

# SCADA System

Benutzerhandbuch

*Version 1.0 — April 2026*

# Inhaltsverzeichnis

Vorwort

Schnellstart — 15 Minuten

**1. Anmeldung und Registrierung**

**2. Dashboard**

**3. Mnemoniksschemata**

**4. Schema Builder**

**5. Schema Viewer**

**6. Geräte**

**7. SPS**

**8. Sequenzen**

**9. Rezepte und Variablen**

**10. Ereignisse und Störungen**

**11. Optionen**

**12. Pulsmodule**

**13. Chargenverwaltung**

**14. Produktionslinien**

**15. Trends**

**16. Alarme**

**17. Mathematische Funktionen**

**18. E/A-Punkte**

**19. E/A-Provisionierung**

**20. Grafische Zustände**

**21. Hardware-Konfigurator**

**22. Benutzerverwaltung**

**23. Rollenverwaltung**

**24. Aktivitätsprotokoll**

**25. Einstellungen**

**26. KI-Assistent**

**27. SPS-Projekt-Export**

**28. Projekt-Export und -Import**

**29. Tanks**

Anwendungsnavigation (Seitenleiste)

Globale UI-Elemente

Tastenkombinationen

Praktische Beispiele

Fehlerbehebung und FAQ

Sicherheit und Best Practices

Glossar

Anhang A: WebSocket-Kanäle-Referenz

Anhang B: API-Referenz (Übersichtstabelle)

Anhang C: Fehler- und Alarmcodes

Dokumentversion: 1.0

Datum: April 2026

Oberflächensprachen: Mehrsprachig (EN, RU, UK, DE, FR, PL, AR und weitere)

# Vorwort

## Über das System

SCADA System ist eine moderne Plattform für Prozessleittechnik und Datenerfassung (Supervisory Control and Data Acquisition), die von Grund auf für die nächste Generation industrieller Automatisierung entwickelt wurde. Das System ist auf den gesamten Lebenszyklus der Prozesssteuerung ausgelegt — von der Gerätekonfiguration und der Erstellung mnemonischer Schemata bis zur Echtzeit-Bedienung von Produktionslinien.

Im Gegensatz zu klassischen SCADA-Systemen, die aus der Zeit der Desktop-Anwendungen und proprietären Protokolle stammen, basiert diese Plattform auf einem modernen Web-Stack und auf Prinzipien der Microservice-Architektur. Das bedeutet, dass der Bediener das System aus jedem Browser heraus nutzen kann — am Industrie-PC in der Halle, am Tablet im Labor oder am Notebook aus dem Büro — ohne zusätzliche Software zu installieren.

Das System deckt den gesamten Lebenszyklus einer automatisierten Produktion ab:

- Engineering — visueller Konstruktor für mnemonische Schemata, Gerätekonfiguration, Erstellung von Steuerungsalgorithmen (Sequenzen) und Rezepten
- Bereitstellung — automatische Erzeugung von SPS-Code (Siemens SCL/XML), Provisionierung der E/A-Punkte, Deployment über TIA Portal Agent
- Betrieb — Echtzeit-Prozessüberwachung, Gerätebedienung, Trendanalyse, Chargenverwaltung
- Instandhaltung — Aktivitätsprotokoll, Alarme, Diagnose, automatische Updates, Datensicherung

The screenshot displays the SCADA System Dashboard. At the top left, the 'SCADA' logo and 'Dashboard' title are visible. The top right shows system metrics: CPU: 21.4%, RAM: 42.8%, and time: 14:45:39. The user 'Serhii Kudlai' is logged in as Administrator. The dashboard is divided into several sections:

- SYSTEMSTATUS:** A warning icon indicates a system warning.
- Schemas:** A list of schemas including 'Пастеризатор HTST', 'Демо-линия', and 'Пастеризатор'.
- Produktionsbereiche:** A list of production areas including 'Pasteurization HTST', 'Biozid', 'Inhibitor', and 'Пастеризатор', each with a 'Gestoppt' (Stopped) button.
- GERÄTESTATUSÜBERSICHT:** A summary of device status with four categories: NORMAL (77), WARNUNGEN (0), KRITISCH (0), and INAKTIV (8). A bar chart shows the distribution of devices by type, with a total of 85 devices.
- AKTUELLE ALARME:** A list of active alarms, including 'Motor P-2 Alarm' and 'Motor P-6 Alarm', both with 'Hoch' (High) priority.
- AKTUELLE TRENDS:** A line graph showing trends for three parameters: PT100-2, PT100-4, and PT100-8.

Collage aus 4 zentralen Bildschirmen des Systems: Dashboard (oben links), Schema Viewer mit laufendem Schema (oben rechts), Trends mit Diagrammen (unten links), Sequenz-Editor (unten rechts). Demonstriert die Breite der Funktionalität.

## Zielgruppe

Rolle	Nutzung des Systems
PLT-Planungsingenieur	Erstellt mnemonische Schemata, konfiguriert Geräte und Sequenzen, konfiguriert die SPS, erzeugt Code für TIA Portal
Prozessingenieur / Technologe	Entwickelt Produktionsrezepte, konfiguriert Prozessparameter, legt Alarmschwellen fest
Bediener einer Produktionslinie	Beobachtet das mnemonische Schema in Echtzeit, bedient die Ausrüstung (Start/Stop, Sollwerte), arbeitet mit Chargen und Rezepten
Schichtleiter / Meister	Überwacht den Zustand aller Linien über das Dashboard, analysiert Trends, arbeitet mit dem Störungsprotokoll
Systemadministrator	Verwaltet Benutzer und Rollen, aktualisiert das System, konfiguriert Verbindungen zu SPS und Netzwerk
Unternehmensleitung	Sieht Berichte, Leistungstrends, Statuswerte der Produktionslinien von überall über den Browser ein

## Wichtigste Vorteile

### 1. Vollständig webbasierte Plattform

Das System läuft in jedem modernen Browser ohne Client-Installation. Das unterscheidet es grundlegend von klassischen SCADA-Systemen (WinCC, Citect, InTouch), die auf jeder Arbeitsstation einen Fat Client benötigen, an ein konkretes Betriebssystem (meist Windows) gebunden sind und eine separate Lizenz pro Arbeitsplatz erfordern.

- Zugriff aus jedem Punkt des Netzwerks (Intranet oder per VPN)
- Läuft auf jedem Gerät: PC, Tablet, Industrie-Terminal
- Keine Betriebssystem-Bindung
- Sofortige UI-Updates für alle Benutzer gleichzeitig

### 2. Echtzeit-Datenaktualisierung

Eine zweistufige WebSocket-Architektur liefert Änderungen sofort an alle verbundenen Clients:

- Rust WebSocket-Server (Port 8001) — hochperformante Zustellung von Massen-Geräteaktualisierungen (bis zu Tausenden von Parametern pro Sekunde). In Rust geschrieben für minimale Latenz und maximalen Durchsatz.
- Django Channels WebSocket — bidirektionale Interaktion für Steuerbefehle (Motor Start/Stop, Ventil Öffnen/Schließen, PID-Sollwert ändern).

Delta-Updates übertragen nur geänderte Felder, und ein Batching (alle 250 ms) minimiert die Netzwerklast beim gleichzeitigen Aktualisieren Hunderter Parameter.

### 3. Integrierte SPS-Codegenerierung

Ein Alleinstellungsmerkmal, das den meisten SCADA-Systemen fehlt: Aus der Konfiguration von Geräten, Sequenzen, Rezepten und Störungen erzeugt das System automatisch ein vollwertiges Projekt für Siemens TIA Portal:

- 11 Funktionsbausteine (SCL) für jeden Gerätetyp (Motor FB, Valve FB, PID FB, AI/AO/DI/DO FB, COS FB, Counter FB, Timer FB, Tank FB)
- Sequence FB — Zustandsautomat für jede Sequenz
- Event / Accident FBs — Handler für Ereignisse und Störungen
- Data Blocks — Instanz-DBs für Geräte und Rezepte
- OB1 Main — Hauptprogramm mit Aufruf aller FBs in korrekter Reihenfolge
- XML Tag Tables — Tag-Tabellen im SimaticML-Format für den direkten Import in TIA Portal

Das reduziert die Inbetriebnahmezeit einer neuen Produktionslinie von Wochen auf Stunden: Der Ingenieur konfiguriert den Prozess in SCADA, drückt «Generieren», lädt das ZIP in TIA Portal — und die SPS ist

einsatzbereit.

#### 4. Visueller Konstruktor für mnemonische Schemata

Ein leistungsfähiger Editor auf Basis von Konva.js (HTML5 Canvas) ersetzt die Notwendigkeit separater SCADA-Grafikeditoren:

- Drag-and-Drop-Platzierung von Geräten, Rohrleitungen, Texten, Bildern
- Automatische Zustandsvisualisierung von Geräten (Farbe, Strömungsanimation)
- Unterstützung für Hintergrundbilder (P&ID-Unterlage)
- Tank-Konstruktor mit Wahl der Deckelform und automatischer Sensorplatzierung
- Mehrsprachige Beschriftungen und Übersetzungen
- Unbegrenztes Zoomen und Verschieben
- Mehrfachauswahl und Gruppenoperationen

#### 5. Flexibles Rezept- und Sequenzsystem

Das Rezeptsystem setzt den Standard ISA-88 (Batch Control) um und erlaubt die Trennung von:

- Sequenz (Sequence) — Algorithmus: Schrittreihenfolge und Übergangsbedingungen
- Schritt Rezept (Recipe) — konkrete Gerätewerte für jeden Schritt
- Produktionsrezept (Production Recipe) — Parametersatz für ein konkretes Produkt

Eine Sequenz lässt sich mit unterschiedlichen Rezepten für verschiedene Produkte auf derselben Ausrüstung wiederverwenden. Rezeptvariablen erlauben die Parametrisierung des Prozesses, ohne den Algorithmus zu verändern.

#### 6. Chargenverwaltung (Batch Management)

Vollwertiges System zur Verwaltung von Produktionschargen:

- Chargenerstellung mit Zuordnung zu Rezept und Produktionslinie
- Verfolgung der Charge durch die Linien-Stufen
- Chargen-Warteschlange mit Prioritäten
- Automatischer Transfer zwischen Stufen
- Historie jeder Charge

#### 7. KI-Assistent

Ein integrierter KI-Helfer auf LLM-Basis (über OpenRouter) beschleunigt die Arbeit mit dem System erheblich:

- Steuerung per natürlicher Sprache: «Starte Motor MOT\_001», «Welche Geräte sind aktuell in Störung?»
- Automatische Analyse von P&ID-Zeichnungen: Bild hochladen → Geräte und Verbindungen erkennen → Vorschlag zur Platzierung im Schema
- Kontextabhängige Hilfe und Erklärungen
- Agenten-Schleife mit Tools (bis zu 15 Iterationen) für komplexe Aufgaben

#### 8. Mehrsprachigkeit

Vollständige Unterstützung für 10+ Oberflächensprachen samt Übersetzung nicht nur der UI, sondern auch nutzergenerierter Inhalte:

- Oberfläche: EN, UK, RU, DE, FR, PL, AR (mit RTL) und weitere
- Namen von Geräten, Sequenzen, Schritten, Rezepten, Optionen — über das Feld `name_translations`
- Texte auf Schemata — über das Feld `translations`
- Namen der Schemata — über `name_translations`
- Sprachumschaltung im laufenden Betrieb ohne Reload

## 9. Skalierbares Rollenmodell für den Zugriff

Ein granulares Berechtigungssystem erlaubt die exakte Steuerung, wer was darf:

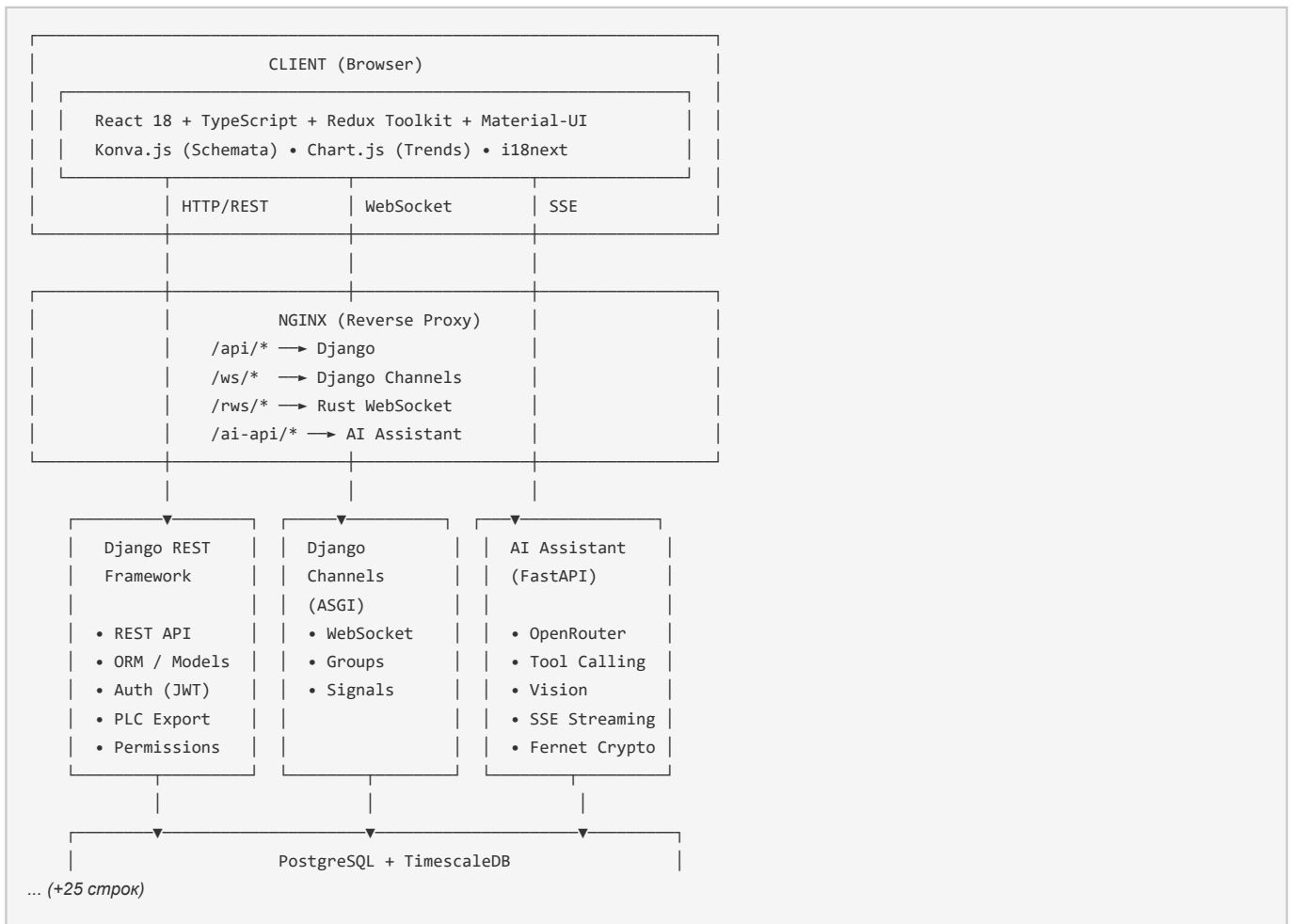
- Rollen mit Berechtigungsmatrix über 12 Kategorien und zahlreiche Aktionen
- Zugriffssteuerung pro Schema (per-schema access)
- Aktivitätsprotokoll aller Benutzeraktionen (Audit)
- IP-basierte Zugriffssteuerung

## 10. Industrielle Zuverlässigkeit

- Automatisches WebSocket-Reconnect mit exponentiellem Backoff
- Dual WebSocket: Rust für Geschwindigkeit + Django für Steuerung (Fallback)
- Vollständige Datenbank-Sicherung und -Wiederherstellung
- Projekt-Export/Import zur Migration zwischen Installationen
- Updatesystem mit Rollback und Dry-Run
- Smoke-Tests nach Updates

## Systemarchitektur

Das System basiert auf einer Microservice-Architektur mit klarer Trennung der Verantwortlichkeiten:



## Komponentenbeschreibung

Frontend (React + TypeScript)

Die Client-Anwendung läuft im Browser des Benutzers. Aufgebaut auf React 18 mit TypeScript für Typsicherheit, Redux Toolkit für das State Management und Material-UI für UI-Komponenten. Schemata werden über Konva.js (HTML5 Canvas) gezeichnet, Trends über Chart.js. Die Mehrsprachigkeit wird über i18next mit Unterstützung für 10+ Sprachen einschließlich RTL (Arabisch) umgesetzt. Die Kommunikation mit dem Backend erfolgt über REST API (axios), WebSocket (native API) und SSE (für KI-Streaming).

### NGINX (Reverse Proxy)

Einstiegspunkt für alle Anfragen. Routet den Verkehr zwischen Microservices:

- /api/\* → Django REST API
- /ws/\* → Django Channels (WebSocket)
- /rws/\* → Rust WebSocket Server (hochperformant, read-only)
- /ai-api/\* → AI Assistant (FastAPI)
- /media/\* → Statische Dateien (Icons, Bilder)

### Django REST Framework (Backend API)

Kern der Geschäftslogik:

- Alle CRUD-Operationen für über 40 Ressourcentypen
- Authentifizierung und Autorisierung (JWT + RBAC)
- Verwaltung von Benutzern, Rollen und Berechtigungen
- SPS-Codegenerierung (11 Funktionsbaustein-Typen, Zustandsautomat, Data Blocks, Tag Tables)
- Projekt-Export/Import
- Systemwartung (Updates, Backups, Diagnose)

### Django Channels (ASGI WebSocket)

Bidirektionale Echtzeit-Interaktion:

- Über 17 WebSocket-Kanaltypen (Geräte, Schemata, Sequenzen, Chargen, Störungen, Math-Funktionen, Aktivität, System)
- Redis Channel Layer für Group Broadcasting
- Verarbeitung von Steuerbefehlen (Start/Stop von Geräten, Chargensteuerung)
- Update-Signale bei Änderungen an Django-Modellen

### AI Assistant (FastAPI)

Autonomer Microservice für den KI-Helfer:

- FastAPI auf Python 3.11 (Port 8080)
- Anbindung an die OpenRouter API (LLM mit Tool-Calling)
- Agentenorchestrator-Schleife (bis zu 15 Iterationen) mit 4 Tool-Kategorien
- Analyse von P&ID-Bildern über ein Vision-Modell
- SSE-Streaming der Antworten (token, toolcall, toolresult, done)
- Verschlüsselung der API-Schlüssel (Fernet/AES)
- JWT-Authentifizierung (per Proxy über Django /api/users/me/)
- Eigene Tabellen in der gemeinsamen PostgreSQL-DB (Präfix ai\_assistant\_)
- Alembic-Migrationen

### PostgreSQL + TimescaleDB

Eine Datenbank für alle Dienste:

- Basistabellen (core\_\*) — Geräte, Schemata, Sequenzen, Rezepte, Rollen, Benutzer
- TimescaleDB-Hypertabellen — Langzeitdaten (Trends) mit automatischer Zeitpartitionierung und Kompression
- KI-Assistent-Tabellen (ai\_assistant\_\*) — Sessions, Nachrichten, Einstellungen

TimescaleDB liefert bei großen Datenmengen deutlich bessere Abfragegeschwindigkeit für Zeitreihen als PostgreSQL allein.

## Redis

### Kommunikationsbus:

- Channel Layer für Django Channels (WebSocket-Gruppen)
- Pub/Sub für Benachrichtigungen zwischen Django und Rust Worker
- Caching häufiger Abfragen

### Rust Worker (scada-worker)

#### Hochperformanter Echtzeit-Verarbeitungsservice in Rust:

- Sequence Engine (engine.rs) — Sequenzausführung: Verfolgung des aktuellen Schritts, Prüfung der Übergangsbedingungen, Aktivierung/Deaktivierung von Geräten, Arbeit mit Rezepten
- Events Processor — Verarbeitung von Sequenz-Ereignissen
- Accident Processor — Erkennung von Störungen, Anlegen von Störungsdatensätzen
- Tank Volume Calculator — periodische Volumenberechnung der Tanks (alle 1000 ms) aus Füllstand- und Drucksensordaten
- Math Function Calculator — Neuberechnung mathematischer Funktionen
- DB Batchter — Batched Writes nach PostgreSQL (I/O-Optimierung)
- WebSocket Batchter — Batched WebSocket-Nachrichten (Intervall 250 ms)
- Rust WebSocket Server (Port 8001) — dedizierter Server für read-only Streaming von Gerätezuständen. Deutlich schneller als Django Channels bei Massenaktualisierungen.

Die Wahl von Rust für diese Komponente ergibt sich aus den Anforderungen an minimale Latenz, vorhersehbaren Speicherverbrauch und die Fähigkeit, Tausende Aktualisierungen pro Sekunde ohne GC-Pausen zu verarbeiten.

## Vorteile gegenüber klassischen SCADA-Systemen

Kriterium	Klassisches SCADA	Dieses System
Client	Fat Client (Installation pro PC)	Webbrowser (keine Installation)
Betriebssystem	Nur Windows	Beliebig (Windows, Linux, macOS, Android, iOS)
Lizenzierung	Per-Seat (pro Arbeitsplatz)	Serverlizenz (unbegrenzte Clients)
Update	Manuell auf jedem PC	Automatisch, zentral (alle Clients gleichzeitig)
Mobiler Zugriff	Nicht vorhanden oder separates Modul	Responsive UI von Haus aus
SPS-Codegenerierung	Keine (manuelle Programmierung)	Automatisch aus SCADA-Konfiguration
KI-Helfer	Keiner	Integriert (P&ID-Analyse, Steuerung, Hilfe)
Mehrsprachigkeit	Begrenzt (UI)	Vollständig (UI + Inhalte + RTL)
Echtzeit	Proprietäre Protokolle	WebSocket + Rust (Sub-Sekunden-Latenz)
Rezepte	Basisunterstützung	ISA-88 (Sequence + Recipe + Production Recipe)
Batch Management	Separates Modul (Zusatzlizenz)	Eingebaut
Trendanalyse	Eingebaut, begrenzt	TimescaleDB (Petabytes, Autokompression)
Audit	Grundlegend	Vollständiges Protokoll mit WebSocket-Updates
Backup/Restore	Externe Tools	Eingebaut (Backup, Export, Import, Rollback)
Bereitstellungskosten	Hoch (Lizenzen + Installation + Konfiguration)	Niedrig (Docker, automatisierte Bereitstellung)
Erweiterbarkeit	Geschlossenes Ökosystem	Offene REST API + WebSocket + SSE

## Systemanforderungen

## Serverseite

Komponente	Minimum	Empfohlen
Prozessor	4 Kerne	8+ Kerne
Arbeitsspeicher	8 GB	16+ GB
Festplattenspeicher	20 GB SSD	100+ GB SSD (für Trends)
Betriebssystem	Linux (Ubuntu 22.04+) oder Windows Server 2019+	Linux (Ubuntu 22.04 LTS)
Docker	Docker Engine 24+	Docker Engine 24+ mit Docker Compose v2
PostgreSQL	15+ mit TimescaleDB	16+ mit TimescaleDB 2.x
Redis	7+	7+

## Clientseite (Browser)

Komponente	Anforderung
Browser	Chrome 90+, Firefox 90+, Edge 90+, Safari 15+
Bildschirmauflösung	Minimum 1280×720, empfohlen 1920×1080+
Netzwerk	Stabile Verbindung zum Server (LAN/VPN)

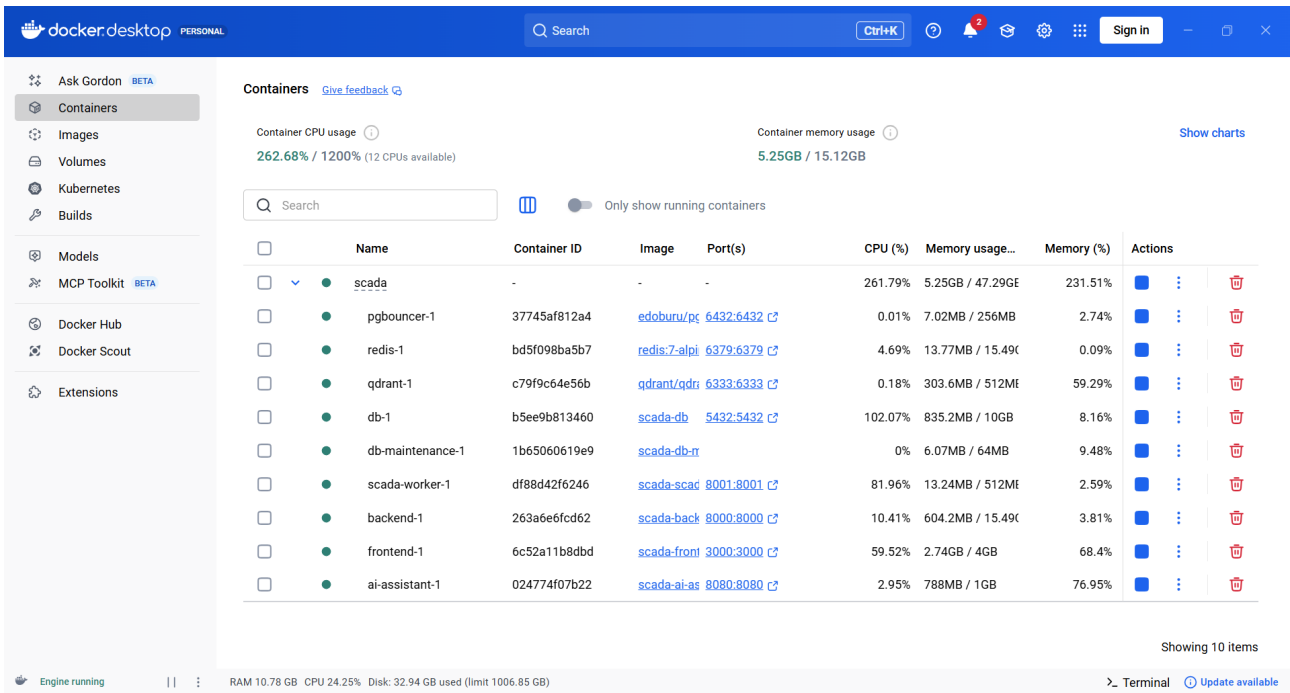
## Bereitstellung

Das System wird als Docker-Container-Bundle ausgeliefert und mit einem einzigen Befehl ausgerollt:

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

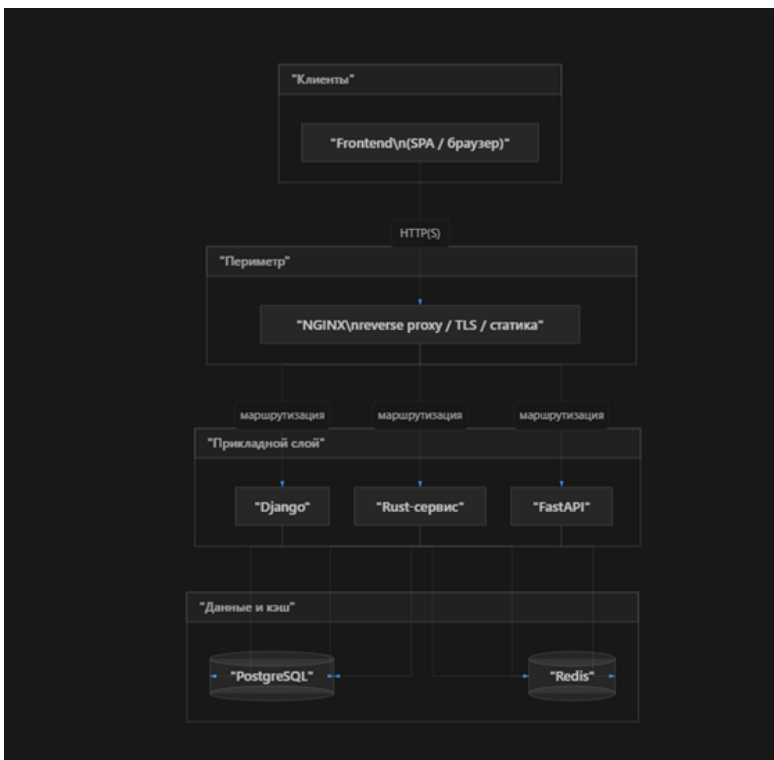
Container-Zusammensetzung:

- backend — Django REST API + Django Channels
- scada-worker — Rust Worker
- ai-assistant — FastAPI KI-Service
- postgres — PostgreSQL + TimescaleDB
- redis — Redis
- nginx — Reverse Proxy



Docker Containers

Terminal mit Ausgabe von `docker-compose ps`: alle 6 Container im Status «Up», Ports, Laufzeit.



Architecture Diagram

Architekturdiagramm des Systems: Komponenten-Blöcke mit Verbindungen (REST, WebSocket, Redis Pub/Sub, SQL). Farbcodierung: blau — Frontend, grün — Django, orange — Rust, violett — KI, grau — Infrastruktur.

## Schnellstart — 15 Minuten

Dieser Abschnitt führt Sie von einer frischen Installation zu einem funktionierenden mnemonischen Schema mit einem Motor und einem Ventil. Wenn Sie SCADA zum ersten Mal starten, beginnen Sie hier.

### Schritt 1. Erste Anmeldung (2 Min)

- Öffnen Sie die Installationsadresse im Browser (z. B. <http://localhost> oder <http://scada.local>).
- Geben Sie auf dem Anmeldebildschirm die vom Installer erstellten Administratoranmeldedaten ein (Standardmäßig admin und das bei der Installation gesetzte Passwort).
- Beim ersten Anmelden kann das System die Lizenzaktivierung anfordern — geben Sie den Schlüssel ein und drücken Sie Aktivieren. Ohne aktive Lizenz sind die meisten Funktionen eingeschränkt.
- Nach erfolgreicher Anmeldung gelangen Sie zum Dashboard.

Tipp. Wählen Sie die Oberflächensprache unten links im Anmeldeformular — sie wird pro Benutzer gespeichert.

### Schritt 2. Erste SPS erstellen (2 Min)

Auch wenn Sie zunächst nur in der Simulation arbeiten, benötigt das System mindestens eine SPS, an die Geräte gebunden werden.

- Öffnen Sie in der Seitenleiste PLCs.
- Drücken Sie «+» oben rechts.
- Füllen Sie aus:
- Name: PLC\_DEMO
- IP-Adresse: 192.168.0.100 (kann später geändert werden)
- Typ: Siemens S7-1200 (oder eine andere verfügbare Option)
- Rack / Slot: 0 / 1
- Drücken Sie Speichern. Der SPS-Status ist «Offline» — das ist für eine Demo normal.

### Schritt 3. Erste Geräte anlegen (3 Min)

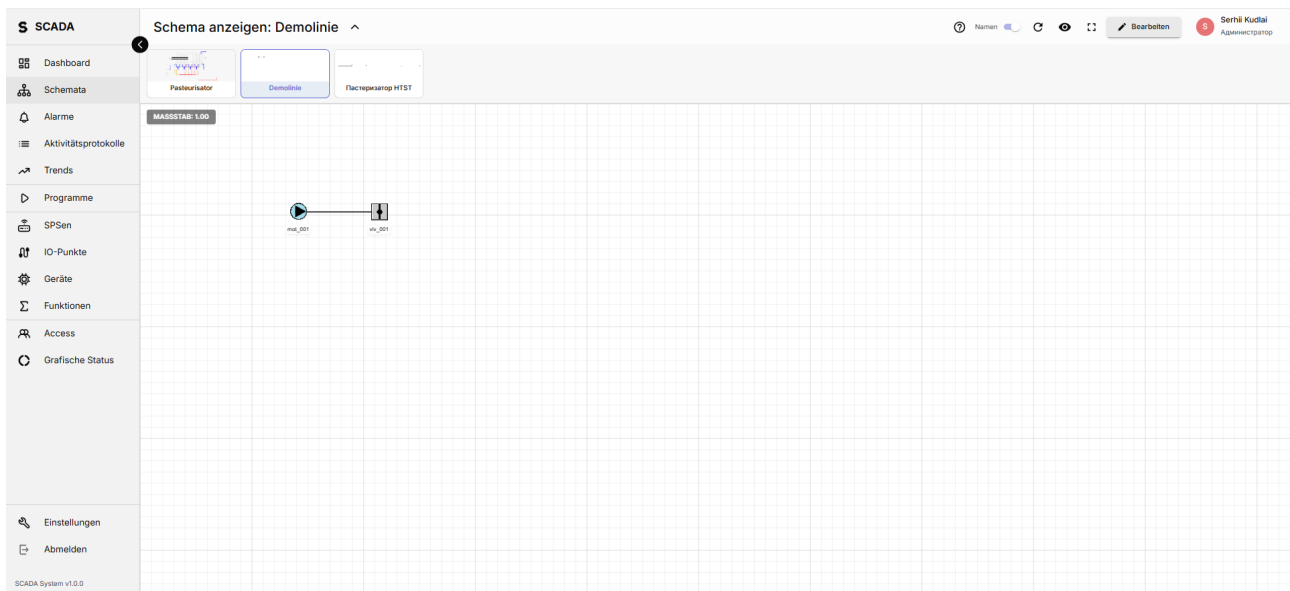
Fügen wir einen Motor und ein Ventil hinzu.

- Gehen Sie zu Devices → «+» → Motor.
- Füllen Sie aus:
- ID: mot\_001
- Name: Zulaufpumpe
- PLC: PLC\_DEMO
- Beschreibung: «Demomotor»
- Drücken Sie Speichern.
- Erstellen Sie analog ein Ventil («+» → Valve):
- ID: val\_001
- Name: Leitungsventil
- PLC: PLC\_DEMO

Wichtig. Die Geräte-ID ist ein Textbezeichner (mot\_001, keine Zahl). Sie wird in allen WebSocket-Kanälen, beim PLC-Export und in Übergangsbedingungen verwendet. Legen Sie von Anfang an ein einheitliches Namensschema fest.

## Schritt 4. Erstes mnemonisches Schema (3 Min)

- Öffnen Sie Schemas → «+».
- Tragen Sie Name: «Demo-Linie» ein und drücken Sie Erstellen.
- Klicken Sie in der Schemaliste auf das Stift-Symbol (Edit) — der Schema Builder öffnet sich.
- Drücken Sie in der linken Leiste Gerät (); wählen Sie im Dialog mot\_001 → Hinzufügen. Der Motor erscheint auf der Leinwand — verschieben Sie ihn mit der Maus.
- Erneut Gerät → val\_001 → Hinzufügen — platzieren Sie das Ventil daneben.
- Wählen Sie Rohrleitung (), klicken Sie auf den Motor → danach auf das Ventil — eine Verbindungslinie erscheint.
- Drücken Sie Strg+S oder den Speichern-Knopf oben.



Quickstart First Schema

Einfaches Demo-Schema: ein Motor (grüner Kreis) verbunden über eine Rohrleitung mit einem Ventil (gelbes Dreieck), im Hintergrund das Gitter der Leinwand.

## Schritt 5. Viewer starten und bedienen (3 Min)

- Drücken Sie oben Anzeigen — Sie wechseln in den Schema Viewer.
- Klicken Sie das Motor-Symbol an — ein Bedien-Popup öffnet sich. Stellen Sie den Modus auf Manuell und drücken Sie Start. Die Anzeige wird grün — der Motor «läuft».
- Klicken Sie das Ventil → Öffnen. Wenn am Rohr Prozessüberwachung aktiviert ist, erscheint eine Strömungsanimation.
- Drücken Sie Stop am Motor — die Animation hält an.

Glückwunsch — Sie haben Ihr erstes funktionierendes mnemonisches Schema mit Live-Bedienung.

## Schritt 6. Was als Nächstes (2 Min)

Sie haben nun ein Grundgerüst. Die nächsten Schritte hängen von Ihrem Ziel ab:

Ziel	Wo weiter
Prozess automatisieren (Start/Stopp über Bedingungen)	Abschnitt 8 «Sequenzen» und Beispiel «Milchpasteurisierer»
Rezepte für verschiedene Produkte konfigurieren	Abschnitt 9 «Rezepte und Variablen»
Reale SPS-E/A anbinden	Abschnitte 18 «I/O Points» und 19 «E/A-Provisionierung»
Alarmschwellen konfigurieren	Abschnitt 10 «Ereignisse und Störungen»
TIA-Portal-Projekt erzeugen	Abschnitt 27 «SPS-Projekt-Export»
Trends analysieren	Abschnitt 15 «Trends»

Tipp. Jede Seite hat oben einen «?»-Button — er startet eine interaktive Tour mit hervorgehobenen Elementen.

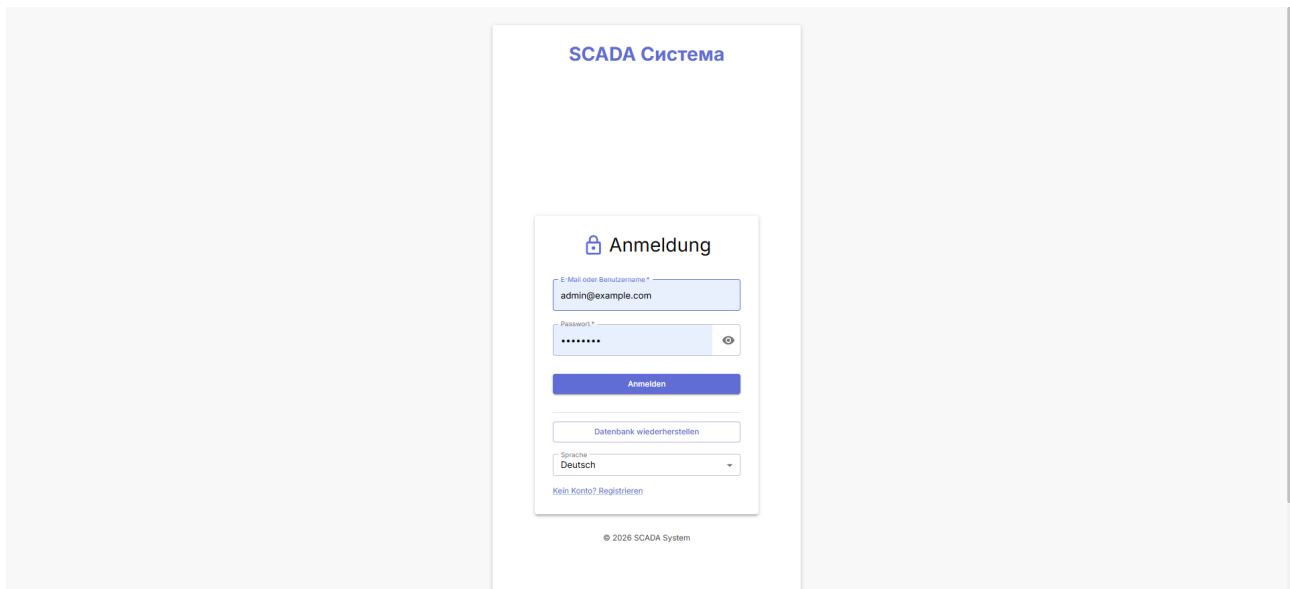
# 1. Anmeldung und Registrierung

## Für Benutzer

Der Anmeldebildschirm ist das Erste, was Sie beim Öffnen des Systems sehen.

