Cistern 4.0

Fast Prototyping with ESP32 and Security in Mind

Stefan Smarzly

Outline

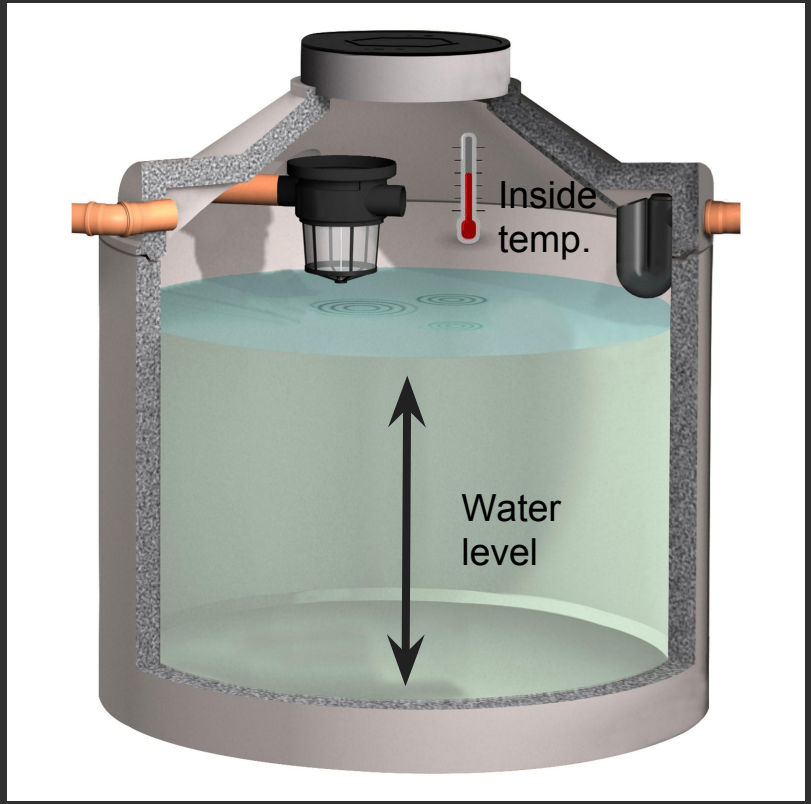
1. Motivation & Goals
2. System Architecture
3. Challenges & Lessons Learned
4. Demo

Current State & Motivation

- Cistern collects rainwater for garden watering

- Being too empty causes trouble with pump
- **But:** being quite empty can provide some buffer against heavy rain falls with an over-occupied sewerage

→ Live water level is useful

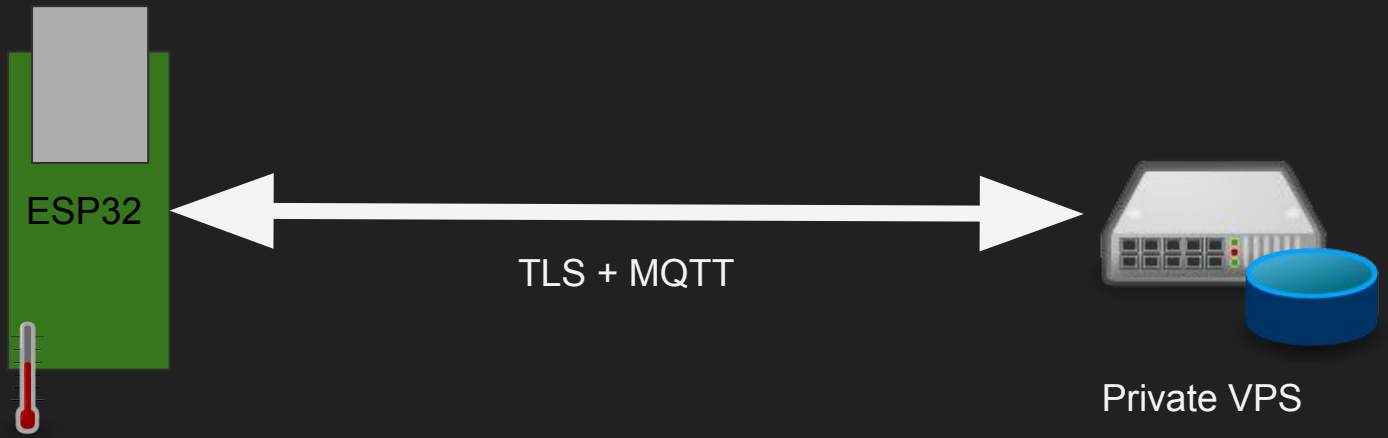


Goals

- Measure & log water level and temperature
- Secure (Network & infrastructure layer)
- Individual update & rollback
- Easy to build with few building blocks (HW & SW)

