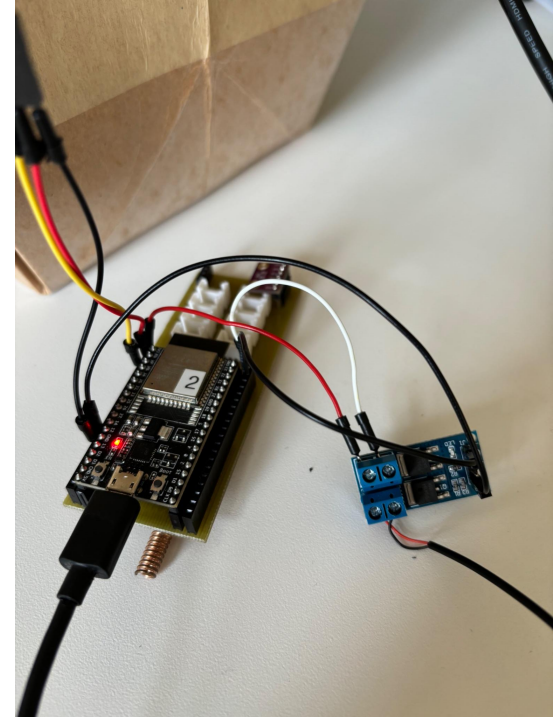
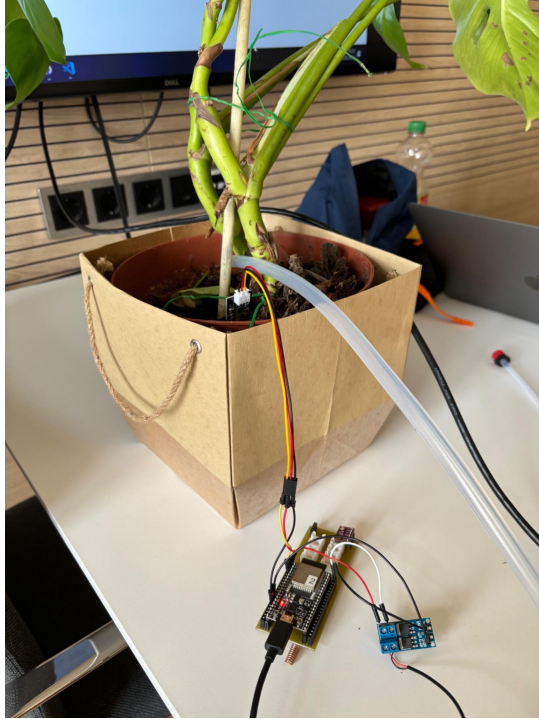


