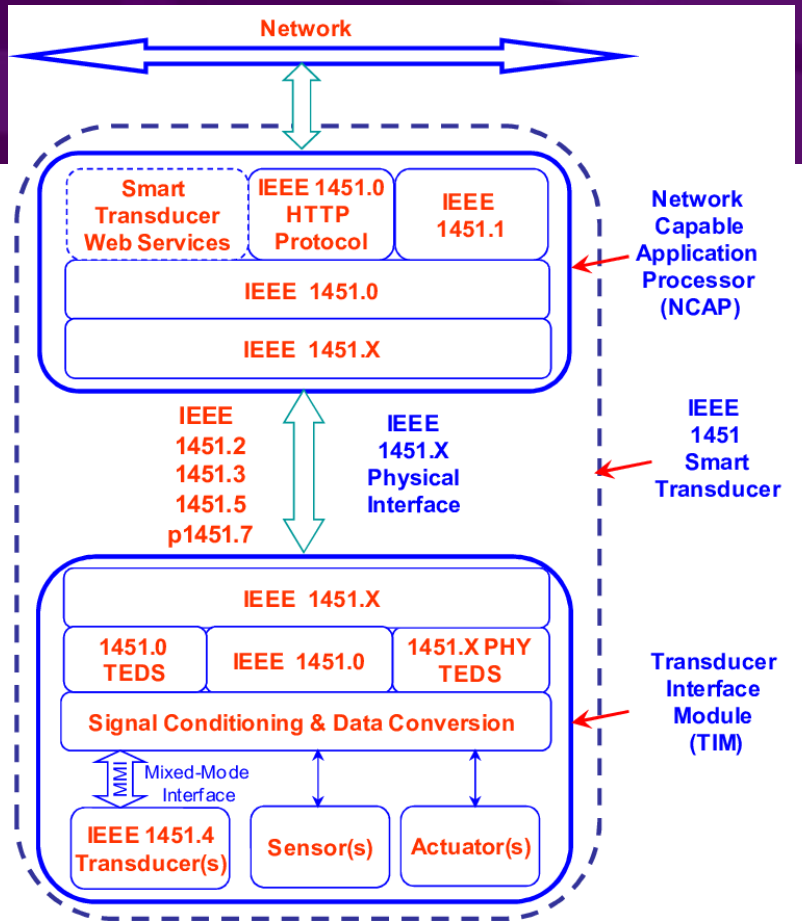




Sampling | Sensors | µController (Power) | Connectivity

IEEE 1451 smart transducers

- **1451.0–2007** Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats
- **1451.1–1999** Network Capable Application Processor Information Model
- **1451.2-1997** Transducer to Microprocessor Communication Protocols & TEDS Formats
- **1451.3-2003** Digital Communication & TEDS Formats for Distributed Multidrop Systems
- **1451.4-2004** Mixed-Mode Communication Protocols & TEDS Formats
- **1451.5-2007** Wireless Communication Protocols & Transducer Electronic Data Sheet (TEDS) Formats
- **1451.7-2010** Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats



T. C. O'Reilly *et al.*, "Instrument interface standards for interoperable ocean sensor networks," *OCEANS 2009-EUROPE*, Bremen, Germany, 2009, pp. 1-10, doi: 10.1109/OCEANSE.2009.5278251. https://ieeexplore.ieee.org/document/5278251/