Architecture

- Centralized (or typical IoT) approach for sake of easiness during ramp-up
- Self-hosted infrastructure

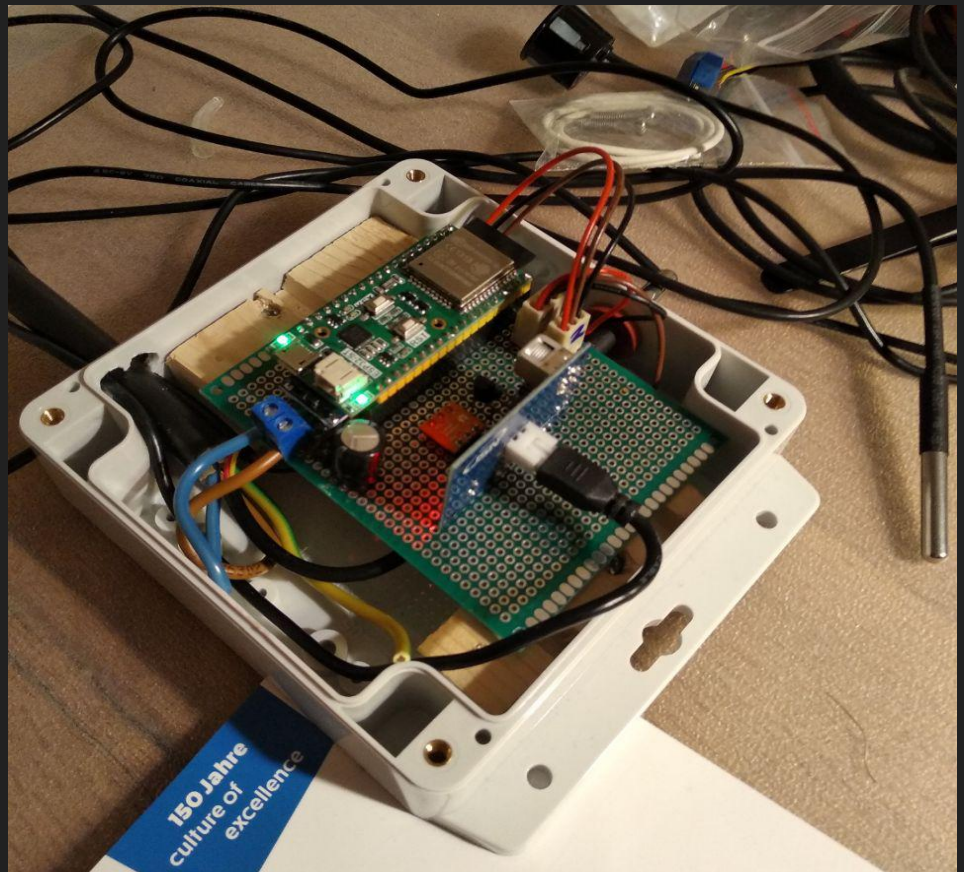


Hardware: ESP32 + Ultrasound + Temperature

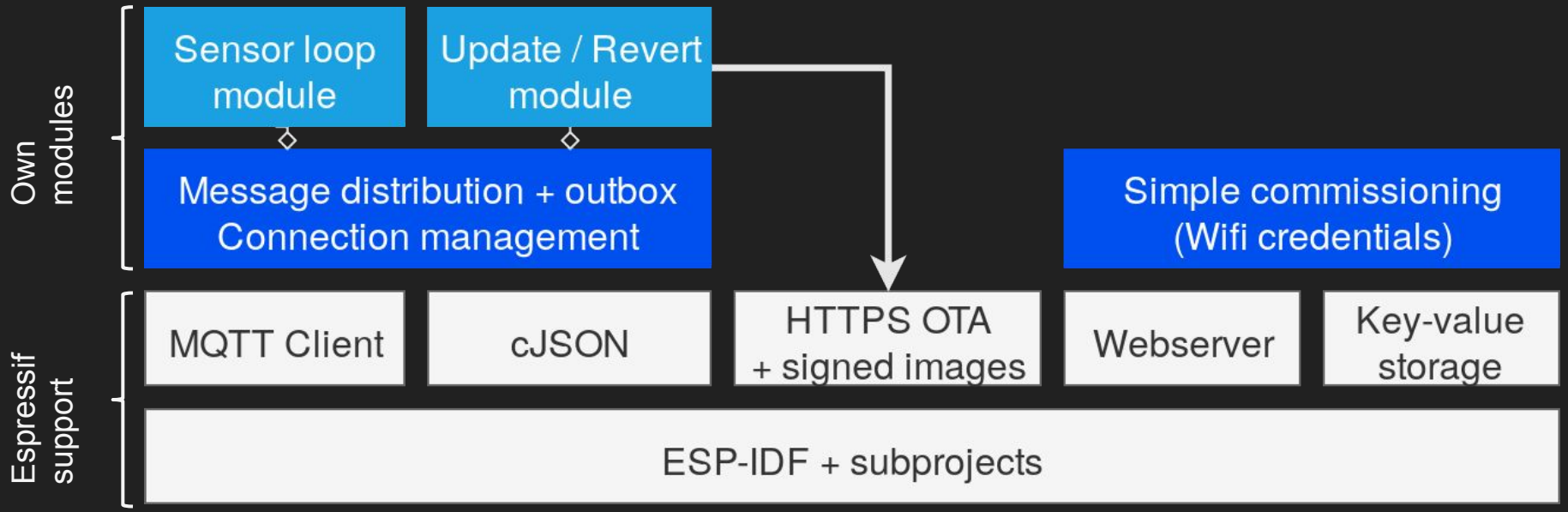
Few components to make a full-featured product

- ESP32-WROOM-32 board
- Ultrasound distance module: JSN-SR04T 2.0
- One-wire temperature sensors: DS18B20
- Buck converter: 12V → 5V
- Few other small parts: logic level shifter, capacitor, transistor, resistors

Put everything in a weather-resistant box



ESP32 Software: Building Blocks



ESP-IDF: FreeRTOS with multi-core support adapted for ESP32

Insight: Sensor Loop

- Adopt Zephyr's
  - Device model
  - Sensor API
- Enables batch processing and clean code

```
static const struct sensor_query sensor_query_table[]
= {
    [0] = {
        .dev = "ULTRASOUND_0",
        .ch = SENSOR_CHAN_DISTANCE,
        .precision = 3,
        .update_threshold = 0.007,
        .sensor_type = "level",
        .unit = "m",
    },
    [1] = {
        .dev = "TEMPERATURE_0",
        .ch = SENSOR_CHAN_AMBIENT_TEMP,