# Bewässerungsprojekt

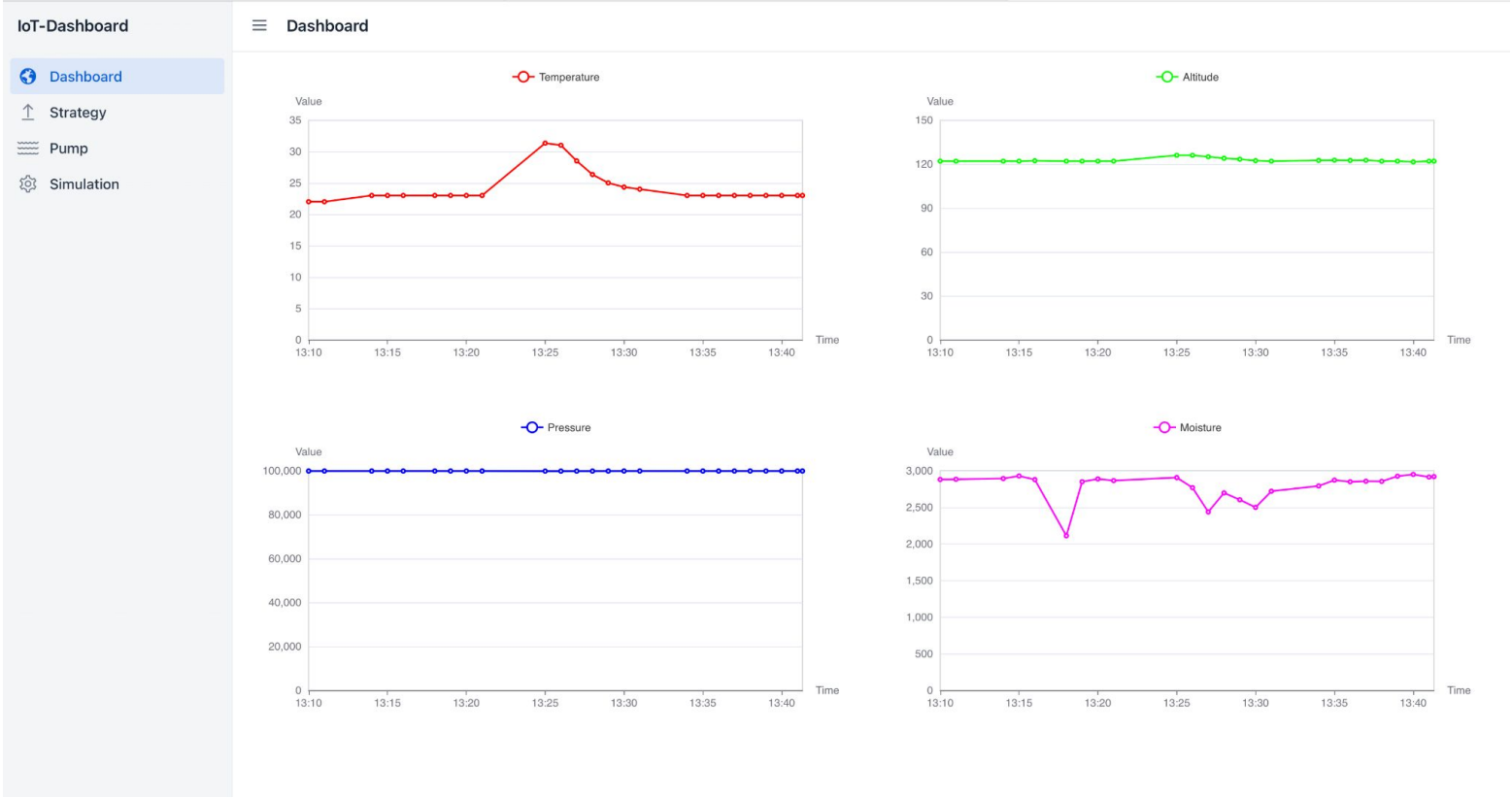


Ibrahim Al Krad, Oliver Greiner-Petter, Yannick Schössow

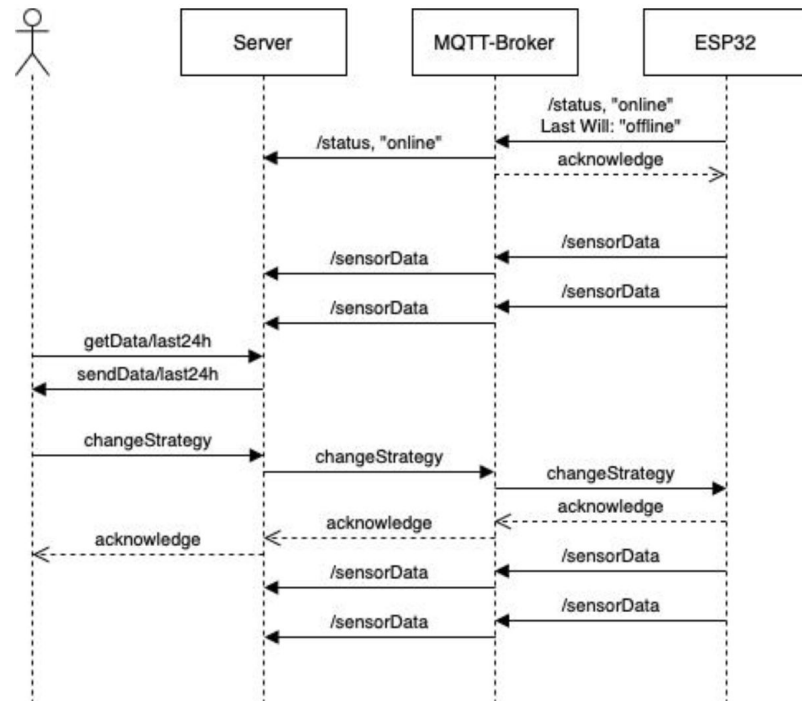
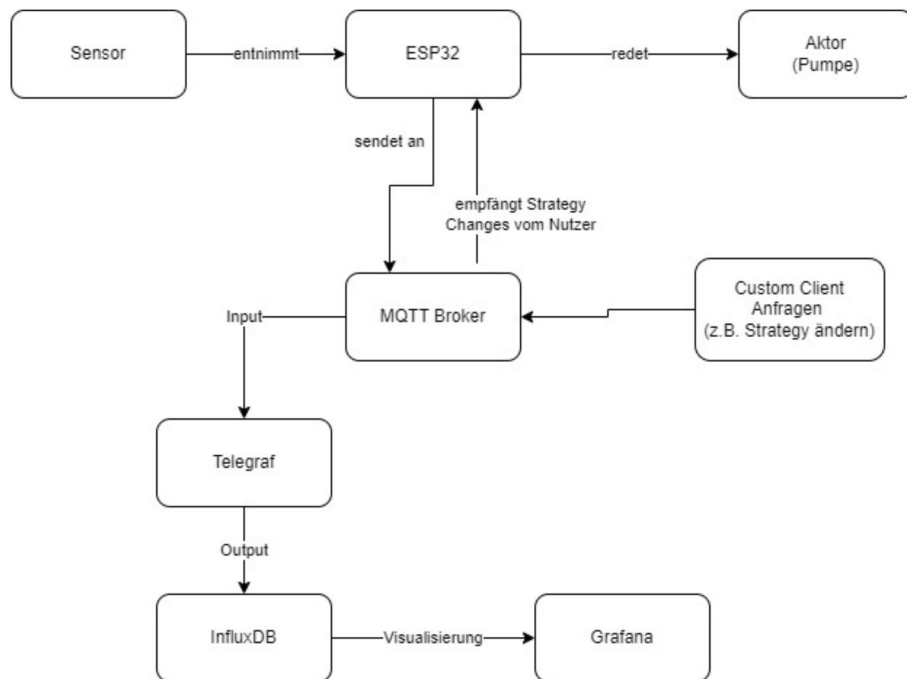
# Resultat 🏆