An IoT Environment

Distributed Nodes

Message communication

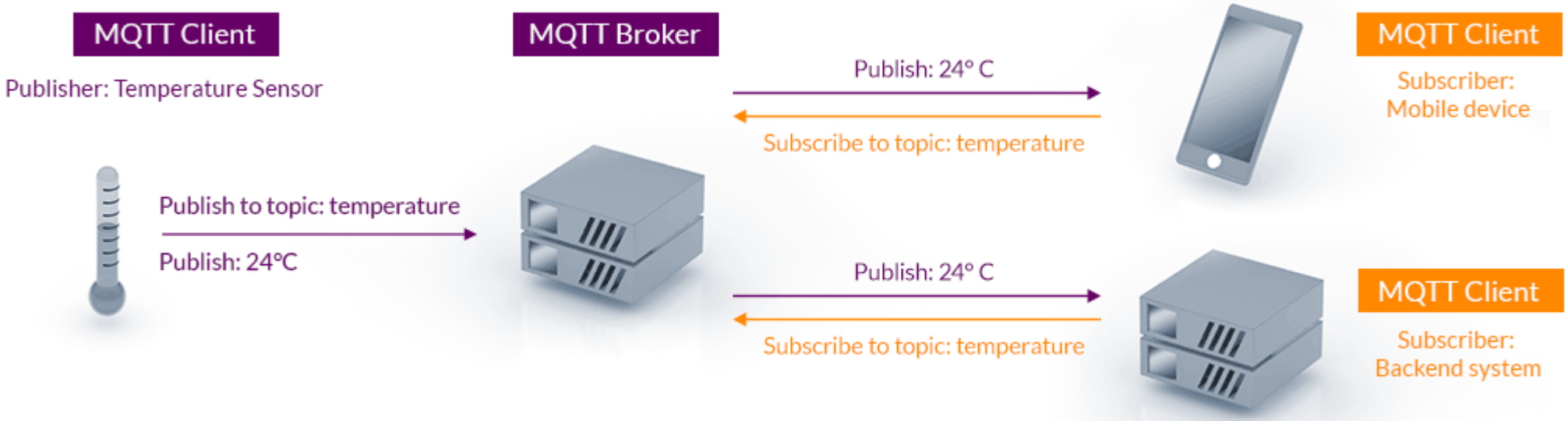
Application Layer / User Interface

IoT Tools MQTT

MQTT is an **OASIS standard** messaging protocol for the Internet of Things (IoT). It is designed as an **extremely lightweight publish/subscribe** messaging transport that is ideal for connecting remote devices with a **small code footprint** and **minimal network bandwidth**. MQTT today is used in a wide variety of industries, such as automotive, manufacturing, telecommunications, oil and gas, etc.

Source: https://mqtt.org/

What is MQTT?



Main Properties:

Lightweight and Efficient

Bi-directional Communications

Scale to Millions of Things

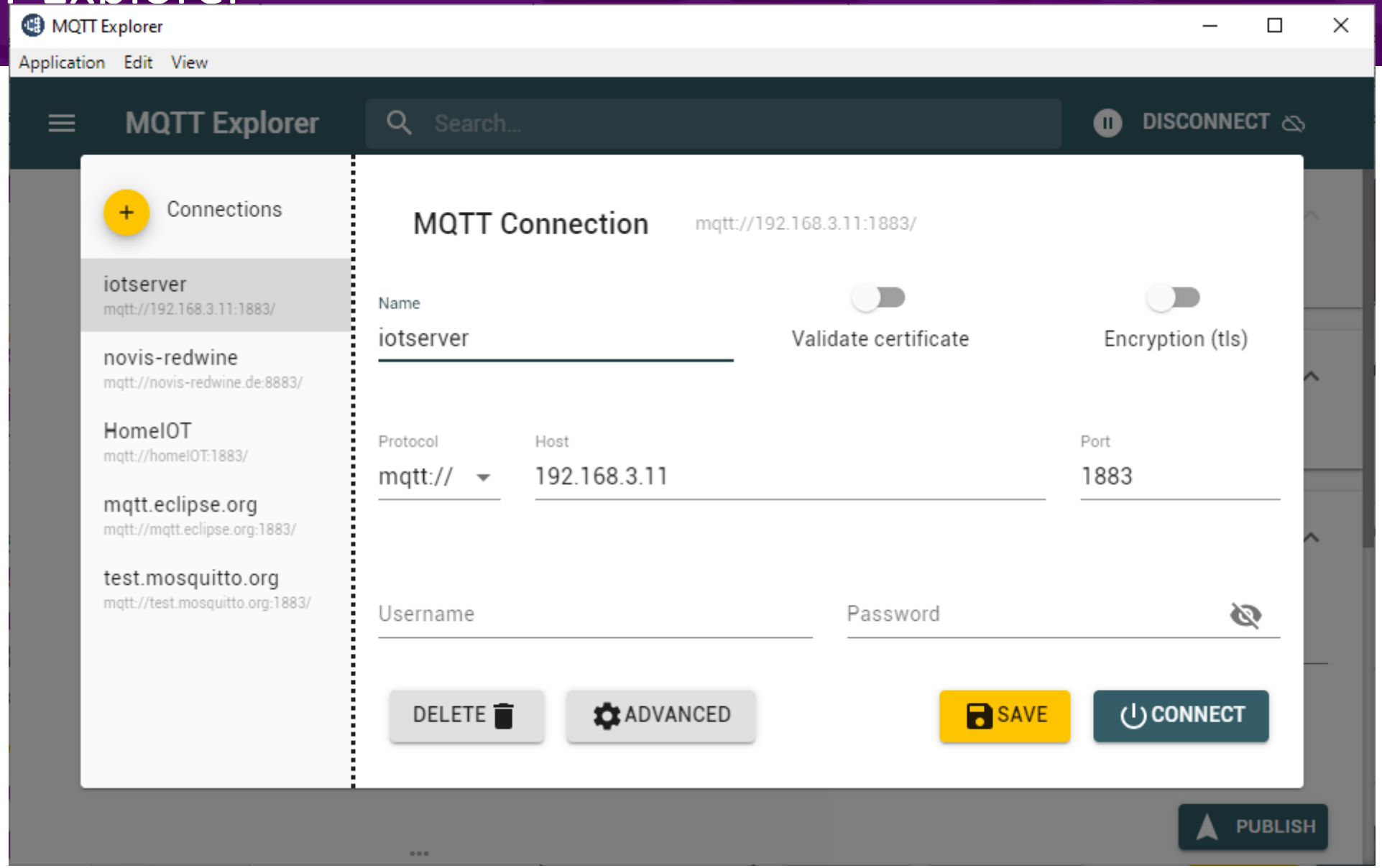
Reliable Message Delivery

Support for Unreliable Networks

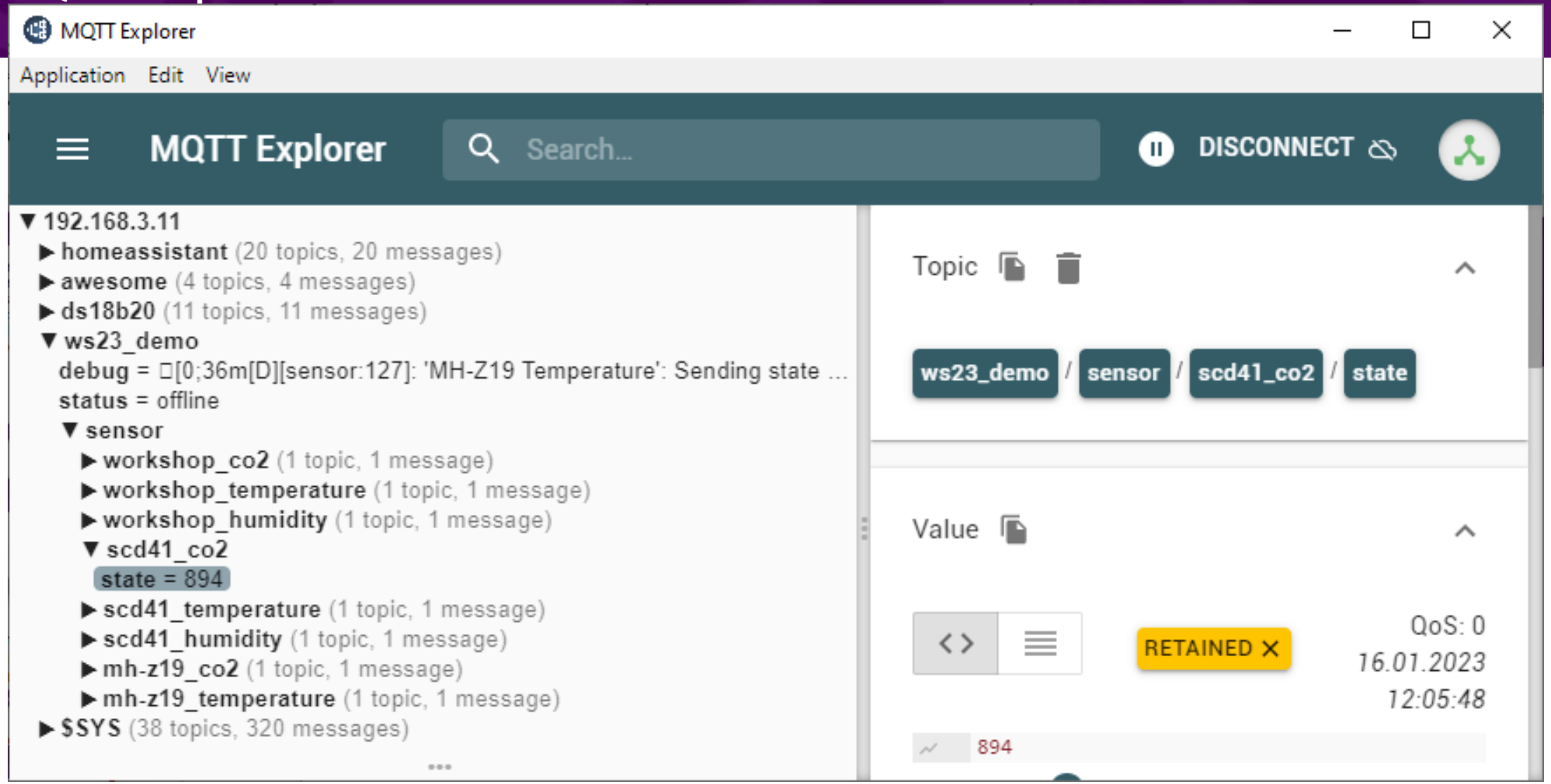
Security Enabled

Source: https://mqtt.org/

MQTT Explorer



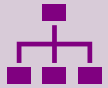
MQTT Topics



Topic: ws23_demo/sensor/scd41_co2/state

MQTT services / topics

Topics are hierarchically ordered



MQTT has 3 defined quality of service levels:

0 - at most once,
1- at least once,
2 - exactly once

MQTT publish messages contain a flag to retain the data

Clients can subscribe to topics using wildcards

MQTT Bridges / Adapters

A range of ready to use services exists to connect various sources into MQTT

- Zigbee2MQTT
- ESP32-ble2mqtt

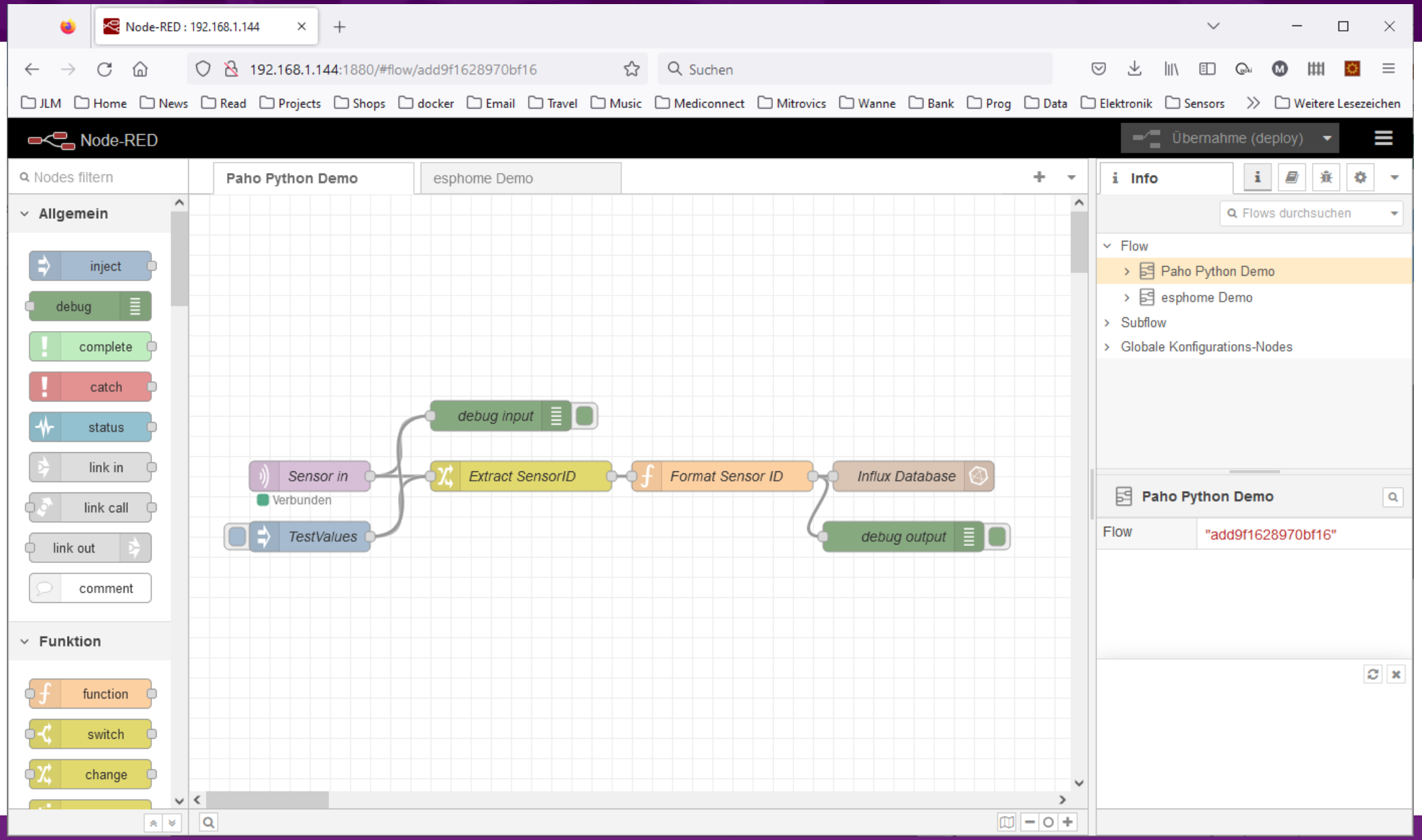
MQTT is supported by many automation systems