        .precision = 2,
        .update_threshold = 0.07,
        .sensor_type = "inside_temp",
        .unit = "C",
    },
    [2] = {
        .dev = "TEMPERATURE_1",
        .ch = SENSOR_CHAN_AMBIENT_TEMP,
        .precision = 2,
        .update_threshold = 0.07,
        .sensor_type = "outside_temp",
        .unit = "C",
    },
};
```

Securing the Connection

- **CIA**: Confidentiality, Integrity, Authenticity

- TLS Mutual Authentication with x509 client certificates
- Client certs encode serial number as CNAME for authentication
- Certificate Chain of Trust

Certificate Chain of Trust

Server Cert: **mqtt1.ingress.mycistern.de**
Issuer's name
Public key
Signature

Intermediate CA: **i1.ingress.mycistern.de**
Issuer's name
Public key
Signature

Signs

Intermediate CA: **i1.dev.mycistern.de**
Issuer's name
Public key
Signature

Signs

Device Cert: **ZISTERNE0001**
Issuer's name
Public key
Signature

1 Device identity *(server-side verify)*

1

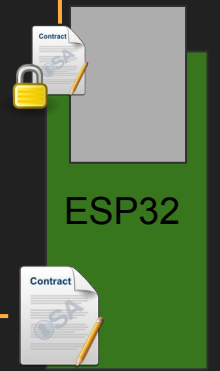
Intention:
Intermediate CAs for automated cert generation without putting RootCA at risk