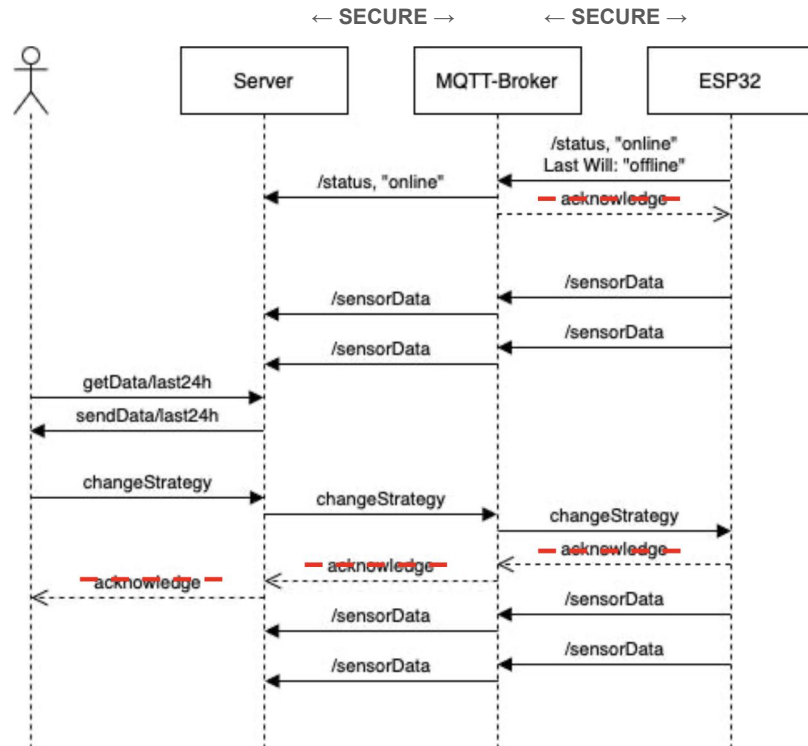
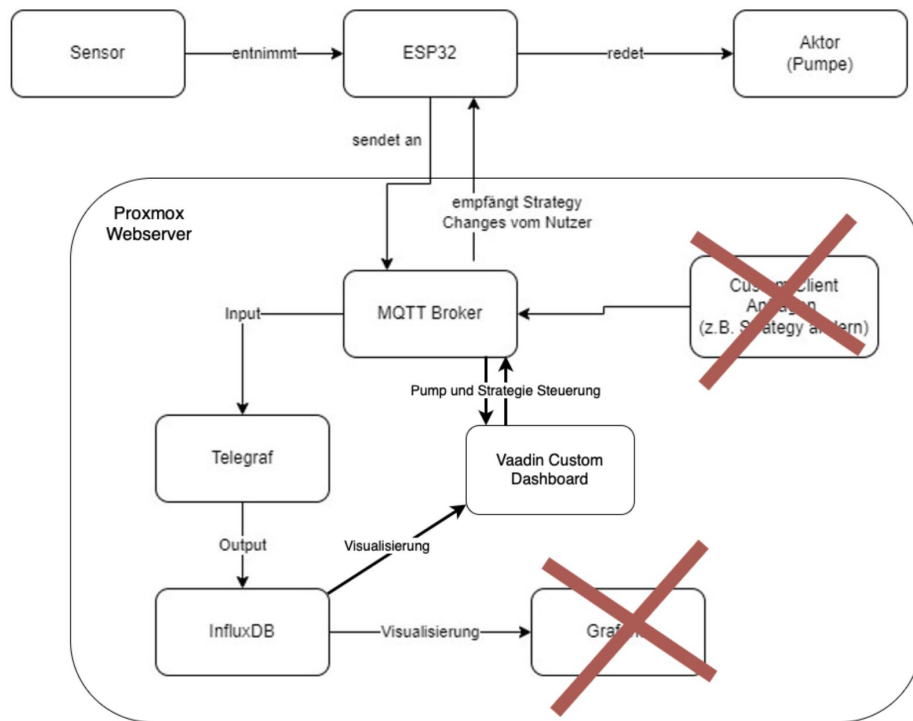
# Dashboard View