e.g. FHEM, OpenHab, Homeassistant

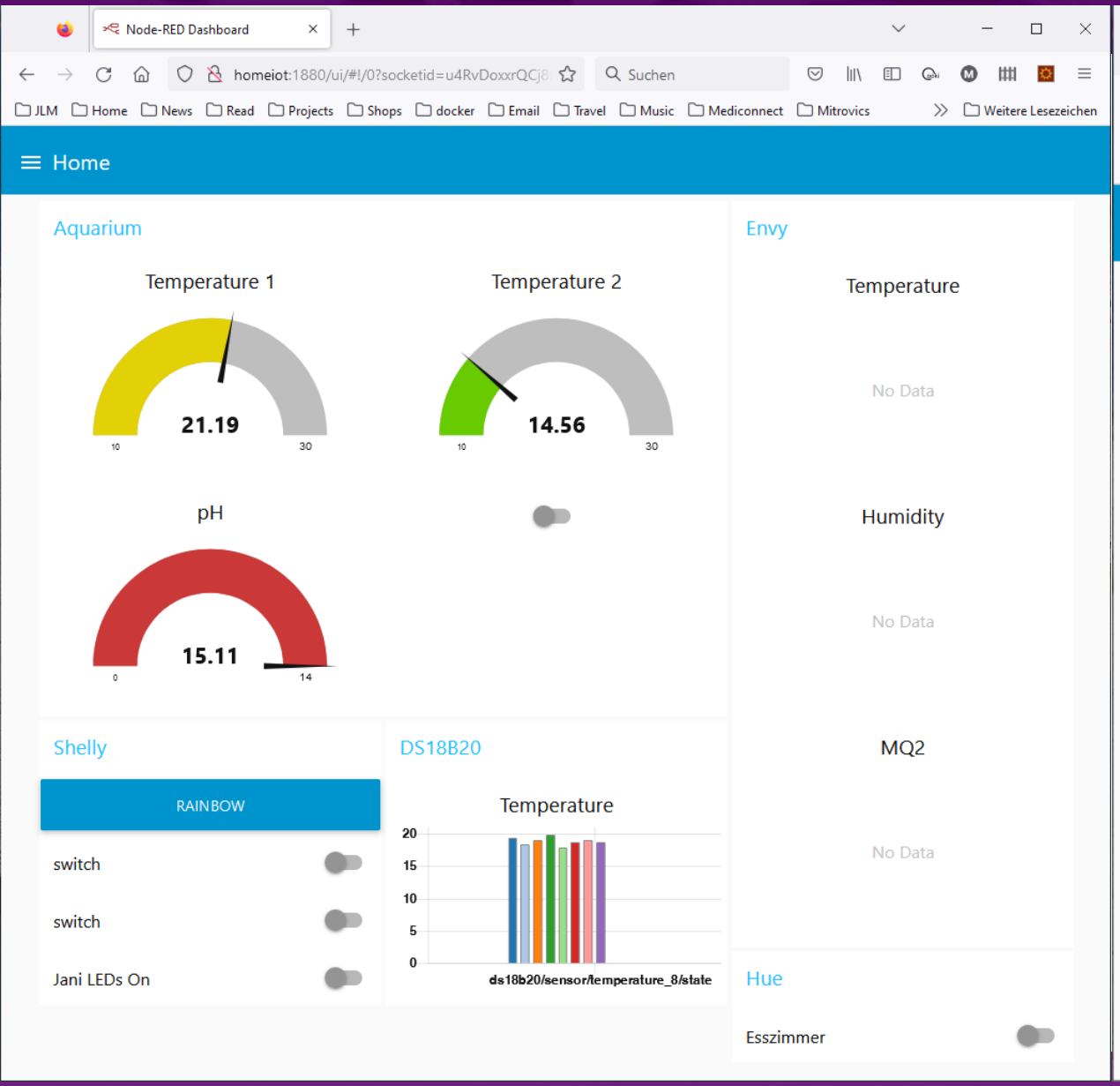
MQTT is easily integrated in Arduino, Python, C#, Java, …