Signs

RootCA: **mycistern.de**
RootCA: public key
RootCA: signature

Signs

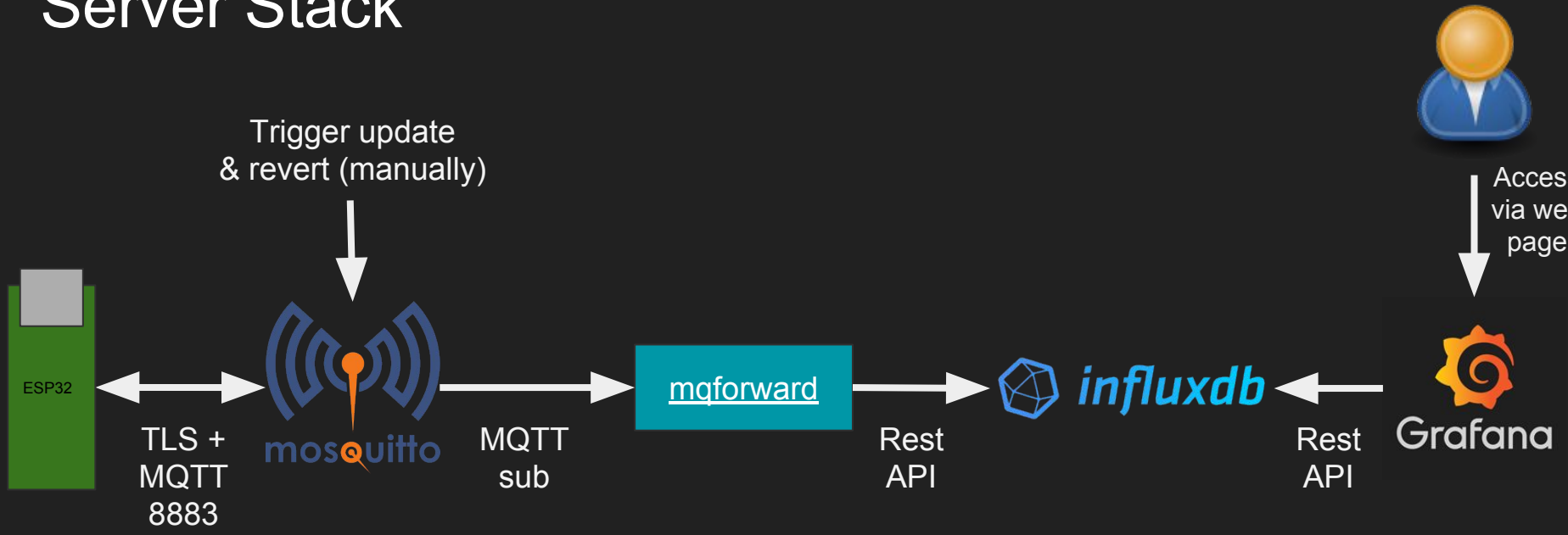
Verify server identity via
Cert + CN check (*.ingress)



Cistern 4.0 | Stefan Smarzly

Note: use dedicated RootCAs for higher security

Server Stack



Some MQTT Messages

Periodic sensor updates

sensor/zisterne0001/level {"value":0.942,"type":"level","unit":"m"}
sensor/zisterne0001/temperature {"value":8.25,"type":"inside_temp","unit":"C"}
sensor/zisterne0001/temperature {"value":9.06,"type":"outside_temp","unit":"C"}

Update control

sensor/zisterne0001/update

{"url":"https://myzisterne.de/firmware/zisterne_abcdefg_v0.1.bin"}

sensor/zisterne0001/update/revert

What renders the project challenging and unexpected (for me) in real-life?

Constructional Conditions

- Massive three-part division causes unexpected ultrasound reflection

Constructional Conditions (2)