# Vorläufiger Plan



# Tatsächlicher Plan





# User Stories


Rolle	Task	Grund	Akzeptanzkriterien
Benutzer	Aktuelle Messwerte für Helligkeit, Temperatur und Bodenfeuchtigkeit meiner Pflanze sehen	um ihren Zustand zu überwachen	<ul style="list-style-type: none"><li>- Benutzeroberfläche zeigt die aktuellen Werte für jede Pflanze an.</li><li>- Werte in alle 5 Minuten aktualisiert.</li><li>- Benutzeroberfläche benutzerfreundlich und leicht zu verstehen.</li></ul>
Benutzer	Bewässerungs Strategie für jede seiner Pflanzen anpassen können	um ihre Bedürfnisse zu erfüllen	<ul style="list-style-type: none"><li>- Benutzeroberfläche ermöglicht Bewässerung Häufigkeit und -dauer für jede Pflanze einzustellen.</li><li>- Benutzer kann Schwellenwerte für Bodenfeuchtigkeit festlegen, ab denen die Bewässerung ausgelöst wird.</li><li>- Änderungen an der Bewässerungs Strategie werden innerhalb von einer Stunde wirksam.</li></ul>
Benutzer	System aktiviert die Bewässerungspumpe automatisch aktiviert, wenn die Bodenfeuchtigkeit unter einen bestimmten Schwellenwert fällt	Um die Pflanze zu bewässern	<ul style="list-style-type: none"><li>- System überwacht regelmäßige Bodenfeuchtigkeit mithilfe entsprechendes Sensors.</li><li>- Wenn Bodenfeuchtigkeit unter den vom Benutzer festgelegten Schwellenwert fällt, wird Bewässerungspumpe automatisch eingeschaltet.</li><li>- Pumpe bleibt für vom Benutzer festgelegte Bewässerungsdauer aktiv.</li><li>- Nach Ablauf der Bewässerungsdauer wird Pumpe automatisch ausgeschaltet.</li><li>- System protokolliert alle Bewässerungsvorgänge, einschließlich Zeitpunkt der Aktivierung und Deaktivierung der Pumpe sowie der gemessenen Bodenfeuchtigkeit.</li></ul>
Benutzer	Historie der Pump Vorgänge einsehen	um mögliche Unregelmäßigkeiten zu sehen	<ul style="list-style-type: none"><li>- Benutzeroberfläche zeigt liste der letzten Pump Aktivierungen an</li><li>- Einträge sind sortierbar</li><li>- Wenn seit 3 Tagen die Pump nicht Aktiviert wurde wird eine Nachricht im Dashboard angezeigt</li></ul>


# Strategy View


IoT-Dashboard

 Dashboard

 Strategy

 Pump

 Simulation

 Strategy

Pump Duration

1500

Low

Medium

High

Moisture Threshold

2800


Low

Medium

High

Sleep Duration

20000

 Upload Strategy

# Strategy Beurteilung

## Vorteile:

- Sehr simple Logik (nur abhängig von 3 Parametern)
- Einfach für den end user zu verstehen
- Presets für schnelle Einrichtung