Bildschirmelemente:

Element	Beschreibung
Feld «Benutzername / E-Mail»	Geben Sie Ihren Login oder die E-Mail ein
Feld «Passwort»	Passwort eingeben. Das Auge-Symbol rechts schaltet die Sichtbarkeit um
Sprachauswahl	Dropdown zur Wahl der Oberflächensprache (wird nach Anmeldung gespeichert)
Schaltfläche «Anmelden»	Führt die Anmeldung aus
Link «Passwort vergessen?»	Wechsel zur Passwort-Wiederherstellung
Link «Registrieren»	Wechsel zum Registrierungsformular



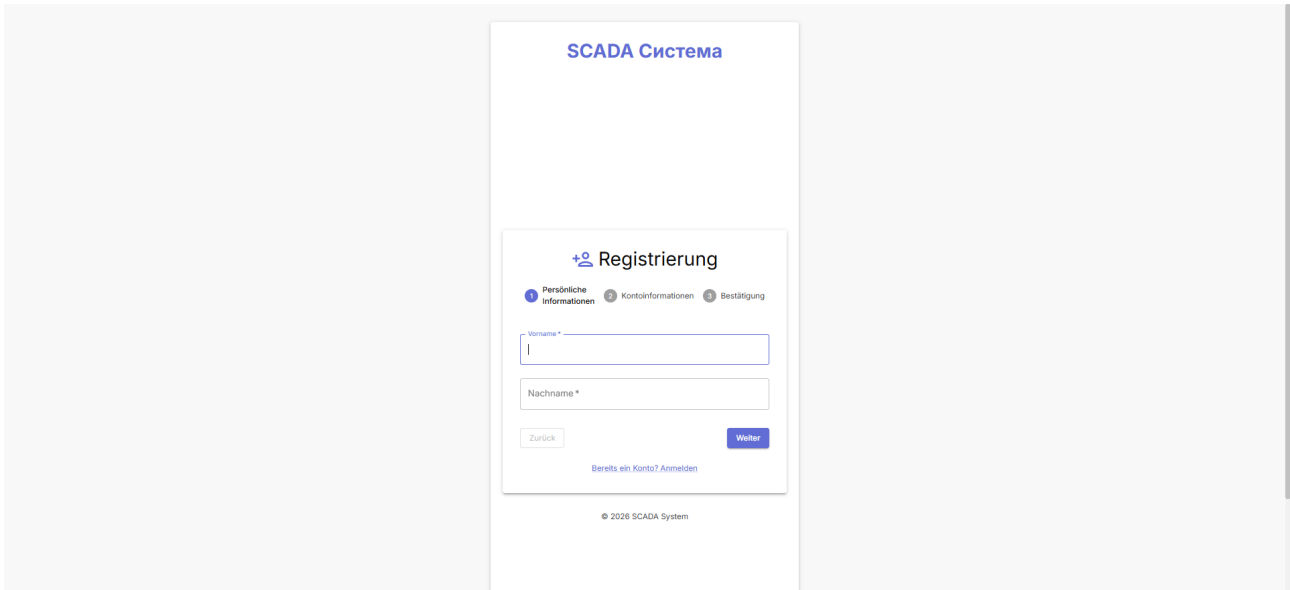
Login Page

Anmeldeseite: Felder «Username/E-Mail» und «Passwort» (mit Augen-Symbol), Sprachauswahl, «Anmelden», Links «Passwort vergessen?» und «Registrieren». Dunkles/helles Design je nach Einstellung.

Registrierungsbildschirm:

Element	Beschreibung
Feld «Benutzername»	Eindeutiger Login-Name
Feld «E-Mail»	E-Mail-Adresse
Feld «Passwort»	Passwort mit Stärkeanzeige
Feld «Passwort bestätigen»	Wiederholte Eingabe
Feld «Vollständiger Name»	Vor- und Nachname

Element	Beschreibung
Schaltfläche «Registrieren»	Legt das Konto an



Register Page

Registrierungsformular: Benutzername, E-Mail, Passwort (mit Stärkeanzeige), Passwort bestätigen, Vollständiger Name, Schaltfläche «Registrieren».

## Technische Dokumentation

### Authentifizierungs-API:

Methode	Endpoint	Beschreibung	Request-Body
POST	/api/auth/login	Anmeldung	{"username": "...", "password": "..."}
POST	/api/auth/register	Registrierung	{"username": "...", "email": "...", "password": "...", "first_name": "...", "last_name": "..."}
GET	/api/auth/me	Aktueller Benutzer	—
POST	/api/auth/token/refresh/	JWT-Token erneuern	{"refresh": "..."}

### Antwort beim Login:

```

{
  "access": "eyJhbGciOi...",
  "refresh": "eyJhbGciOi...",
  "user": {
    "id": 1,
    "username": "admin",
    "email": "admin@example.com",
    "roles": [...]
  }
}
    
```

Autorisierung: Alle geschützten Endpoints benötigen den Header Authorization: Bearer .

## 2. Dashboard

### Für Benutzer

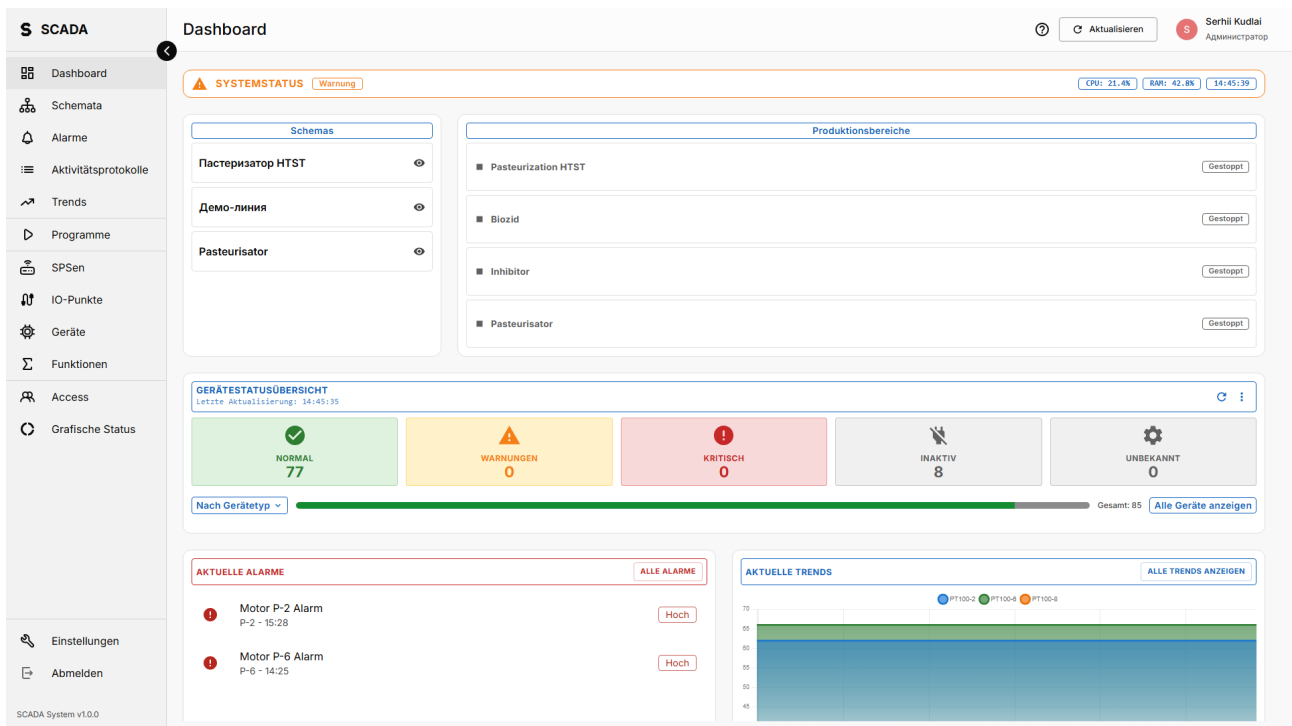
Das Dashboard ist die zentrale Seite des Systems und zeigt den Gesamtzustand.

Bildschirmelemente:

Element	Beschreibung
Statistikkarten	Kurzübersicht: Anzahl Geräte, aktive Schemata, Störungen, Sequenzen
Zustandsübersicht	Visuelle Zustandsanzeigen für Geräte und System
Störungsliste	Neueste Alarme mit farbcodierter Schwere
Mini-Trends	Miniaturlinien der wichtigsten Parameter
Sequenzstatus	Aktueller Schritt und Fortschritt jeder laufenden Sequenz
Systemmetriken	CPU, RAM, Uptime, Temperatur, Festplattenspeicher
Schnellzugriff-Schaltflächen	Direkter Sprung zu den Hauptfunktionen

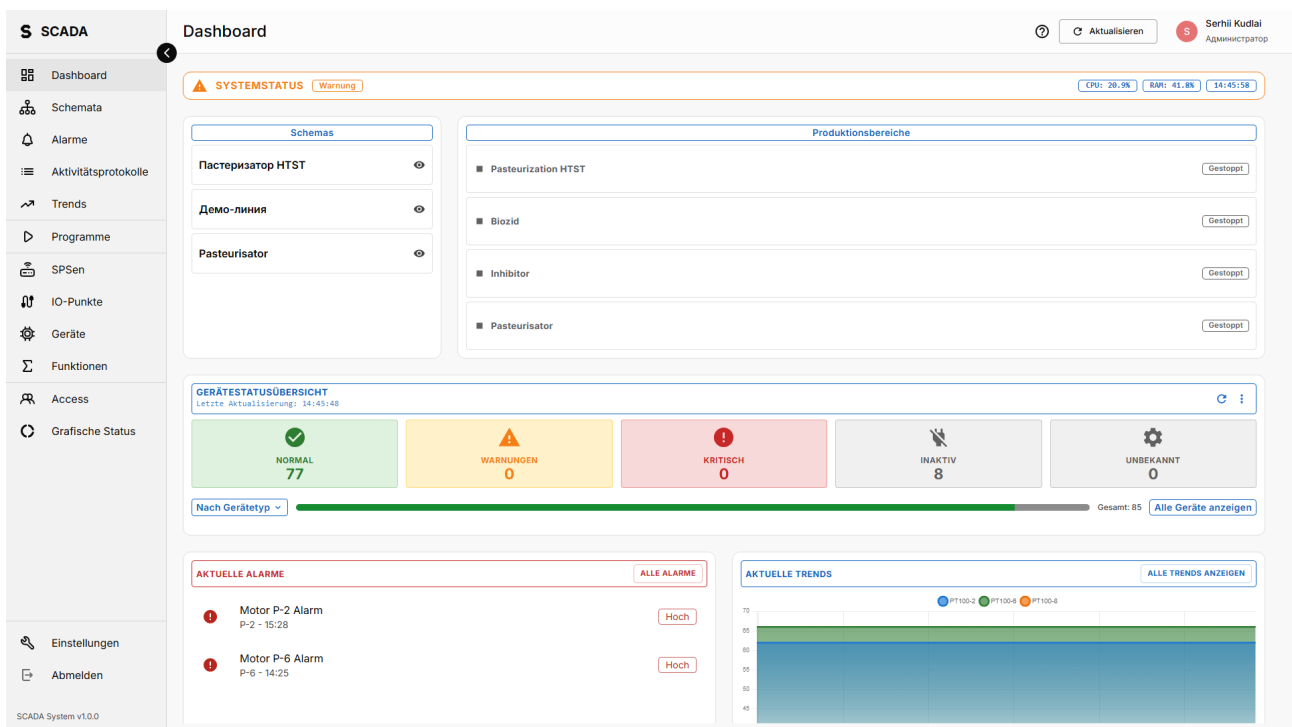
Schaltflächen in der Kopfleiste:

Schaltfläche	Beschreibung
Aktualisieren	Dashboard-Daten aktualisieren
Tour	Startet die interaktive Tour über die Dashboard-Elemente
Projekt exportieren	Öffnet den Exportdialog des gesamten Projekts
Projekt importieren	Öffnet den Importdialog aus Datei



Dashboard Overview

Vollständiges Dashboard: Statistikkarten oben, Störungsblock links, Mini-Trends rechts, Sequenzstatus unten. Navigations-Seitenleiste links.



Dashboard System Metrics

Block der Systemmetriken: CPU usage (%), RAM usage (%), Disk usage, Uptime, Network stats. Jeder Parameter mit farbcodierter Anzeige (grün/gelb/rot).

## Technische Dokumentation

API für Systemmetriken:

Methode	Endpoint	Beschreibung
GET	/api/system/info/	Systeminformationen
GET	/api/system/health-status/	Systemzustand

WebSocket: Systemmetriken (Streaming alle 2 s)

Verbindung: ws:///ws/system

```
{
  "type": "system_metrics",
  "cpu": 45.2,
  "memory": 67.8,
  "disk": 34.1,
  "network": {"rx": 1024, "tx": 512},
  "uptime": 86400,
  "temperature": 52.0,
  "lastUpdate": "2026-04-07T12:00:00Z"
}
```

WebSocket: SPS-Status (Streaming alle 2 s)

```
{
  "type": "plc_status",
  "is_active": true,
  "status": "running",
  "service_running": true,
  "timestamp": "2026-04-07T12:00:00Z"
}
```

## 3. Mnemonikschemata

### Für Benutzer

Mnemonikschemata sind visuelle Diagramme Ihres Produktionsprozesses. Sie sind das zentrale Werkzeug für Überwachung und Bedienung.

Bildschirmelemente der Liste:

Element	Beschreibung
Schema-Liste / Raster	Karten oder Tabellenzeilen mit Schemata
Ansichtsumschalter	Raster / Liste (Icons oben rechts)
Schaltfläche «+»	Neues Schema erstellen
Suche	Suchfeld nach Name
Aktualisieren	Liste neu laden

Aktionen pro Schema:

Schaltfläche	Symbol	Beschreibung
Anzeigen		Öffnet das Schema im SchemaViewer
Bearbeiten		Öffnet das Schema im SchemaBuilder
Löschen		Löscht das Schema (mit Bestätigung)
Zugriffsverwaltung		Konfiguriert den Zugriff von Benutzern
Umbenennen		Ändert Namen und Übersetzungen
Einstellungen		Konfiguriert Schema-Parameter (Beschreibung, Übersetzungen)

ID	Name	Beschreibung	Erstellt	Geräte	Aktionen
1	Pasteurisateur	-	11.11.2025	59	[Anzeigen] [Bearbeiten] [Löschen]
3	Демо-линия	-	8.5.2026	2	[Anzeigen] [Bearbeiten] [Löschen]
4	Пастеризатор HTST	-	8.5.2026	10	[Anzeigen] [Bearbeiten] [Löschen]

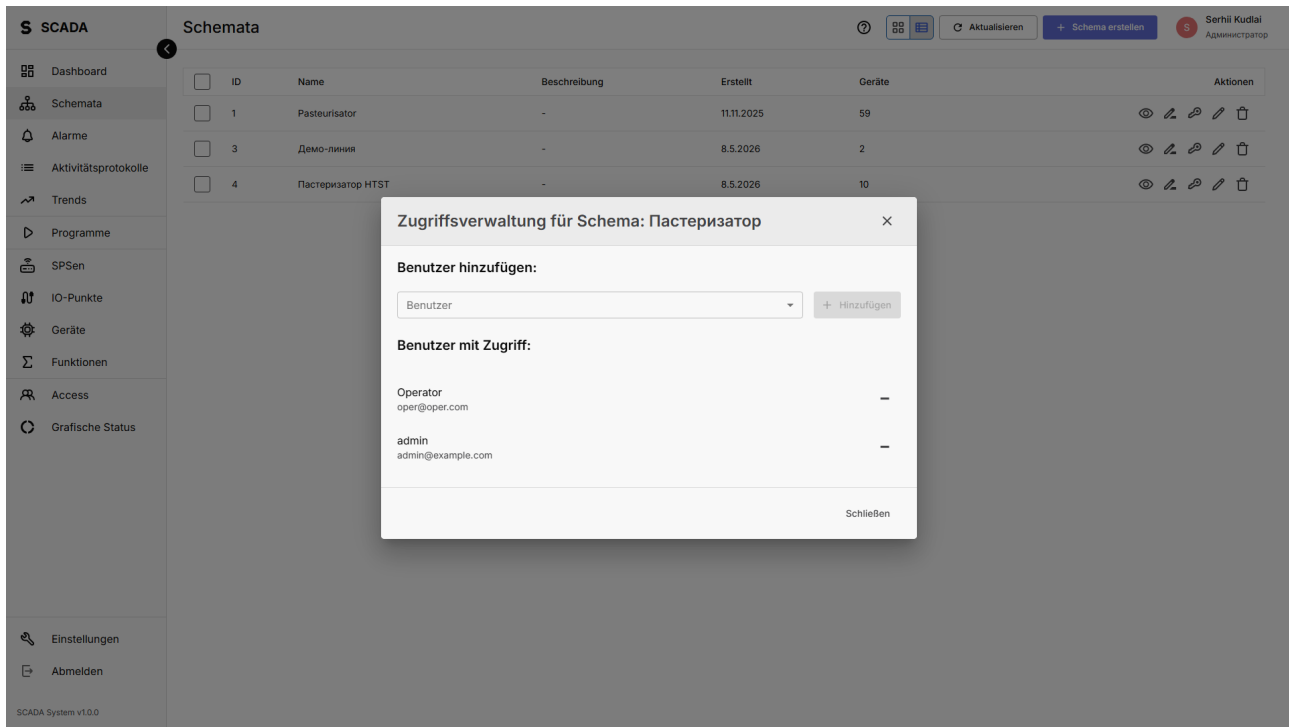
Schemas List Grid

Schema-Listenseite als Kartenraster. Jede Karte zeigt Vorschau, Name, Änderungsdatum und Aktionsschaltflächen (Anzeigen, Bearbeiten, Löschen). Schaltfläche «+» oben rechts.

Name	Beschreibung	Erstellt am	Zuletzt geändert	Aktionen
Pasteurisateur	Keine Beschreibung	11.11.2025	0	[Anzeigen] [Bearbeiten] [Löschen]
Демо-линия	Keine Beschreibung	08.05.2026	2	[Anzeigen] [Bearbeiten] [Löschen]
Пастеризатор HTST	Keine Beschreibung	08.05.2026	0	[Anzeigen] [Bearbeiten] [Löschen]

Schemas List Table

Dieselbe Seite in Tabellenansicht: Spalten «Name», «Beschreibung», «Erstellt am», «Zuletzt geändert», «Aktionen».



Schema Access Dialog

Dialog zur Zugriffsverwaltung: Benutzerliste mit Checkboxes, Schaltflächen «Hinzufügen» und «Speichern».

## Technische Dokumentation

### Schemata-API:

Methode	Endpoint	Beschreibung	Parameter
GET	/api/schemas/	Liste aller Schemata	?search=...&ordering=...
POST	/api/schemas/	Schema erstellen	{"name": "...", "description": "...", "name_translations": {...}}
GET	/api/schemas/{id}/	Schema abrufen	—
PUT	/api/schemas/{id}/	Schema aktualisieren	Vollständiges Schema-Objekt
PATCH	/api/schemas/{id}/	Teilaktualisierung	{"name": "...", "description": "...", "name_translations": {...}}
DELETE	/api/schemas/{id}/	Schema löschen	—
GET	/api/schemas/{id}/user_access/	Liste berechtigter Benutzer	—
POST	/api/schemas/{id}/user_access/	Zugriff hinzufügen	{"user_id": 1}
DELETE	/api/schemas/{id}/user_access/	Zugriff entziehen	{"user_id": 1}

## 4. Schema Builder

### Für Benutzer

Der Schema-Editor ist ein leistungsfähiges visuelles Werkzeug zum Erstellen und Bearbeiten technologischer Diagramme. Er arbeitet auf einer Leinwand (Canvas) mit Drag-and-Drop, Zoom und Rotation von Elementen.

#### Linke Werkzeuggestreife

Schaltfläche	Symbol	Beschreibung
Auswahl		Modus zum Auswählen und Verschieben von Elementen
Gerät		Fügt ein Gerät aus dem Katalog auf das Schema ein
Rohrleitung/Signal		Zeichnet eine Rohrleitung (Verbindung zwischen Geräten)
Text	T	Fügt Textanmerkungen zum Schema hinzu
Bild		Fügt ein Hintergrundbild hinzu (P&ID-Unterlage)
Sequenz		Bindet eine Sequenz an das Schema
Option		Fügt einen Options-Button einer Sequenz hinzu
Rezeptvariable		Fügt eine Anzeige einer Rezeptvariablen hinzu
Produktionslinie		Fügt eine Produktionslinie hinzu
Math-Funktion	$f$	Fügt eine mathematische Funktion hinzu
Tank		Öffnet den Tank-Konstruktor (separate Schaltfläche)
Gerät erstellen		Schnelle Erstellung eines neuen Geräts

#### Obere Werkzeuggestreife

Schaltfläche	Beschreibung
Speichern	Speichert den aktuellen Zustand des Schemas
Rückgängig (Strg+Z)	Letzte Aktion widerrufen
Wiederholen (Strg+Y)	Widerrufene Aktion wiederherstellen
Anzeigen	Wechsel in den Schema Viewer
Umschalter «Namen»	Gerätenamen auf dem Schema ein-/ausblenden
Schema-Name	Anzeige des aktuellen Schema-Namens

#### Leinwand-Steuerung

Schaltfläche	Beschreibung
Vergrößern (+)	Zoom vergrößern
Verkleinern (-)	Zoom verkleinern
Gitter	Gitter auf der Leinwand ein/aus
Hintergrundfarbe	Hintergrundfarbe des Schemas ändern
Hintergrundtransparenz	Schieberegler für die Transparenz
Einklappen	Linke Leiste einklappen/ausklappen

## Rechte Eigenschafts-Leiste

Die Leiste erscheint beim Auswählen eines Elements und hängt vom Elementtyp ab:

Für Gerät (DevicePropertyPanel):

Eigenschaft	Beschreibung
Position X, Y	Koordinaten auf der Leinwand
Drehwinkel	Rotation des Elements (0-360°)
Größe W, H	Breite und Höhe
Farbe	Farbe des Elements (Palette)
Spiegelung	Horizontal/vertikal spiegeln
Namen anzeigen	Geräte-Beschriftung ein/aus
Grafischer Zustand	Bindung an ein Zustandsschablone
Konfiguration	Gerätespezifische Parameter

Für Rohrleitung (PipePropertyPanel):

Eigenschaft	Beschreibung
Name	Name der Rohrleitung
Farbe	Linienfarbe
Dicke	Linienstärke (px)
Namensanzeige	Name ein-/ausblenden
Prozessüberwachung	Strömungsanimation aktivieren

Für Text (TextPropertyPanel):

Eigenschaft	Beschreibung
Text	Inhalt
Schriftgröße	Textgröße
Farbe	Textfarbe
Fett / Kursiv	Formatierung
Übersetzungen	Mehrsprachige Textübersetzungen

Für Sequenz (SequencePropertyPanel):

Eigenschaft	Beschreibung
Position	Koordinaten
Größe	Breite × Höhe
Darstellung	Anzeigeoptionen (Name, Status, Schritte)

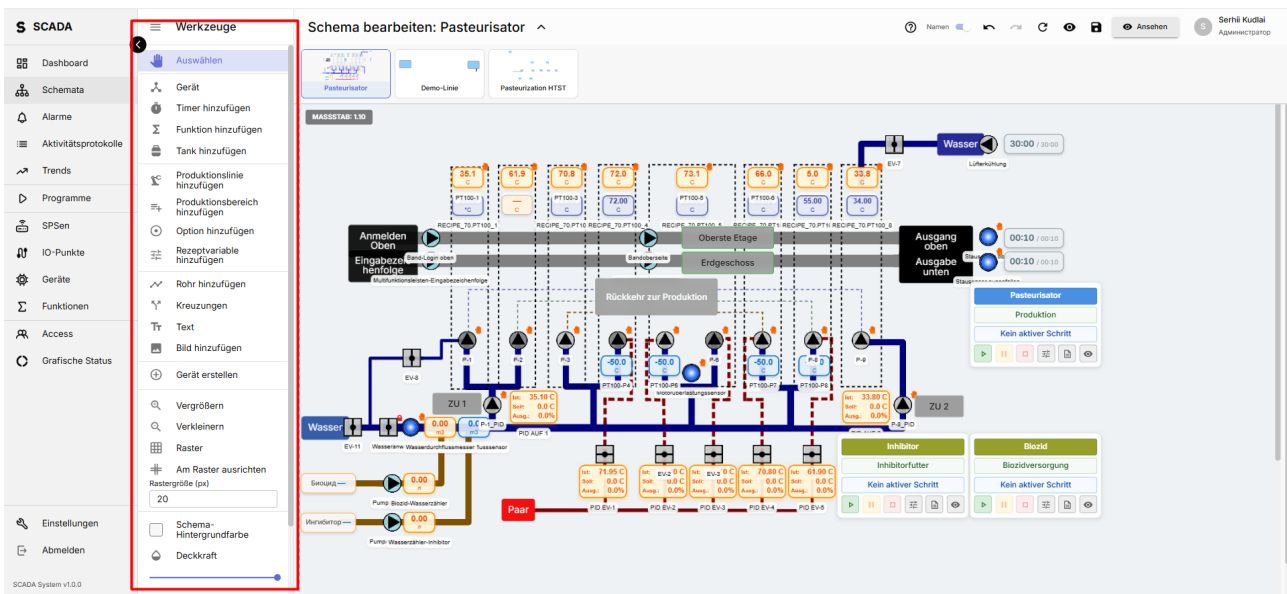
## Dialoge des Editors

Dialog	Beschreibung
DeviceDialog	Auswahl eines Geräts aus dem Katalog für die Platzierung. Filter nach Typ.
PipeDialog	Name, Typ und Sichtbarkeit einer Rohrleitung
AddSequenceDialog	

g

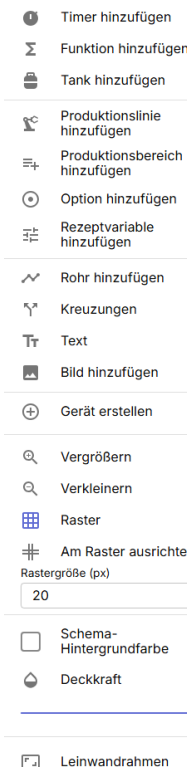
Auswahl einer Sequenz zur Bindung an das Schema

Dialog	Beschreibung
AddOptionDialog	Auswahl einer Sequenz-Option
AddProductionLineDialog	Auswahl einer Produktionslinie
AddRecipeVariableDialog	Auswahl einer Rezeptvariablen
MathFunctionDialog	Auswahl einer mathematischen Funktion
TankConstructorDialog	Tank-Konstruktor: Geräteauswahl, Deckeltypen (FLAT/DOME/DISH/CONE), Maße, SVG-Vorschau, automatische Sensorplatzierung
ImageDialog	Hochladen eines Hintergrundbilds
TextDialog	Text hinzufügen/bearbeiten



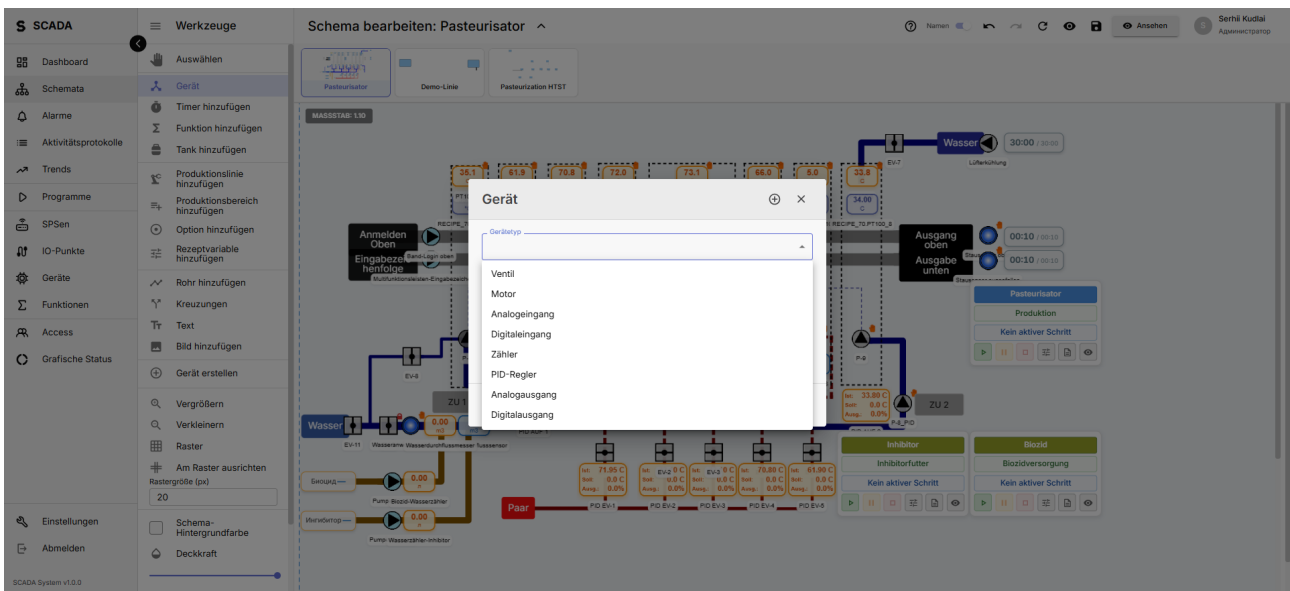
Schema Builder Full

Vollständiger Editor: linke Werkzeugleiste mit Symbolen, obere Leiste mit «Speichern», «Rückgängig», «Wiederholen», zentrale Leinwand mit dem Schema (Geräte, Rohrleitungen, Texte), rechte Eigenschaftsleiste des gewählten Elements.



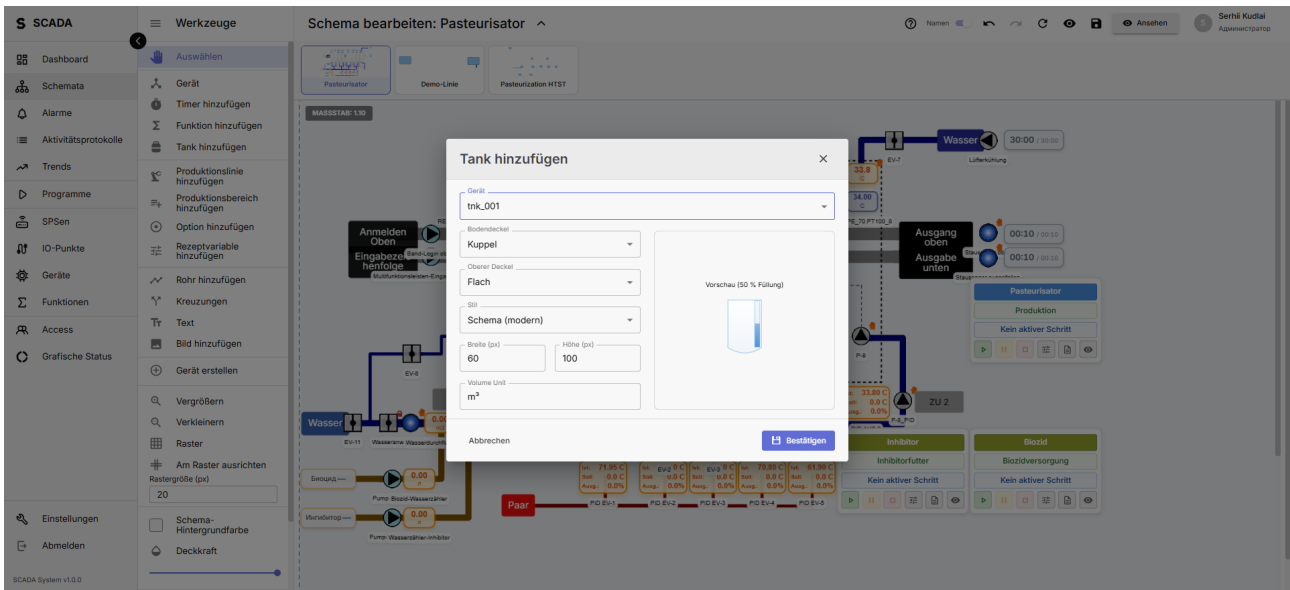
Schema Builder Toolbar Left

Nahaufnahme der linken Werkzeugleiste: jede Schaltfläche beschriftet (Auswahl, Gerät, Rohrleitung, Text, Bild, Sequenz, Option, Rezeptvariable, Produktionslinie, Math-Funktion, Tank).



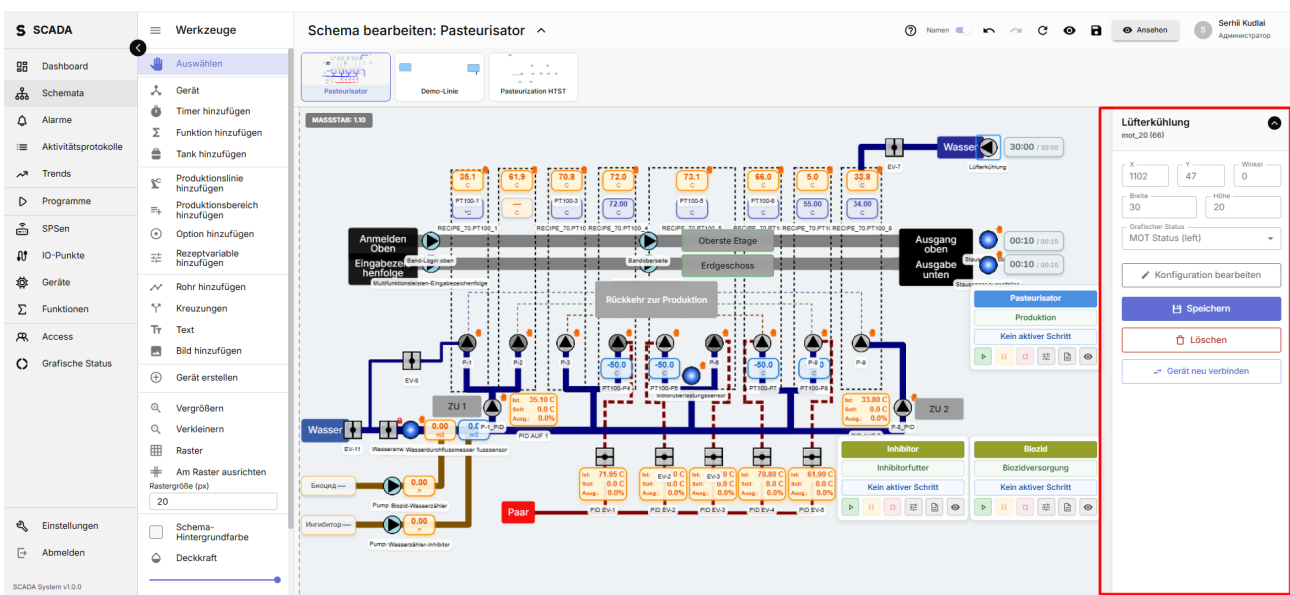
Schema Builder Device Dialog

Dialog Geräteauswahl: Geräteliste mit Typfilter (Motor, Valve, AI, AO, DI, DO, PID, Counter, Timer, Tank, COS), Suchfeld, «Hinzufügen»-Button.



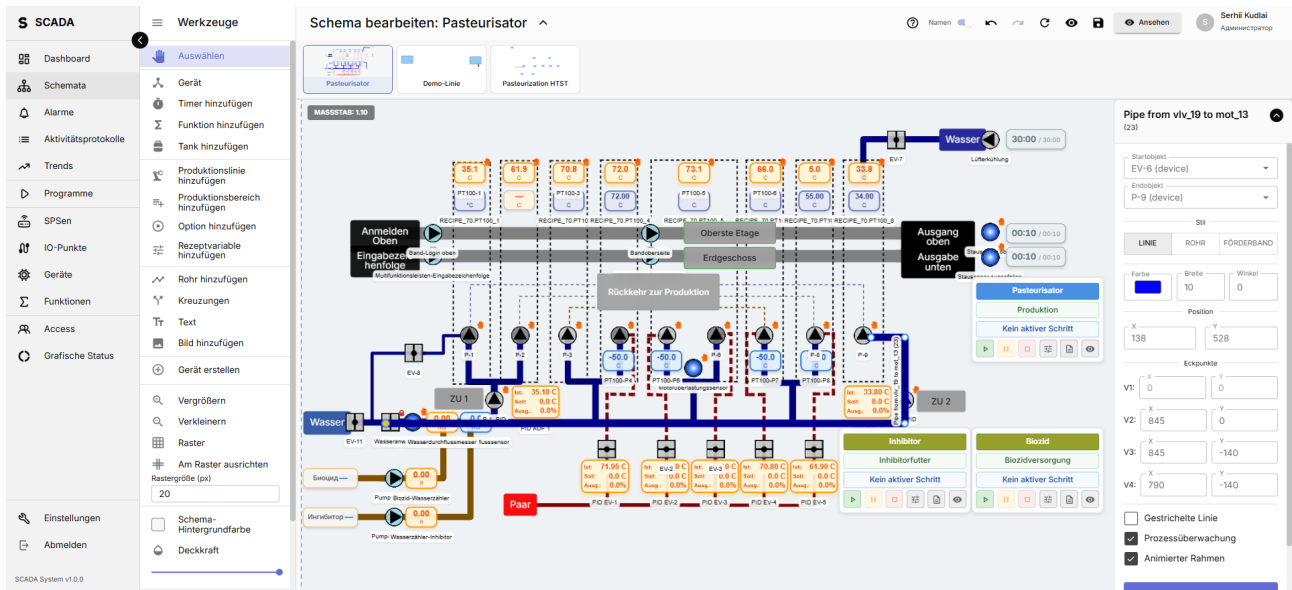
Schema Builder Tank Constructor

Tank-Konstruktor: Dropdowns für oberen und unteren Deckeltyp (FLAT, DOME, DISH, CONE), Breite/Höhe-Felder, SVG-Vorschau der Tankform, Tankauswahl.



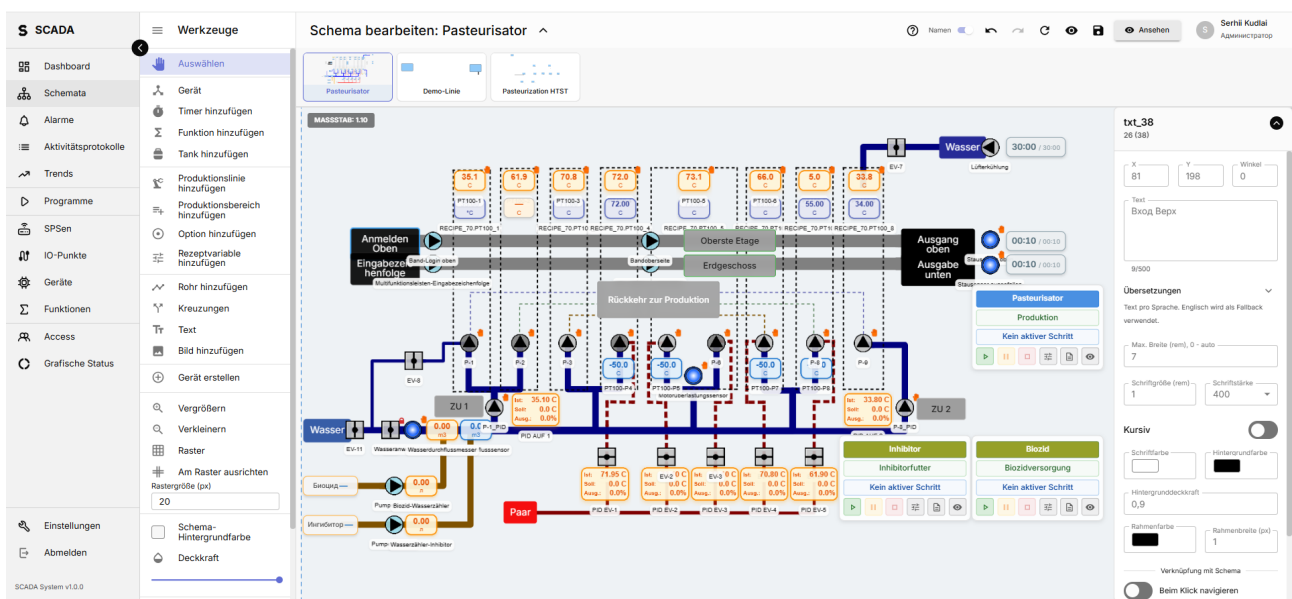
Schema Builder Properties Device

Rechte Eigenschaftsleiste bei ausgewähltem Gerät: Felder Position X/Y, Angle, Width/Height, Farwähler, Mirror/Show Name-Umschalter.