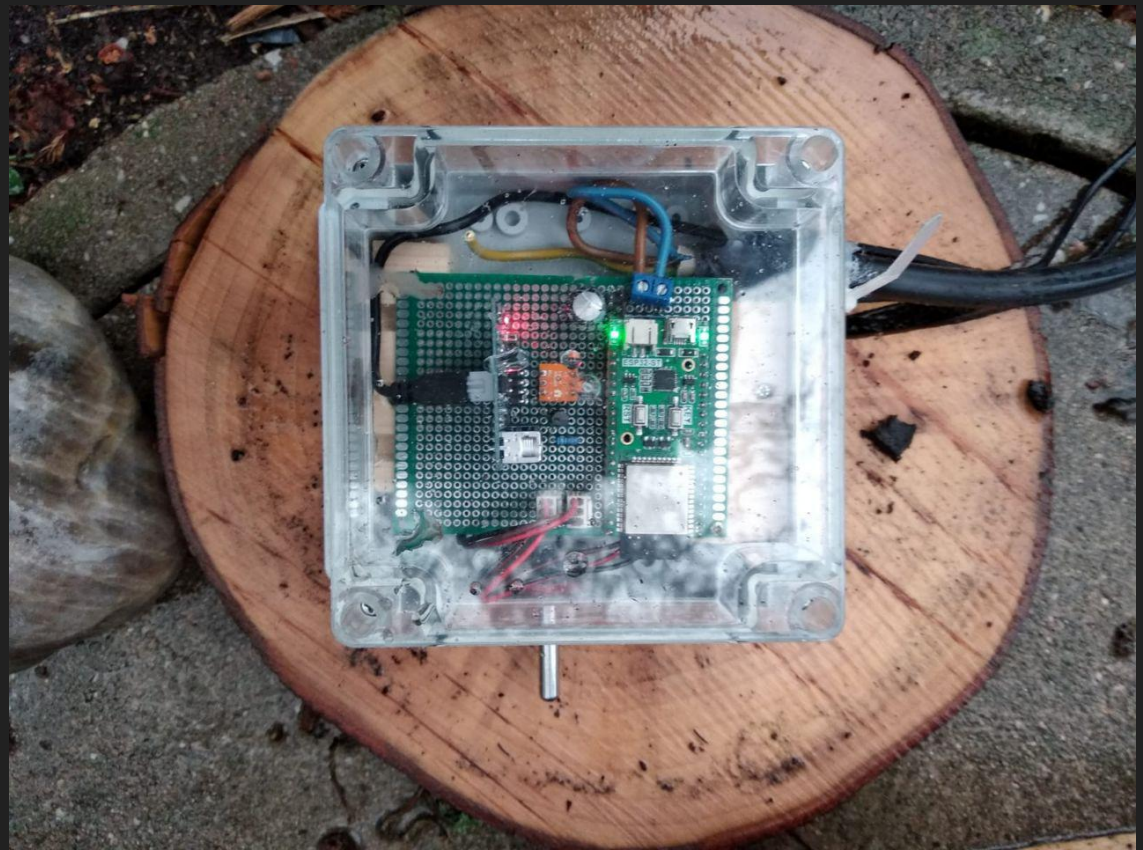
→ Offset sensor to align it as good as possible

Digging an Underground Cable into the Soil

- Harder than expected for a distance of 3m and depth around 15-25cm
- No worries: only 12V applied

Harsh Weather Conditions

- Sensor box lives outside of cistern due to very bad Wifi reception inside of it
- Keep it dry with a good housing
- Use a flower pot upside-down as secondary guard



Electronics Side: Lessons Learned

- Cheap Ultrasound module fails measuring at some point in time
- No reset line exists

- Add transistor to cut power if measurement fails
- Bi-directional logic-level shifter will provide power via IO pins if done wrongly

→ Certain state of pins needed to actually reset it

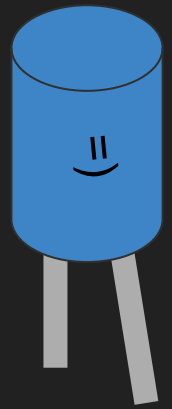
Copyright: https://hackspark.fr/en/electronics/1395-waterproof-ultrasonic-module-jsn-sr04t.html

Disclaimer: Computer Science background here ;)

Electronics Side: Lessons Learned (2)

- Expensive buck converter dies after applying 5V to V_OUT (for debugging)
- Cheap buck converter replacement **almost** does the job

- Connect to ESP32 board → WIFI very unreliable or even not working at all
- Add proper capacitor to V_OUT → WIFI stable



Disclaimer: Computer Science background here ;)

Copyright:
Fabulous own work

Demo

Kibana View


Zisterne
Last 24 hours
Temperatur
12
10
8
6
4
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
min
max
avg
zisterne0001.temperature.mean {type: inside_temp}
7.77
8.37
8.13
zisterne0001.temperature.mean {type: outside_temp}
5.06
11.52
9.10
Füllstand
0.950
0.945
0.940
0.935
0.930
0.925
0.920
zisterne0001.level.mean
0.92330
0.94650
0.94226
MQTT Verbindungsaufbau
1.25
1.00
0.75
0.50
0.25
0
zisterne0001.status.count

Long-term Goals

- Make it lasting (observation ongoing ;)
- Make my parents happy

Vision

- Make cloud connection optional
- Move processing logic into decentralized components

Resources

- OSA Icon Library 13.05: icons for graphical illustrations
  - Website: http://www.opensecurityarchitecture.org/cms/library/icon-library
  - License: Creative Commons Share-alike
- Awesome tutorial about creating a Certificate Chain of Trust
  - Website: https://jamielinux.com/docs/openssl-certificate-authority/introduction.html

Backup Slides

Ultrasound: Calculate Distance

$$d_{surface} = t_{travel} * (331.4 + 0.6T_C) / 2 \text{ [m]}$$

- Speed of Sound depends on multitude of environmental facts
- In our case: temperature has the biggest impact

→ Not implemented, to be done :)