## Nachteile:

- Jahreszeitenunabhängig
- Thresholds nicht abhängig von anderen Variablen (z.B. Tageszeit)

Zusatz: Feuchtigkeitssensor ionisiert die Erde und kann ggf. die Pflanze töten



# Pump View

IoT-Dashboard

Dashboard

Strategy

Pump

Simulation

Pump

Duration

2000

Activate

Revoke

Time	Duration
05.06.24, 13:15	2000.0
05.06.24, 13:16	3000.0
05.06.24, 13:19	3000.0
05.06.24, 13:20	2500.0
05.06.24, 13:21	1500.0
05.06.24, 13:25	1500.0
05.06.24, 13:26	4333.333333333333
05.06.24, 13:34	1750.0
05.06.24, 13:35	1500.0
05.06.24, 13:36	1500.0

# Herausforderungen

## Problem 1: Certificate Handshake failure

```
iot-webserver-broker-1 1716384120: New connection from 141.41.35.39:59762 on port 8884.  
iot-webserver-broker-1 1716384120: OpenSSL Error[0]: error:14035412:SSL routines:ACCEPT_SR_CERT:sslv3 alert bad certificate  
iot-webserver-broker-1 1716384120: OpenSSL Error[1]: error:140350E5:SSL routines:ACCEPT_SR_CERT:ssl handshake failure  
iot-webserver-broker-1 1716384120: Client <unknown> disconnected: Protocol error.
```

**Lösung:** Zertifikate neu erstellen und Einstellungen überprüfen

---

## Problem 2: PowerManagement verhindert im DeepSleep das Ansprechen

**Lösung:** MQTT messages retained setzten und beim wake up ausgelesen und verarbeitet

---

**Problem 3:** Docker Probleme, config abhängig von automatisch generierten tokens, dashboard nicht containerizable

**Lösung:** Statischen influxDB Token; Shell-Script für docker compose und das Deployment des Vaadin Dashboards

## Definition of Done

Alle Akzeptanzkriterien sind erfüllt:

- ✓ Custom Vaadin Dashboard ist auf Web-Server deployed,
- ✓ ESP32 sendet Daten und setzt die gewählte Strategie um,
- ✓ Einstellungen können über das Dashboard angepasst werden.

# Retrospektive

Was lief gut?

- Versionskontrolle via GitLab
- Aufgabenverteilung
- Austausch per auftretenden Problemen
- Bearbeitung der Tasks

Wo besteht Optimierungsbedarf?

- Aufwandseinschätzung bevor Umsetzung (Unterschätzt)
- Dokumentation der Sprints und Arbeitsprotokolle
- Code Dokumentation

Was haben wir gelernt?

- Securityaspekte beim Design berücksichtigen
- Aufwand großzügiger einschätzen
- Ressourcenplanung bei schwacher Hardware

Nach Sprint I
<ul style="list-style-type: none"><li>- Design und Mockups für das Setup</li><li>- Teamuhr nach Tuckman: Norming</li><li>- Servereinrichtung</li><li>- Austausch über weitere Tasks</li><li>- Anbindung von Grafana, InfluxDB und Telegraf</li><li>- Objektorientierte Entwicklung von Hilfsklassen für die Sensoren</li></ul>
Nach Sprint II
<ul style="list-style-type: none"><li>- Teamuhr nach Tuckman: Performing</li><li>- Umstrukturierung des Setups</li><li>- Umstieg auf Vaadin</li><li>- Entwicklung des Dashboards</li><li>- Umsetzung eines Powermanagementkonzepts</li><li>- Tests Durchführung</li></ul>
Aufwand
<b>60 Std. gesamt Aufwand</b> <ul style="list-style-type: none"><li>- 50 Std. (Pair-)Programming, Implementation, Testing</li><li>- 5 Std. Dokumentation</li><li>- 5 Std. Design, Mockup, etc.</li></ul>



**Live Demo**



# **Fortsetzung Dokumentation**

# Docker

Der Web-Server wird mit Docker deployed.

*# Zum Starten aller Container:*

```
sudo docker compose up -d
```

*# Für das Dashboard:*

```
mvn clean package -Pproduction
```

```
nohup java -jar target/iot-dashboard-1.0-SNAPSHOT.jar &
```