Schema Builder Properties Pipe

Rechte Eigenschaftsleiste bei ausgewähltem Rohr: Feld Name, Farbwähler, Thickness-Schieberegler, Show Name-Umschalter, Process Monitoring-Umschalter.



Schema Builder Properties Text

Rechte Eigenschaftsleiste bei ausgewähltem Text: Feld Text, Font Size, Color, Bold/Italic-Umschalter, Block Translations mit Feldern pro Sprache.

## Technische Dokumentation

API für Schema-Elemente:

Methode	Endpoint	Beschreibung
POST	/api/schemas/{id}/add_motor/	Motor hinzufügen

Methode	Endpoint	Beschreibung
POST	/api/schemas/{id}/add_valve/	Ventil hinzufügen
POST	/api/schemas/{id}/add_analog_input/	AI hinzufügen
POST	/api/schemas/{id}/add_analog_output/	AO hinzufügen
POST	/api/schemas/{id}/add_digital_input/	DI hinzufügen
POST	/api/schemas/{id}/add_digital_output/	DO hinzufügen
POST	/api/schemas/{id}/add_pid_controller/	PID hinzufügen
POST	/api/schemas/{id}/add_counter/	Zähler hinzufügen
POST	/api/schemas/{id}/add_variable_speed_drive/	FU hinzufügen
POST	/api/schemas/{id}/add_sequence/	Sequenz hinzufügen
POST	/api/schemas/{id}/add_option/	Option hinzufügen
POST	/api/schemas/{id}/add_production_line/	Linie hinzufügen
POST	/api/schemas/{id}/add_recipe_variable/	Variable hinzufügen
POST	/api/schemas/{id}/add-math-function/	Math-Funktion hinzufügen
POST	/api/schemas/{id}/bulk_add_schema_devices/	Massenzufügen
PATCH	/api/schemas/{id}/update_positions/	Positionen massenhaft aktualisieren
PATCH	/api/schemas/{id}/devices/{deviceId}/	Element aktualisieren
DELET	/api/schemas/{id}/remove_device/	Element entfernen

E

Request-Body zum Hinzufügen eines Geräts:

```
{
  "device_id": "mot_001",
  "content_type": "motor",
  "position_x": 100,
  "position_y": 200,
  "angle": 0,
  "configuration": {
    "color": "#FF0000",
    "mirror_x": false,
    "mirror_y": false,
    "show_name": true,
    "tank_visual": {
      "bottom_type": "DOME",
      "top_type": "FLAT",
      "width": 60,
      "height": 100
    }
  }
}
```

Rohrleitungs-API:

Methode	Endpoint	Beschreibung
GET	/api/pipes/?schema_id={id}	Alle Rohre eines Schemas
POST	/api/pipes/	Rohr erstellen
PATCH	/api/pipes/{id}/	Rohr aktualisieren
DELETE	/api/pipes/{id}/	Rohr löschen

## 5. Schema Viewer

### Für Benutzer

Der Anzeige-Modus liefert eine Live-Visualisierung des Prozesses mit Echtzeit-Datenaktualisierung.

Funktionen:

Funktion	Beschreibung
Live-Gerätestatus	Farben und Animationen aktualisieren sich in Echtzeit
Klick auf Gerät	Öffnet Popup mit Bedienung und Informationen
Zoomen	Mausrad oder Schaltflächen «+» / «-»
Verschieben (Panning)	Rechte Maustaste gedrückt halten + ziehen
Alarmleiste	Oben/unten am Bildschirm — aktuelle aktive Alarmer
Namen-Umschalter	Gerätebeschriftungen ein-/ausblenden

### Popup zur Gerätesteuerung

Beim Klick auf ein Gerät öffnet sich ein Popup mit Tabs:

Tab «Steuerung» (Control):

Für jeden Gerätetyp:

Typ	Bedienelemente
Motor	Start/Stop, Umschalter Auto/Manuell, Frequenzsteuerung
Ventil	Öffnen/Schließen, Auto/Manuell, %, Durchfluss
Analogeingang (AI)	Auto/Manuell, Werteingabe, Skalenanzeige
Analogausgang (AO)	Auto/Manuell, Werteingabe, Min/Max
Digitaleingang (DI)	Auto/Manuell, ON/OFF-Anzeige, Verriegelung
Digitalausgang (DO)	Auto/Manuell, ON/OFF-Schaltflächen, Verriegelung
PID-Regler	Sollwert, Auto/Manuell, P/I/D-Parameter, Fehler und Ausgang
Timer	Start/Stop/Reset, Dauer einstellen
Zähler	Reset, aktueller Wert, Wert setzen
Tank	Füllstand, Volumen, Sensordaten

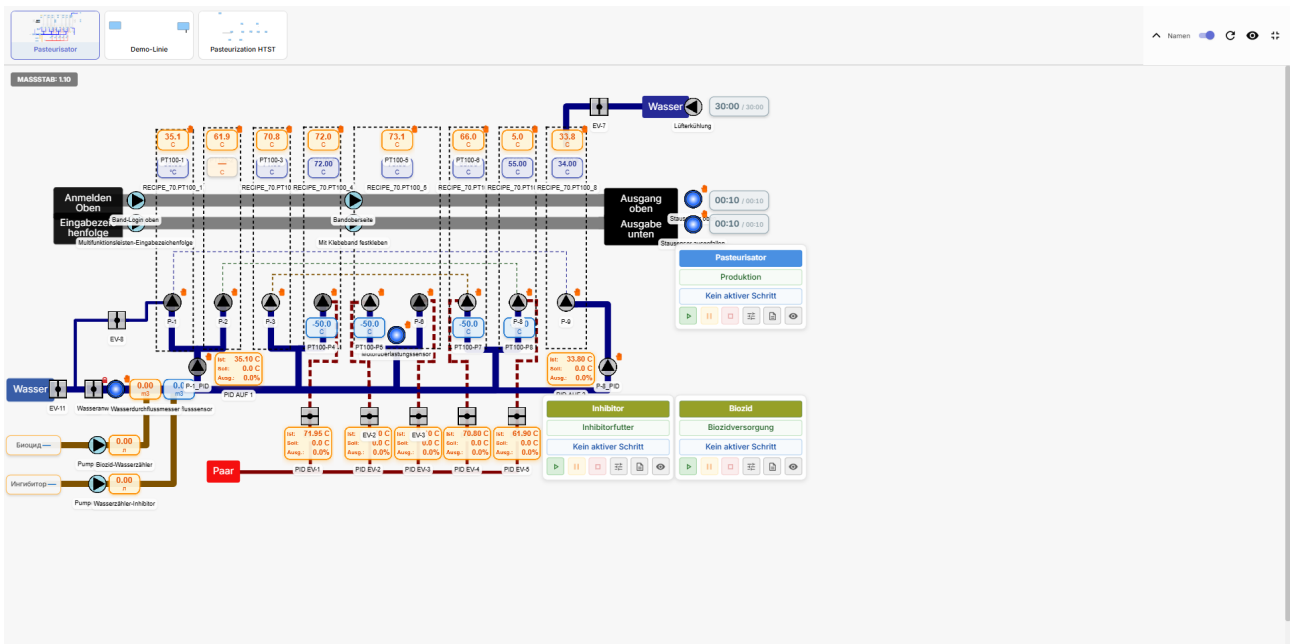
Tab «Status» (Status): Aktueller Zustand in Echtzeit.

Tab «Tabelle» (Table): Parameter in Tabellenform.

Tab «Trend» (Trend): Miniaturgraph der Werteentwicklung.

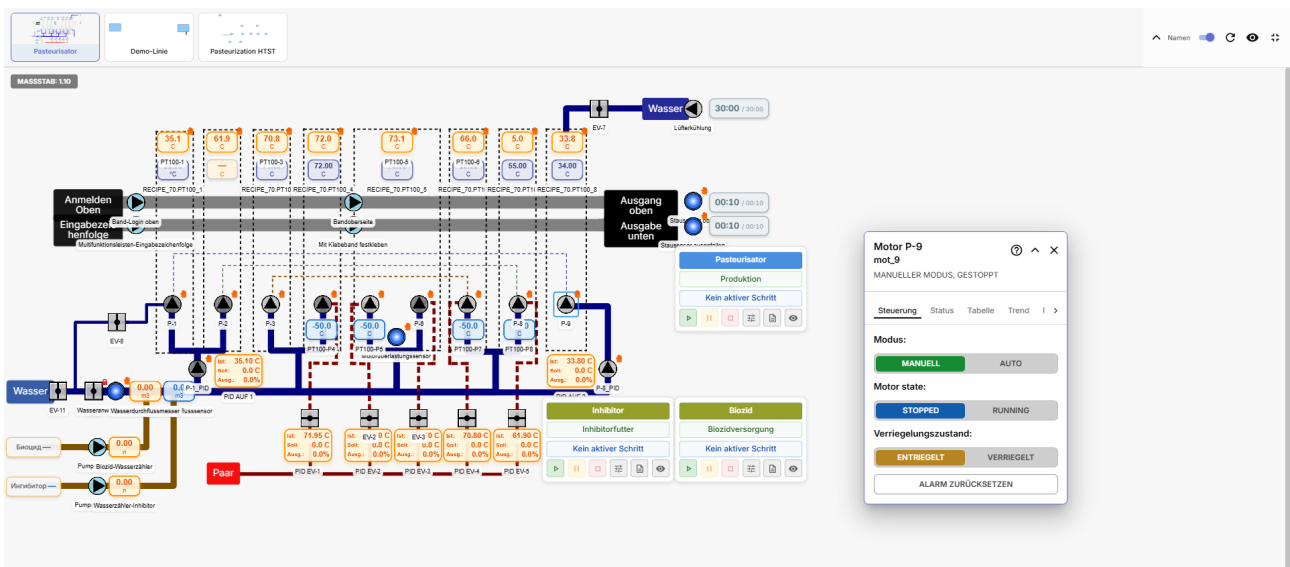
Tab «Logs» (Logs): Ereignisprotokoll des Geräts.

Tab «Übergangsbedingungen» (Transition Conditions): Aktive Sequenzbedingungen (falls das Gerät beteiligt ist).



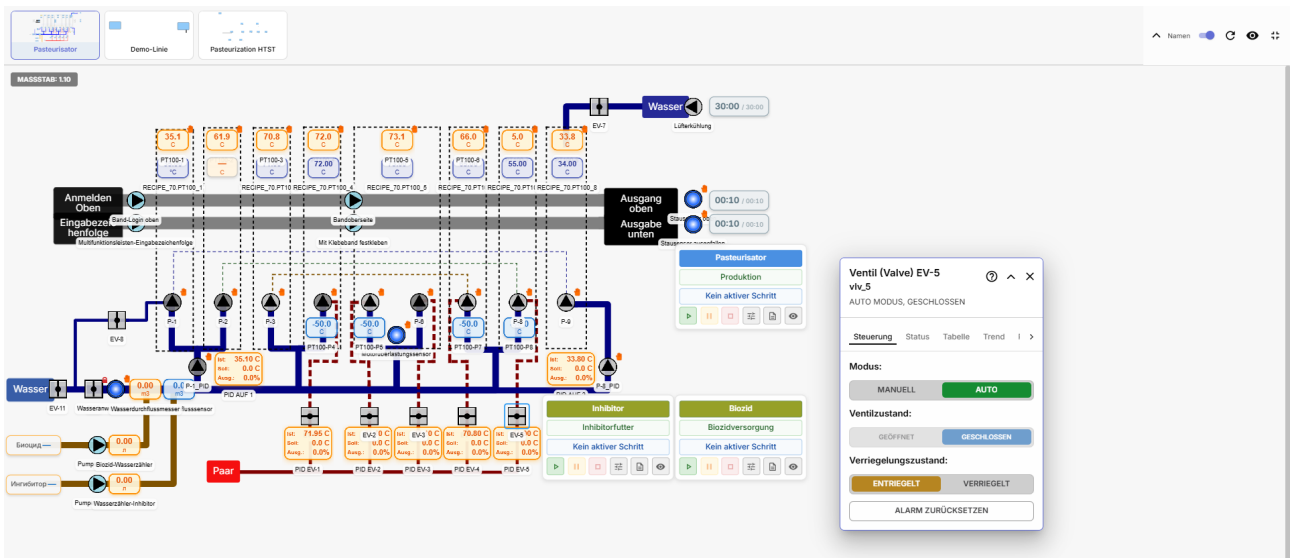
Schema Viewer Full

Vollständiger Schema Viewer: farbige Geräte mit Strömungsanimation auf den Rohren, Sequenzblock mit aktuellem Schritt, Alarmleiste oben.



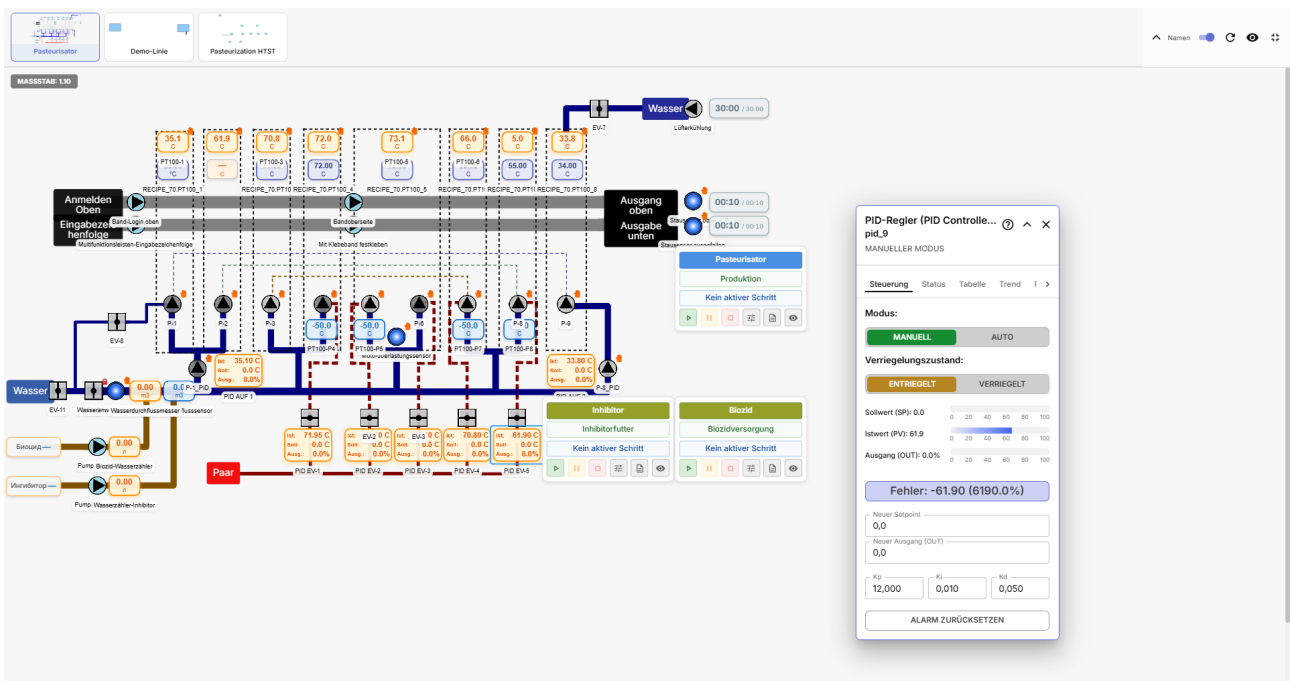
Schema Viewer Motor Popup

Motor-Popup: Tab Control mit Start/Stop, Umschalter Auto/Manual, Frequenz. Grüne Anzeige — Motor läuft.



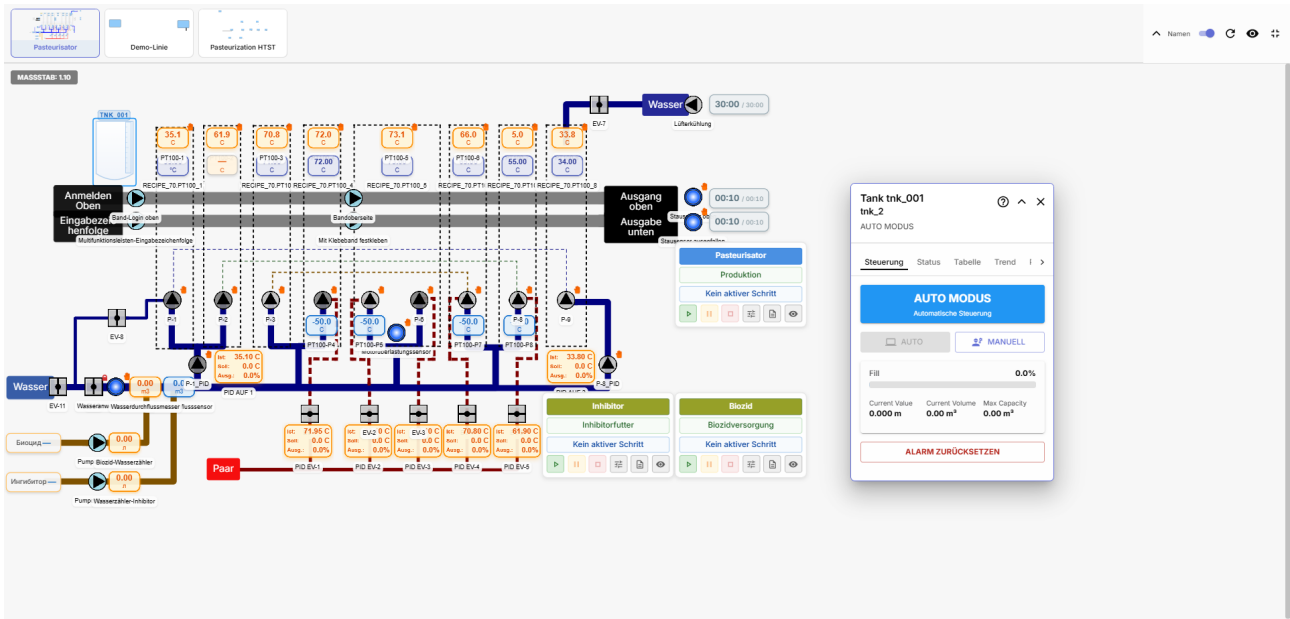
Schema Viewer Valve Popup

Ventil-Popup: Open/Close-Schaltflächen, Slider für Öffnungsgrad, Auto/Manual-Modus.



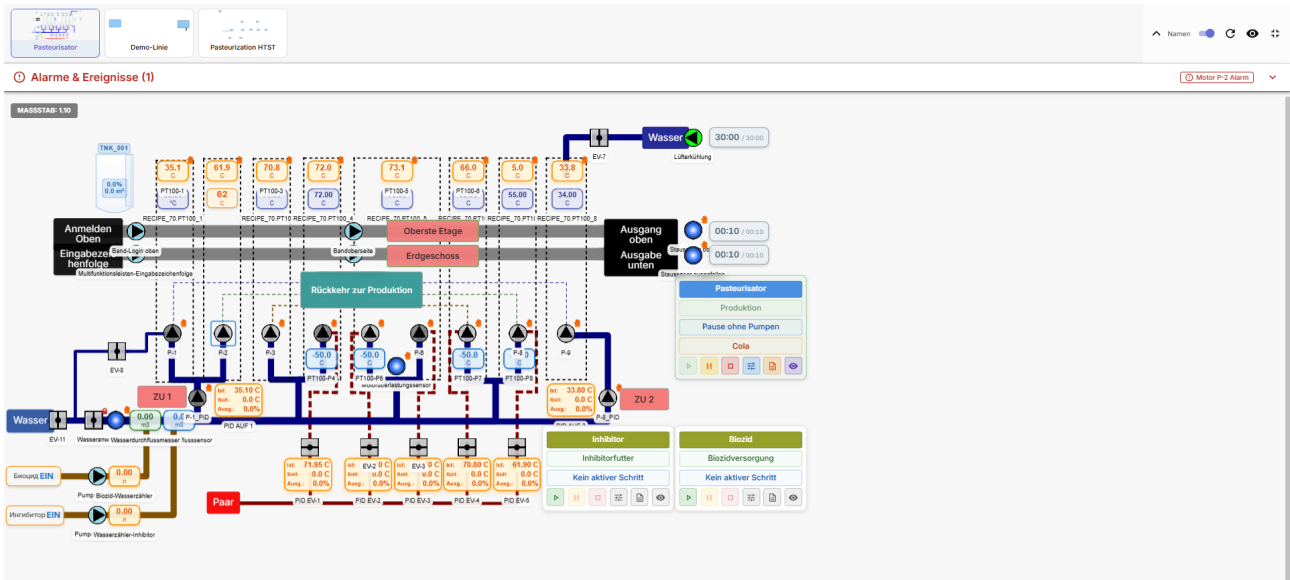
Schema Viewer Pid Popup

PID-Popup: Felder Setpoint, P, I, D, Anzeigen Error und Output, Umschalter Auto/Manual.



Schema Viewer Tank Popup

Tank-Popup: Füllstandsvisualisierung, Sensor-Tabelle (Füllstand, Druck), Volumen.



Schema Viewer Alarm Bar

Alarmliste oben: rote/gelbe Nachrichten mit Zeitstempel, Severity und Geräteame.

## Technische Dokumentation

WebSocket-Verbindungen für die Schema-Anzeige:

WebSocket-Pfad	Beschreibung
ws/schema/{schemald}/devices	Alle Geräte des Schemas (Realtime)
ws/schema/{schemald}/pipes	Alle Rohrleitungen (Strömungsanimation)
ws/schema/{schemald}/sequences	Alle Sequenzen des Schemas
ws/schema/{schemald}/batches	Chargen und Produktionslinien
ws/schema/{schemald}/accidents	Störungen dieses Schemas

### Format eines Geräteupdate-Events:

```
{
  "type": "schema_devices_status",
  "devices": {
    "mot_001": {
      "device": "mot_001",
      "current_status": {
        "value": 1,
        "is_running": true,
        "mode": "auto",
        "frequency": 50.0,
        "interlock": false
      },
      "status_icon": "/media/icons/motor_running.svg",
      "position_x": 100,
      "position_y": 200,
      "angle": 0,
      "updated_at": "2026-04-07T12:00:00Z"
    }
  }
}
```

### Delta-Updates (nur geänderte Felder):

```
{
  "type": "device_status_update",
  "device": "mot_001",
  "delta": true,
  "changes": {
    "frequency": 45.0
  }
}
```

### API zur Gerätesteuerung:

Methode	Endpoint	Beschreibung	Body
POST	/api/{device-type}/{id}/set_mode/	Auto/Manuell	{"mode": "manual"}
POST	/api/{device-type}/{id}/set_manual_value/	Wert setzen	{"value": 50.0}
POST	/api/{device-type}/{id}/set_simulation/	Simulation	{"simulation": true}
POST	/api/{device-type}/{id}/set_interlock/	Verriegelung	{"interlock": true}
POST	/api/{device-type}/{id}/update_status/	Status aktualisieren	{"current_status": {...}}

### WebSocket-Befehle (über Frontend websocketService):

```
{
  "type": "device_command",
  "device": "mot_001",
  "command": "start",
  "params": {},
  "options": {}
}
```

**Antwort:**

```
{
  "type": "command_response",
  "device": "mot_001",
  "command": "start",
  "status": "success",
  "data": {...},
  "timestamp": "2026-04-07T12:00:00Z"
}
```

## 6. Geräte

### Für Benutzer

Die Geräteverwaltung erlaubt das Erstellen, Bearbeiten und Löschen aller Gerätetypen.

Gerätetypen:

Typ	Code	Beschreibung
Ventil	VLV	Absperr- bzw. Regelarmaturen
Motor	MOT	Elektromotoren
Analogeingang	AI	Analogensensoren (Temperatur, Druck, Durchfluss)
Analogausgang	AO	Analogausgänge (4-20 mA, 0-10 V)
Digitaleingang	DI	Ein/Aus-Signale (Endschalter, Freigaben)
Digitalausgang	DO	Digitale Befehle
Koordinationsignal	COS	Systemübergreifende Koordinationssignale
PID-Regler	PID	Regelkreise
Zähler	CNT	Impulszähler
Tank	TNK	Behälter mit Volumenberechnung
Timer	TMR	Softwaretimer

Bildschirmelemente:

Element	Beschreibung
Ansichtsumschalter	Raster / Tabelle
Schaltfläche «+»	Neues Gerät anlegen
Typfilter	Dropdown Gerätetypen
Suche	Suche nach Name
Aktualisieren	Liste neu laden

Geräteformular (Erstellen/Bearbeiten):

Feld	Beschreibung
Geräte-ID	Eindeutiger Bezeichner (z. B. mot_001)
Name	Lesbarer Name
Namensübersetzungen	Mehrsprachige Namen
Typ	Gerätetyp auswählen
Beschreibung	Freitext
Standardwert	Startwert
Einheit	Einheit
Min / Max	Zulässiger Wertebereich
Skalierung	Signalumrechnungsparameter
Schaltfläche «Speichern»	Speichern
Schaltfläche «Abbrechen»	Zurück ohne Speichern

Feld	Beschreibung
Schaltfläche «Löschen»	Gerät löschen (im Bearbeitungsmodus)

ID	Name	Typ	IP	Status	Aktivierungsorte	Aktionen
vlv_1	EV-1	Ventil (Valve)	Nicht angegeben	OK	Pasteurisateur → Aktivieren des EV-1-Ventils [Offnen]	
vlv_10	EV-10	Ventil (Valve)	Nicht angegeben	OK	—	
vlv_11	EV-11	Ventil (Valve)	Nicht angegeben	OK	Pasteurisateur → Wasserset [Offnen]	
vlv_12	EV-12	Ventil (Valve)	Nicht angegeben	OK	—	
vlv_2	EV-2	Ventil (Valve)	Nicht angegeben	OK	Pasteurisateur → Einschalten des EV-2-Ventils [Offnen]	
vlv_3	EV-3	Ventil (Valve)	Nicht angegeben	OK	Pasteurisateur → Aktivieren des EV-3-Ventils [Offnen]	
vlv_4	EV-4	Ventil (Valve)	Nicht angegeben	OK	Pasteurisateur → Einschalten des EV-4-Ventils [Offnen]	
vlv_5	EV-5	Ventil (Valve)	Nicht angegeben	OK	Pasteurisateur → Aktivieren des EV-5-Ventils [Offnen]	
vlv_6	EV-6	Ventil (Valve)	Nicht angegeben	OK	Pasteurisateur → Wasserset [Offnen] Pasteurisateur → Wasserset [Offnen]	
vlv_7	EV-7	Ventil (Valve)	Nicht angegeben	OK	—	
vlv_8	EV-8	Ventil (Valve)	Nicht angegeben	OK	—	
vlv_9	EV-9	Ventil (Valve)	Nicht angegeben	OK	—	
mot_L	P-1	Motor	Nicht angegeben	OK	Pasteurisateur → Start der Pumpen 1, 9 [Start] Pasteurisateur → Start der Pumpen 2, 8 [Start]	
mot_1R	P-1 DIN	Motor	Nicht angegeben	OK	Pasteurisateur → Start der Pumpen 2, 8 [Start]	

Devices List Grid

Geräteseite als Raster: Karten mit Gerätesymbolen (Motor, Ventil usw.), Name, Typ, Wert. Typfilter oben.

ID	Name	Typ	IP	Status	Aktionen
vlv_1	EV-1	Ventil (Valve)	Nicht angegeben	OK	
vlv_10	EV-10	Ventil (Valve)	Nicht angegeben	OK	
vlv_11	EV-11	Ventil (Valve)	Nicht angegeben	OK	
vlv_12	EV-12	Ventil (Valve)	Nicht angegeben	OK	
vlv_2	EV-2	Ventil (Valve)	Nicht angegeben	OK	
vlv_3	EV-3	Ventil (Valve)	Nicht angegeben	OK	
vlv_4	EV-4	Ventil (Valve)	Nicht angegeben	OK	
vlv_5	EV-5	Ventil (Valve)	Nicht angegeben	OK	
vlv_6	EV-6	Ventil (Valve)	Nicht angegeben	OK	
vlv_7	EV-7	Ventil (Valve)	Nicht angegeben	OK	
vlv_8	EV-8	Ventil (Valve)	Nicht angegeben	OK	
vlv_9	EV-9	Ventil (Valve)	Nicht angegeben	OK	

Devices List Table

Tabellenansicht: Spalten ID, Name, Typ, Standardwert, Einheit, Aktionen.

**S SCADA** Geräte Serhii Kudlai  
Администратор

### Neues Gerät erstellen

1 Grundlegende Informationen — 2 Gerätekonfiguration

Gerätename \*

**Namensübersetzungen** v  
Name pro Sprache. Englisch wird als Fallback verwendet.

Gerätetyp \* v

SPS v  
Wählen Sie die SPS aus, zu der das Gerät gehört

Beschreibung

Abbrechen Weiter >

SCADA System v1.0.0

Device Form Motor 1

Motorformular: Grundfelder — ID, Name, PLC, Description, Gerätetyp.

**S SCADA** Geräte Serhii Kudlai  
Администратор

### Gerät bearbeiten P-1 (mot\_1)

✓ Grundlegende Informationen — 2 Gerätekonfiguration

**E/A-Punkte**

Startbefehl (Punkt) + 0  
P-1\_DO\_Start

Stoppbefehl (Punkt) + 0

Rückmeldung EIN (Punkt) + 0  
P-1\_FB

Geschwindigkeitssollwert (Punkt) + 0

Aktuelle Geschwindigkeit (Punkt) + 0

**Verknüpfte Geräte** + 0  
Von PID-Regler gesteuert

**Parameter**

Mindestgeschwindigkeit (Parameter) 0

Höchstgeschwindigkeit (Parameter) 0

Inversion (für Befehle?) 0

**Alarme**

Alarm generieren 0

Alarm mit Status 0

Alarmbestätigung 0

Alarmverzögerung (s) 00:00:00:05

**Zustand**

Gerät arbeitet 0

Gerät gestoppt 0

**System**

Gerät aktiviert 0

Manueller Modus 0

Alarm aktiv 0

Simulationen aktiv 0

**Zeitverzögerungen**

EIN-Verzögerung (s) 00:00:00:00

AUS-Verzögerung (s) 00:00:00:00

SCADA System v1.0.0

Device Form Motor 2

Motorformular: min/max speed, Verzögerungen, generate\_alarm, Speichern.

**S SCADA** Geräte Serhii Kudlai  
Администратор

**Neues Gerät erstellen**

1 Grundlegende Informationen — 2 Gerätekonfiguration

Gerätename \*

**Namensübersetzungen**  
Name pro Sprache. Englisch wird als Fallback verwendet.

Gerätetyp \*

SPS

Wählen Sie die SPS aus, zu der das Gerät gehört

Beschreibung

Abbrechen Weiter >

SCADA System v1.0.0

Device Form Analog 1

AI-Formular: Min/Max, Units, Scaling, Raw Min/Max.

**S SCADA** Geräte Serhii Kudlai  
Администратор

**Gerät bearbeiten PT100-1 (ai\_1)**

✓ Grundlegende Informationen — 2 Gerätekonfiguration

**E/A-Punkte**  
Analog-Eingangspunkt  
PT100-1\_LALValue

**Protokollierung**  
 Protokollierung

**Skalierung**  
Datentyp  
Mindestwert  
Höchstwert  
150  
Untere RAW-Grenze  
Obere RAW-Grenze  
28500  
Maßeinheiten  
C  
Skalierungsfaktor  
Versatz

**System**  
 Gerät aktiviert  
 Manueller Modus

**Alarime**  
 Allgemeiner Alarm  
 Alarm: Wert unter unterer Grenze  
 Alarm: Wert über oberer Grenze  
 Alarm: Signal außerhalb des Bereichs  
 Grenzwertkontrolle (SPS-Anfrage)  
 Grenzwertkontrolle (Konfiguration)  
 Warnung: hoher Wert  
 Warnung: niedriger Wert  
 Warnung: übermäßig niedriger Wert  
HH threshold (high-high)  
H threshold (high)  
L threshold (low)  
LL threshold (low-low)

SCADA System v1.0.0

Device Form Analog 2

AI-Formular: Warnungsschwellen, Standardwert und weitere Felder.

## Technische Dokumentation

Geräte-API (einheitlich für alle Typen):

Methode	Endpoint	Beschreibung
GET	/api/devices/all_devices/	Alle Geräte aller Typen
GET	/api/{device-type}/	Liste der Typgeräte
POST	/api/{device-type}/	Gerät erstellen
GET	/api/{device-type}/{id}/	Gerät abrufen
PUT	/api/{device-type}/{id}/	Aktualisieren
PATCH	/api/{device-type}/{id}/	Teilaktualisierung
DELETE	/api/{device-type}/{id}/	Löschen

### Endpoint-Typen:

- /api/analog-inputs/
- /api/analog-outputs/
- /api/digital-inputs/
- /api/digital-outputs/
- /api/motors/
- /api/valves/
- /api/pid-controllers/
- /api/counters/
- /api/timers/
- /api/coordination-signals/
- /api/variable-speed-drives/

### Zusätzliche Aktionen (für jeden Typ):

Methode	Endpoint	Beschreibung
POST	/{id}/set_mode/	Auto/Manuell umschalten
POST	/{id}/set_simulation/	Simulation ein/aus
POST	/{id}/set_interlock/	Verriegelung setzen/aufheben
POST	/{id}/update_status/	Aktuellen Status aktualisieren
POST	/{id}/set_manual_value/	Manuellen Wert setzen
POST	/{id}/trigger_websocket_update/	WS-Update erzwingen (AllowAny)

### WebSocket: Geräte-Update

Verbindung: ws://ws/device/{device\_id}

```
{
  "type": "device_status_update",
  "device": "ai_001",
  "data": {
    "current_status": {
      "value": 23.5,
      "mode": "auto",
      "simulation": false,
      "interlock": false,
      "quality": "good"
    },
    "status_icon": null,
    "updated_at": "2026-04-07T12:00:00Z"
  }
}
```

## 7. SPS

### Für Benutzer

Verwaltungsseite für speicherprogrammierbare Steuerungen.

Bildschirmelemente:

Element	Beschreibung
SPS-Liste	Karten/Tabelle mit SPSen
Statusanzeige	Grün — Online, grau — Offline, rot — Fehler
Schaltfläche «+»	Neue SPS anlegen
Schaltfläche «Hardware Config»	Wechsel zum Hardware-Konfigurator
Umschalter Run Mode	SPS-Ausführungsmodus ein/aus

Informationen auf der SPS-Karte:

Feld	Beschreibung
Name	SPS-Name
Protokoll	Kommunikationsart (Siemens S7, Modbus, OPC UA)
Status	Online / Offline
Zykluszeit	Abfragegeschwindigkeit
IP-Adresse	Netzwerkadresse
Geräteanzahl	Anzahl gebundener Geräte

SPS-Formular:

Feld	Beschreibung
Name	SPS-Name
Beschreibung	Freitext
Protokoll	Kommunikationsprotokoll
IP / Port	Netzwerkeinstellungen
Rack / Slot	Für Siemens S7
Schaltfläche «Verbindung testen»	Verbindungstest
Schaltfläche «Speichern»	Konfiguration speichern

ID	Name	Adresse	Beschreibung	Status	Zykluszeit	SPS-Betriebsmodus	Aktionen
1	Paster	192.168.50.150	Контроллер пастеризатора	Gestoppt	-	<input type="checkbox"/>	

Plc List

SPS-Liste: Karten mit Statusanzeige (grün/grau), Name, Protokoll, IP, Run-Mode-Umschalter, Button Hardware Config.

**SPS erstellen**

Name \*

Adresse \*

Beschreibung

Protokoll

Kommunikationsprotokoll für diese SPS auswählen

Abbrechen SPS erstellen

Plc Form

SPS-Formular: Felder Name, Protokoll-Dropdown, IP-Adresse, Port, Rack, Slot, Schaltfläche Test Connection.

## Technische Dokumentation

### SPS-API:

Methode	Endpoint	Beschreibung
GET	/api/plcs/	SPS-Liste
POST	/api/plcs/	SPS erstellen
GET	/api/plcs/{id}/	SPS abrufen
PUT	/api/plcs/{id}/	Aktualisieren
DELETE	/api/plcs/{id}/	Löschen
GET	/api/plcs/{id}/status/	SPS-Status
POST	/api/plcs/{id}/set-run-mode/	Run Mode umschalten
GET	/api/plcs/status/	Status aller SPSen

### SPS-Simulation:

Methode	Endpoint	Beschreibung
POST	/api/plc/simulation/run/	Simulation starten
GET	/api/plc/simulation/status/	Simulationsstatus
POST	/api/plc/service/start/	SPS-Dienst starten
POST	/api/plc/service/stop/	SPS-Dienst stoppen

## 8. Sequenzen

### Für Benutzer

Sequenzen beschreiben den Ablauf der Produktionsausrüstung — Schritte, Übergänge zwischen ihnen und Umschaltbedingungen.