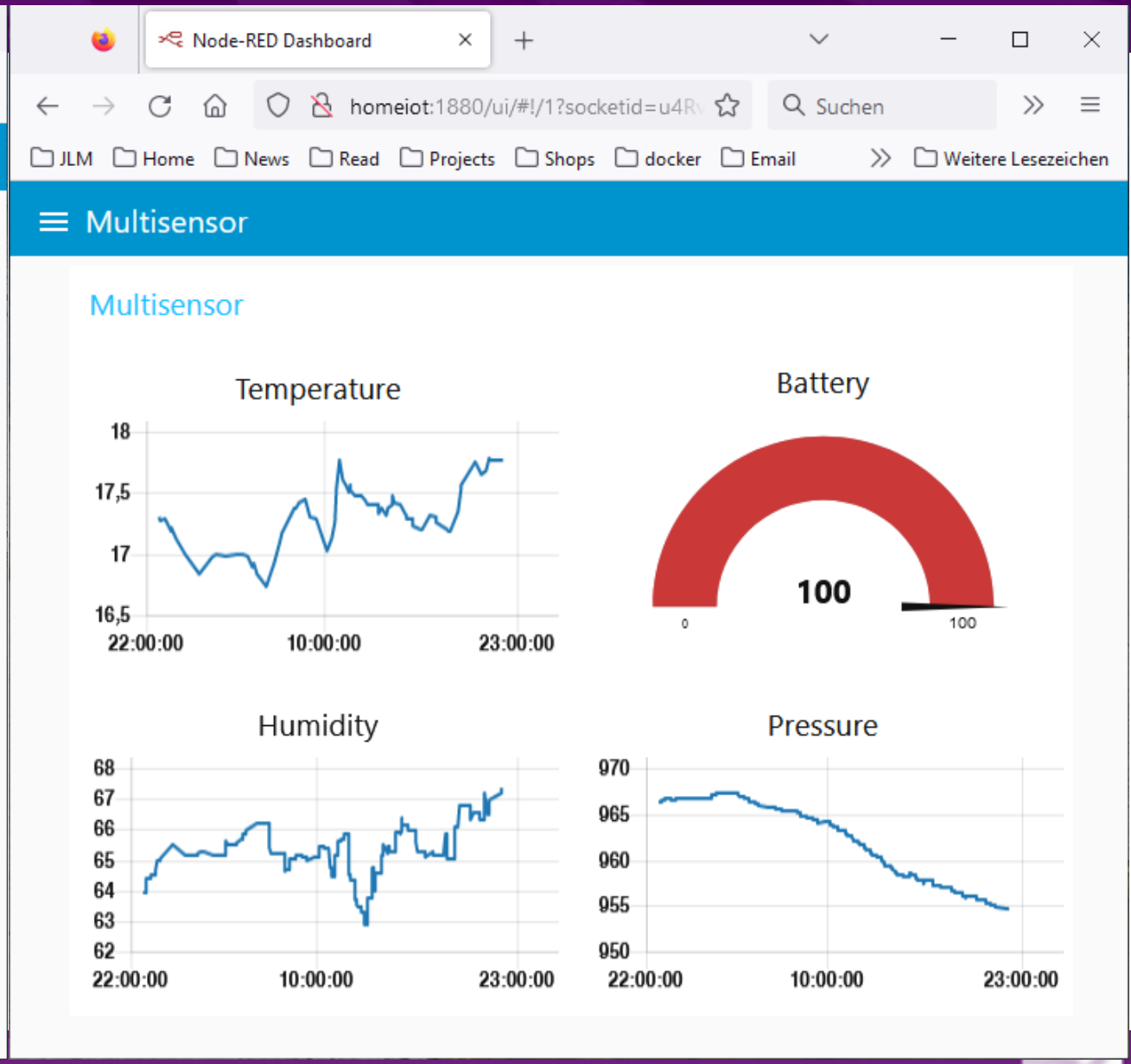
MQTT often directly supported by IOT devices (e.g. Tasmota, Shelly, ESPHome, …)

