

# SentraQ Plattform

Eine Open-Source Software unter der GPL 3-0 Lizenz

Technische Dokumentation der web-basierten Plattform zur Visualisierung von Versorgungseinrichtungen (Wasser/Energie).

Version 1.2

Erstellt durch

The logo for Watermann IT, featuring a stylized blue 'W' followed by the text 'atermann IT' in a blue sans-serif font.

Stefan Watermann

Am Margland 4

31749 Auetal

[www.watermann-it.de](http://www.watermann-it.de)

## Änderungsverlauf

Datum	Version	Grund der Änderung	Autor(en)
9.1.2026	1.0	Initialerstellung	S. Watermann
4.2.2026	1.1	Anpassung in Abschnitt „Datenübertragung“	S. Watermann
9.2.2026	1.2	Ergänzungen	S. Watermann

SentraQ wird als Open-Source Software in Github verwaltet: <https://github.com/stefanwatermann/SentraQ>

Irrtum und Änderungen vorbehalten. Der Autor und Watermann IT übernehmen keine Haftung für die Vollständigkeit von Angaben, Hinweisen und Ratschlägen, die in dieser Dokumentation aufgeführt sind.

Genannte Warenzeichen oder Produktnamen sind ggf. Markenzeichen der jeweiligen Unternehmen.

Alle Rechte vorbehalten. Die unautorisierte Vervielfältigung dieser Dokumentation, auch auszugsweise, ist nicht gestattet.

© 2026 Watermann IT, Auetal.

## Inhaltsverzeichnis

1. Definitionen.....	2
2. Architektur .....	4
2.1. Logische Architektur .....	4
2.2. Technische Architektur.....	5
2.3. Kommunikation .....	6
3. Konfiguration Siemens LOGO8 .....	7
3.1. Beispiel-Station.....	7
3.2. Datenübertragung .....	8
3.3. Cloud-Verbindung .....	9

# 1. Definitionen

## **SentraQ**

Ein web-basiertes Software-System für den Empfang, die Speicherung und Visualisierung von Signalen und Zuständen von Versorgungseinrichtungen (Stationen).

## **Station**

### **A) Physisch**

Eine örtliche Versorgungseinrichtung, wie ein Pumpwerk, eine Abwasserstation oder Ähnliches. Hauptmerkmal ist ihr geografischer Ort (Länge und Breite).

### **B) Logisch**

Eine digitale Ausprägung einer physischen Einrichtung (digitaler Zwilling).

Die logische Station

- besitzt einen eindeutigen Namen
- hat eine eindeutige **UID**
- hat eine geografische Position (Latitude, Longitude)
- hat 1-n Komponenten

## **Komponente**

### **A) Physisch**

(Hardware-) Komponente einer örtlichen Versorgungsanlage, z.B. eine Pumpe oder ein Behälter, die ein Signal sendet (Sender).

### **B) Logisch**

Eine digitale Ausprägung einer physischen Komponente zur Darstellung der Zustände.

Die logische Komponente

- ist entweder ein Aktor, Sensor oder Zähler oder eine Störung (Typ)
- hat einen (Anzeige-) Namen
- hat eine eindeutige **HardwareID**
- ist einer einzelnen Station zugeordnet
- kann einen Ist-Wert und einen Soll-Wert haben
- hat je nach Typ eine visuelle Darstellung in der Web-Oberfläche

Komponententypen sind:

- Sensor -> liefert einen analogen Wert (z.B. Füllstand oder Durchfluss)
- Aktor -> liefert die Zustände 0 oder 1
- Zähler -> liefert einen statischen Wert
- Störung -> liefert die Zustände „0 = keine Störung“ oder „1 = Störung“

Jedes von einer Station zu sendende Signal (Schalter, Analogwert, ...) wird in der Plattform durch eine eindeutige Komponente repräsentiert.