Bildschirmelemente:

Element	Beschreibung
Sequenzliste	Karten/Tabelle mit Filter
Schaltfläche «+»	Neue Sequenz erstellen
Statusanzeige	Grün — läuft, grau — gestoppt, gelb — Pause
Statusfilter	Alle / laufend / gestoppt
Suche	Nach Name

Sequenz-Editor (Tabs):

Tab	Beschreibung
Eigenschaften	Allgemeine Einstellungen (Name, Beschreibung, Gerätebindung)
Schritte	Liste der Schritte in Ausführungsreihenfolge
Übergangsbedingungen	Konfiguration der Übergänge zwischen Schritten
Ereignisse	Ereignisverarbeitung
Störungen	Störungsverarbeitung
Optionen	Optionen konfigurieren
Rezeptvariablen	Variablendefinitionen
Produktionsrezepte	Rezepte für diese Sequenz
Pulsmodule	Verwaltung von Pulsmodulen
Chargen	Chargenverwaltung (Batches)

Schritteinstellungen:

Feld	Beschreibung
Ordnungsnummer	Schrittnummer in der Sequenz
Name	Schrittname (+ Übersetzungen)
Beschreibung	Freitext
Geräte	Verknüpfte Geräte und ihre Werte im Schritt
Nächste Schritte	JSON der Übergänge: {"target_step_order": [condition_id, ...]}
Chargenaktion	NONE / TRANSFER / COMPLETE

Übergangsbedingung (Transition Condition):

Feld	Beschreibung
Typ	Device (Gerätewert), Timer (Zeit), Option (Bedieneroption)
Gerät	Überwachtes Gerät (bei Device)

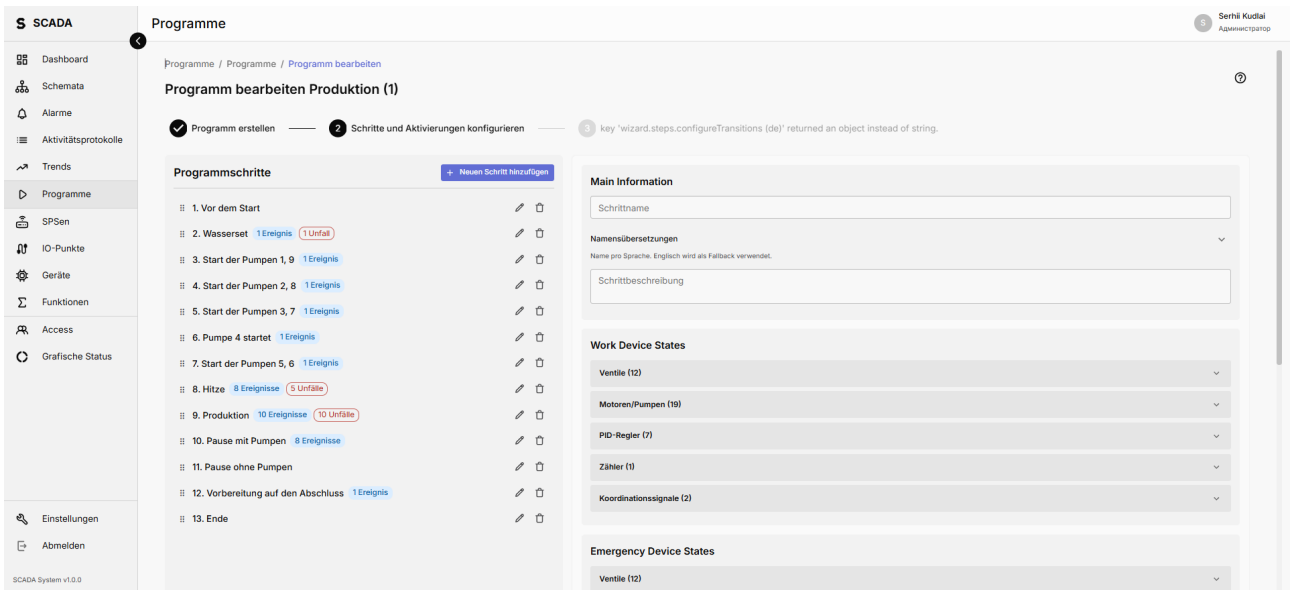
Feld	Beschreibung
Operator	==, !=, >, =, <=
Wert	Zielwert für den Vergleich
Dauer	Zeit (bei Timer)
Simulation	Bedingung zum Testen simulieren

The screenshot shows the 'Programme' section of the SCADA system. It features a sidebar with navigation options like Dashboard, Schemata, Alarmer, and Einstellungen. The main area displays a list of 4 programs. At the top right, there are buttons for 'Aktualisieren' and 'Programm erstellen'. A search bar is present above the table.

ID ↓	Name	Produktionsbereich	Erstellt	Beschreibung	Aktionen
6	Пастеризатор	Pasteurization HTST	8.5.2026	-	[Edit] [Copy] [Delete]
3	Biozidversorgung	Biozid	16.12.2025	-	[Edit] [Copy] [Delete]
2	Inhibitorfutter	Inhibitor	16.12.2025	-	[Edit] [Copy] [Delete]
1	Produktion	Pasteurisateur	19.11.2025	-	[Edit] [Copy] [Delete]

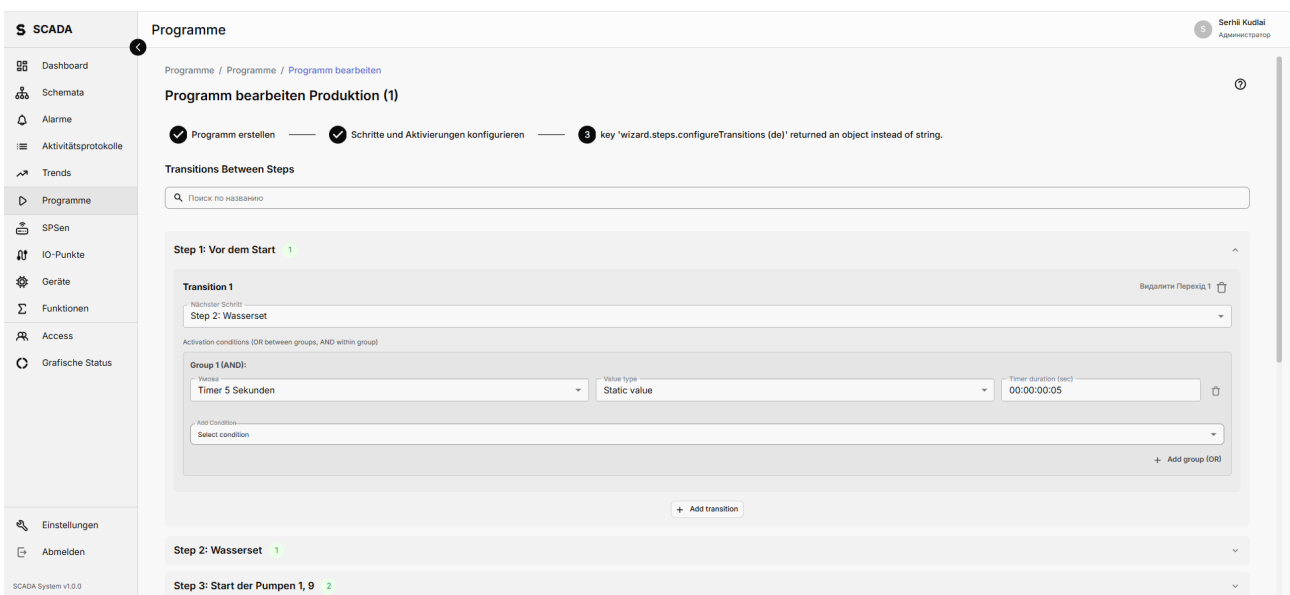
Sequences List

Sequenzliste: Karten mit Name, Statusanzeige (grün/grau), aktuellem Schritt, Edit/Delete-Schaltflächen.



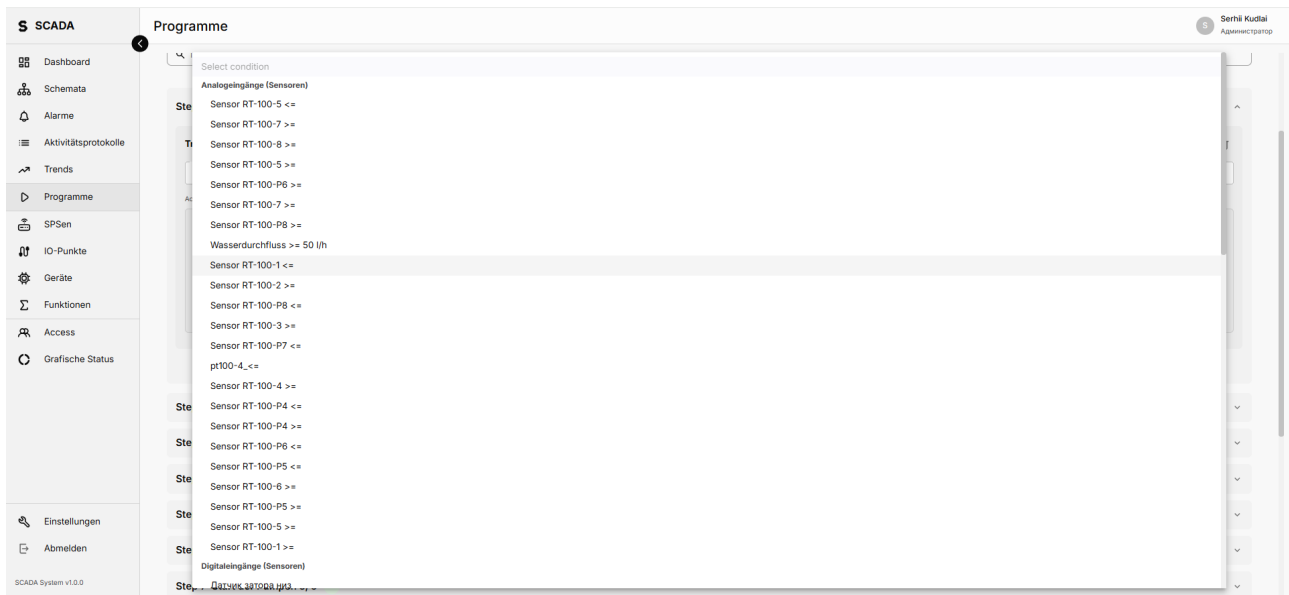
Sequence Editor Steps

Schritt-Editor: nummerierte Schrittliste, pro Schritt Name, gebundene Geräte, Übergangspfeile, Schaltflächen Add/Delete Step.



Sequence Editor Conditions

Übergangsbedingungen: Tabelle mit Spalten Type, Device, Operator, Value, Duration, Simulation-Umschalter.



Transition Condition Dialog

Dialog zum Anlegen einer Bedingung: Typauswahl (Device/Timer/Option), Gerät-Dropdown, Vergleichsoperator, Wertfeld.

## Technische Dokumentation

### Sequenz-API:

Methode	Endpoint	Beschreibung
GET	/api/sequences/	Sequenzliste
POST	/api/sequences/	Erstellen
GET	/api/sequences/{id}/	Abrufen
PUT	/api/sequences/{id}/	Aktualisieren
DELETE	/api/sequences/{id}/	Löschen
GET	/api/sequences/{id}/steps/	Schrittliste
POST	/api/sequences/{id}/add_step/	Schritt hinzufügen
POST	/api/sequences/{id}/set_steps_order/	Schritt Reihenfolge ändern
POST	/api/sequences/{id}/add_transition_conditions/	Bedingungen hinzufügen

### Schritt-API:

Methode	Endpoint	Beschreibung
GET	/api/sequence-steps/	Alle Schritte
POST	/api/sequence-steps/	Schritt erstellen
PUT	/api/sequence-steps/{id}/	Aktualisieren
DELETE	/api/sequence-steps/{id}/	Löschen

### Übergangsbedingungen-API:

Methode	Endpoint	Beschreibung
GET	/api/transition-conditions/	Alle Bedingungen
POST	/api/transition-conditions/	Erstellen
PUT	/api/transition-conditions/{id}/	Aktualisieren
DELETE	/api/transition-conditions/{id}/	Löschen
POST	/api/transition-conditions/{id}/simulate/	Bedingung simulieren
POST	/api/transition-conditions/clear_simulation/	Schritt-Simulation zurücksetzen

### WebSocket: Sequenzstatus

Verbindung: ws://ws/sequence/{sequence\_id}

```
{
  "type": "sequence_status",
  "id": 1,
  "name": "Production A",
  "is_running": true,
  "is_pause": false,
  "is_stop": false,
  "active_step": 3,
  "active_step_data": {
    "id": 3,
    "order": 3,
    "name": "Heating",
    "name_translations": {"de": "Heizen"},
    "description": "..."
  },
  "previous_step": 2,
  "current_step_start_time": "2026-04-07T11:30:00Z",
  "monitored_devices": {
    "ai_001": {"value": 72.3, "mode": "auto"}
  },
  "activated_devices": {
    "mot_001": {"value": 1, "is_running": true}
  },
  "conditions": [
    {
      "id": 10,
      "type": "device",
      "device_id": "ai_001",
      "operator": ">=",
      "value": 80.0,
      ... (+9 cмpok)
    }
  ]
}
```

## 9. Rezepte und Variablen

### Für Benutzer

Rezepte definieren konkrete Geräteparameter für jeden Schritt einer Sequenz. Rezeptvariablen erlauben, diese Werte zu parametrisieren.

Rezepttypen:

Typ	Beschreibung
Recipe (Schrittzept)	Legt Gerätewerte je Sequenzschritt fest
Production Recipe (Produktionsrezept)	Variablensatz für einen konkreten Produktionsauftrag

Formular Schrittzept:

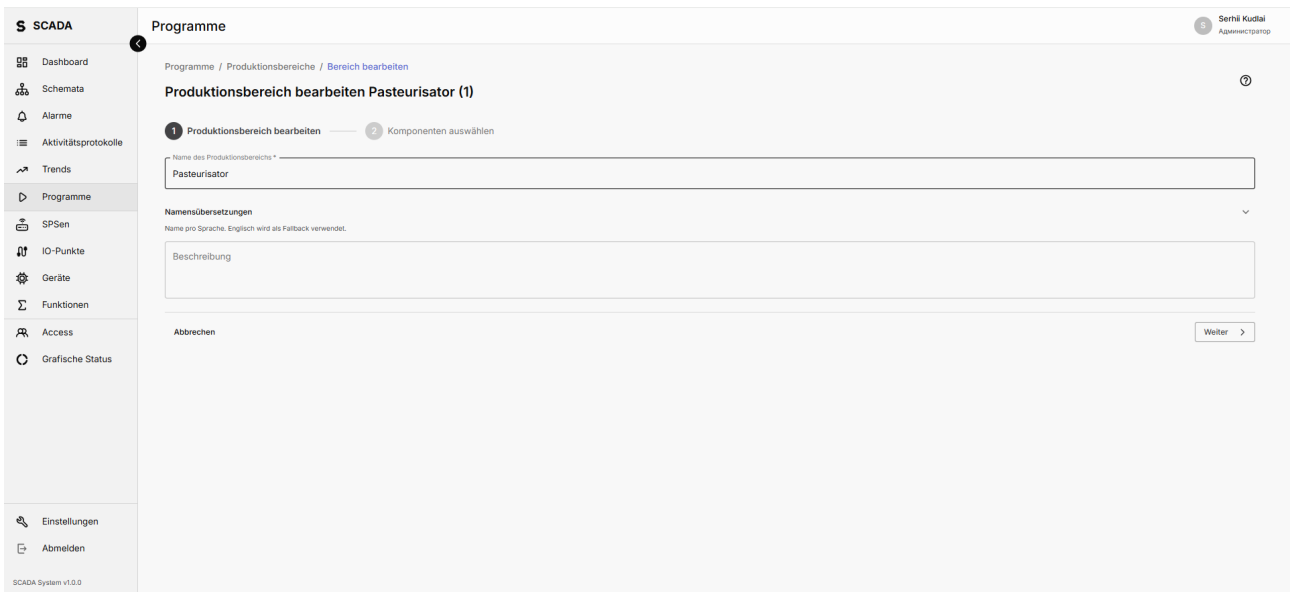
Feld	Beschreibung
Name	Rezeptname (+ Übersetzungen)
Sequenz	Bindung an eine Sequenz
Schritte	Pro Schritt ein Satz «Gerät → Wert»

Formular Produktionsrezept:

Feld	Beschreibung
Name	Rezeptname
Bindung	An Sequenz ODER Produktionslinie (XOR)
Variablen	Liste der Variablen mit Werten

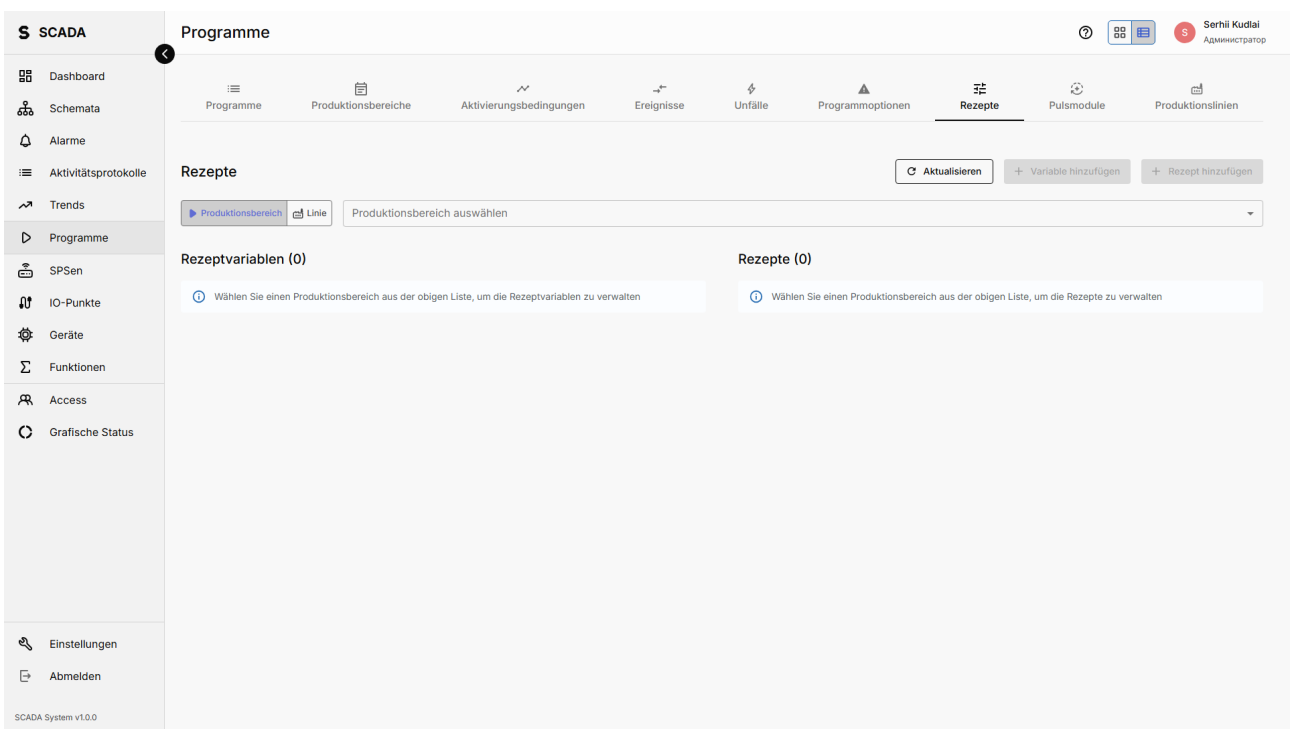
Rezeptvariablen:

Feld	Beschreibung
Name	Variablenname (+ Übersetzungen)
Bindung	An Sequenz ODER Produktionslinie (XOR)
Typ	Numerisch, Text
Standardwert	Startwert
Min / Max	Grenzwerte



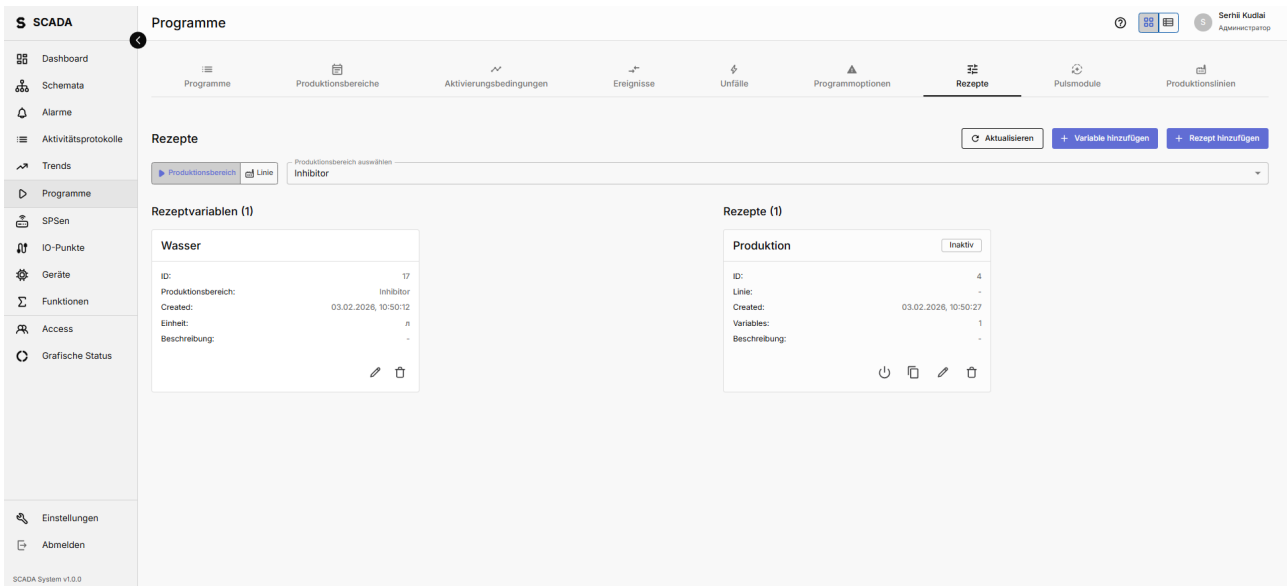
Recipe Creation Wizard

Rezept-Assistent: Schritt 1 — Name und Sequenz, Schritt 2 — je Schritt Tabelle «Gerät → Wert».



Production Recipes Management

Produktionsrezepte: Tabelle mit Rezepten, Schaltflächen Create/Edit/Delete/Duplicate, Umschalter Sequence/Production Line.



Recipe Variables Management

Variablen: Liste mit Name, Typ, Standardwert, Min/Max. Schaltflächen zum Hinzufügen/Löschen.

## Technische Dokumentation

### Rezept-API:

Methode	Endpoint	Beschreibung
GET	/api/recipes/	Liste der Schrittrezepte
POST	/api/recipes/	Rezept erstellen
PUT	/api/recipes/{id}/	Aktualisieren
DELETE	/api/recipes/{id}/	Löschen

### Produktionsrezept-API:

Methode	Endpoint	Beschreibung	Parameter
GET	/api/production-recipes/	Alle Rezepte	?sequence=ID oder ?production_line=ID
POST	/api/production-recipes/	Erstellen	—
PUT	/api/production-recipes/{id}/	Aktualisieren	—
DELETE	/api/production-recipes/{id}/	Löschen	—
POST	/api/production-recipes/{id}/select-production-recipe/	Aktives Rezept wählen	—

### Variablen-API:

Methode	Endpoint	Beschreibung	Parameter
GET	/api/production-recipe-variables/	Alle Variablen	?sequence=ID oder ?production_line=ID
POST	/api/production-recipe-variables/	Erstellen	—
PUT	/api/production-recipe-variables/{id}/	Aktualisieren	—
DELETE	/api/production-recipe-variables/{id}/	Löschen	—

E

# 10. Ereignisse und Störungen

## Für Benutzer

Ereignisse (Events) — Bedingungen, die während der Sequenzausführung Aktionen auslösen (z. B. Umschalten beim Erreichen eines Wertes).

Störungen (Accidents) — kritische Bedingungen, die den Prozess bei Gefahrensituationen stoppen.

Ereignisformular:

Feld	Beschreibung
Name	Ereignisname (+ Übersetzungen)
Typ	Auslösetyp
Gerät	Überwachtes Gerät
Bedingung	Vergleichsoperator und Wert
Aktion	Was bei Auslösung passiert

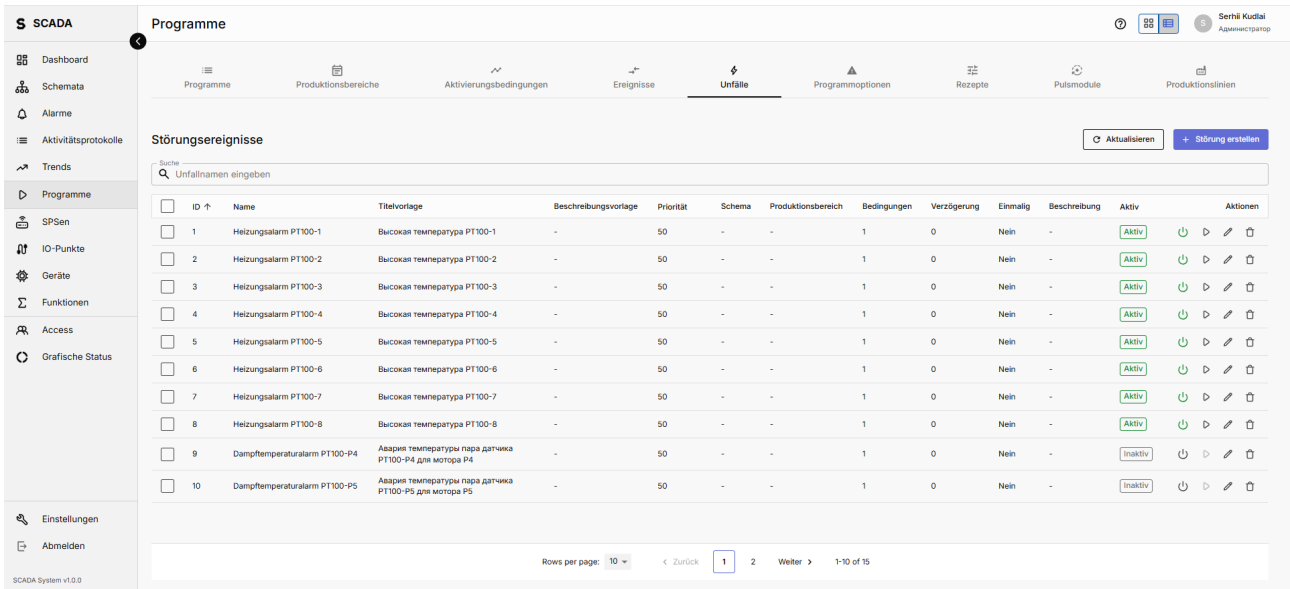
Störungsformular:

Feld	Beschreibung
Name	Störungsname (+ Übersetzungen)
Severity	INFO / WARNING / ERROR / CRITICAL
Gerät	Überwachtes Gerät
Bedingung	Operator und Wert
Verriegelung	Welche Geräte zu sperren sind

The screenshot shows the SCADA system's 'Ereignisse' (Events) management interface. It features a sidebar with navigation options and a main content area with a table of 12 events. The table columns are: ID, Name, Aktionstyp, Area, Aktivierungsbedingungen, Status, and Aktionen. The events listed include conditions like 'Wasseranwesenheitssensor', 'Pasteur-Wahlschaller „Oberste Etage“', and various 'Sensor RT-100' readings.

Events Management

Ereignisverwaltung: Tabelle mit Ereignissen, Spalten Name, Device, Condition, Action, Schaltflächen Add/Edit/Delete.



Accident Events Management

Störungsverwaltung: Tabelle mit Farbcodierung nach Severity (INFO-blau, WARNING-gelb, ERROR-rot, CRITICAL-dunkelrot).

## Technische Dokumentation

### Ereignis-API:

Methode	Endpoint	Beschreibung
GET	/api/events/	Ereignisliste
POST	/api/events/	Erstellen
PUT	/api/events/{id}/	Aktualisieren
DELETE	/api/events/{id}/	Löschen
POST	/api/events/{id}/set_active/	Aktivieren/deaktivieren

### Störungs-API:

Methode	Endpoint	Beschreibung
GET	/api/accident-events/	Liste der Störungsdefinitionen
POST	/api/accident-events/	Definition erstellen
GET	/api/accidents/	Liste der aufgetretenen Störungen
POST	/api/accidents/{id}/acknowledge/	Störung quittieren
POST	/api/accidents/{id}/resolve/	Störung beheben
POST	/api/accidents/resolve-all/	Alle beheben
GET	/api/accidents/active_events/	Aktive Störungen

### WebSocket: Störungen

Verbindung: ws://ws/accidents

```
{
  "type": "accident_created",
  "accident_id": 42,
  "title": "Hohe Temperatur",
  "description": "AI_001 hat 95°C überschritten",
  "status": "NEW",
  "severity_level": "CRITICAL",
  "device_id": "ai_001",
  "device_name": "Temperature Sensor 1",
  "schema_id": 1,
  "occurred_at": "2026-04-07T12:00:00Z"
}
```

# 11. Optionen

## Für Benutzer

Optionen sind Schaltflächen/Umschalter, die dem Bediener während der Sequenzausführung zur Verfügung stehen. Sie erlauben die Wahl alternativer Ausführungspfade.

Optionsformular:

Feld	Beschreibung
Name	Optionsname (+ Übersetzungen)
Sequenz	Bindung an eine Sequenz
Beschreibung	Freitext

Auf dem Schema (Schema Viewer):

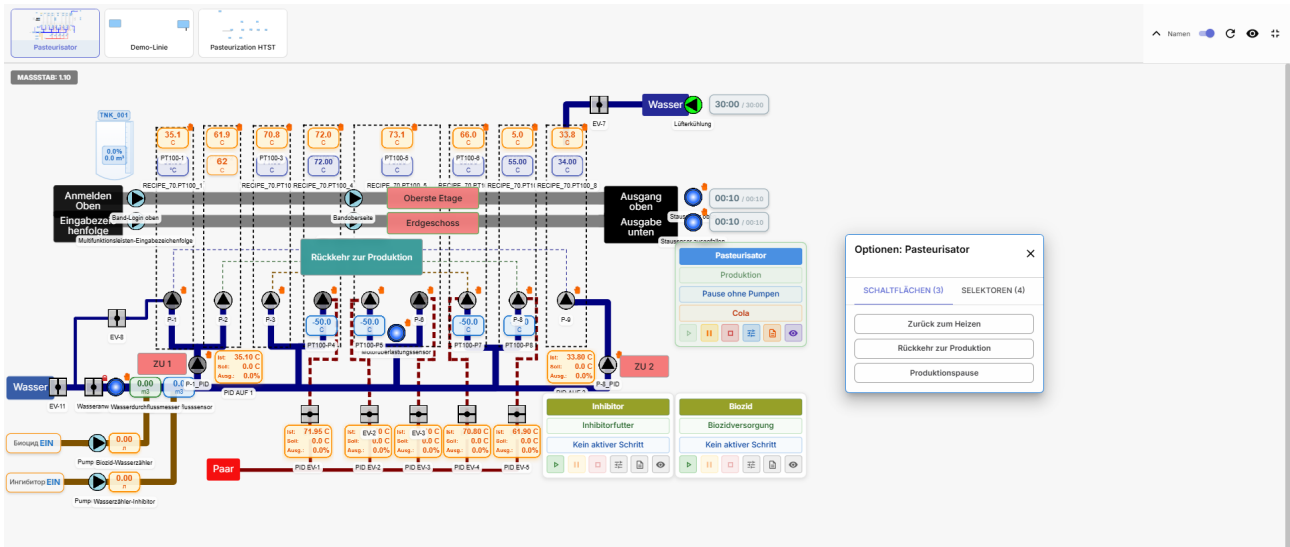
- Die Option wird als Schaltfläche dargestellt
- Ein Klick schaltet die Option um (ein/aus)
- Die Farbe ändert sich je nach Zustand

The screenshot shows the 'Programmoptionen (7)' management interface. It includes a search bar, a table of options, and various control buttons. The table data is as follows:

ID	Name	Beschreibung	Type	Area	Aktiviert	Bedingungen	Availability	Status	Aktionen
1	Produktionspause	-	Schaltfläche (Trigger)	Pasteurisator	31.03.2026, 13:43:31	1	Nicht verfügbar - Bedingungen nicht erfüllt	Inaktiv	[Icons]
2	Oberste Etage	-	Selektor	Pasteurisator	-	1	Nicht verfügbar - Bedingungen nicht erfüllt	Inaktiv	[Icons]
3	Erdgeschoss	-	Selektor	Pasteurisator	-	1	Nicht verfügbar - Bedingungen nicht erfüllt	Inaktiv	[Icons]
4	Rückkehr zur Produktion	-	Schaltfläche (Trigger)	Pasteurisator	09.04.2026, 14:01:30	1	Nicht verfügbar - Bedingungen nicht erfüllt	Inaktiv	[Icons]
5	Zurück zum Heizen	-	Schaltfläche (Trigger)	Pasteurisator	24.03.2026, 15:44:42	1	Nicht verfügbar - Bedingungen nicht erfüllt	Inaktiv	[Icons]
6	ZU 1	-	Selektor	Pasteurisator	-	1	Nicht verfügbar - Bedingungen nicht erfüllt	Inaktiv	[Icons]
7	ZU 2	-	Selektor	Pasteurisator	-	1	Nicht verfügbar - Bedingungen nicht erfüllt	Inaktiv	[Icons]

Options Management

Optionenverwaltung: Liste der Optionen mit Name, Sequenz, Beschreibung, Edit/Delete-Schaltflächen.



Schema Option Button

Options-Schaltfläche auf dem Schema: rechteckige Schaltfläche mit Namen, grün — aktiv, grau — inaktiv.

## Technische Dokumentation

Optionen-API:

Methode	Endpoint	Beschreibung
GET	/api/sequence-options/	Optionsliste
POST	/api/sequence-options/	Erstellen
PUT	/api/sequence-options/{id}/	Aktualisieren
DELETE	/api/sequence-options/{id}/	Löschen

## 12. Pulsmodule

### Für Benutzer

Pulsmodule erzeugen zyklische Ein-/Ausschaltfolgen für Geräte (z. B. zyklische Wasserzufuhr).

Struktur:

Ebene	Beschreibung
Modul	Container für Schritte
Schritt	Eine Zyklusetape
Phase	Zeitintervall innerhalb eines Schritts (ein/aus)
Phasengerät	Welches Gerät ein-/ausgeschaltet wird

The screenshot shows the 'Pulsmodule' management page in the SCADA system. The left sidebar contains a navigation menu with items like 'Dashboard', 'Schemata', 'Alarmer', 'Aktivitätsprotokolle', 'Trends', 'Programme', 'SPSen', 'IO-Punkte', 'Geräte', 'Funktionen', 'Access', and 'Grafische Status'. The main content area is titled 'Pulsmodule' and features a search bar with the placeholder 'Suche Modulnamen eingeben'. Below the search bar, there are two buttons: 'Aktualisieren' and '+ Modul erstellen'. The central message reads 'Keine Module gefunden' with a '+ Erstes Modul erstellen' button below it. The top navigation bar includes 'Programme', 'Produktionsbereiche', 'Aktivierungsbedingungen', 'Ereignisse', 'Unfälle', 'Programmooptionen', 'Rezepte', 'Pulsmodule', and 'Produktionslinien'. The user's name 'Serhii Kudlai' and role 'Администратор' are visible in the top right corner.

Pulse Module Management

Pulsmodule-Verwaltung: hierarchische Liste — Modul → Schritte → Phasen → Geräte. Schaltflächen Add/Edit/Delete auf jeder Ebene.

### Technische Dokumentation

Pulsmodul-API:

Methode	Endpoint	Beschreibung
GET	/api/pulse-modules/	Modulliste
POST	/api/pulse-modules/	Erstellen
POST	/api/pulse-modules/{id}/start/	Starten

Methode	Endpoint	Beschreibung
POST	/api/pulse-modules/{id}/stop/	Stoppen
GET	/api/pulse-module-steps/	Modulschritte
GET	/api/pulse-module-step-phases/	Schrittphasen
GET	/api/pulse-module-devices/	Phasengeräte

## 13. Chargenverwaltung

### Für Benutzer

Die Chargenverwaltung verfolgt Produktionschargen (Batches) über den gesamten Prozess hinweg.

Elemente der Chargen-Leiste:

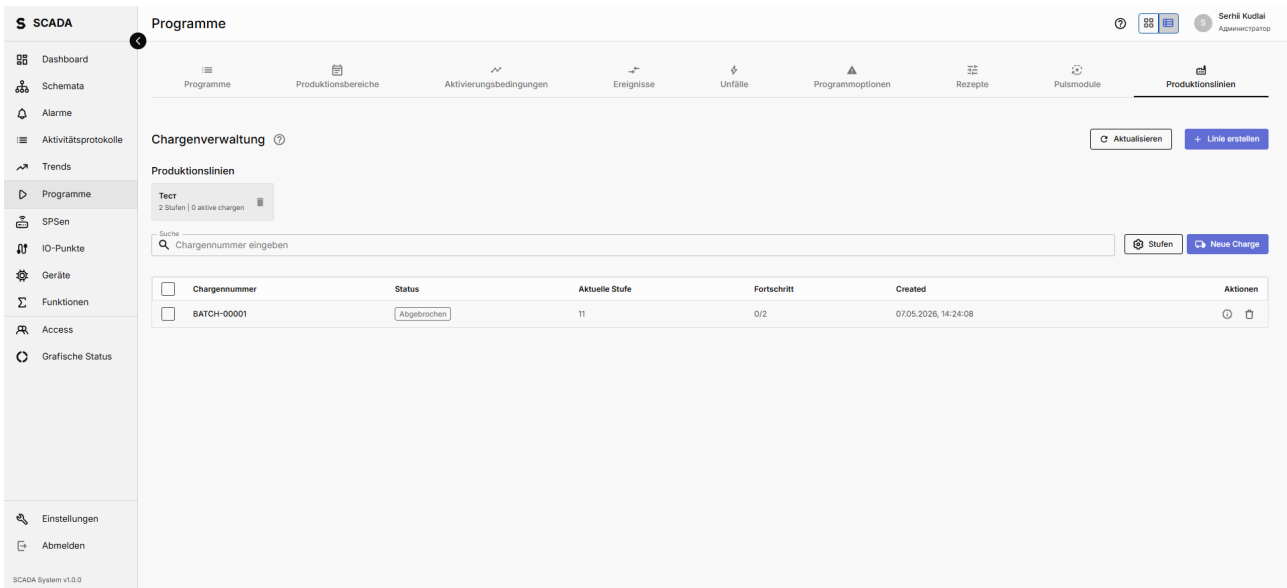
Element	Beschreibung
Charge erstellen	Schaltfläche zum Anlegen einer Charge
Chargen-Tabelle	Liste mit Nummer, Rezept, Status, Fortschritt
Chargenstatus	Pending / Queued / In Progress / Paused / Completed / Failed / Cancelled
Warteschlange	Bearbeitungsreihenfolge der Chargen

Chargenaktionen:

Schaltfläche	Beschreibung
Play	Start/Fortsetzung der Charge
Pause	Pausieren
Stop	Vollständiger Stopp
Skip	Aktuellen Schritt überspringen
Cancel	Charge abrechnen
Delete	Aus der Liste entfernen

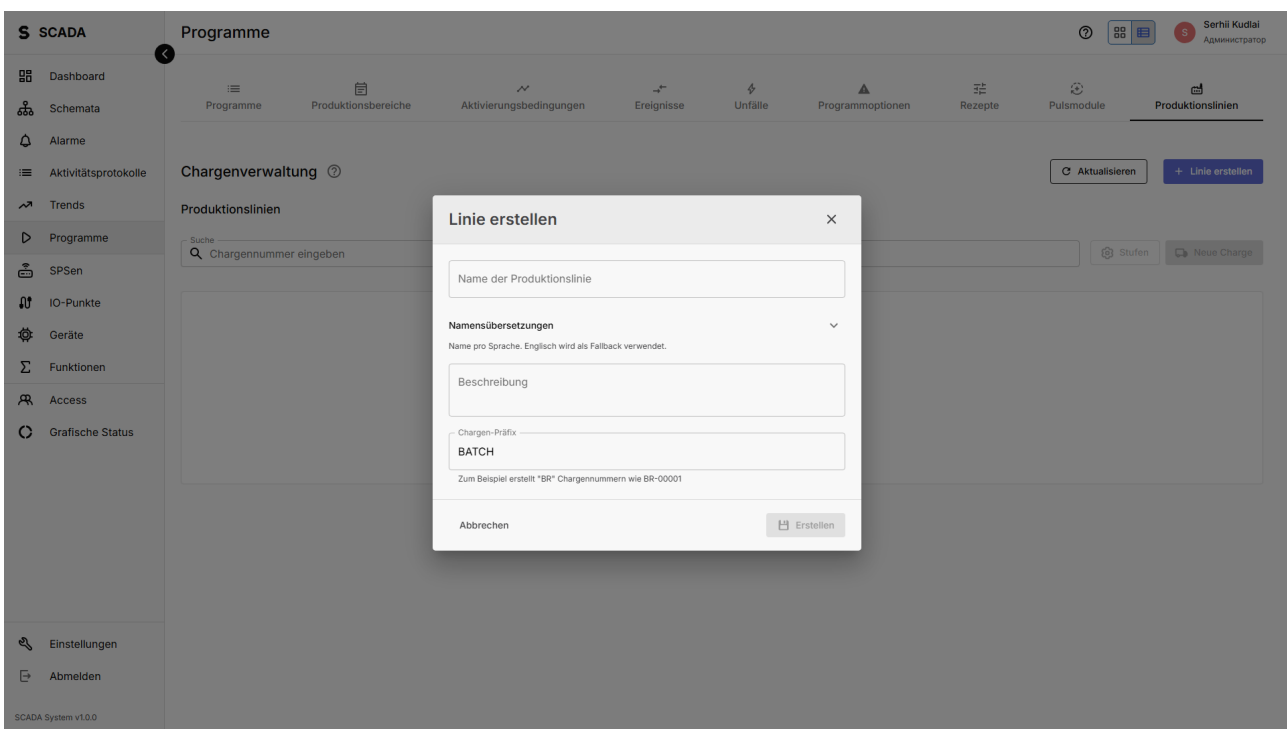
Chargen-Erstelldialog:

Feld	Beschreibung
Produktionslinie	Linie auswählen
Rezept	Rezept auswählen
Chargennummer	Eindeutige Nummer (BAT-001)
Auto-Start	Sofort starten



Batch Management Panel

Chargenverwaltungspanel: Tabelle mit Chargen, farbcodiertem Status (grün — In Progress, grau — Pending, blau — Completed, rot — Failed), Steuerschaltflächen.