IoT Tools Node Red



Node-Red Dashboard



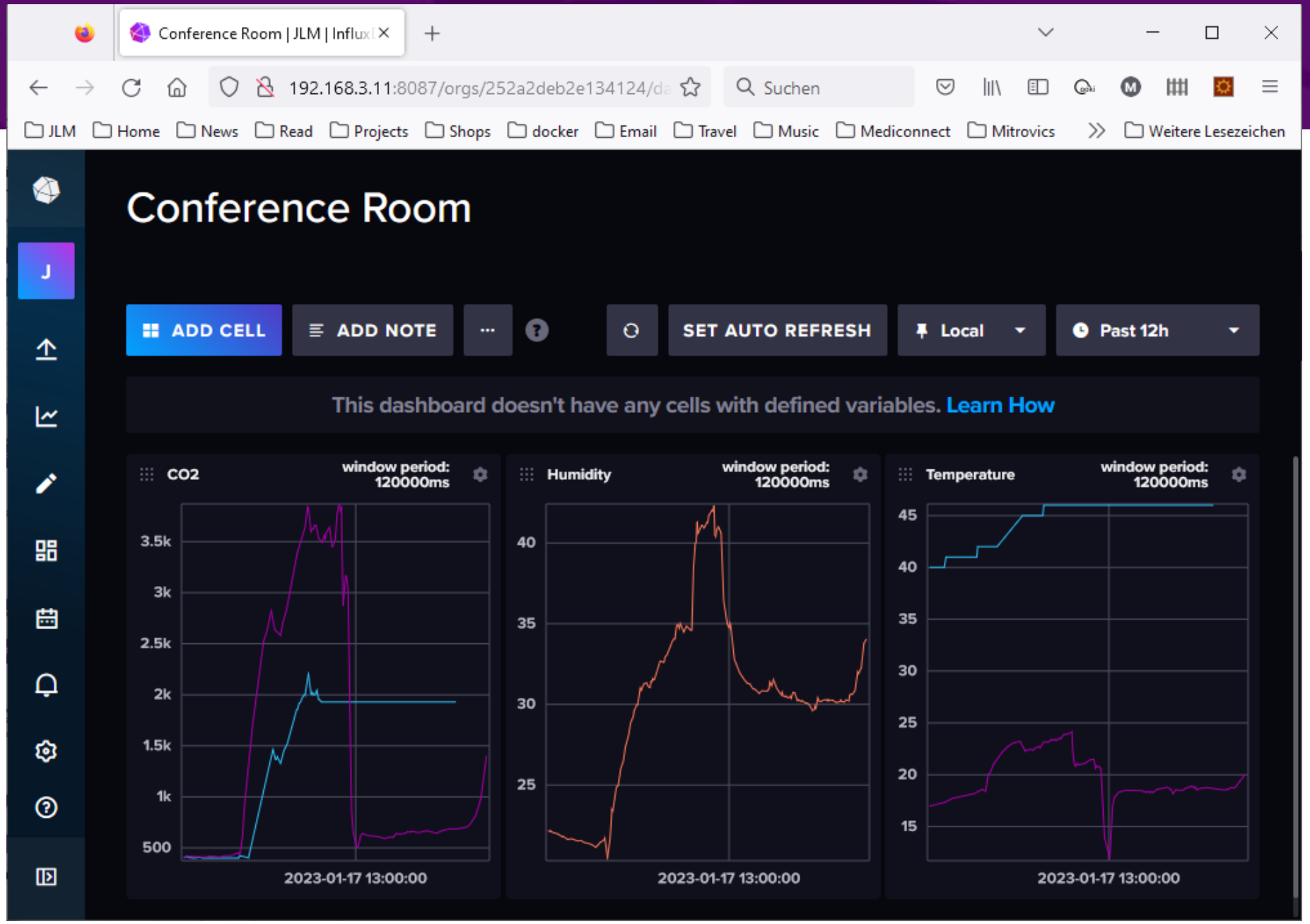


IoT Tools: Time Series Databases InfluxDB2

- Influx-DB is optimized to store data (typically JSON) with TimeTags
- InfluxDB2 adds an optimized query language and integrates various services (e.g. Telegraph to automatically pull data from other sources.
- Data can be separated by Organizations and Buckets
- Buckets may contain different types of measurements
- Data can be deleted automatically after a defined period of time
- Annotations can be added to describe events at certain point of time

InfluxDB 2

Integrated Grafana to create dashboards with graphs of data from different sources