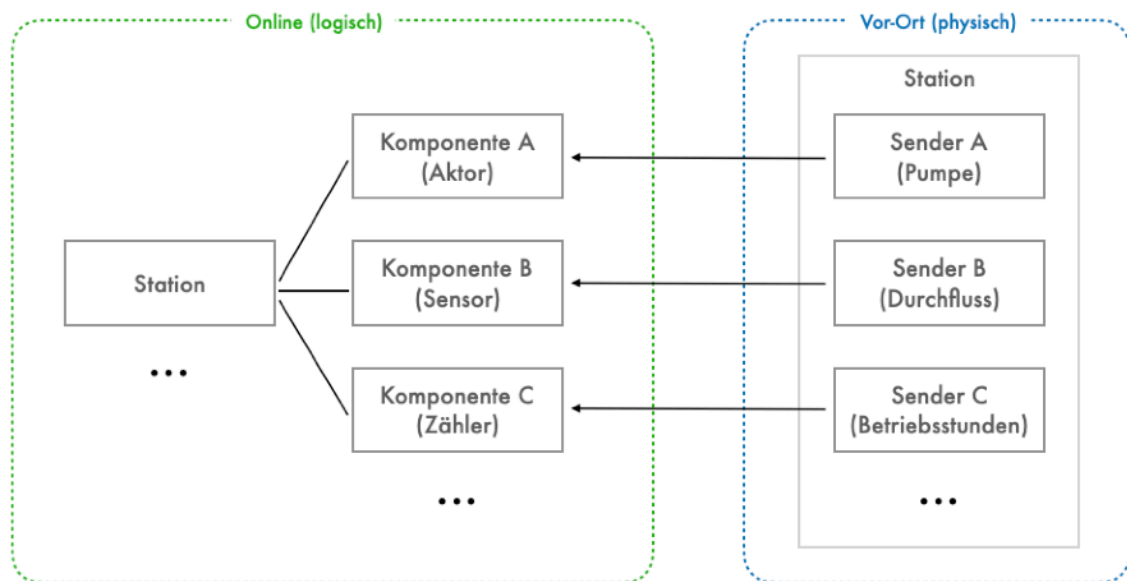
Eine Komponente ist stets nur einer einzelnen Station zugeordnet.

UIDs und HardwareIDs werden ausschließlich durch die Plattform vergeben, da nur diese die Eindeutigkeit dieser sicherstellen kann.

## 2. Architektur

### 2.1. Logische Architektur

Die logische Architektur beschreibt den Zusammenhang zwischen Station und Komponente. Jede Station verfügt über mindestens eine Komponente und kann beliebig viele<sup>1</sup> Komponenten enthalten. Jeder logischen Komponente (Online) ist ein physischer Sender (Vor-Ort) zugeordnet, wie auch jede logische Station die Repräsentation einer physischen Station ist.



Die WwP kann beliebig viele Stationen verwalten.

<sup>1</sup> Wird in diesem Dokument die Formulierung „beliebig viele“ verwendet, so ist damit die jeweils technisch sinnvolle Maximalmenge gemeint (nicht unendlich). Als Maximalwerte gelten somit die sich ergebenden Grenzen je nach technischer Ausstattung (Speicherplatz, CPU Leistung, Bandbreite) des WwP Hosting-Systems.

## 2.2. Technische Architektur

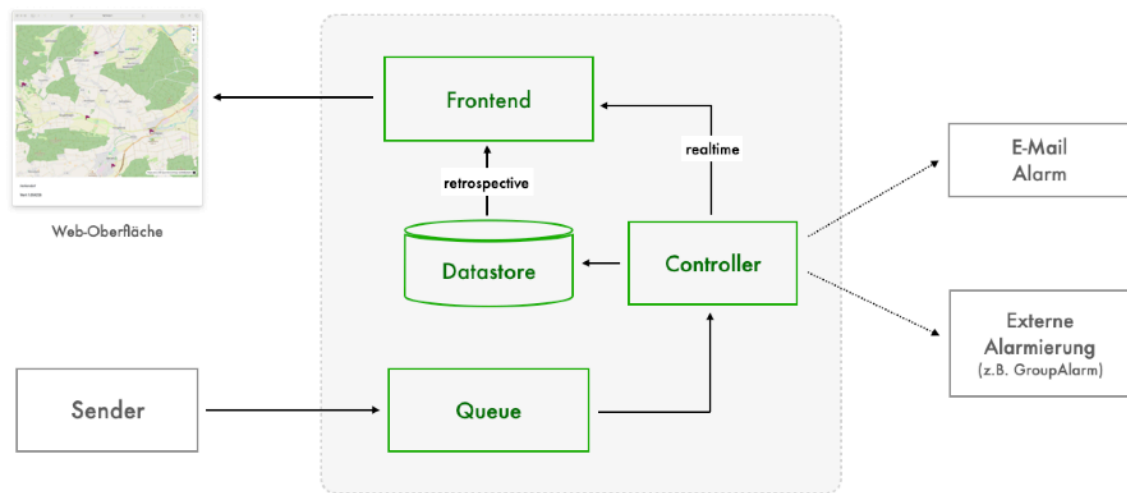
Die technische Architektur beschreibt die Software-Bausteine der WwP.