Create Batch Dialog

Ersteldialog: Linienauswahl, Rezept, Nummer, Checkbox «Auto-Start», Schaltflächen «Erstellen» und «Abbrechen».

## Technische Dokumentation

Chargen-API:

Methode	Endpoint	Beschreibung
GET	/api/batches/	Alle Chargen
POST	/api/production-lines/{id}/create_batch/	Charge erstellen
POST	/api/batches/{id}/start/	Starten
POST	/api/batches/{id}/pause/	Pause
POST	/api/batches/{id}/resume/	Fortsetzen
POST	/api/batches/{id}/cancel/	Abbrechen
POST	/api/batches/{id}/transfer/	Zur nächsten Stufe übergeben
GET	/api/batches/{id}/history/	Stufenhistorie
GET	/api/batch-queue/	Chargen-Warteschlange
POST	/api/batch-queue/reorder/	Warteschlangen-Reihenfolge ändern

### WebSocket: Chargenstatus

Verbindung: `ws://ws/schema/{schemald}/batches`

```
{
  "type": "batch_status",
  "batch_id": "uuid-...",
  "batch_number": "BAT-001",
  "status": "IN_PROGRESS",
  "current_stage": "Mixing",
  "progress": 65,
  "production_line_id": "uuid-...",
  "recipe_name": "Recipe A",
  "updated_at": "2026-04-07T12:00:00Z"
}
```

### WebSocket-Befehle (Client → Server):

```
{
  "type": "batch_command",
  "command": "start",
  "batch_id": "uuid-...",
  "production_line_id": "uuid-..."
}
```

# 14. Produktionslinien

## Für Benutzer

Eine Produktionslinie fasst mehrere Sequenzen zu einer Produktionsstrecke mit Stufen zusammen.

Elemente:

Feld	Beschreibung
Name	Linienname (+ Übersetzungen)
Stufen	Liste der Stufen (Production Line Stages), jede an eine Sequenz gebunden
Stufenreihenfolge	Definiert den Durchlauf der Charge
Status	Aktiv / Inaktiv

Production Lines Management

Linienverwaltung: Tabelle mit Name, Stufenzahl, Status, Edit/Delete-Schaltflächen. Detailansicht — Stufenliste mit Drag-and-Drop für die Reihenfolge.

## Technische Dokumentation

Produktionslinien-API:

Methode	Endpoint	Beschreibung
GET	/api/production-lines/	Linienliste
POST	/api/production-lines/	Erstellen
PATCH	/api/production-lines/{id}/	Aktualisieren
DELETE	/api/production-lines/{id}/	Löschen
POST	/api/production-lines/{id}/activate/	Aktivieren
POST	/api/production-lines/{id}/deactivate/	Deaktivieren

Methode	Endpoint	Beschreibung
GET	/api/production-line-stages/	Alle Stufen
POST	/api/production-line-stages/	Stufe erstellen
PATCH	/api/production-line-stages/{id}/	Aktualisieren
DELETE	/api/production-line-stages/{id}/	Löschen

## 15. Trends

### Für Benutzer

Trends visualisieren historische Gerätedaten als Diagramme.

Bildschirmelemente:

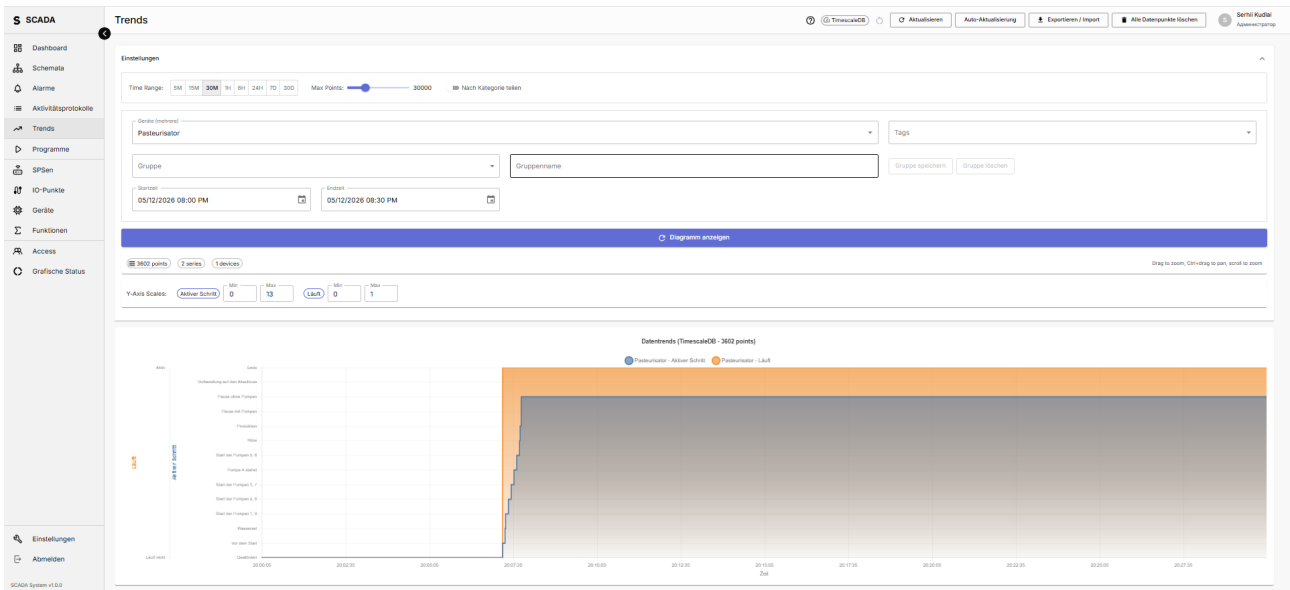
Element	Beschreibung
Geräteauswahl	Multi-Select Geräte/Parameter für das Diagramm
Zeitbereich	Presets: 5 min, 15 min, 30 min, 1 h, 6 h, 24 h, 7 Tage, 30 Tage
Benutzerdefinierter Bereich	Freie Datum/Uhrzeit-Auswahl
Diagramm	Liniendiagramm (Chart.js) mit Zoom
Gerätegruppen	Sätze von Geräten speichern/laden
Speicher	Information zur Speicherplatznutzung

Diagramm-Interaktion:

Aktion	Beschreibung
Drag (Ziehen)	Bereich zum Zoomen markieren
Mausrad	Zoom
RMT + Ziehen	Verschieben (Panning)
Mouseover	Tooltip mit Werten
Reset zoom	Zoom auf Anfang zurücksetzen

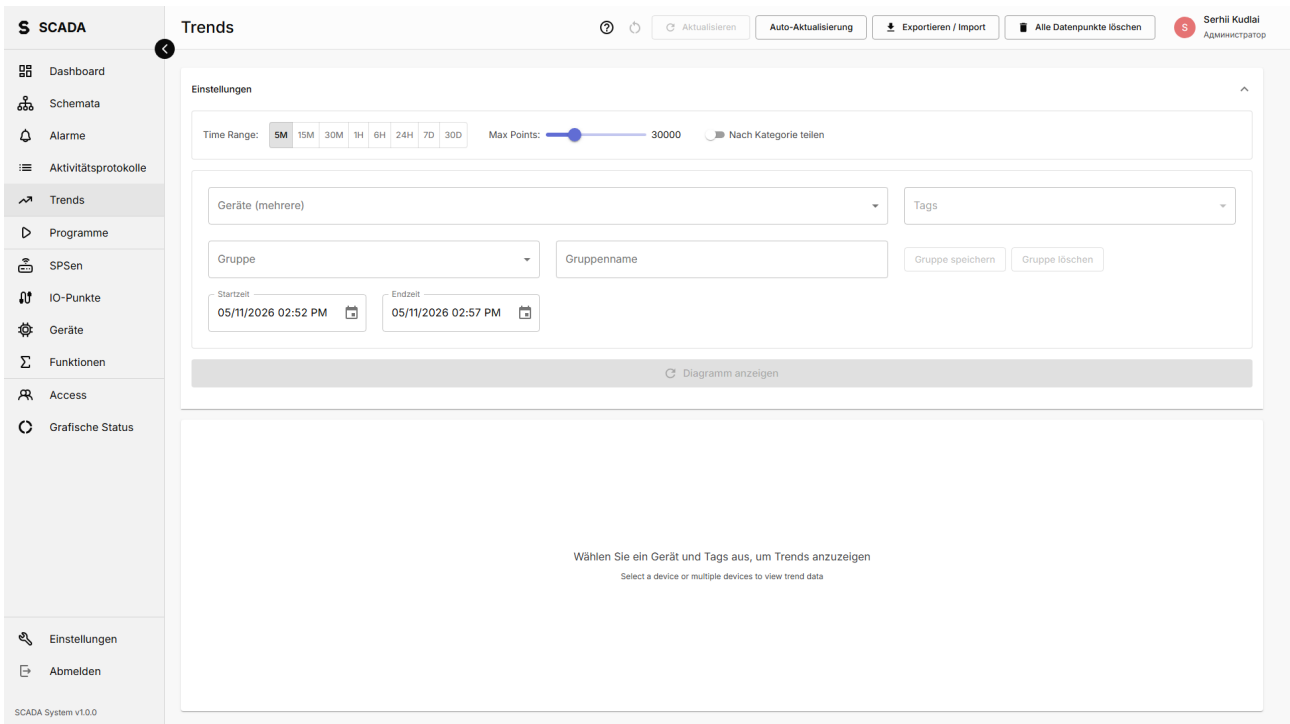
Schaltflächen:

Schaltfläche	Beschreibung
Geräte hinzufügen	Öffnet Multi-Select
Gruppe speichern	Speichert das aktuelle Geräteset
Gruppe laden	Wählt eine gespeicherte Gruppe
Gruppe löschen	Löscht eine Gruppe
Download	Daten exportieren (CSV/JSON)
Diagramm herunterladen	Exportiert Diagramm als Bild
Daten löschen	Entfernt Datenpunkte (Superuser)
Aktualisieren	Daten neu laden



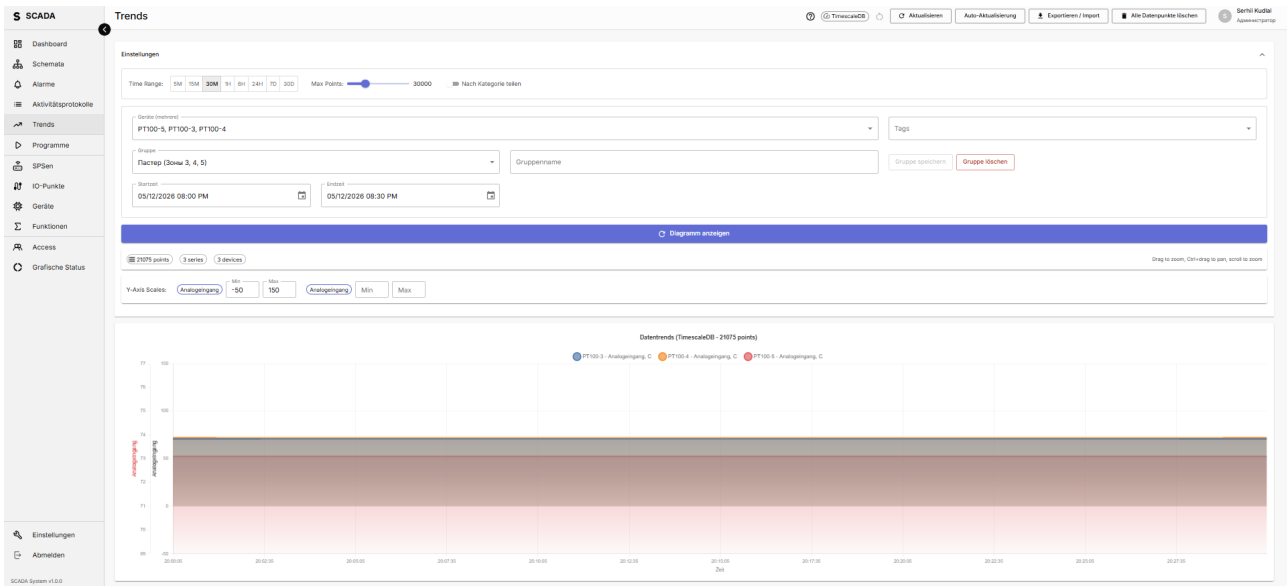
Trends Full

Vollständiges Trends-Panel: oben Geräteauswahl (Multi-Select-Chips), Zeitbereich (Schaltflächen 5m/15m/1h/6h/24h/7d/30d und DateTimePicker). Mittig Liniendiagramm mit mehreren Farblinien. Rechts die Legende.



Trends Zoom

Diagramm nach Zoom: markierter Bereich vergrößert, Reset-zoom-Button sichtbar.



Trends Device Groups

Dropdown gespeicherter Gerätegruppen: «Gruppe 1 — Temperaturen», «Gruppe 2 — Drücke», Schaltflächen Save/Load/Delete.

## Technische Dokumentation

Trend-API (TimescaleDB):

Methode	Endpoint	Beschreibung	Parameter
GET	/api/graphs/graph/	Diagramm Daten	?start_time=ISO&end_time=ISO&device_ids=id1,id2
GET	/api/graphs/available-parameters/	Verfügbare Parameter	—
GET	/api/graphs/timescale-status/	TimescaleDB-Status	—
DELET	/api/graphs/delete-all/	Alle Daten löschen	—
GET	/api/data-points/	Datenpunkte	?limit=100&offset=0

Exportdaten:

Methode	Endpoint	Beschreibung
POST	/api/project/export-graph-data/	Diagramme exportieren
POST	/api/project/import-graph-data/	Diagramme importieren
GET	/api/project/graph-data-stats/	Datenstatistik

## 16. Alarme

### Für Benutzer

Zentrales Protokoll aller Alarmereignisse im System.

Tabellenelemente:

Spalte	Beschreibung
Zeit	Datum und Uhrzeit des Auftretens
Severity	Error (rot) / Warning (gelb)
Meldung	Beschreibung
Gerät	Quelle
Status	Active / Acknowledged / Resolved
Aktionen	Steuerschaltflächen

Filter:

Filter	Beschreibung
Nach Severity	Warning / Error / All
Nach Status	Active / Acknowledged / Resolved / All
Nach Gerät	Geräteauswahl
Suche	Textsuche

Schaltflächen:

Schaltfläche	Beschreibung
Acknowledge	Quittieren (zur Kenntnis genommen)
Resolve	Beheben (Problem behoben)
Resolve All	Alle Alarme beheben
Delete	Eintrag löschen
Refresh	Liste aktualisieren

**S SCADA** Alarmer & Ereignisse

Serhii Kudlai  
Администратор

<input type="checkbox"/>	Auftrittszeitpunkt	Gerät	Unfalltyp	Beschreibung	Schweregrad	Status	Aktionen
<input type="checkbox"/>	8.4.2026, 15:28:21	P-2	Motor P-2 Alarm	Motor P-2: läuft, aber keine Rückmeldung (automatisch Modu...	Hoch	Gelöst	
<input type="checkbox"/>	8.4.2026, 14:25:56	P-6	Motor P-6 Alarm	Motor P-6: läuft, aber keine Rückmeldung (automatisch Modu...	Hoch	Gelöst	

SCADA System v1.0.0

Alarms Page

Alarmseite: Tabelle mit farbcodierter Severity (roter Streifen links — Error, gelber — Warning). Filter oben. Schaltflächen Acknowledge/Resolve pro Zeile.

## Technische Dokumentation

Alarm-API: Siehe Abschnitt 10.

WebSocket: Verbindung ws/accidents — neue Alarmer und Statusaktualisierungen in Echtzeit.

## 17. Mathematische Funktionen

### Für Benutzer

Mathematische Funktionen erlauben berechnete Werte auf Basis von Gerätewerten und Rezeptvariablen.

Bildschirmelemente:

Element	Beschreibung
Funktionsliste	Tabelle/Karten der Funktionen
Schaltfläche «+»	Funktion erstellen
Suche	Nach Name

Ausdrucks-Editor (blockbasiert):

Der Ausdruck besteht aus sequenziell eingefügten Blöcken:

Blocktyp	Beschreibung
Zahl	Numerische Konstante
Gerät	Aktueller Gerätewert
Rezeptvariable	Variablenwert
Operator	+, -, ×, ÷, %, ^
Klammern	( und ) zur Gruppierung
Funktion	sqrt, log10, round, floor, ceil

Schaltflächen:

Schaltfläche	Beschreibung
Block hinzufügen	Neuen Block ergänzen
Block entfernen	Block aus dem Ausdruck löschen
Prüfen	Syntaxvalidierung
Test	Mit aktuellen Werten berechnen
Speichern	Funktion speichern

ID	Name	Ausdruck	Ergebnis	Einheiten	Status	Letzte Berechnung	Aktionen
func_014	Alarm PT-100-5 Cold	Pasteurisorator - PT100_5 Pasteurisorator - Cold_diff	72.0000	C	Aktiv	10.04.2026, 15:08:02	[Edit] [Delete] [Test]
func_016	RECIPE_70.PT100_1	Pasteurisorator - PT100_1	35.0000	C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_020	RECIPE_70.PT100_2	Pasteurisorator - PT100_2	62.0000	C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_019	RECIPE_70.PT100_3	Pasteurisorator - PT100_3	71.0000	C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_018	RECIPE_70.PT100_4	Pasteurisorator - PT100_4	72.0000	C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_017	RECIPE_70.PT100_5	Pasteurisorator - PT100_5	73.0000	C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_021	RECIPE_70.PT100_6	Pasteurisorator - PT100_6	65.0000	C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_022	RECIPE_70.PT100_7	Pasteurisorator - PT100_7	55.0000	C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_023	RECIPE_70.PT100_8	Pasteurisorator - PT100_8	34.0000	C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_015	pt100-4_cold	Pasteurisorator - PT100_4 Pasteurisorator - Cold_diff	71.0000	C	Aktiv	08.04.2026, 15:32:31	[Edit] [Delete] [Test]
func_026	test	Pasteurisorator - PT100_2 + 222	284.0000	-	Aktiv	14.04.2026, 13:50:50	[Edit] [Delete] [Test]
func_001	Авария PT100-1	Pasteurisorator - PT100_1 Pasteurisorator - Alarm_diff	90.0000	°C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_002	Авария PT100-2	Pasteurisorator - PT100_2 Pasteurisorator - Alarm_diff	117.0000	°C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]
func_003	Авария PT100-3	Pasteurisorator - PT100_3 Pasteurisorator - Alarm_diff	126.0000	°C	Aktiv	30.03.2026, 17:15:36	[Edit] [Delete] [Test]

Math Functions List

Funktionsliste: Tabelle mit Spalten Name, Expression, Result, Unit, Schaltflächen Edit/Delete/Test.

**S SCADA**

- Dashboard
- Schemata
- Alarmer
- Aktivitätsprotokolle
- Trends
- Programme
- SPSen
- IO-Punkte
- Geräte
- Funktionen
- Access
- Grafische Status
- Einstellungen
- Abmelden

**Mathematische Funktionen**

ID	Name	Ausdruck	Ergebnis	Einheiten	Status	Letzte Berechnung	Aktionen
func_014	Alarm PT-100-5 Cold	Pasteurisorator - PT100_5 Pasteurisorator - Cold_diff	72.0000	C	Aktiv	10.04.2026, 15:08:02	[Edit] [Delete] [Test]

**Funktion bearbeiten**

Name \*  
Alarm PT-100-5 Cold

Namensübersetzungen  
Name pro Sprache. Englisch wird als fallback verwendet.

Beschreibung

Mathematischer Ausdruck \*  
Pasteurisorator - PT100\_5 + Pasteurisorator - Cold\_diff

Maßeinheiten  
C

Aktualisierungsintervall (Sekunden)  
 Leer lassen für Berechnung auf Anfrage

Aktiv

Abbrechen Speichern

Math Function Editor

Ausdrucks-Editor: Blockreihe — [AI\_001] [x] [2.5] [+] [sqrt] [(] [AI\_002] [)] . Block-Zuordnung unten. Testergebnis rechts.

## Technische Dokumentation

### Math-Funktionen-API:

Methode	Endpoint	Beschreibung
GET	/api/math-functions/	Funktionsliste
POST	/api/math-functions/	Erstellen
PUT	/api/math-functions/{id}/	Aktualisieren
DELETE	/api/math-functions/{id}/	Löschen
POST	/api/math-functions/{id}/calculate/	Berechnen
POST	/api/math-functions/{id}/validate_expression/	Validierung
POST	/api/math-functions/{id}/test_expression/	Testlauf
POST	/api/math-functions/validate_all/	Alle validieren

### WebSocket: Math-Funktion-Update

Verbindung: ws:///ws/math-functions

```
{
  "type": "math_function_update",
  "function_id": 5,
  "name": "Total Flow",
  "expression": "ai_001 * 2.5 + sqrt(ai_002)",
  "result": 157.3,
  "units": "L/h",
  "last_calculated": "2026-04-07T12:00:00Z",
  "calculation_error": null,
  "is_active": true
}
```

# 18. E/A-Punkte

## Für Benutzer

E/A-Punkte verbinden physische SPS-Signale mit logischen SCADA-Geräten.

Tabs:

Tab	Beschreibung
Analog Inputs	Analogeingänge (4-20 mA, 0-10 V)
Analog Outputs	Analogausgänge
Digital Inputs	Digitaleingänge
Digital Outputs	Digitalausgänge

E/A-Punkt-Formular (IoPointDialog):

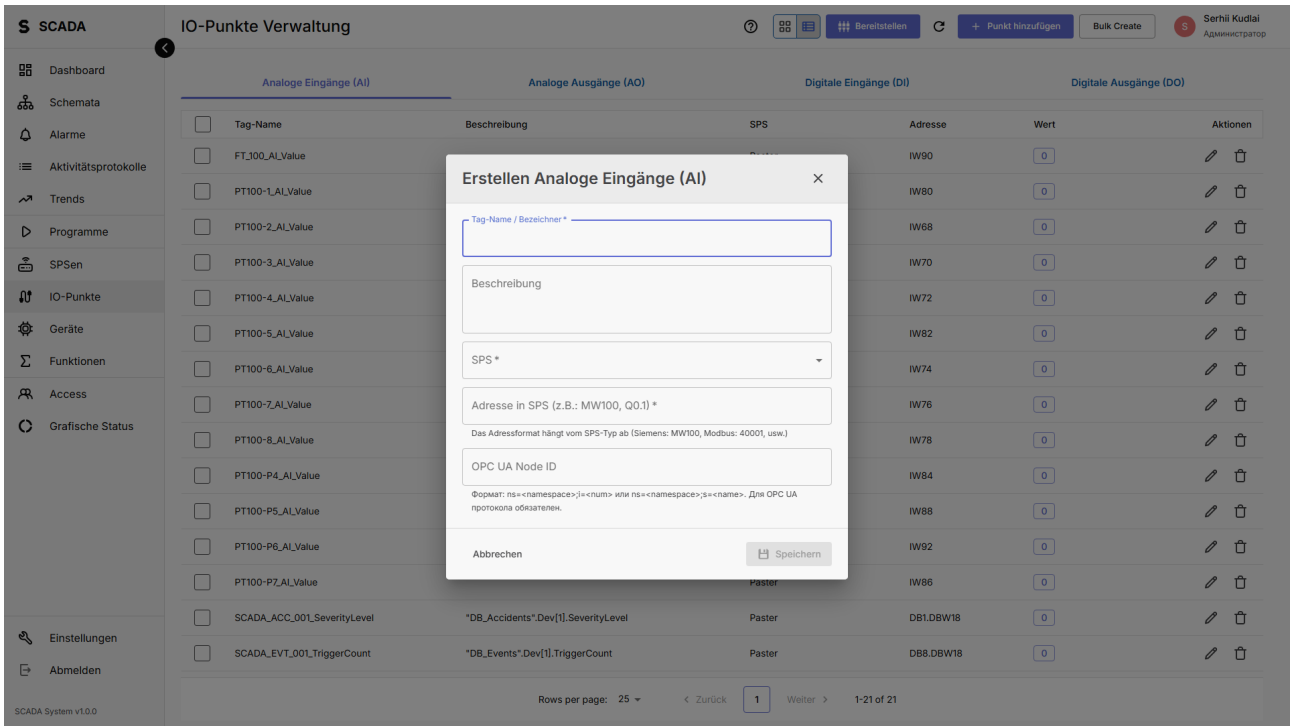
Feld	Beschreibung
SPS	Steuerung auswählen
Nummer/Adresse	Adresse in der SPS (IW0, QW0, IO.0, Q0.0)
Gerät	Bindung an logisches Gerät
Skalierung	Min/Max (für Analog)
Datentyp	INT, REAL, BOOL
Einheit	Einheit

The screenshot shows the 'IO-Punkte Verwaltung' interface in the SCADA system. The left sidebar contains navigation options like Dashboard, Schemata, Alarmer, and IO-Punkte. The main area displays a table of 'Analoge Eingänge (AI)' with columns for Tag-Name, Beschreibung, SPS, Adresse, Wert, and Aktionen. The table lists 17 entries, including PT100-1\_AI.Value through PT100-P7\_AI.Value, and two database-linked entries for severity and trigger counts. The bottom of the interface shows pagination controls for 21 rows.

Tag-Name	Beschreibung	SPS	Adresse	Wert	Aktionen
<input type="checkbox"/>	FT_100_AI.Value	Paster	IW90	0	
<input type="checkbox"/>	PT100-1_AI.Value	Paster	IW80	0	
<input type="checkbox"/>	PT100-2_AI.Value	Paster	IW68	0	
<input type="checkbox"/>	PT100-3_AI.Value	Paster	IW70	0	
<input type="checkbox"/>	PT100-4_AI.Value	Paster	IW72	0	
<input type="checkbox"/>	PT100-5_AI.Value	Paster	IW82	0	
<input type="checkbox"/>	PT100-6_AI.Value	Paster	IW74	0	
<input type="checkbox"/>	PT100-7_AI.Value	Paster	IW76	0	
<input type="checkbox"/>	PT100-8_AI.Value	Paster	IW78	0	
<input type="checkbox"/>	PT100-P4_AI.Value	Paster	IW84	0	
<input type="checkbox"/>	PT100-P5_AI.Value	Paster	IW88	0	
<input type="checkbox"/>	PT100-P6_AI.Value	Paster	IW92	0	
<input type="checkbox"/>	PT100-P7_AI.Value	Paster	IW86	0	
<input type="checkbox"/>	SCADA_ACC_001_Severity.Level	Paster	DB1.DBW18	0	
<input type="checkbox"/>	SCADA_EVT_001_TriggerCount	Paster	DB8.DBW18	0	

Io Points Manager

E/A-Punkte-Seite: 4 Tabs (AI, AO, DI, DO), Tabelle mit Spalten PLC, Address, Device, Scale, Unit, Schaltflächen Add/Edit/Delete.



Io Point Dialog

Dialog zum Anlegen eines E/A-Punkts: SPS-Dropdown, Adressfeld, Geräte-Dropdown, Min/Max Scale, Unit.

## Technische Dokumentation

E/A-Punkt-API:

Methode	Endpoint	Beschreibung
GET	/api/points/analog-inputs/	AI-Punkte
POST	/api/points/analog-inputs/	AI erstellen
GET	/api/points/analog-outputs/	AO-Punkte
POST	/api/points/analog-outputs/	AO erstellen
GET	/api/points/digital-inputs/	DI-Punkte
POST	/api/points/digital-inputs/	DI erstellen
GET	/api/points/digital-outputs/	DO-Punkte
POST	/api/points/digital-outputs/	DO erstellen

Für alle Typen: Standard-CRUD (GET, POST, PUT, PATCH, DELETE) + /validate/ und /test/.

# 19. E/A-Provisionierung

## Für Benutzer

Massenhaftes Anlegen von E/A-Punkten auf Basis der Gerätekonfiguration.

Bildschirmelemente:

Element	Beschreibung
SPS-Auswahl	Ziel-Steuerung
SPS-Typ	S7-1200 oder S7-1500
Adressen erhalten	Checkbox — bestehende Adressen nicht überschreiben
Entitätsliste	Checkboxen für jeden Typ (Motoren, Ventile, PID, AI, AO, DI, DO, COS, Zähler, Timer, Tanks, Sequenzen, Ereignisse, Störungen)
Vorschau	Vorschau des Ergebnisses
Provisioniere	Erzeugung der Punkte starten

n

Io Provision Page

Provisionierungsseite: SPS-Dropdown, Radio S7-1200/S7-1500, Entitätsliste mit Checkboxen, Schaltflächen Preview und Provision.

## Technische Dokumentation

Provisionierungs-API:

Methode	Endpoint	Beschreibung
POST	/api/plc-export/provision-io-points/preview/	Vorschau
POST	/api/plc-export/provision-io-points/	Ausführen
GET	/api/plc-export/provision-io-points/entities/	Verfügbare Entitäten

## 20. Grafische Zustände

### Für Benutzer

Grafische Zustände sind Icon-Schablonen zur Darstellung des Gerätestatus auf Schemata.

Elemente:

Element	Beschreibung
Schablonenliste	Karten/Tabelle
Schaltfläche «+»	Schablone erstellen

Erstell-Assistent:

- Schritt 1: Name, Beschreibung, Gerätetyp
- Schritt 2: Zustände und Icons
- Statuscode → Icon (URL oder Upload)
- Galerie verfügbarer Icons
- Hochladen eigener Icons

Schaltfläche	Beschreibung
Status hinzufügen	Neue Zeile «Code → Icon»
Status entfernen	Zeile löschen
Icon hochladen	Icon-Datei-Upload
Galerie	Aus zuvor hochgeladenen wählen
Zurück/Weiter	Assistenten-Navigation
Speichern	Schablone speichern

ID	Name	Beschreibung	Anzahl der Status	Erstellt	Aktualisiert	Aktionen
18	DI Green/Red	Keine Beschreibung	10	12.02.26 09:55:21	29.03.26 23:35:43	👁️ ✎️ 🗑️
17	VLV BF Status (NC vertical)	Keine Beschreibung	10	08.12.25 12:16:54	30.03.26 00:44:45	👁️ ✎️ 🗑️
14	DI Red/Green	Keine Beschreibung	10	27.10.25 19:21:49	29.03.26 23:53:18	👁️ ✎️ 🗑️
11	VLV RV Status (vertical)	Keine Beschreibung	10	27.10.25 19:07:54	29.03.26 23:21:34	👁️ ✎️ 🗑️
10	Valve RV Status (horizontal)	Keine Beschreibung	10	27.10.25 19:03:27	29.03.26 23:22:15	👁️ ✎️ 🗑️
9	VLV DS Status	Keine Beschreibung	12	27.10.25 18:48:51	29.03.26 23:21:11	👁️ ✎️ 🗑️
8	VLV Status BF (NO horizontal)	Keine Beschreibung	10	27.10.25 18:37:14	30.03.26 00:11:00	👁️ ✎️ 🗑️
6	MOT Status (up)	Keine Beschreibung	11	27.10.25 18:22:51	29.03.26 23:17:55	👁️ ✎️ 🗑️
5	MOT Status (right)	Keine Beschreibung	11	27.10.25 18:18:30	29.03.26 23:21:18	👁️ ✎️ 🗑️
4	MOT Status (left)	Keine Beschreibung	11	27.10.25 18:14:42	29.03.26 23:18:45	👁️ ✎️ 🗑️
3	MOT Status (down)	Keine Beschreibung	11	27.10.25 18:09:25	29.03.26 23:18:53	👁️ ✎️ 🗑️
2	Agitator Status	Keine Beschreibung	11	27.10.25 15:31:36	08.12.25 12:16:54	👁️ ✎️ 🗑️

Graphical Statuses List

Liste grafischer Zustände: Karten mit Icon-Vorschau, Schablonenname, Gerätetyp.

Graphical Status Wizard

Erstell-Assistent: links die Statusliste (Code + Icon-Vorschau), rechts die Icon-Galerie.

## Technische Dokumentation

Grafische-Zustände-API:

Methode	Endpoint	Beschreibung
GET	/api/graphical-statuses/	Schablonenliste
POST	/api/graphical-statuses/	Erstellen
PUT	/api/graphical-statuses/{id}/	Aktualisieren
DELETE	/api/graphical-statuses/{id}/	Löschen
POST	/api/graphical-statuses/upload-icon/	Icon hochladen (multipart)
GET	/api/graphical-statuses/list-icons/	Icon-Liste

## 21. Hardware-Konfigurator

### Für Benutzer

Visueller Editor der SPS-Hardwarekonfiguration für das Deployment in TIA Portal.