Building a Stack for IoT using Open Source Tools

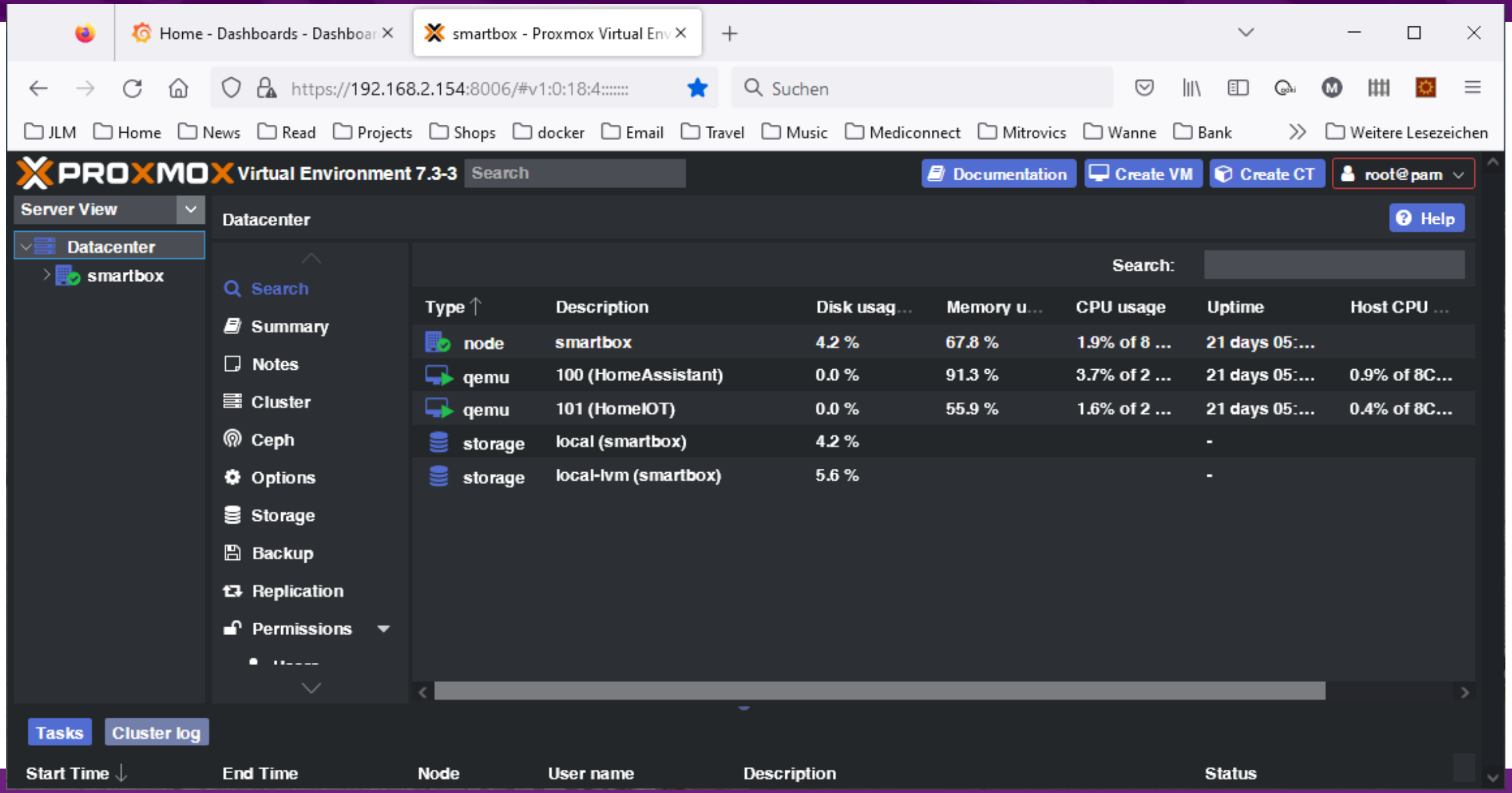
Server

- Cloud Server / Web Services: AWS / Azure / …
- Root Server running Linux
- Single Board Computers: Raspberry Pi

Services Infrastructure

- Virtual Machines (HyperV, Proxmox, VirtualBox, …)
- Containers (e.g. Docker)

Virtualization with Proxmox



Docker Containers

Advantages over Virtual Machines:

- Lightweight (share a common kernel)
- Easy to configure complete stack (via docker-compose)
- One container per service
  - Dependencies can be met per container!
- Virtual networks connecting containers
- Configurable network bridge to the outside
- Easy to move containers (e.g. from Docker on Windows to Linux)
- Huge repository of ready to use containers
  https://hub.docker.com/
- Easy to separate (and access) data via volumes

Creating a stack of services

A stack with all required services can be defined in one simple configuration file

All service can be started with one command docker-compose up –d

Containers can expose ports to the outside. Traffic between containers can be kept within the host

Images are automatically pulled from the hub.docker.com.

Data can be stored persistant in volumes that are also accessible on the host.

```
services:
  node-red:
    image: nodered/node-red:latest
    restart: unless-stopped
    networks:
      - node-red-net
    ports:
      - "1880:1880"
    volumes:
      - node-red-data:/data

  MQTTbroker:
    image: eclipse-mosquitto:1.6.13
    restart: unless-stopped
    networks:
      - node-red-net

volumes:
  node-red-data:

networks:
  node-red-net:
```

IOTstack a simple configuration tool to create a docker-compose.yamp file for IOT

https://sensorsiot.github.io/IOTstack/

Optimized for Rapberry PI

Simply select different services from a menu to build a complete stack

Can run on other Linux platforms, but requires small changes to remove Raspberry Pi specific parts

Management of Docker Containers via Portainer