Das **Frontend** stellt die Benutzeroberfläche zur Visualisierung der Stationen und Komponenten bereit.

Die **Queue** empfängt kontinuierlich MQTT Nachrichten von den physischen Komponenten.

Der **Datastore** speichert alle empfangenen Daten der Komponenten dauerhaft.

Der **Controller** dient als Vermittler zwischen den anderen Bausteinen und ist für Alarmierungen und Automatisierungen zuständig.



Der **Sender** ist ein aktives System, welches MQTT Nachrichten erzeugt und über das Internet an die Queue der Plattform sendet.

Die Plattform unterstützt generische MQTT Nachrichten, wie auch Nachrichten der Kleinststeuerung Siemens LOGO8.

## 2.3. Kommunikation

Die Kommunikation zwischen physischen Komponenten und der Plattform basiert auf MQTT Nachrichten.

Das Payload einer MQTT Nachricht enthält mindestens 2 verpflichtende Elemente:

1. `hid` = eindeutige HardwareID der Komponente aus der Plattform
2. `value` = Istwert der Komponente (z.B. 0 oder 1 bei einem Aktor)

Optional:

3. Zeitstempel im UNIX Format oder als String „YYYY-MM-DD hh:mm:ss“

Die UID der Station ist für die Kommunikation nicht relevant, da zwischen Komponente und Station eine 1:1 Beziehung besteht.

Beispiel für ein generisches MQTT Payload:

```
{ "value":0, "hid":"Q3DCMDNFEH", "ts":"2025-09-29 15:38:56" }
```

Beispiel einer Siemens LOGO8 MQTT Payload:

```
{ "state": { "reported": { "Q3DCMDNFEH": { "desc": "M-bit-1-1", "value": [0] },  
"$logotime": 1767973154 } }
```

Der Bezeichner des JSON Objektes „reported“ wird als HardwareID (hid) verwendet. Der Wert (value) wird als Array erwartet (Einstellung in der Siemens LOGO8 Datenübertragung). Nur das erste Element des Arrays wird verwendet.

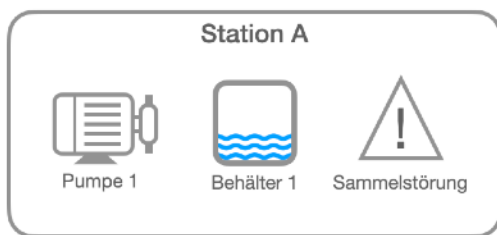
Den Zeitstempel \$logotime sendet die Siemens LOGO8 nur für zeitgesteuerte Übertragungen. Änderungsübertragungen senden keinen Zeitstempel. Daher ist die zeitgesteuerte Übertragung vorzuziehen.

### 3. Konfiguration Siemens LOGO8

Die erforderliche Konfiguration der Datenübertragung einer Siemens LOGO8 Kleinststeuerung wird anhand der nachfolgenden Beispiel-Station illustriert.