Tabs:

Tab	Beschreibung
Rack View	Visueller Rack-Editor mit Modulen
PROFINET Topology	Netzwerktopologie der Geräte
Address Table	Adresstabelle der E/A

Rack-Editor:

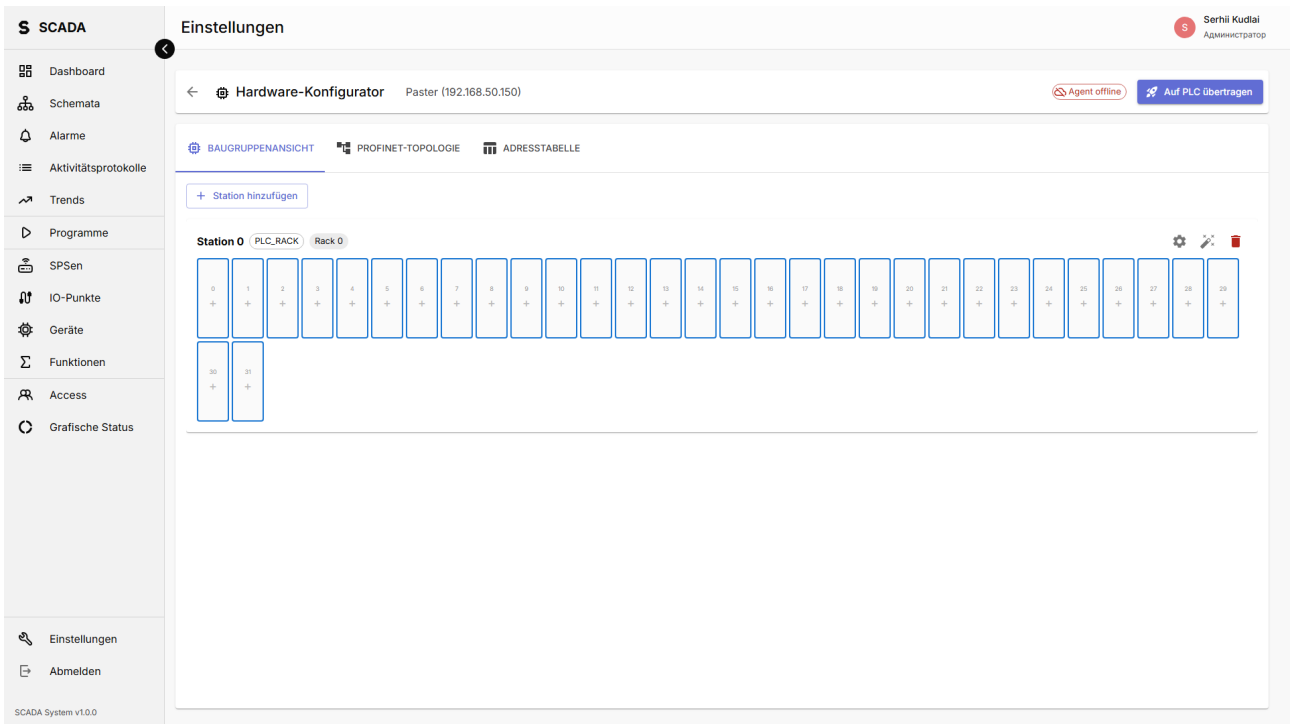
Element	Beschreibung
Rack-Slots	Visuelle Darstellung der Module im Rack
Modulkatalog	Seitenleiste mit verfügbaren Modulen (drag-drop)
Eigenschaften	Konfiguration des gewählten Moduls
Modul hinzufügen	Ziehen aus dem Katalog
Modul entfernen	Aus dem Slot entfernen

Schaltflächen der Werkzeugleiste:

Schaltfläche	Beschreibung
Deploy	Konfiguration in TIA Portal ausrollen
Download	Konfiguration herunterladen
Import GSD	GSD/GSDML-Datei importieren
TIA Agent Status	Verbindungsanzeige zum TIA Agent
Back	Zurück zur SPS-Liste

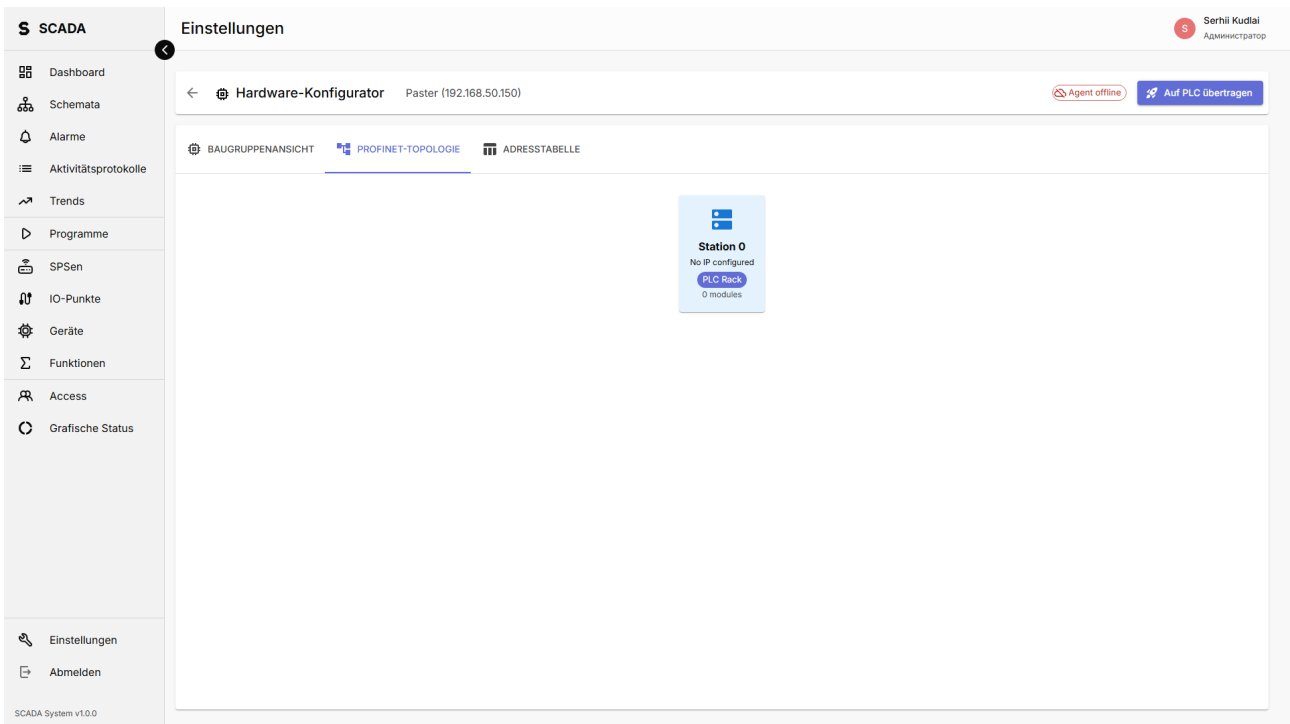
Dialoge:

Dialog	Beschreibung
TiaDeployDialog	Deployment: Zielsystem, Fortschritt, Status
GsdImportDialog	GSD-Import: Datei-Upload, Modul-Vorschau, Bestätigung
NetworkConfigDialog	Netzeinstellungen: IP, Maske, Gateway, PROFINET



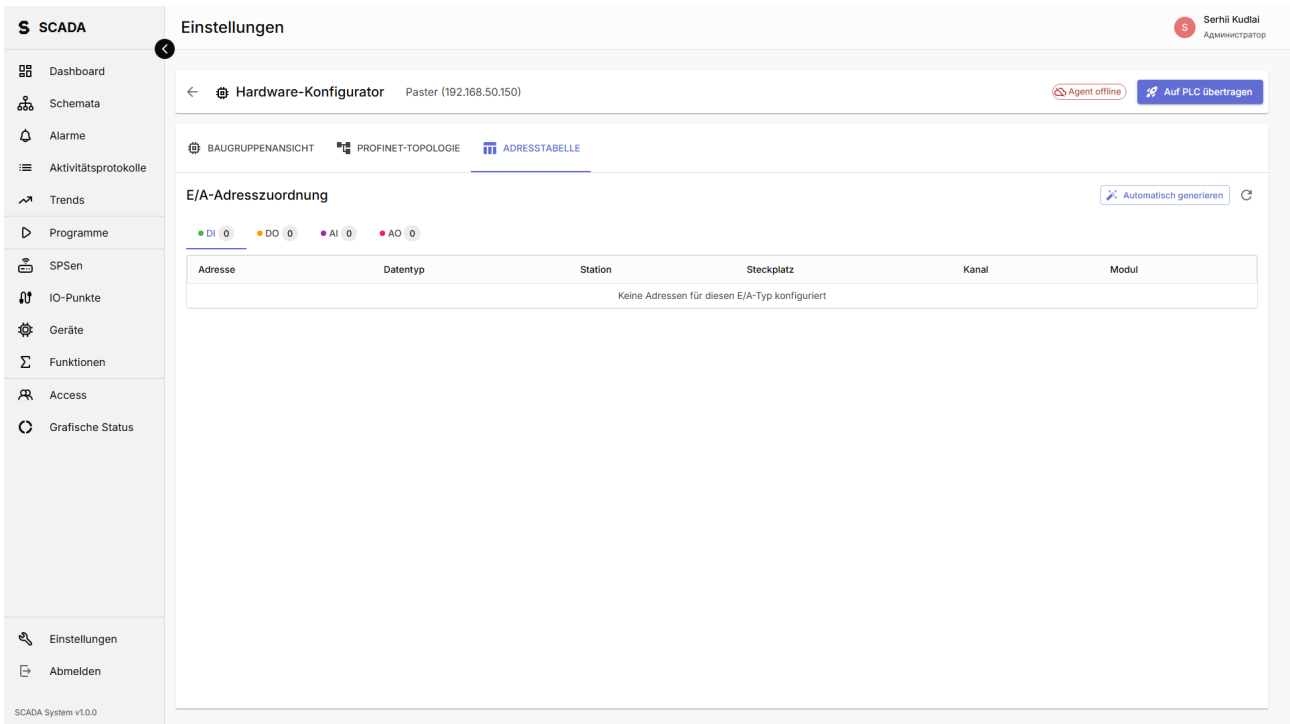
Hardware Configurator Rack

Rack-Editor: visuelles Rack mit Modulen in Slots, Modulkatalog links (ausklappbar), Eigenschaften rechts, Deploy in der Toolbar.



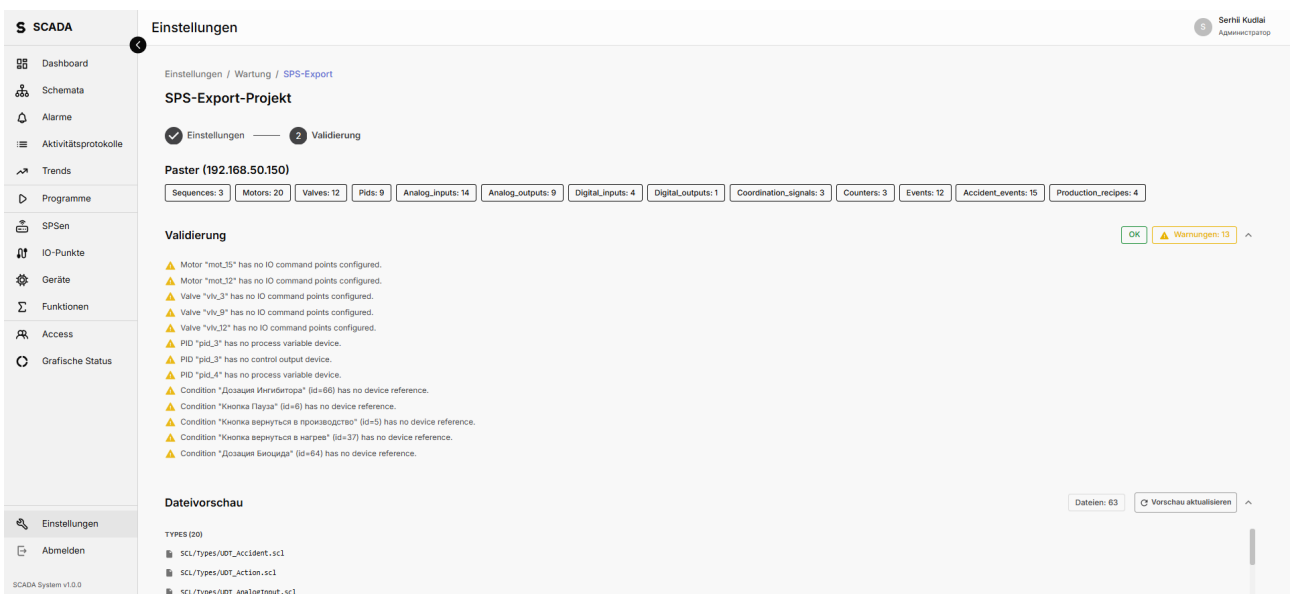
Hardware Configurator Profinet

PROFINET-Topologie: Netzwerkdiagramm mit Geräteknoten und Verbindungen.



Hardware Configurator Addresses

Adresstabelle: Spalten Module, Slot, Channel, Address, Type (I/O), Device.



Tia Deploy Dialog

Deploy-Dialog: TIA-Agent-Auswahl, Fortschrittsbalken, Operationslog, Schaltflächen Deploy/Cancel.

## Technische Dokumentation

Konfigurator-API:

Methode	Endpoint	Beschreibung
GET	/api/hw-catalog/	Modulkatalog
POST	/api/hw-catalog/import-gsd/	GSD-Import (multipart)
GET	/api/hw-stations/	Stationen
POST	/api/hw-stations/	Station erstellen
POST	/api/hw-stations/{id}/auto-assign-addresses/	Automatische Adressvergabe
GET	/api/hw-slots/	Slots
POST	/api/hw-slots/	Slot erstellen
POST	/api/hw-slots/bulk-update/	Massenaktualisierung
GET	/api/hw-network/	Netzkonfiguration
POST	/api/tia/deploy/	In TIA Portal ausrollen
GET	/api/tia-deploy-jobs/{id}/progress/	Deploy-Fortschritt
GET	/api/tia-agent-config/	TIA-Agent-Konfigurationen
GET	/api/tia-agent-config/{id}/check-connection/	Verbindung testen

## 22. Benutzerverwaltung

### Für Benutzer

Erstellen, Bearbeiten und Löschen von Benutzerkonten.

Benutzertabelle:

Spalte	Beschreibung
Benutzername	Login
Vollständiger Name	Vor- und Nachname
E-Mail	E-Mail-Adresse
Rollen	Zugewiesene Rollen
Letzte Anmeldung	Datum und Uhrzeit
Status	Active / Inactive
Aktionen	Edit / Delete

Benutzerformular:

Feld	Beschreibung
Username	Eindeutiger Login
E-Mail	E-Mail-Adresse
Passwort	Passwort (beim Anlegen)
Bestätigung	Wiederholte Passwort-Eingabe
Vor-/Nachname	Echter Name
Avatar	Profilbild hochladen
Rollen	Rollen-Multiselect
Status	Active/Inactive-Umschalter

Schaltfläche	Beschreibung
Speichern	Änderungen speichern
Abbrechen	Zurück ohne Speichern
Löschen	Konto löschen (nur beim Bearbeiten)
Passwort ändern	Benutzerpasswort ändern

The screenshot shows the 'Access' management interface. On the left is a navigation menu with options like Dashboard, Schemata, Alarme, etc. The main area is titled 'Access' and contains a search bar and a table of users. The table has the following data:

ID	Benutzername	E-Mail	Vollständiger Name	Rollen	Letzte Anmeldung	Status	Aktionen
2	Operator	oper@oper.com	Anton Partika	Оператор	Nie	Aktiv	[edit] [delete]
1	admin	admin@example.com	Serhii Kudlai	Администратор	11.5.2026, 14:44:44	Aktiv	[edit] [delete]

Users List

Benutzertabelle mit Spalten, Avataren, farbigen Rollen-Badges, Statusanzeige.

The screenshot shows the 'Benutzer erstellen' (Create User) form. It includes the following fields and controls:

- Avatar:** A placeholder for the user's avatar with an 'Upload Image' button.
- Benutzername \*:** Required text input field.
- E-Mail \*:** Required text input field.
- Vorname:** Text input field for the first name.
- Nachname:** Text input field for the last name.
- Passwort \*:** Required password input field with a visibility toggle.
- Passwort bestätigen \*:** Required confirmation password input field with a visibility toggle.
- Rollen \*:** A dropdown menu for selecting the user's role, with the text 'Rollen auswählen (optional)'. The current selection is 'Администратор'.
- Aktiv:** A toggle switch currently set to 'on'.
- Abbrechen:** A button to cancel the creation process.
- Benutzer erstellen:** A blue button to create the user.

User Form

Erstell-/Bearbeitungsformular: Felder username, email, password, role selector, avatar upload, status toggle.

## Technische Dokumentation

---

### Benutzer-API:

Methode	Endpoint	Beschreibung
GET	/api/users/	Benutzerliste
POST	/api/users/	Erstellen
GET	/api/users/{id}/	Abrufen
PUT	/api/users/{id}/	Aktualisieren
DELETE	/api/users/{id}/	Löschen
GET	/api/users/me/	Aktueller Benutzer
POST	/api/users/{id}/set_password/	Passwort ändern
POST	/api/users/{id}/upload-avatar/	Avatar hochladen (multipart)
GET	/api/users/{id}/roles/	Benutzerrollen
POST	/api/users/{id}/roles/	Rolle zuweisen
DELETE	/api/users/{id}/roles/	Rolle entziehen

## 23. Rollenverwaltung

### Für Benutzer

Rollenbasiertes Zugriffsmodell. Jede Rolle definiert einen Satz von Berechtigungen.

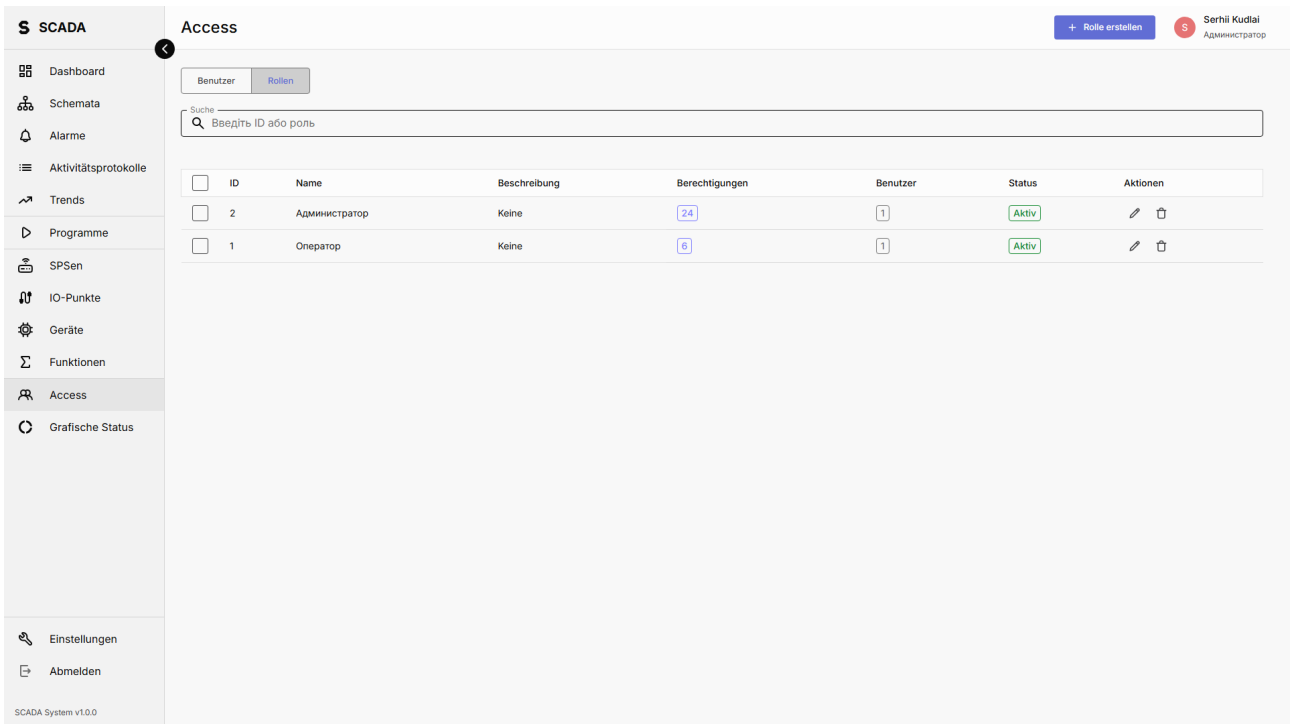
Rollentabelle:

Spalte	Beschreibung
Name	Rollenname
Beschreibung	Freitext
Benutzer	Anzahl Benutzer mit dieser Rolle
Aktionen	Edit / Delete

Berechtigungsmatrix (im Rollenformular):

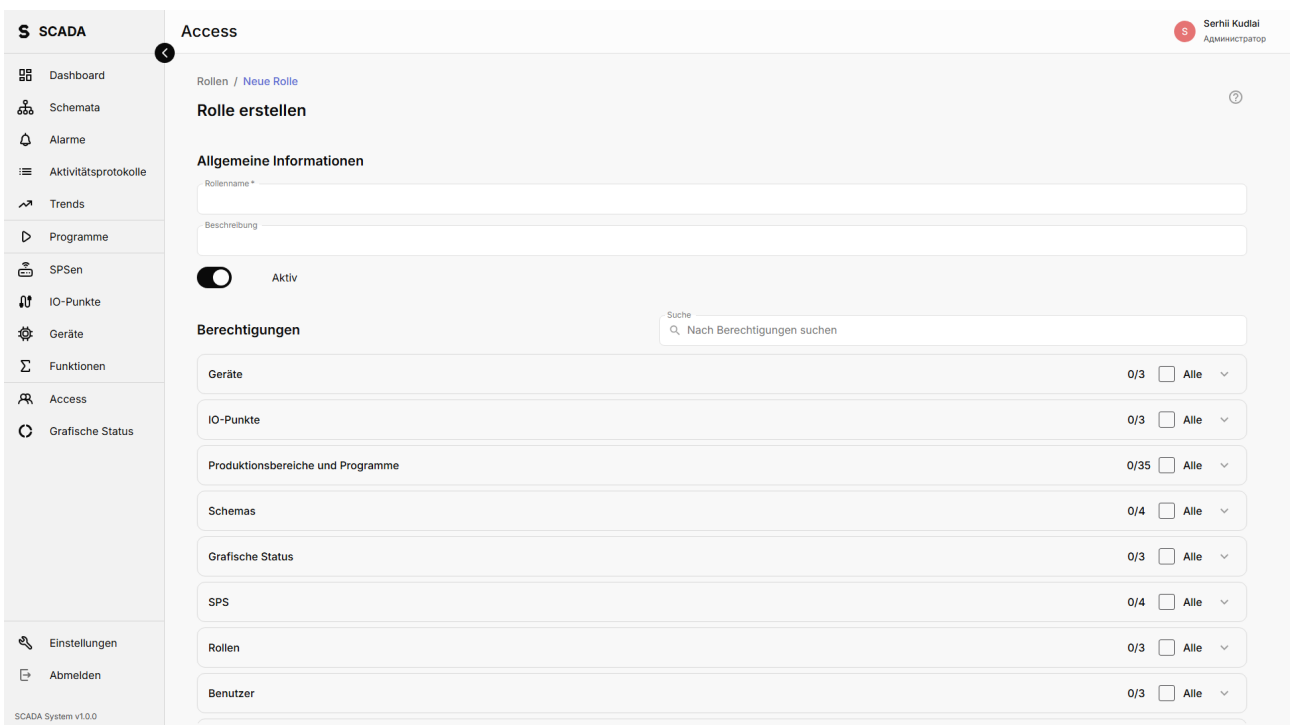
Kategorie	Berechtigungen
Geräte	Erstellen, Anzeigen, Bearbeiten, Löschen
SPS	Erstellen, Anzeigen, Bearbeiten, Löschen, Hardware Config
Sequenzen	Erstellen, Anzeigen, Bearbeiten, Löschen, Duplizieren
Schemata	Erstellen, Anzeigen, Bearbeiten, Löschen
E/A-Punkte	Erstellen, Anzeigen, Bearbeiten, Löschen
Benutzer	Erstellen, Anzeigen, Bearbeiten, Löschen
Rollen	Erstellen, Anzeigen, Bearbeiten, Löschen
Störungen	Quittieren, Beheben, Löschen
Trends	Anzeigen, Exportieren, Daten löschen
Grafische Zustände	Erstellen, Anzeigen, Bearbeiten, Löschen
Verwaltung	Verbindungen, Lizenz, Updates

Schaltfläche	Beschreibung
Select All	Alle Berechtigungen auswählen
Deselect All	Alle abwählen
Speichern	Rolle speichern



Roles List

Rollenliste: Tabelle mit Name, Beschreibung, Benutzeranzahl, Edit/Delete-Schaltflächen.



Role Form Permissions

Rollenformular: Name, Beschreibung, Checkbox-Matrix pro Kategorie (Devices, PLCs, Sequences, Schemas...). Schaltflächen Select All/Deselect All.

## Technische Dokumentation

---

Rollen-API:

Methode	Endpoint	Beschreibung
GET	/api/roles/	Rollenliste
POST	/api/roles/	Erstellen
GET	/api/roles/{id}/	Abrufen
PATCH	/api/roles/{id}/	Aktualisieren
DELETE	/api/roles/{id}/	Löschen
GET	/api/roles/{id}/permissions/	Rollen-Berechtigungen
GET	/api/permissions/categories/	Berechtigungskategorien
GET	/api/permissions/actions/	Aktionen

## 24. Aktivitätsprotokoll

### Für Benutzer

---

Audit aller Benutzeraktionen im System.

Bildschirmelemente:

Spalte	Beschreibung
Zeit	Datum und Uhrzeit der Aktion
Benutzer	Wer hat die Aktion ausgeführt
Aktion	Operationstyp (create, update, delete, login...)
Beschreibung	Details der Aktion
IP-Adresse	Von welcher Adresse

Filter:

Filter	Beschreibung
Nach Benutzer	Benutzerauswahl
Nach Aktion	Operationstyp
Nach Datum	Datumsbereich
Suche	Textsuche

Statistik:

- Gesamtanzahl der Aktionen
- Übersicht nach Benutzern
- Übersicht nach Aktionstypen

**S** SCADA
Benutzeraktivitätsprotokolle
Alle Protokolle löschen
Serhii Kudlai  
Администратор

- Dashboard
- Schemata
- Alarme
- Aktivitätsprotokolle**
- Trends
- Programme
- SPSen
- IO-Punkte
- Geräte
- Funktionen
- Access
- Grafische Status
- Einstellungen
- Abmelden

**Gesamtaktionen (7 Tage)**  
47

Поиск:

**Häufigste Aktionen**

Gerät erstellt: 11    Gerät gelöscht: 1    Produktionsbereich erstellt: 1    Produktionsbereich aktualisiert: 10    Produktionsbereich gestartet: 2

Користувач: Alle

Аktionstyp: Alle

Zeitraum: 11.05.2025 - 11.05.2026

Datum/Zeit	Benutzer	Aktionstyp	Beschreibung	Objekt	IP-Adresse
11.05.2026 12:25:08	admin	Produktionsbereich aktualisiert	Produktionsbereich Pasteurisator aktualisiert	Produktionsbereich: Pasteurisator	172.18.0.1
11.05.2026 11:49:20	admin	Produktionsbereich gestoppt	Produktionsbereich Pasteurization HTST gestoppt	Produktionsbereich: Pasteurization HTST	-
11.05.2026 11:49:00	admin	Produktionsbereich gestoppt	Produktionsbereich Pasteurisator gestoppt	Produktionsbereich: Pasteurisator	-
11.05.2026 11:47:34	admin	Produktionsbereich gestartet	Produktionsbereich Pasteurisator gestartet	Produktionsbereich: Pasteurisator	-
08.05.2026 14:36:23	admin	Programm aktualisiert	Програма Пастеризатор актуалізовано	Програма: Пастеризатор	172.18.0.1
08.05.2026 14:36:15	admin	Produktionsbereich aktualisiert	Produktionsbereich Pasteurization HTST aktualisiert	Produktionsbereich: Pasteurization HTST	172.18.0.1
08.05.2026 14:35:50	admin	Schrittwechsel	Schritt auf 2 für Produktionsbereich Pasteurization HTST geändert	Produktionsbereich: Pasteurization HTST	172.18.0.1
08.05.2026 14:35:44	admin	Produktionsbereich gestartet	Produktionsbereich Pasteurization HTST gestartet	Produktionsbereich: Pasteurization HTST	-
08.05.2026 14:35:43	admin	Produktionsbereich aktualisiert	Produktionsbereich Pasteurization HTST aktualisiert	Produktionsbereich: Pasteurization HTST	172.18.0.1
08.05.2026 14:35:23	admin	Programm aktualisiert	Програма Пастеризатор актуалізовано	Програма: Пастеризатор	172.18.0.1
08.05.2026 14:33:51	admin	Programm aktualisiert	Програма Пастеризатор актуалізовано	Програма: Пастеризатор	172.18.0.1
08.05.2026 14:33:20	admin	Produktionsbereich aktualisiert	Produktionsbereich Pasteurization HTST aktualisiert	Produktionsbereich: Pasteurization HTST	172.18.0.1
08.05.2026 14:33:11	admin	Produktionsbereich aktualisiert	Produktionsbereich Pasteurisator aktualisiert	Produktionsbereich: Pasteurisator	172.18.0.1
08.05.2026 14:27:58	admin	Produktionsbereich aktualisiert	Produktionsbereich {{(sequence_name)}} aktualisiert	SequenceStep: Удержание	172.18.0.1
08.05.2026 14:25:58	admin	Programm aktualisiert	Програма Пастеризатор актуалізовано	Програма: Пастеризатор	172.18.0.1
08.05.2026 14:25:46	admin	Produktionsbereich aktualisiert	Produktionsbereich Pasteurization HTST aktualisiert	Produktionsbereich: Pasteurization HTST	172.18.0.1
08.05.2026 14:25:26	admin	Schema aktualisiert	Schema Пастеризатор HTST актуалізовано	Schema: Пастеризатор HTST	172.18.0.1

Rows per page: 25    < Zurück    1    2    3    4    5    Weiter    >    1-25 of 106

Activity Logs

Aktivitätsprotokoll: Tabelle mit Filtern oben, Statistik rechts (Diagramme nach Aktionstyp und Benutzer). Paginierung unten.

## Technische Dokumentation

### Aktivitätsprotokoll-API:

Methode	Endpoint	Beschreibung	Parameter
GET	/api/user-activity-logs/	Log-Liste	?user_id=X&action_type=X&limit=100&offset=0
GET	/api/user-activity-logs/recent/	Letzte 24 h	—
GET	/api/user-activity-logs/statistics/	Statistik	—
DELET	/api/user-activity-logs/clear/	Löschen (admin)	—

E

WebSocket: Protokoll in Echtzeit

Verbindung: ws:///ws/activity-log

```
{
  "type": "activity_log_entry",
  "id": 1234,
  "action_type": "device_update",
  "description": "Updated motor MOT_001 mode to manual",
  "user": {"id": 1, "username": "admin"},
  "content_object": {"type": "motor", "id": "mot_001"},
  "metadata": {},
  "created_at": "2026-04-07T12:00:00Z"
}
```

## 25. Einstellungen

### Für Benutzer

Die Einstellungsseite enthält mehrere Tabs:

#### Tab «Allgemein»

Parameter	Beschreibung
Design	Farbschema (Blue, Green, Orange, Pink, Purple, Red)
Dunkler Modus	Light/Dark-Umschalter
Sprache	Oberflächensprache (EN, UK, RU, DE, FR, PL, AR...)
Designversion	V1 / V2
KI-Assistent	KI-Helfer aktivieren/deaktivieren

#### Tab «Benachrichtigungen»

Parameter	Beschreibung
E-Mail-Benachrichtigungen	Ein/Aus
Push-Benachrichtigungen	Ein/Aus
SMS-Benachrichtigungen	Ein/Aus
Severity-Filter	Minimaler Schweregrad für Benachrichtigungen

#### Tab «Verbindungen»

Element	Beschreibung
TIA-Agent-Liste	TIA-Portal-Verbindungskonfigurationen
Hinzufügen	Neuer TIA Agent (Host, Port)
Test	Verbindungstest
Statusanzeige	Grün/Rot

#### Tab «Lizenz»

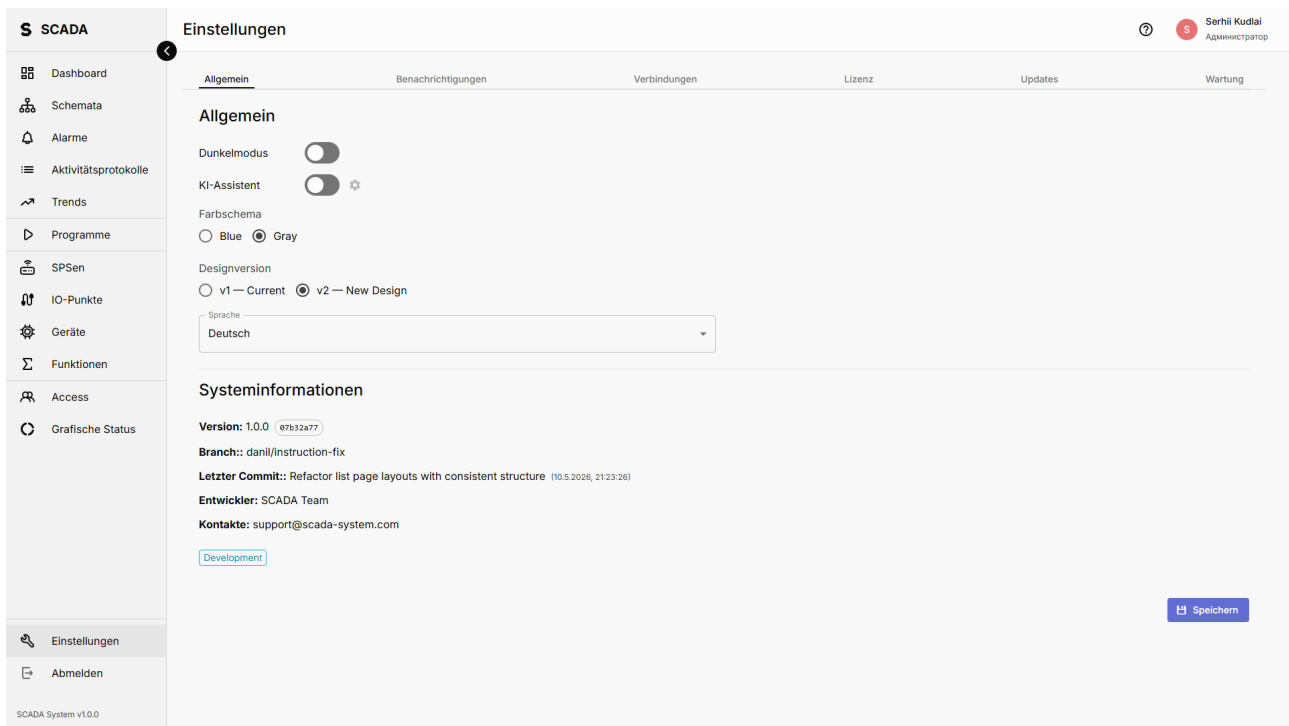
Feld	Beschreibung
Status	Valid / Invalid / Trial
Typ	Lifetime / Trial / Subscription
Tage übrig	Countdown
Firma	Firmenname
Ablaufdatum	Lizenzablauf
Installations-ID	Eindeutiger Bezeichner

#### Tab «Updates»

Element	Beschreibung
Aktuelle Version	Versionsnummer
Nach Updates suchen	Schaltfläche
Update herunterladen	Schaltfläche
Fortschritt	Fortschrittsanzeige
Auto-Update	Umschalter

### Tab «Wartung»

Element	Beschreibung
Datenbank	DB-Wartungsaktionen
Cache	Cache leeren
Logs	Log-Verwaltung
Systemzustand	Health-Checks
Backup/Restore	Sicherung und Wiederherstellung
Festplatte bereinigen	Temporäre Dateien löschen
Smoke-Test	Funktionstest



Settings General

Tab General: Dark-Mode-Umschalter, Color Picker (6 Farben), Sprach-Dropdown, Umschalter Design V1/V2, Toggle AI Assistant.

The screenshot shows the 'Verbindungen' (Connections) tab in the SCADA settings. It is divided into two main sections: 'API- und MQTT-Einstellungen' and 'TIA Portal Agent'.  
 - **API- und MQTT-Einstellungen:** Includes fields for 'API URL' (http://localhost:8000/api) and 'MQTT Broker' (mqtt://localhost:1883).  
 - **TIA Portal Agent:** Includes fields for 'Agentname' (tia-agent), 'Agent URL' (http://192.168.0.10:5100), 'API-Schlüssel (optional)', 'TIA Portal Version' (V17), and 'Projektverzeichnis auf dem Agent' (C:\TIA\_Projects\SCADA). It also shows a 'Backend-Rückruf-URL' (http://192.168.0.1:8000).  
 - **Status:** An 'Aktiv' toggle switch is turned on. Below it are buttons for 'Speichern', 'Verbindung testen', and a trash icon. A 'Letzter Heartbeat' timestamp is shown as 29.3.2026, 14:25:46.  
 - **Footer:** A 'Speichern' button is located at the bottom right of the main content area.

Settings Connections

Tab Connections: TIA-Agent-Liste mit Host/Port, Schaltflächen Add/Edit/Delete/Test, Statusanzeigen.

The screenshot shows the 'Lizenz' (License) tab in the SCADA settings. It contains the following information:  
 - **Lizenzstatus:** Shows 'Lizenztyp: Lifetime License' and 'Status: Aktiv' (with a green badge). It also indicates 'Gültig bis: Lifetime' and 'Verbleibende Tage: Lifetime'.  
 - **Installations-ID:** 48fa1f6b992e212a79a288cc890aac6.  
 - **Neue Lizenz aktivieren:** A form with a 'Lizenzschlüssel eingeben' field and an 'Aktivieren' button.  
 - **Systeminformationen:** Lists 'Version: 1.0.0' (with a commit hash), 'Branch: danil/instruction-fix', 'Letzter Commit: Refactor list page layouts with consistent structure (10.5.2026, 21:23:26)', 'Entwickler: SCADA Team', and 'Kontakte: support@scada-system.com'.  
 - **Footer:** A 'Development' badge is visible at the bottom.

Settings License

Tab License: Lizenzstatus (grünes Badge), Typ, Tage übrig, Firma, Ablaufdatum.

The screenshot shows the 'Einstellungen' (Settings) page in the SCADA system, specifically the 'Updates' tab. The left sidebar contains navigation options like 'Dashboard', 'Schemata', 'Alarmer', etc. The main content area is titled 'System-Updates' and displays the following information:

- Aktuelle Version:** 07b32a77 (development)
- Branch:** danil/instruction-fix
- Letzter Commit:** Refactor list page layouts with consistent struct... (10.5.2026, 21:23:26)
- Letzte Prüfung:** Nie
- Hostname:** e1564ff4f189
- Betriebssystem:** Debian GNU/Linux 13 (trixie)
- Kernel:** 5.15.167.4-microsoft-standard-WSL2
- Architektur:** x86\_64

Below this information, there is a status message: 'Status file not found. Update check has not run yet.' with two buttons: 'Nach Updates suchen' and 'Offline-Update'. Further down, there are sections for 'Update-Protokoll' (showing 'Kein Update-Protokoll verfügbar.'), 'Systemzustand', 'Datenbank-Sicherungen', and 'Update-Branch'. At the bottom, there are two toggle switches: 'Automatische Update-Prüfungen' (deactivated) and 'Automatische Update-Anwendung' (deactivated).

Settings Updates

Tab Updates: aktuelle Version, Check-for-Updates-Button, Fortschrittsbalken, Update-Logs.

The screenshot shows the 'Einstellungen' (Settings) page in the SCADA system, specifically the 'Wartung' (Maintenance) tab. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Projekt Export / Import' and contains buttons for 'Projekt exportieren' and 'Projekt importieren'. Below this is the 'Dienstprogramme' section with buttons for 'IDs normalisieren', 'Namen automatisch übersetzen', and 'SPS-Export'. The 'Datenbank' section has buttons for 'DB Dump' and 'DB wiederherstellen'. A large red warning box is present, stating: 'Aktionen in diesem Abschnitt können zu irreversiblen Datenverlust führen.' Below the warning is a button labeled 'Datenbank leeren'.

Settings Maintenance

Tab Maintenance: Schaltflächen Backup DB, Restore DB, Clear Cache, Disk Cleanup, Smoke Test mit Beschreibungen.

## Technische Dokumentation

### Benutzer-Einstellungen-API:

Methode	Endpoint	Beschreibung
GET	/api/user/language/	Sprache abrufen
POST	/api/user/language/	Sprache setzen
GET	/api/user/dark-mode/	Dark Mode abrufen
POST	/api/user/dark-mode/	Dark Mode setzen

### Lizenz-API:

Methode	Endpoint	Beschreibung
GET	/api/license/status/	Lizenzstatus
POST	/api/license/activate/	Aktivierung
POST	/api/license/generate/	Schlüssel erzeugen

### Update-API:

Methode	Endpoint	Beschreibung
GET	/api/system/update-status/	Update-Status
POST	/api/system/update-check/	Nach Updates suchen
POST	/api/system/update-apply/	Update anwenden
GET	/api/system/update-progress/	Fortschritt
POST	/api/system/update-cancel/	Abbrechen
POST	/api/system/update-rollback/	Rollback
GET	/api/system/update-history/	Update-Historie
GET	/api/system/update-backups/	Backups
POST	/api/system/update-backups/{filename}/restore/	Aus Backup wiederherstellen
POST	/api/system/disk-cleanup/	Festplatte bereinigen
GET	/api/system/smoke-test/	Smoke-Test
GET	/api/system/diagnostics/	Diagnose (Download)

### WebSocket: Update-Benachrichtigungen

```
{
  "type": "system_update",
  "event": "update_progress",
  "progress_percent": 45,
  "current_step": "Applying migrations...",
  "status": "in_progress"
}
```

## 26. KI-Assistent

### Für Benutzer

Integrierter KI-Helfer zur Bedienung von SCADA per natürlicher Sprache.

Elemente:

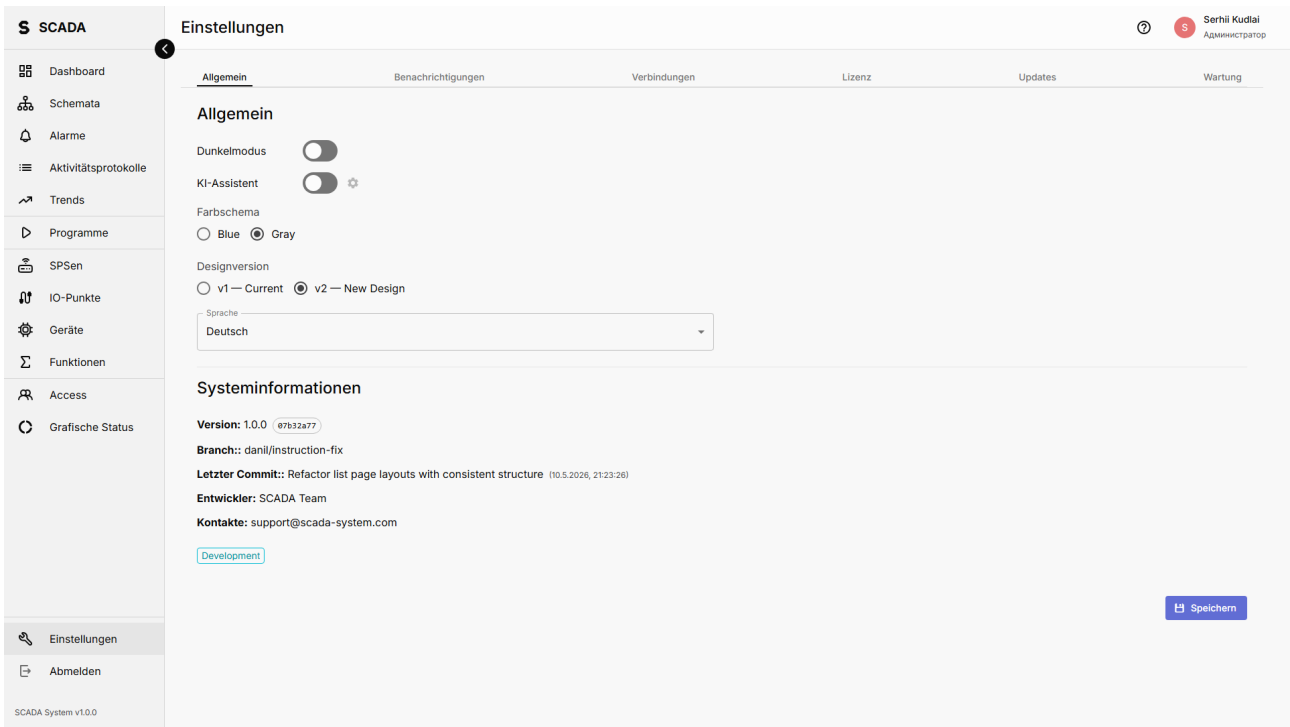
Element	Beschreibung
KI-Button (FAB)	Schwebende Schaltfläche unten rechts
Chat-Panel	Ausklapp-/ziehbares Panel
Sessions-Liste	Seitenleiste mit Gesprächshistorie
Eingabefeld	Textfeld für Nachrichten
Senden	Nachricht senden
Anhängen	Bild hochladen (P&ID-Zeichnung)
Einstellungen	KI-Konfiguration (Modell, API-Key)

KI-Funktionen:

Funktion	Beschreibung
Gerätesteuerung	«Starte Motor MOT_001»
Informationsabfrage	«Welche Geräte sind gerade in Störung?»
Elemente anlegen	«Erstelle einen neuen Analogeingang AI_010»
P&ID-Analyse	Bild hochladen → Elemente erkennen → Platzierungs-Vorschlag
Hilfe	Fragen zum System beantworten

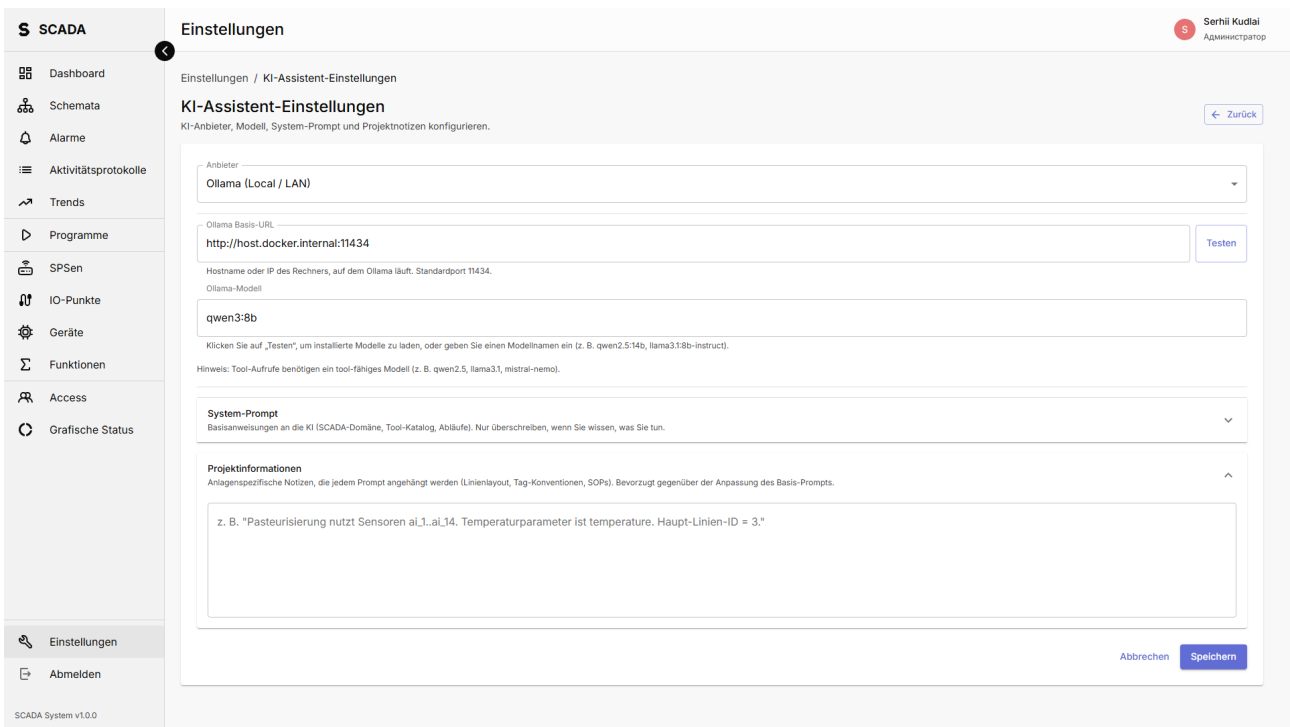
KI-Einstellungen (AiSettingsDialog):

Feld	Beschreibung
API-Key	OpenRouter-Schlüssel (AES/Fernet-verschlüsselt)
Modell	LLM-Modellauswahl
Temperatur	Antwort-Kreativität (0-1)
Max tokens	Maximale Antwortlänge



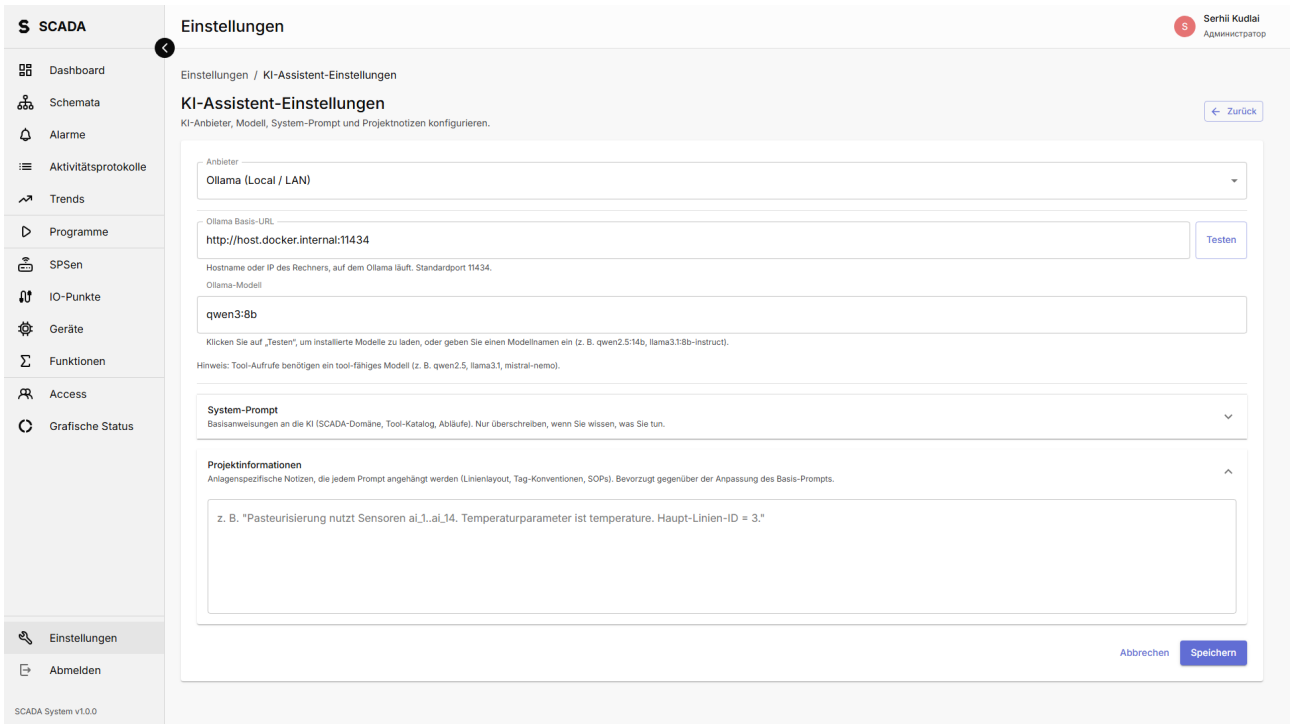
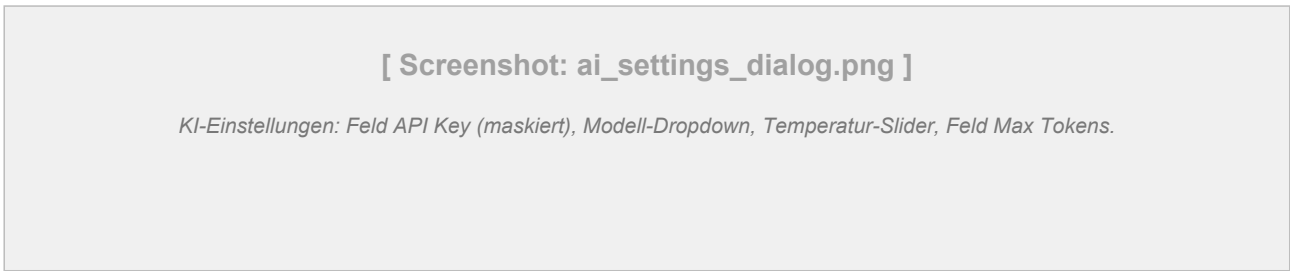
Ai Chat Button

Schwebender KI-Button unten rechts: rund, mit Roboter-/Chat-Symbol.



Ai Chat Panel

KI-Chat-Panel: links Sessions-Liste, rechts der Dialog (Nutzer- und KI-Nachrichten), unten das Eingabefeld mit Send- und Attach-Buttons.



P&ID-Analyse: hochgeladenes Bild links, rechts erkannte Elemente (Geräte, Rohre) als strukturierter Vorschlag, Schaltfläche «Anwenden».

## Technische Dokumentation

KI-API (FastAPI-Microservice, Base URL: `/ai-api/v1`):

Methode	Endpoint	Beschreibung
POST	/sessions	Session erstellen
GET	/sessions	Session-Liste
DELETE	/sessions/{id}	Session löschen
PATCH	/sessions/{id}	Umbenennen
GET	/sessions/{id}/messages	Session-Nachrichten
POST	/chat/{id}/messages	Nachricht senden (SSE-Stream)
POST	/images/analyze	Bildanalyse (multipart)
POST	/images/apply	Vorschlag anwenden
GET	/settings	Einstellungen abrufen
PATCH	/settings	Einstellungen aktualisieren
GET	/health	Health-Check

SSE-Events (bei stream=true):

Event	Beschreibung
token	Antwort-Text-Token (Streaming)
tool_call	KI ruft ein Tool auf (devicetools, schematools, query_tools)
tool_result	Tool-Ergebnis
error	Fehler
done	Antwort abgeschlossen

### SSE-Format:

```
event: token
data: {"content": "Motor "}

event: token
data: {"content": "MOT_001 "}

event: tool_call
data: {"name": "get_device_status", "arguments": {"device_id": "mot_001"}}

event: tool_result
data: {"result": {"status": "running", "value": 1}}

event: token
data: {"content": "läuft aktuell."}

event: done
data: {}
```

## 27. SPS-Projekt-Export

### Für Benutzer

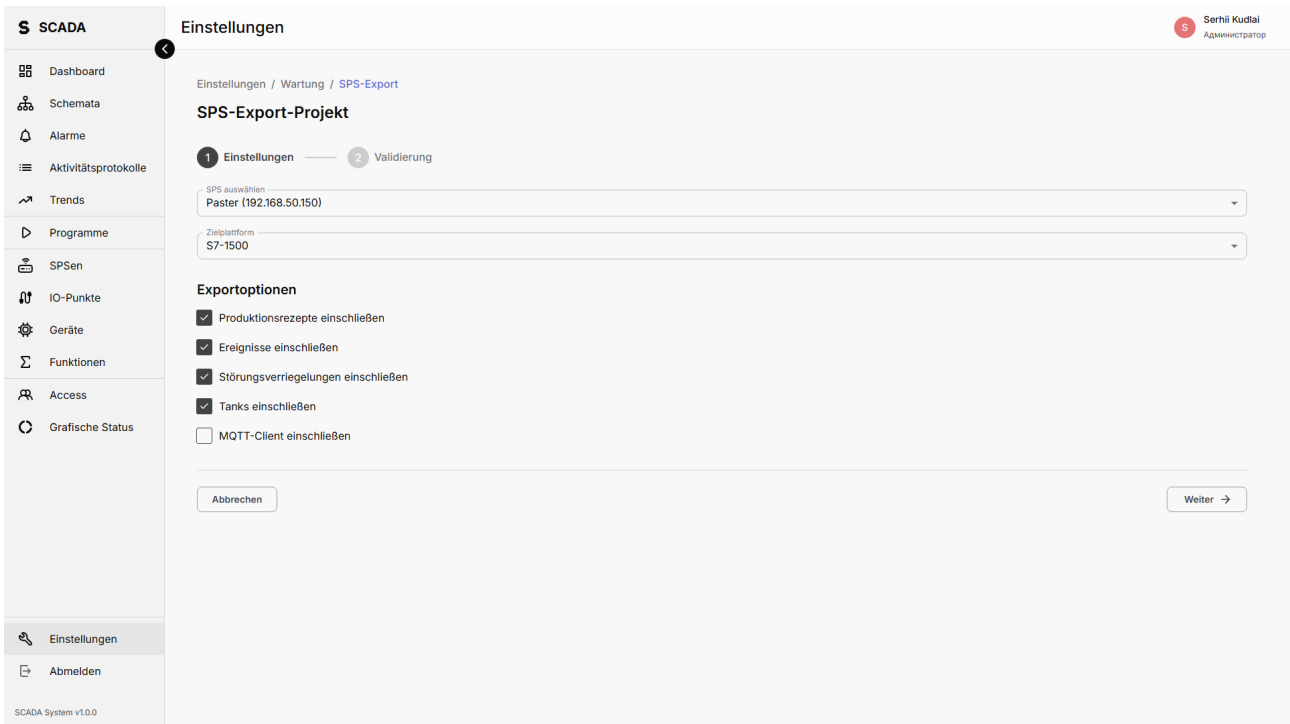
Erzeugung eines Siemens-TIA-Portal-Projekts aus der SCADA-Konfiguration.

Exportdialog (PlcExportDialog):

Element	Beschreibung
SPS-Auswahl	Ziel-Steuerung
SPS-Typ	S7-1200 / S7-1500
Rezepte einbinden	Checkbox — Rezepte hinzufügen
Ereignisse einbinden	Checkbox — Ereignisse
Störungen einbinden	Checkbox — Störungen
Tanks einbinden	Checkbox — Tanks
MQTT einbinden	Checkbox — MQTT-Kommunikation
MQTT Broker IP	IP des MQTT-Brokers
MQTT Topic Prefix	Topic-Präfix
Vorschau	Vorschau des SCL-Codes
Generieren	ZIP-Datei des Projekts herunterladen
Deploy to TIA	Über TIA Agent ausrollen

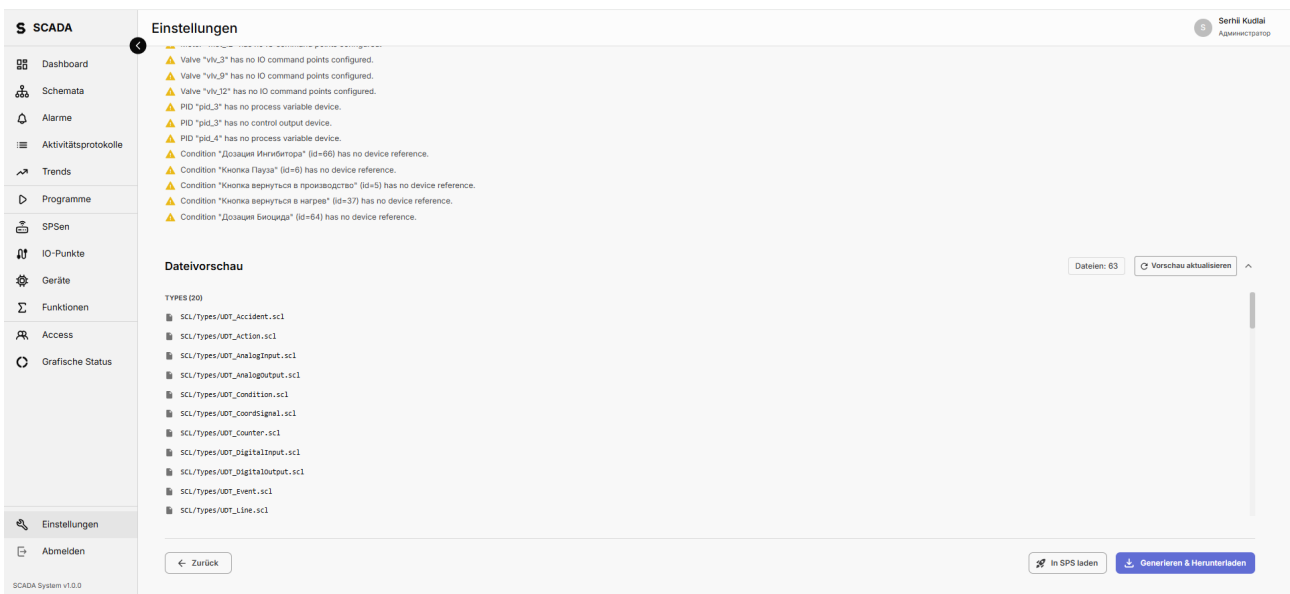
Inhalt des erzeugten Projekts:

- SCL-Code der Funktionsbausteine (Motor FB, Valve FB, PID FB, AI FB, AO FB, DI FB, DO FB, COS FB, Counter FB, Timer FB, Tank FB)
- Sequence FB (Zustandsautomat)
- Event/Accident FBs
- Data Blocks (Instanz-DBs, Rezept-DBs)
- OB1 Main (Aufruf aller FBs)
- XML Tag Tables (SimaticML)



Plc Export Dialog

Export-Dialog: SPS-Dropdown, Radio S7-1200/S7-1500, Include-Checkboxes (Recipes, Events, Accidents, Tanks, MQTT), MQTT-Config-Felder, Schaltflächen Preview/Generate/Deploy.



Plc Export Preview

SCL-Vorschau: links Dateibaum (FBs, DBs, Tag Tables), rechts SCL-Datei mit Syntax-Highlighting.

## Technische Dokumentation

---

PLC-Export-API:

Methode	Endpoint	Beschreibung
GET	/api/plc-export/preview/	Code-Vorschau
POST	/api/plc-export/validate/	Validierung
POST	/api/plc-export/generate/	ZIP erzeugen
GET	/api/plc-export/address-map/	Adress-Map

## 28. Projekt-Export und -Import

### Für Benutzer

Vollständiger SCADA-Projekt-Export/-Import zur Migration zwischen Installationen.

Exportdialog (ExportProjectDialog):

Element	Beschreibung
Objekt-Checkboxen	Typenauswahl: Devices, Schemas, Sequences, Recipes, Events, Accidents, Options, Trends, Alarms, Users, Roles
Select All	Alle auswählen
Graphen einbinden	Trenddaten exportieren
Suche	Objekt-Filter
Export	Exportdatei herunterladen
Fortschritt	Fortschrittsanzeige

Schaltflächen:

Schaltfläche	Beschreibung
Projekt exportieren	Export der gewählten Objekte starten
Projekt importieren	Datei laden → Vorschau → Anwenden
DB-Backup	Datenbank-Sicherung erstellen
DB-Restore	Aus Sicherung wiederherstellen

The screenshot shows the 'Export Project Dialog' in the SCADA System. The interface is in German and includes a sidebar with navigation options like Dashboard, Schemata, Alarmer, and Einstellungen. The main area is titled 'Projekt exportieren' and features a checkbox for 'Geräte-diagramme einschließen' (checked). Below this is a table titled 'Objekte zum Exportieren auswählen' with columns for object type and a progress indicator. The objects listed are: Schemata (3/3), Sequenzen (4/4), Motoren (21/21), Ventile (15/15), PID-Regler (11/11), Frequenzumrichter (0/0), Zähler (3/3), Timer (0/0), Analogeingänge (17/17), Analogausgänge (9/9), Digitaleingänge (4/4), Digitalausgänge (1/1), and Koordinationssignale (3/3). A button 'Ausgewählt: 247' and 'Alle abwählen' are visible at the top right of the table.

Export Project Dialog

Export-Dialog: Typenliste mit Checkboxes, Select-All, Toggle «Include Graphs», Export-Button, Fortschrittsbalken.

## Technische Dokumentation

### Projekt-API:

Methode	Endpoint	Beschreibung
GET	/api/project/export-models/	Liste exportierbarer Modelle
POST	/api/project/export/	Export
POST	/api/project/import/preview/	Import-Vorschau
POST	/api/project/import/	Import
POST	/api/project/backup-db/	DB-Backup
POST	/api/project/restore-db/	Restore
POST	/api/project/normalize-ids/	ID-Normalisierung
POST	/api/project/clear-database/	DB leeren
POST	/api/project/export-graph-data/	Graphen exportieren
POST	/api/project/import-graph-data/	Graphen importieren
GET	/api/project/graph-data-stats/	Graph-Statistik

## 29. Tanks

### Für Benutzer

Tanks sind Behälter mit automatischer Volumenberechnung anhand von Sensordaten.

Tank-Parameter:

Feld	Beschreibung
Füllstandssensor	Bindung an ein AI-Gerät (Füllstandsmesser)
Drucksensor oben	Druck im oberen Bereich (optional)
Drucksensor unten	Druck im unteren Bereich (für $\Delta P$ )
Alarmschwellen	HH, H, L, LL — Alarmpegel

Volumenberechnungsmethoden:

Methode	Beschreibung
Strapping Table	Peiltabelle (Interpolation sensor_value → Volumen)
Geometric	Geometrisch (Sektionen: Zylinder, Kegel, Kugel, Klöpperboden)

Tank-Konstruktor auf dem Schema:

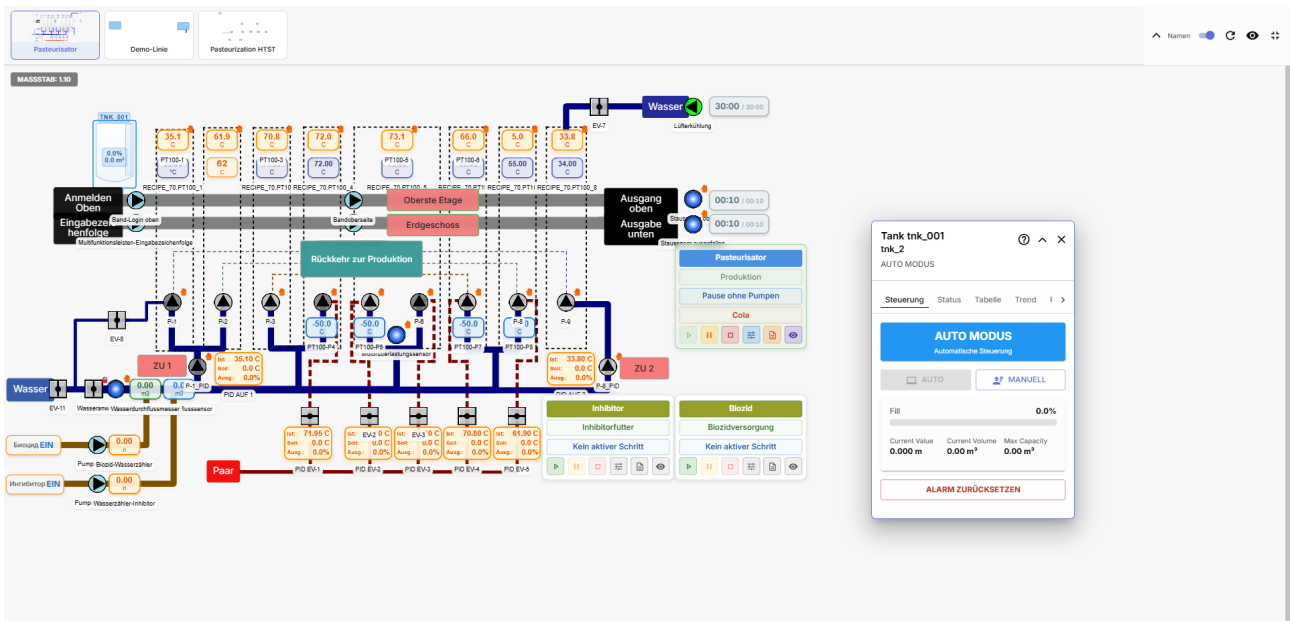
Element	Beschreibung
Geräteauswahl	Dropdown vorhandener TNK-Geräte
Bodentyp	FLAT / DOME / DISH / CONE
Deckeltyp	FLAT / DOME / DISH
Breite / Höhe	Visuelle Maße
SVG-Vorschau	Vorschau der Tankform
Sensoren automatisch platzieren	Sensoren entlang der rechten Tankseite

Tank Device Form

Tankformular: Sensorbindung (Level Sensor Dropdown, Pressure Top, Pressure Bottom), Alarmschwellen (HH, H, L, LL — numerische Felder).

Tank Strapping Table

Peiltabelle: zwei Spalten — Sensor Value und Volume, Schaltflächen Add Row, Delete Row, Import CSV, Export CSV.



Tank Constructor Dialog

Tank-Konstruktor: links Einstellungen (Gerät, Deckel, Maße), rechts SVG-Vorschau der Form.

## Technische Dokumentation

Tank-API:

Methode	Endpoint	Beschreibung
GET	/api/tanks/	Tankliste
POST	/api/tanks/	Erstellen
PUT	/api/tanks/{id}/	Aktualisieren
DELETE	/api/tanks/{id}/	Löschen
POST	/api/tanks/{id}/calculate_volume/	Volumen neu berechnen
POST	/api/tanks/{id}/import_strapping_table/	Tabelle importieren (CSV)
GET	/api/tanks/{id}/export_strapping_table/	Tabelle exportieren
GET	/api/tanks/{id}/volume_at_level/	Volumen bei Füllstand (?sensor_value=X)
GET	/api/tank-sections/	Sektionen (?tank=ID)
GET	/api/tank-strapping-entries/	Tabellenzeilen (?tank=ID)

Rust Worker — periodische Berechnung (alle 1000 ms):

Datei `scada-worker/src/tasks/tank_volume.rs` — fragt Füllstands- und Drucksensoren ab, berechnet Volumen nach Peiltabelle oder Geometrie, schreibt das Ergebnis in die DB und streamt es über Redis.

## Anwendungsnavigation (Seitenleiste)

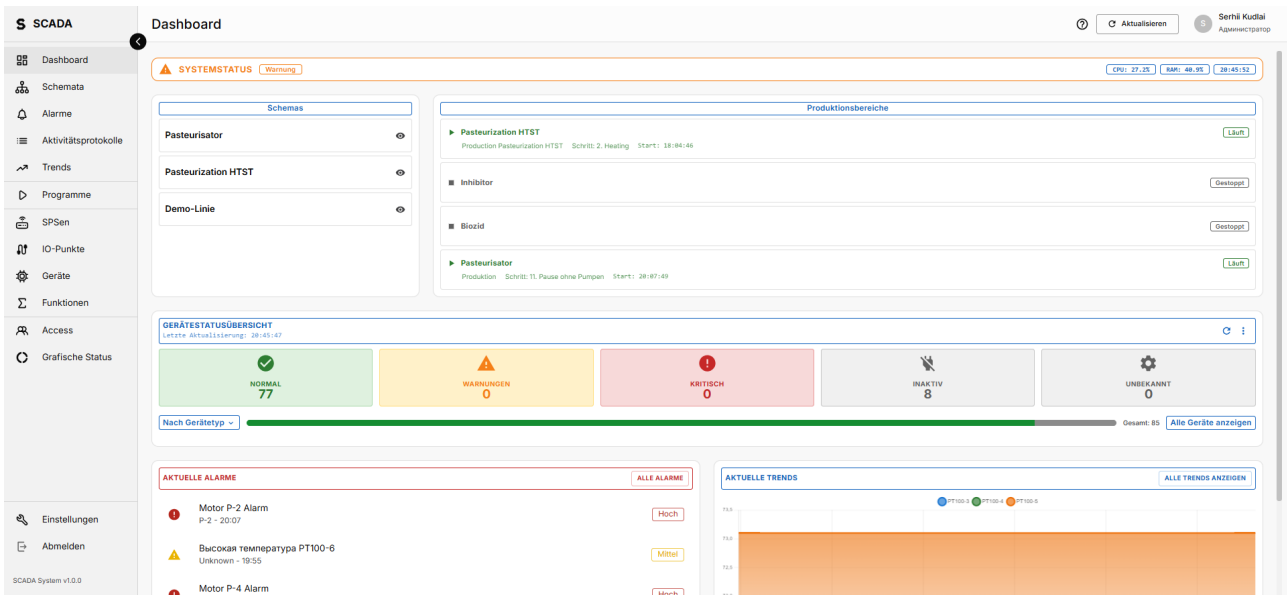
### Für Benutzer

Die Navigations-Seitenleiste ist auf allen Seiten verfügbar. Die Menüpunkte werden abhängig von den Berechtigungen angezeigt.

Menüpunkt	Seite	Benötigte Berechtigung
Dashboard	Dashboard	— (für alle)
Schemas	Mnemoniksschemata	—
Alarms	Alarme	—
Activity Logs	Aktivitätsprotokoll	—
Trends	Trends	—
Sequences	Sequenzen	sequences_recipes
PLCs	SPS	plcs
I/O Points	E/A-Punkte	io_points
Devices	Geräte	devices
Functions	Math-Funktionen	devices
Users	Benutzer	users
Roles	Rollen	roles
Graphical Statuses	Grafische Zustände	graphical_statuses
Settings	Einstellungen	— (für alle)
Logout	Abmelden	—

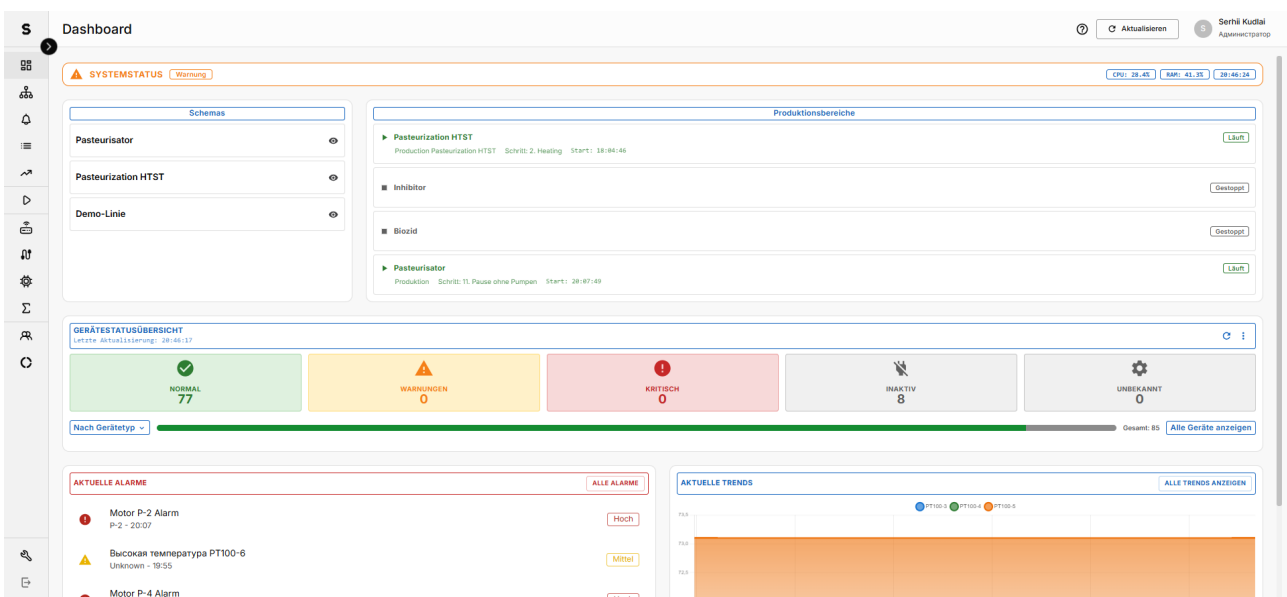
Weitere Elemente der Seitenleiste:

Element	Beschreibung
Benutzer-Avatar	Zeigt Foto und Name
Ein-/Ausklappen	Leiste auf Icons verkleinern
Versionsnummer	Aktuelle Anwendungsversion



Sidebar Expanded

Seitenleiste ausgeklappt: Avatar oben, Menüpunkte mit Icons und Text, Logout unten.



Sidebar Collapsed

Seitenleiste eingeklappt: nur Icons der Menüpunkte.

# Globale UI-Elemente

## Für Benutzer

Kopfleiste (Header):

Element	Beschreibung
Seitentitel	Aktueller Abschnittsname
Hilfe-Button	Startet die Tour
Update-Indikator	Badge bei verfügbaren Updates
Verbindungsanzeige	Status der WebSocket/MQTT-Verbindung
Kontextuelle Aktionen	Seitenspezifische Schaltflächen

Benachrichtigungen (Snackbar):

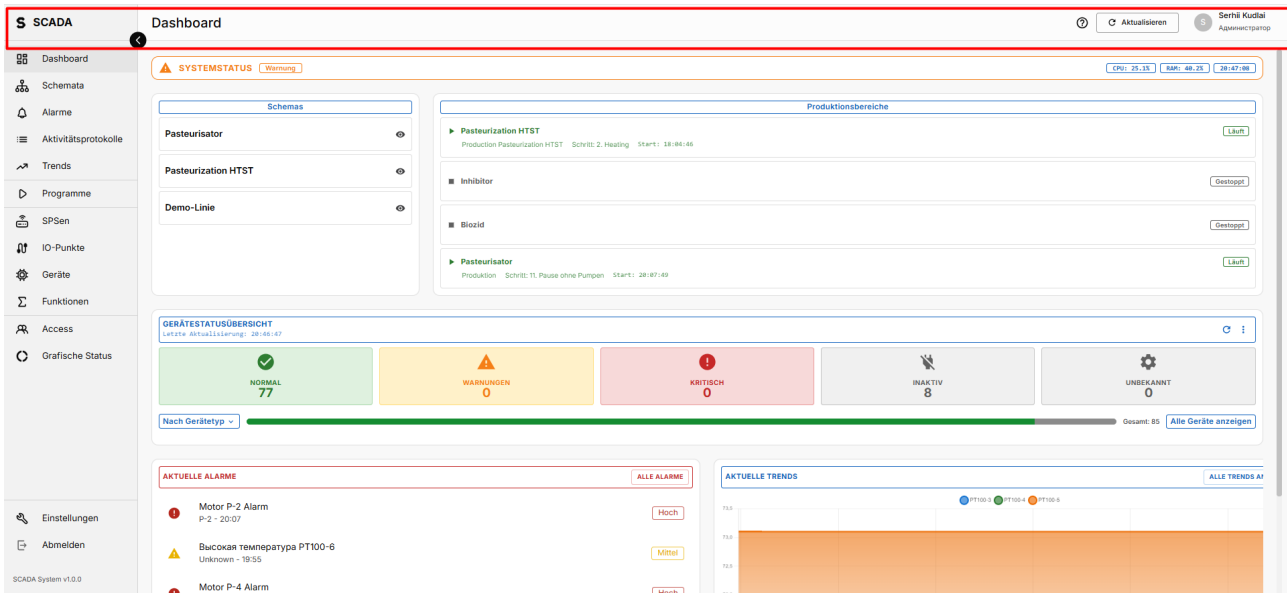
Typ	Beschreibung
Success (grün)	Operation erfolgreich
Error (rot)	Fehler
Warning (gelb)	Warnung
Info (blau)	Information

Banner:

Banner	Beschreibung
Connection Status	Gelbes Banner bei Verbindungsverlust (nach 15 s)
Maintenance	Blaues Banner im Wartungsmodus
Version Mismatch	Oranges Banner bei Versionskonflikt Frontend/Backend
Update Available	Grünes Badge im Header bei verfügbarem Update

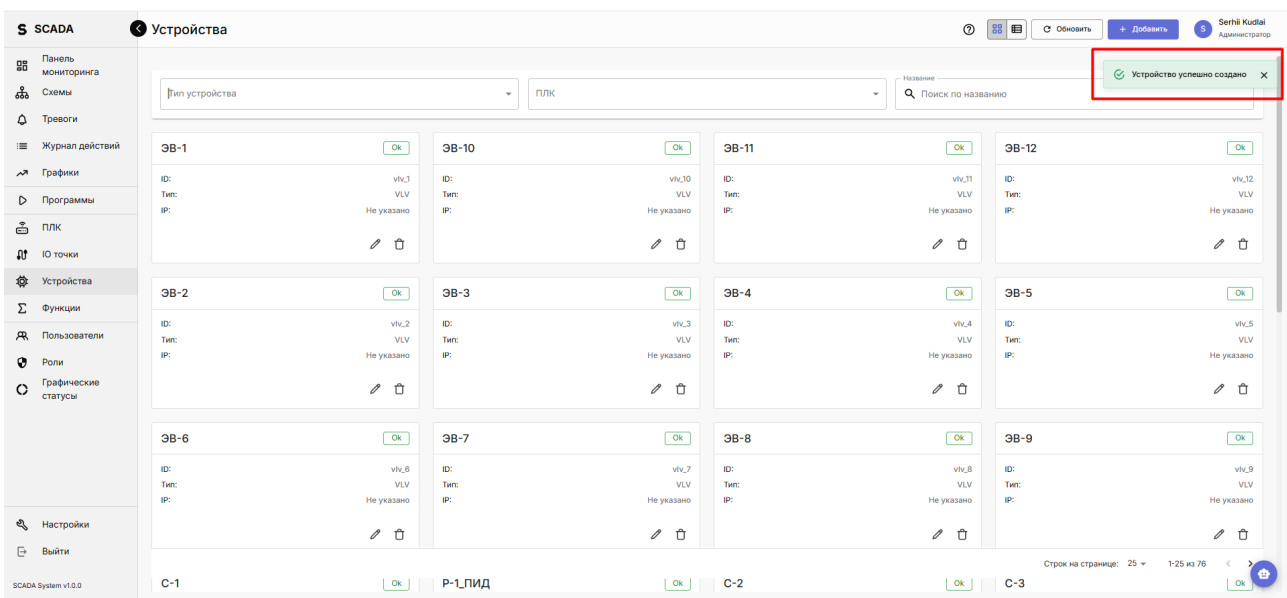
Lerntouren (Guided Tours):

Auf jeder Seite über den «?»-Button im Header verfügbar. Schrittweise interaktive Anleitung mit Element hervorhebung und Erläuterungen.