#### 3.1. Beispiel-Station

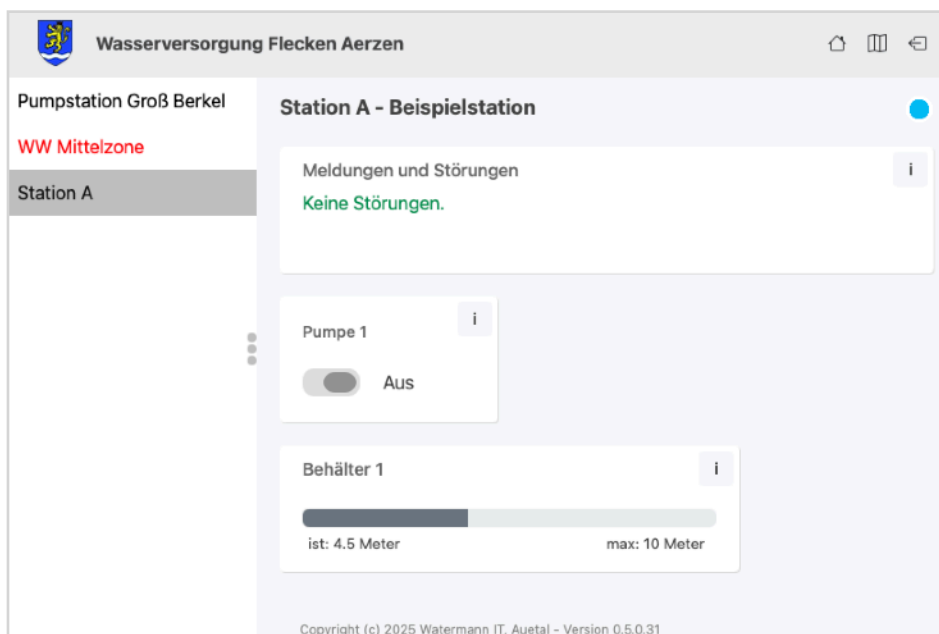
Angenommen eine Pumpstation (Station A) verfügt über eine Wasserpumpe (Pumpe 1), einen Störkontakt (Sammelstörung) und den Füllstand eines Behälters (Behälter 1).



Schema physische Beispiel-Station

In der Plattform ist dazu „Station A“ mit folgenden Komponenten eingerichtet:

1. „Pumpe 1“ als Aktor mit HardwareID „9HQAUF76DW“
2. „Sammelstörung“ als Störung mit HardwareID „H3KFZ222ZZ“
3. „Behälter 1“ als Sensor mit HardwareID „7WYZUEQHTK“



Darstellung logische Beispiel-Station in der Plattform

### 3.2. Datenübertragung

Die Siemens LOGO8 Steuerung verbindet die Signale der physischen Komponenten mit den logischen HardwareIDs über die „Einstellungen für die Cloud-Datenübertragung“.

In der Beispiel-Station ist die Sammelstörung in der Steuerung durch den Merker *M1* definiert. In der Konfiguration der Datenübertragung wird die HardwareID der logischen Sammelstörungskomponente als *Name* verwendet und der Adresse *M1* zugeordnet.

In gleicher Weise wird mit allen weiteren Signalen und Komponenten verfahren.

[illegible]

## LOGO8 Konfiguration Beispiel-Station

Wird für alle Komponenten der gleiche Zeitraum (hier 15 Sekunden) verwendet, sendet die Siemens LOGO8 zu diesem Zeitpunkt eine einzelne MQTT Nachricht mit allen Werten zusammen. Die Nachricht enthält in diesem Falle auch den Sendezeitpunkt im UNIX Format. Diese Konfiguration ist in jedem Fall zu bevorzugen, da somit das Nachrichtenvolumen reduziert wird.

Hinweis: Eine CSV Datei für den Import der Konfiguration zur Cloud-Datenübertragung kann aus der Plattform über den Stations-Info Dialog heruntergeladen werden (Button „LOGO8 CSV“).

### 3.3. Cloud-Verbindung

Die Kommunikation der Kleinststeuerung zur Plattform wird über die „Einstellung für Cloud-Verbindung“ eingerichtet.

Um einen sicheren Datenaustausch zu gewährleisten, muss die Kommunikation per SSL (TCP Port 8883) erfolgen.

#### ***Parameter für die Cloud-Verbindung zur Plattform.***

Cloud-Typ	MQTT
URL des MQTT-Brokers	<i>[Plattform URL]</i>
Port	8883
MQTT-Client-ID	<i>[UID der Station]</i>
Benutzername	<i>[MQTT Nutzer]</i>
Passwort	<i>[Nutzer-Passwort]</i>
Authentifizierungstyp	TLS Ohne Authentifizierung
Publish-Thema	client/send/ <i>[Publish-GUID]</i>
Publish-QoS	QoS0
Subscribe-Thema	client/receive/ <i>[Subscribe-GUID]</i> / <i>[UID der Station]</i>

**Wichtig:** Das „Publish-Thema“ **muss** sich vom „Subscribe-Thema“ unterscheiden, damit keine ungewollten MQTT Nachrichten von der Kleinststeuerung versendet werden.