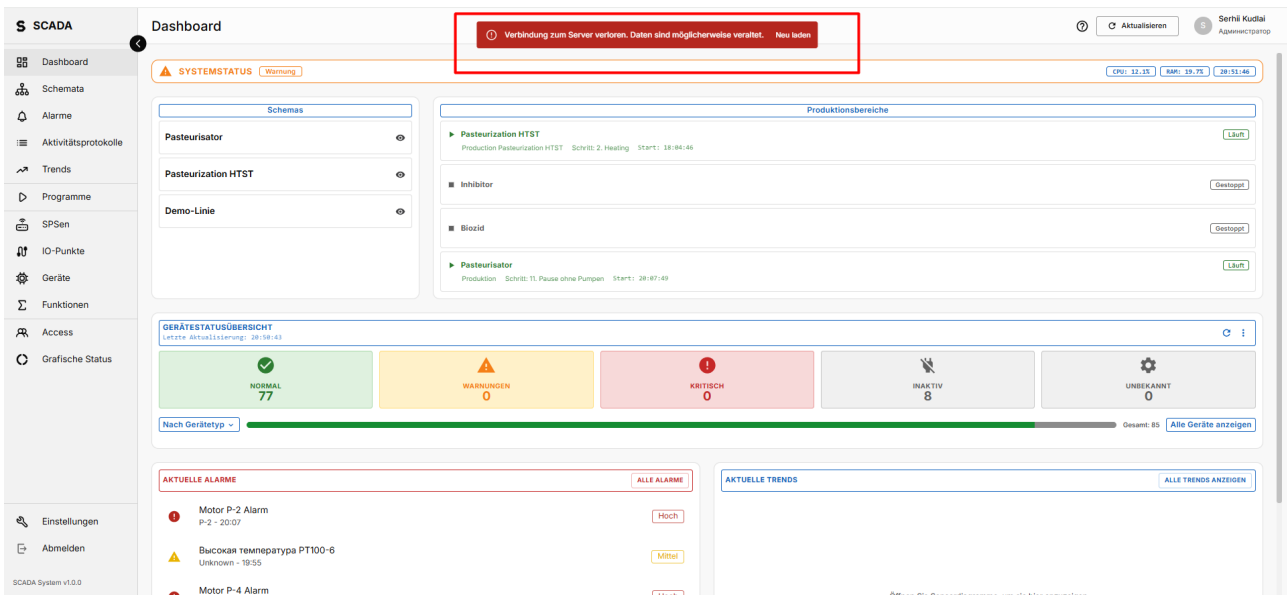
Header Bar

Anwendungs-Header: Seitentitel links, Tour-Button (i), Update-Indikator (rotes Badge), Verbindungsstatus (grüner/roter Punkt).



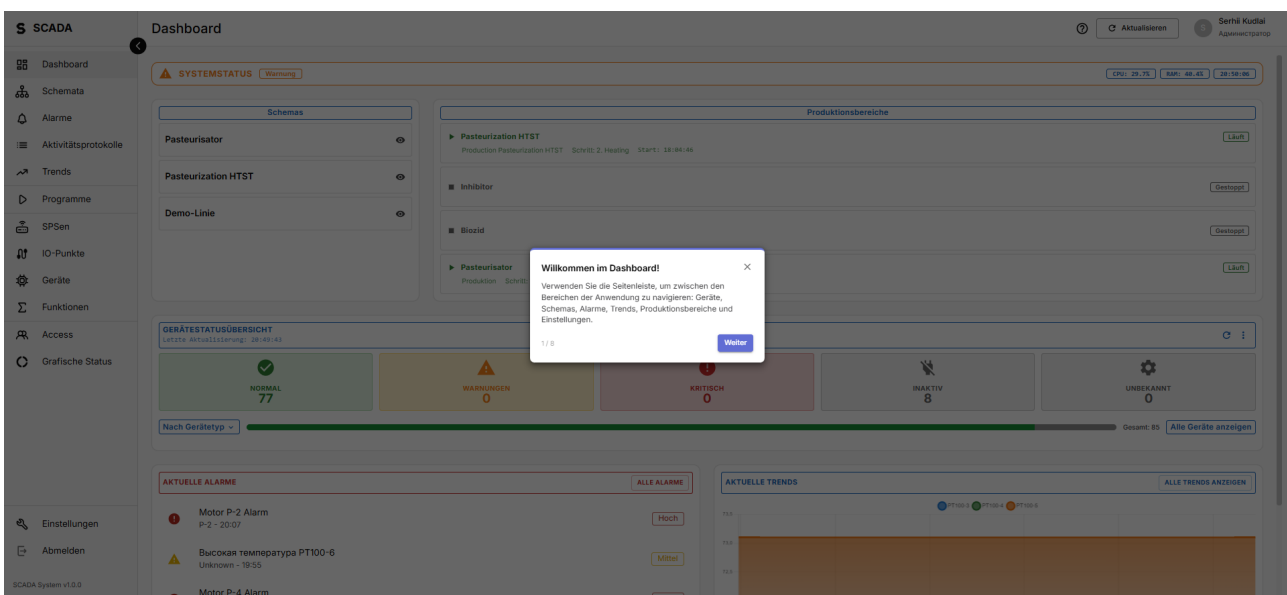
Snackbar Notifications

Benachrichtigungs-Beispiele: grünes «Device saved successfully», rotes «Error: Connection lost», gelbes «Warning: License expires in 7 days».



Connection Lost Banner

Gelbes Banner oben: «Connection lost. Attempting to reconnect...» mit Animation.



Guided Tour

Tour: hervorgehobenes Element mit Tooltip, Schaltflächen «Next», «Back», «Skip», Fortschrittsbalken.

## Tastenkombinationen

Taste	Aktion	Wo aktiv
Strg+Z	Rückgängig	Schema Builder
Strg+Y	Wiederholen	Schema Builder
Strg+S	Speichern	Schema Builder
Entf	Element löschen	Schema Builder
Strg+A	Alles auswählen	Schema Builder
Strg+C	Kopieren	Schema Builder
Strg+V	Einfügen	Schema Builder
Esc	Auswahl aufheben / Dialog schließen	Überall
Mausrad	Zoom	Schema Builder / Viewer, Trends

## Praktische Beispiele

Dieser Abschnitt enthält End-to-End-Automatisierungsszenarien, die zeigen, wie Geräte, Schema, Sequenz, Rezept und Ereignisse für reale Prozesse zusammenspielen.

### Beispiel 1. Automatisierung eines Milchpasteurisierers

Prozessziel. Milch auf 75 °C erhitzen, 15 Sekunden halten, auf 4 °C abkühlen. Das ist der klassische HTST-Prozess (High Temperature Short Time), weit verbreitet in der Molkereindustrie.

Prozessausrüstung:

Gerät	ID	Typ	Funktion
Zulaufpumpe	mot_00	Motor	Fördert Milch durch den Wärmetauscher
Dampfeintrittsventil	val_001	Valve	Versorgt den Heizabschnitt mit Dampf
Kühlerventil	val_002	Valve	Führt Kaltwasser zu
Heiztemperatursensor	ai_001	Analog Input	Temperatur nach Heizsektion
Haltetemperatursensor	ai_002	Analog Input	Temperatur nach Halterohr
Auslass-Temperatursensor	ai_003	Analog Input	Temperatur nach Kühler
Heizregler	pid_001	PID Controller	Hält Heiztemperatur (Sollwert → val_001)
Kühlregler	pid_002	PID Controller	Hält Auslasstemperatur (Sollwert → val_002)
Füllstand Empfangstank	tnk_001	Tank	Füllstandsüberwachung
3-Wege-Umleniventil	val_003	Valve	Führt untererwärmte Milch zur Wiederaufheizung zurück

### Etappe 1. Geräteanlage

Legen Sie alle Geräte über Devices → «+» an. Für den PID-Regler pid\_001:

```
POST /api/pid-controllers/
{
  "id": "pid_001",
  "name": "Heizregler",
  "plc": 1,
  "setpoint": 75.0,
  "kp": 2.5,
  "ki": 0.1,
  "kd": 0.05,
  "output_min": 0,
  "output_max": 100
}
```

Tank mit automatischer Volumenberechnung über eine Peiltabelle:

```
POST /api/tanks/
{
  "id": "tnk_001",
  "name": "Empfangstank",
  "plc": 1,
  "volume_method": "STRAPPING",
  "level_sensor": "ai_004",
  "alarm_hh": 95.0,
  "alarm_h": 85.0,
}
```

```
"alarm_1": 15.0,  
"alarm_11": 5.0  
}
```

## Etappe 2. Sequenz

Eine 5-Schritt-Sequenz beschreibt den gesamten Pasteurisationszyklus:

#	Schritt	Zweck	Übergangsbedingung zum nächsten Schritt
1	Idle	Warten auf Start	Option «Start»
2	Heating	Aufheizen auf Sollwert	ai_001 >= 75.0 für 5 s
3	Holding	Halten bei Temperatur	Timer 15 s
4	Cooling	Abkühlen auf 4 °C	ai_003 <= 4.0
5	Complete	Abschluss	Rückkehr zu Idle

Sequenz über die API anlegen:

```
POST /api/sequences/
{
  "name": "Pasteurization HTST",
  "name_translations": {"de": "Pasteurisation HTST", "ru": "Пастеризация HTST"},
  "description": "Aufheizen 75°C → Halten 15 s → Abkühlen 4°C"
}
```

Schritte hinzufügen:

```
POST /api/sequence-steps/
{
  "sequence": 1,
  "order": 2,
  "name": "Heating",
  "name_translations": {"de": "Heizen"},
  "next_steps": {"3": [10]}
}
```

Im Feld `next_steps` ist der Schlüssel "3" die Zielschrittnummer und der Wert [10] die Liste der Übergangsbedingungen-IDs.

Übergangsbedingung «Temperatur ≥ 75°C»:

```
POST /api/transition-conditions/
{
  "id": 10,
  "type": "device",
  "device_id": "ai_001",
  "content_type": "analoginput",
  "operator": ">=",
  "value": 75.0,
  "duration_seconds": 5
}
```

Wichtig. `duration_seconds` sorgt dafür, dass die Bedingung die angegebene Zeit durchgehend erfüllt ist — das schützt vor Fehlübergängen durch kurze Signalspitzen.

## Etappe 3. Produktionsrezept

Wir heben produktabhängige Parameter in Rezeptvariablen heraus, damit dieselbe Sequenz für verschiedene Molkereiprodukte (Vollmilch, Sahne, Magermilch) verwendet werden kann.

## Rezeptvariablen:

```
POST /api/production-recipe-variables/
{
  "name": "target_heating_temp",
  "name_translations": {"de": "Ziel-Heiztemperatur"},
  "sequence": 1,
  "default_value": 75.0,
  "min_value": 70.0,
  "max_value": 90.0
}
```

```
POST /api/production-recipe-variables/
{
  "name": "holding_time_sec",
  "name_translations": {"de": "Haltezeit, Sekunden"},
  "sequence": 1,
  "default_value": 15.0,
  "min_value": 10.0,
  "max_value": 30.0
}
```

## Rezept für Vollmilch:

```
POST /api/production-recipes/
{
  "name": "Whole Milk 3.5%",
  "name_translations": {"de": "Vollmilch 3,5%"},
  "sequence": 1,
  "variables": {
    "target_heating_temp": 75.0,
    "holding_time_sec": 15.0
  }
}
```

## Rezept für Sahne (strengerer Modus):

```
POST /api/production-recipes/
{
  "name": "Cream 30%",
  "name_translations": {"de": "Sahne 30%"},
  "sequence": 1,
  "variables": {
    "target_heating_temp": 85.0,
    "holding_time_sec": 20.0
  }
}
```

Bei Rezeptaktivierung wird der PID-Sollwert `pid_001.setpoint` automatisch an die Variable `target_heating_temp` und der Halte-Timer an `holding_time_sec` gebunden.

## Etappe 4. Störungseignisse

Definieren wir zwei kritische Störungen:

### 1. Überhitzung (über 90 °C):

```
POST /api/accident-events/
{
  "name": "Overheat protection",
  "name_translations": {"de": "Überhitzungsschutz"},
  "device_id": "ai_001",
  "content_type": "analoginput",
  "operator": ">",
  "value": 90.0,
  "severity_level": 4,
  "action": "STOP_SEQUENCE"
}
```

severity\_level: 4 ist CRITICAL — sofortiger Stopp der Sequenz.

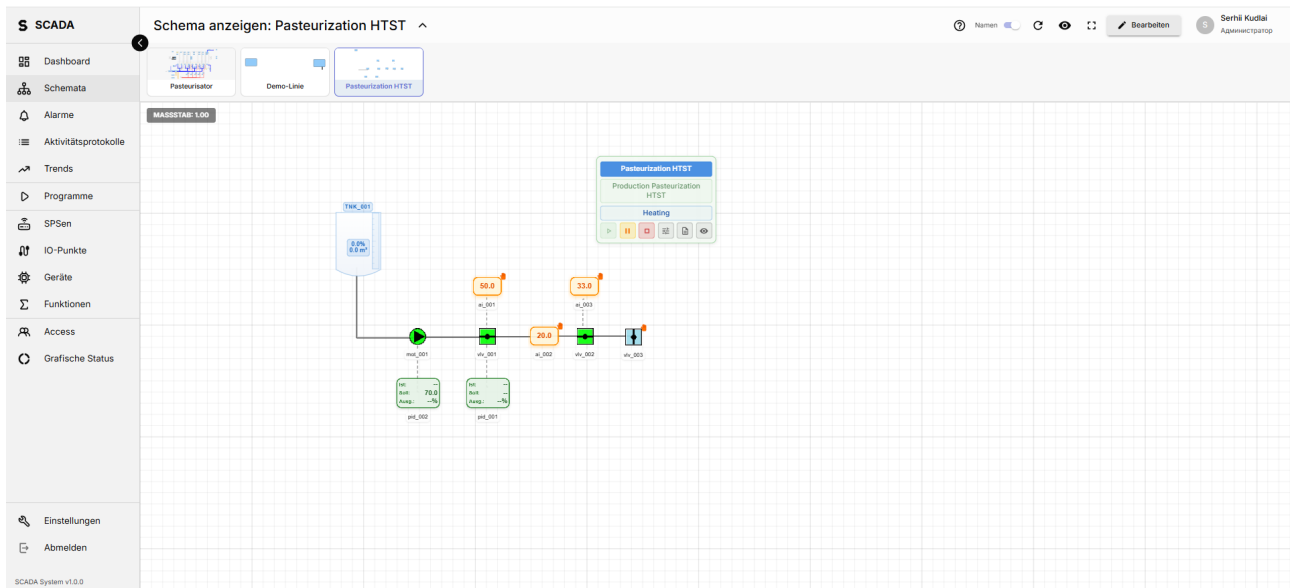
## 2. Unterpasteurisierung (unter 73 °C im Holding-Modus):

```
POST /api/accident-events/
{
  "name": "Insufficient pasteurization",
  "name_translations": {"de": "Unzureichende Pasteurisation"},
  "device_id": "ai_002",
  "content_type": "analoginput",
  "operator": "<",
  "value": 73.0,
  "severity_level": 3,
  "action": "TRIGGER_DIVERT"
}
```

Bei Auslösung aktiviert das System automatisch das Umlenkventil val\_003 — unzureichend pasteurisierte Milch wird zur erneuten Erwärmung geleitet.

## Etappe 5. Mnemonisches Schema

- Erstellen Sie ein Schema «Pasteurisierer HTST».
- Platzieren Sie im Schema Builder:
- Empfangstank tnk\_001 über Tank — wählen Sie im Konstruktor Form FLAT / DOME und Maße 80×120.
- Pumpe mot\_001, Ventile val\_001, val\_002, val\_003, PIDs pid\_001, pid\_002.
- Sensoren ai\_001, ai\_002, ai\_003.
- Verbinden Sie mit Rohrleitungen: Tank → Pumpe → Heizer (mit val\_001 und ai\_001) → Halterohr (ai\_002) → Kühler (val\_002, ai\_003) → Auslass.
- Binden Sie die Sequenz ein: Sequenz → «Pasteurization HTST» wählen.
- Fügen Sie Options-Buttons «Start», «Pause», «Stop» hinzu (eigener Option-Block).



Example Pasteurizer Schema

Schema des Pasteurisierers: links der Empfangstank mit Füllstand, davon eine Rohrleitung zur Pumpe, dann durch die Heizsektion mit Temperatursensor und Dampfventil, das Halterohr, die Kühlsektion mit Kühlmittelventil. Rechts der Sequenzblock mit Hervorhebung des aktiven Schritts «Heating».

## Etappe 6. Prozess starten

- Öffnen Sie das Schema im Schema Viewer.
- Wählen Sie auf dem Sequenz-Block das aktive Rezept — z. B. «Whole Milk 3.5%».
- Drücken Sie Start — die Sequenz wechselt von Idle zu Heating.
- Verfolgen Sie den Prozess in Echtzeit:
- Hervorhebung des aktiven Schritts (gelb → grün bei erfolgreichem Übergang).
- Aktueller Temperaturwert im Sensor-Popup.
- Öffnungsgrad val\_001 (durch PID gesteuert).
- Bei 75 °C und 5 s Haltezeit erfolgt automatisch der Übergang zu Holding.
- Nach 15 s — Übergang zu Cooling.
- Bei Auslasstemperatur  $\leq 4$  °C — Complete → zurück zu Idle.

## Etappe 7. Analyse nach der Charge

- Öffnen Sie Trends → neues Diagramm erstellen.
- Fügen Sie Trendlinien hinzu: ai\_001, ai\_002, ai\_003, pid\_001.output.
- Stellen Sie den Bereich auf die letzten 30 Minuten.
- Prüfen Sie:
- Form der Heizkurve (Zeit bis Sollwert).
- Stabilität der Haltetemperatur (Welligkeit  $\leq 0,5$  °C erwartet).
- Abkühlgeschwindigkeit.
- Exportieren Sie das Diagramm als PNG oder CSV für den Bericht.

## Etappe 8. PLC-Projekt erzeugen

- Öffnen Sie Dashboard → Button PLC-Export (oder Settings → PLC Export).
- Wählen Sie im Dialog:

- SPS: PLC\_DEMO
- Geräte, Sequenzen, Rezepte und Störungen — alle.
- Drücken Sie Preview — das System zeigt die zu erzeugenden Dateien:
- Motor\_FB.scl, Valve\_FB.scl, PID\_FB.scl — Geräte-Funktionsbausteine
- Sequence\_Pasteurization\_FB.scl — Zustandsautomat
- Accident\_FB.scl — Störungshandler
- OB1\_Main.scl — Hauptprogramm
- Tags\_Inputs.xml, Tags\_Outputs.xml — Tag-Tabellen für TIA Portal
- Validate — prüft die Konfiguration.
- Generate — erstellt ein ZIP für den TIA-Portal-Import.

## Beispiel 2. CIP-Reinigung mit Chargenverwaltung

Prozessziel. Automatische Reinigung der Prozessanlagen (Tanks, Rohrleitungen) nach einer Produktionsschicht. Mehrstufig: Vorspülung → Laugenwäsche → Zwischenspülung → Säurewäsche → Endspülung.

Besonderheiten dieses Beispiels:

- Verwendet Batch Management — jeder Reinigungszyklus wird als Charge erfasst.
- Nutzt eine Produktionslinie mit Stufen.
- Der Bediener kann einzelne Stufen über Optionen überspringen.
- Zeigt Timer- und Zähler-Einsatz.

### Etappe 1. Ausrüstung

ID	Typ	Funktion
tnk_cip_01	Tank	Laugenvorrattank (NaOH 2%)
tnk_cip_02	Tank	Säurevorrattank (HNO 1,5%)
tnk_water	Tank	Umlaufwassertank
mot_cip_pump	Motor	CIP-Zirkulationspumpe
val_cip_supply	Valve	Zulaufventil in den Reinigungskreis
val_cip_return	Valve	Rücklaufventil
val_alkali	Valve	Laugenzulaufventil
val_acid	Valve	Säurezulaufventil
val_water	Valve	Wasserzulaufventil
ai_cip_temp	Analog Input	Lösungstemperatur
ai_cip_cond	Analog Input	Leitfähigkeit (Konzentrationskontrolle)
cnt_cip_flow	Counter	Durchflusszähler im Zulauf
tim_cip_stage	Timer	Timer der aktuellen Stufe

### Etappe 2. Produktionslinie und Stufen

Erstellen wir eine Produktionslinie mit fünf Stufen — jede hat ihre eigene Sequenz.

```
POST /api/production-lines/
{
  "name": "CIP Line 1",
  "name_translations": {"de": "CIP-Linie 1"},
  "description": "Tankreinigung Produktion 1"
}
```

Stufen (ProductionLineStage):

#	Name	Sequenz	Chargenaktion
1	Pre-rinse	seq_cip_rinse (Wasser 90 s)	TRANSFER zur nächsten Stufe
2	Alkali wash	seq_cip_alkali (NaOH 20 Min bei 75 °C)	TRANSFER
3	Intermediate rinse	seq_cip_rinse	TRANSFER
4	Acid wash	seq_cip_acid (HNO 15 Min bei 65 °C)	TRANSFER
5	Final rinse	seq_cip_rinse	COMPLETE

Das Feld `batch_transfer_action` am letzten Schritt jeder Sequenz steuert den Übergang:

- TRANSFER — Charge wird automatisch zur nächsten Stufe weitergereicht.
- COMPLETE — Charge wird abgeschlossen.
- NONE — Charge bleibt auf der aktuellen Stufe (manuelle Steuerung).

### Etappe 3. Sequenz «Laugenwäsche»

Schritte:

- Fill — Kreis mit Lauge füllen (`val_alkali = OPEN`, `mot_cip_pump = ON`).
- Heat — auf 75 °C heizen (`ai_cip_temp >= 75.0`).
- Circulate — Zirkulation 20 Min (Timer `tim_cip_stage`).
- Drain — Ablassen (`val_cip_return = OPEN` in die Kanalisation).

Rezept «Laugenwäsche — Standard»:

```
POST /api/production-recipes/
{
  "name": "Alkali wash - standard",
  "production_line": 1,
  "variables": {
    "alkali_temp": 75.0,
    "alkali_duration_min": 20,
    "acid_temp": 65.0,
    "acid_duration_min": 15
  }
}
```

Beachten Sie: Das Rezept ist nicht an eine Sequenz gebunden, sondern an die Produktionslinie (`production_line: 1`). Die Variablen sind in allen Sequenzen der Linien-Stufen verfügbar.

### Etappe 4. Bedieneroptionen

Fügen wir Options-Buttons hinzu, die den Prozess beschleunigen oder anpassen:

Option	Beschreibung	Effekt
SKIP_ACID	Säurewäsche überspringen	Bei Aktivierung wird Stufe 4 übersprungen (direkt zu Stufe 5)

Option	Beschreibung	Effekt
DOUBLE_RINSE	Doppelte Endspülung	Stufe 5 wird zweimal wiederholt
LOW_TEMP	Schonmodus (niedrigere Temperatur)	Variablen alkali_temp = 60, acid_temp = 50

Optionen werden über Sequence Options hinzugefügt und als Buttons auf dem Schema platziert.

## Etappe 5. Charge erstellen und starten

- Öffnen Sie das CIP-Linien-Schema im Schema Viewer.
- Drücken Sie am Production Line-Element Charge erstellen — der CreateBatchDialog öffnet sich.
- Füllen Sie aus:
  - Batch name: CIP-2026-04-18-001
  - Recipe: «Alkali wash — standard»
  - Stage: Pre-rinse (Anfangsstufe)
  - Priority: Normal
- Create — die Charge reiht sich in die Warteschlange ein.
- Aktivieren Sie die Charge im BatchManagementPanel → Start.

API-Äquivalent:

```
POST /api/production-lines/1/create_batch/
{
  "name": "CIP-2026-04-18-001",
  "recipe_id": 5,
  "starting_stage_id": 1,
  "priority": "NORMAL"
}
```

## Etappe 6. Chargen-Monitoring

Im BatchManagementPanel werden angezeigt:

- Aktuelle Stufe (z. B. «Alkali wash»).
- Fortschritt innerhalb der Stufe (Countdown-Timer).
- Aktuelle Werte der Schlüsselsensoren (Temperatur, Leitfähigkeit, Durchfluss).
- Gesamtfortschritt der Charge über alle Stufen.

Stufenübergänge werden in BatchStageHistory gespeichert — die vollständige Historie lässt sich später abrufen:

```
GET /api/batches/{id}/history/
```

## Etappe 7. Validierung der erfolgreichen Reinigung

HACCP/ISO 22000 verlangen dokumentarische Bestätigung der Reinigung. Nutzen wir ein Ereignis in entgegengesetzter Richtung:

```
POST /api/events/
{
  "name": "CIP validation - conductivity check",
  "device_id": "ai_cip_cond",
  "operator": "<",
  "value": 0.5,
  "duration_seconds": 30,
  "action": "LOG_EVENT"
}
```

Dieses Ereignis protokolliert den Moment, in dem die Leitfähigkeit der Endspülung für 30 s unter 0,5 mS/cm fällt — Nachweis, dass das Reinigungsmittel vollständig ausgespült wurde.

Das Protokoll ist in Activity Logs mit Filter nach Sequenz/Charge verfügbar und kann als CSV für Berichte exportiert werden.

## Checkliste für einen neuen Prozess

---

Nutzen Sie als Vorlage bei jeder neuen Prozesseinrichtung:

### 1. Planung (vor der Eingabe ins System):

- P&ID-Schema des Prozesses gezeichnet
- Geräteliste mit eindeutigen IDs erstellt
- Alle analogen und digitalen Signale definiert
- Betriebslogik beschrieben (Schritte und Übergangsbedingungen)
- Alarmschwellen und Aktionen definiert

### 2. Einrichtung in SCADA:

- SPS mit korrekter Adresse angelegt
- Alle Geräte angelegt und an die SPS gebunden
- PID-Parameter konfiguriert (Kp, Ki, Kd)
- Tank-Peiltabellen konfiguriert (wo zutreffend)
- Mnemonisches Schema mit Geräten und Rohren erstellt
- Sequenz mit Schritten erstellt
- Übergangsbedingungen erstellt
- Rezeptvariablen erstellt
- Mindestens ein Rezept mit Standardwerten erstellt
- Störungsereignisse mit passender Severity konfiguriert

### 3. Test (vor Produktionsaufnahme):

- Jedes Gerät lässt sich manuell steuern (Manual)
- Übergangsbedingungen lösen korrekt aus (Simulation nutzen — `POST /api/transition-conditions/{id}/simulate/`)
- Sequenz durchläuft alle Schritte end-to-end in Manual
- Störungen lösen aus und stoppen den Prozess korrekt
- Rezepte werden angewendet — Sollwerte aktualisieren sich
- Trends schreiben Daten für alle Schlüsselvariablen
- PLC-Export läuft fehlerfrei durch die Validierung

### 4. Inbetriebnahme:

- Bediener geschult
- Projektbackup erstellt (`POST /api/project/export/`)
- Rollen und Berechtigungen zugeordnet (Bediener, Schichtleiter, Ingenieur)
- Erste Charge mit reduzierten Parametern als Sanity-Check gestartet
- Arbeitsplatzdokumentation erstellt

## Fehlerbehebung und FAQ

Dieser Abschnitt enthält typische Probleme und deren Lösungen.

### Verbindung und Authentifizierung

F: Ein gelbes «Verbindung verloren»-Banner bleibt sichtbar.

Mögliche Ursachen:

- Rust-Worker oder Django-Backend steht — `docker-compose ps` ausführen; alle 6 Container müssen Up sein.
- Nginx proxyt WebSocket nicht — DevTools → Network → WS öffnen. Close-Code 1006 oder 502 weist auf nginx hin. Prüfen Sie `nginx.conf`: `proxy_http_version 1.1` sowie Header `Upgrade/Connection` sind erforderlich.
- JWT-Token abgelaufen — ab- und wieder anmelden. Normalerweise wird der Token 1 Minute vor Ablauf automatisch erneuert.
- Eine Firewall blockiert Port 8001 (Rust WebSocket). Das System fällt auf Django-WebSocket zurück, mit zusätzlicher Latenz — Port öffnen.

F: Nach einem Update erscheint ein orangefarbenes «Version mismatch».

Frontend und Backend sind unterschiedlichen Versionen. Browser mit Cache-Bypass neu laden (Strg+Umschalt+R). Hilft das nicht — `docker-compose.yml` prüfen: der `frontend`-Tag muss zum `backend`- und `scada-worker`-Tag passen.

F: Ich kann mich nicht anmelden — «Invalid credentials».

- Tastaturbelegung prüfen (CAPS LOCK).
- Sicherstellen, dass Sie Username, nicht E-Mail, verwenden (falls die Konfiguration spezifisch ist).
- Wenn das Passwort verloren ist und SMTP nicht konfiguriert ist, auf dem Server zurücksetzen: ``bash docker-compose exec backend python manage.py changepassword``
- Prüfen, dass Ihre IP nicht in IP Access gesperrt ist (ein Administrator kann eine Whitelist aktiviert haben).

### Schema und Geräte

F: Ein Gerät bleibt grau und Werte aktualisieren sich nicht.

- SPS offline — Status im Abschnitt PLCs prüfen. Ohne Online bekommt das Gerät keine Daten.
- Gerät nicht an SPS gebunden — Gerätekarte öffnen, Feld PLC prüfen.
- WebSocket-Kanal nicht abonniert — DevTools → Network → WS: es sollte eine `ws/schema/{id}/devices`-Verbindung existieren. Fehlt sie, Seite neu laden.
- Keine Berechtigung für das Gerät — Administrator bitten, Ihre Rollen zu prüfen.

F: Ein Tank zeigt Volumen 0, obwohl der Füllstandssensor arbeitet.

- Peiltabelle nicht konfiguriert (für Methode STRAPPING). Tank → Tab «Strapping Table» → mindestens 2 Punkte eintragen (`sensor_value` → `volume`).
- Falsche Berechnungsmethode — Feld `volume_method` prüfen: STRAPPING oder GEOMETRIC. GEOMETRIC erfordert TankSection-Einträge (Zylinder, Kegel usw.).
- Rust-Worker aktualisiert das Volumen nicht — `docker-compose logs scada-worker | grep tank_volume`. Standardmäßig wird das Volumen sekundlich berechnet (`TANK_VOLUME_INTERVAL_MS`).

F: Ein Rohr animiert keinen Fluss, obwohl der Motor läuft.

Aktivieren Sie Prozessüberwachung am Rohr in der rechten Eigenschafts-Leiste des Schema Builder. Standardmäßig ist es aus.

F: Ein Gerät springt beim Ziehen zurück.

Das passiert bei gleichzeitigem WebSocket-Update. Im Schema Builder blockiert das Bearbeiten Live-Updates; ansonsten direkt nach dem Verschieben speichern (Strg+S).

## Sequenzen und Rezepte

F: Die Sequenz wechselt nicht zum nächsten Schritt, obwohl die Bedingung erfüllt ist.

- Bedingung nicht am Schritt gebunden — aktuellen Schritt öffnen, `next_steps` prüfen. Der Zielschritt muss die Bedingungs-ID enthalten.
- ``duration_seconds`` noch nicht abgelaufen — die Bedingung muss die Zeit durchgehend erfüllt sein. Im Bedingungs-Popup das Feld «Continuous for» prüfen.
- Die Bedingung wird mit einem anderen Wert simuliert — Flag `is_simulated` im WebSocket-Payload der Bedingung prüfen. Zurücksetzen: `POST /api/transition-conditions/clear_simulation/` mit `{"step_id": }`.
- Gerätewert kommt nicht an — Gerät offline (siehe oben).

F: Rezept geändert, aber der PID-Sollwert blieb gleich.

Stellen Sie sicher:

- Das Rezept ist aktiviert — `POST /api/production-recipes/{id}/select-production-recipe/`.
- Die Rezeptvariable ist an ein Gerätefeld gebunden im Sequenz-Editor (Tab «Schritte» → Gerätewerte pro Schritt prüfen).
- Die Sequenz ist nicht in `is_pause = true` — Pause friert Übergänge und Rezeptanwendung ein.

F: Ich möchte die Logik ohne reale Hardware testen.

- Übergangsbedingungs-Simulation nutzen — `POST /api/transition-conditions/{id}/simulate/` mit `{"simulated_value": 80.0}`. Die Bedingung bewertet den übergebenen Wert.
- Für Analogeingänge Simulation mode einschalten: `POST /api/analog-inputs/{id}/set_simulation/` mit `{"is_simulated": true, "simulated_value": 75.0}`.
- Nach dem Test Simulation unbedingt ausschalten — im Betrieb ist sie gefährlich.

## Chargen (Batches)

F: Eine Charge hängt auf einer Stufe.

- `batch_transfer_action` am letzten Schritt der aktuellen Stufen-Sequenz prüfen — muss TRANSFER (für Weitergabe) oder COMPLETE (für Abschluss) sein.
- Prüfen, dass die nächste Stufe existiert und aktiv ist (`is_active: true`).
- Rust-Worker-Log prüfen — Meldung `batch transfer failed` nennt die Ursache.
- Manueller Transfer: `POST /api/batches/{id}/transfer/` mit `{"next_stage_id": }`.

F: Die Chargen-Warteschlange ist in falscher Reihenfolge.

Sortierung: Priorität, dann Erstellungszeit. Priorität ändern: `POST /api/batch-queue/{id}/set_priority/` mit `{"priority": "HIGH"}`. Werte: LOW, NORMAL, HIGH, URGENT.

## PLC-Export

F: Beim Drücken von «Validate» erscheint eine Fehlerliste.

Typische Validierungsfehler:

- Device ID contains invalid characters — ID muss `[a-z0-9_]` entsprechen, keine Bindestriche oder Leerzeichen.
- Sequence references missing device — Gerät gelöscht, aber im Schritt referenziert. Im Sequenz-Editor korrigieren.
- PID `output_max`  $\leq$  `output_min` — PID-Grenzen prüfen.

- Tank strapping entries must be monotonic — sensor\_value muss streng zunehmen.
- No PLC assigned to device X — Gerät an eine SPS binden.

F: Der generierte SCL-Code kompiliert im TIA Portal nicht.

- TIA-Portal-Version prüfen — der Generator zielt auf TIA Portal v16+.
- Sicherstellen, dass beide XML-Tag-Tabellen (Inputs und Outputs) vor dem SCL importiert werden.
- TIA-Portal-Kompilierungsprotokoll lesen — es zeigt den konkreten FB und die Zeile.

## Trends

---

F: Das Diagramm ist leer, obwohl das Gerät arbeitet.

- Zeitbereich zu klein oder in der Zukunft — auf «Last 1 hour» zurücksetzen.
- Timescale offline — GET /api/graphs/timescale-status/ prüfen. disconnected bedeutet TimescaleDB nicht verfügbar.
- Datenpunkte nicht erfasst — im Abschnitt Data Points muss das Gerät eine aktive Abo-Erfassung haben.

F: Trends werden bei großen Bereichen langsam.

Downsampling ist aktiv. Wenn es trotzdem zu viele Samples sind — Bereich verkleinern oder Aggregationsintervall erhöhen. Für Langzeitanalysen nach CSV exportieren und extern verarbeiten.

## KI-Assistent

---

F: Der KI-Assistent gibt «Authentication failed» zurück.

- OpenRouter-API-Schlüssel in Settings → AI Assistant prüfen. Schlüssel werden beim Speichern mit Fernet verschlüsselt — der Fehler bedeutet abgelaufener Schlüssel oder beschädigter DB-Eintrag.
- OpenRouter-Guthaben prüfen — bei Null werden Anfragen abgelehnt.
- Logs: docker-compose logs ai-assistant | tail -50.

F: Die KI antwortet langsam.

Das Modell läuft als Agent mit bis zu 15 Iterationen und Tool-Aufrufen. Komplexe Anfragen («analysiere alle Störungen der letzten Woche und schlage Optimierungen vor») können 30-60 s dauern. Für schnelle Antworten — kurz und spezifisch fragen.

## Leistung

---

F: Die UI ruckelt bei einem Schema mit 100+ Geräten.

- Strömungsanimation an nicht am Prozess beteiligten Rohren deaktivieren.
- In Settings → Performance «Throttle non-critical updates» aktivieren (inaktive Geräte werden alle 2 s statt 250 ms aktualisiert).
- Docker mehr RAM geben (mindestens 4 GB für komfortables Arbeiten).

F: Die Datenbank wächst schnell.

Hauptquellen sind das Aktivitätsjournal und historische Trends. Retention in Settings → Data Management konfigurieren:

- Activity Logs: 90 Tage Standard.
- Trend-Daten: pro Gerät über Data Points einstellbar.

## Sicherheit und Best Practices

Dieser Abschnitt sammelt Empfehlungen für einen sicheren Systembetrieb. Ihre Einhaltung senkt Zwischenfallsrisiken und hilft bei der Einhaltung von Qualitätsstandards (ISO 9001, HACCP, GMP).

### Arbeit mit Auto / Manuell

Manuell-Modus gibt dem Bediener direkte Kontrolle unter Umgehung von Sequenzen und Alarmverriegelungen. Ein mächtiges — und gefährliches — Werkzeug.

Richtige Einsätze von Manuell:

- Nur für Inbetriebnahme und Diagnose.
- Nur an stillstehender Ausrüstung oder in speziell vorgesehenen Zuständen.
- Mit mündlicher Zustimmung des Schichtleiters an laufender Linie.
- Mit Eintrag im Aktivitätsjournal (geschieht automatisch, aber der Bediener sollte es wissen).

Falsche Einsätze von Manuell:

- Umgehen von Schutzbedingungen, um den Prozess zu beschleunigen.
- Manueller Start von Ausrüstung ohne Kenntnis des aktuellen Chargenzustands.
- Ausrüstung am Schichtende in Manuell zu lassen.

Regel. Nach Arbeit in Manuell immer auf Auto zurückstellen. Das System macht es nicht automatisch.

### Simulation: nur für Tests

Simulation von Bedingungen, Analogeingängen und Timern ist Werkzeug des Planers, nicht des Bedieners.

Szenario	Erlaubt?
Sequenzlogik auf dem Prüfstand testen	Ja
Alarmauslösung im Simulator prüfen	Ja
Bedingung «überspringen» während der Produktion zum schnelleren Start	Kategorisch nein
Sensor simulieren, weil der echte kaputt ist	Nein — Sensor reparieren oder Prozess stoppen

Jede aktive Simulation während der Produktion ist eine potenzielle Zwischenfallsursache. Vor dem Start einer Produktionscharge prüfen: `GET /api/transition-conditions/?is_simulated=true` — die Liste darf nichts zur aktiven Sequenz enthalten.

### Backups und Wiederherstellung

Was sichern:

- Projektkonfiguration — Geräte, Schemata, Sequenzen, Rezepte, Störungen. Über `POST /api/project/export/` → ZIP-Datei.
- Datenbank — vollständiger PostgreSQL-Dump (enthält Journal, historische Trends, Chargendaten). Befehl: ``bash docker-compose exec postgres pgdump -U scada scadadb > backup_$(date +%Y%m%d).sql``
- Benutzer-Uploads — Volume `ai_uploads` (Bilder für die KI-Analyse).
- nginx-Einstellungen und `docker-compose.yml` — in Git halten.

Wie oft:

- Projektkonfiguration — vor jeder signifikanten Änderung.
- Datenbank — täglich automatisch + vor System-Updates.
- Vor einem SCADA-Versions-Update — beide Backup-Typen sind Pflicht.

Restore testen. Einmal im Quartal ein Backup auf einem Teststand ausrollen und verifizieren. Ein nie geprüft Backup ist ein unzuverlässiges Backup.

## Zugangsverwaltung

Least-Privilege-Prinzip. Jeder Nutzer erhält nur die Rechte, die er für seine Aufgabe braucht.

Typische Rollenmatrix:

Rolle	Rechte
Bediener	Schemas anzeigen + Geräte in Auto bedienen + Störungen quittieren. Kein Bearbeitungszugriff.
Schichtleiter	Alles vom Bediener + Chargen starten/stoppen + Rezepte aus einer erlaubten Liste ändern.
Technologie	Alles oben + Rezepte und Variablen erstellen/bearbeiten.
Automatisierungingenieur	Alles oben + Schemata, Sequenzen, Geräte bearbeiten.
Administrator	Vollzugriff inklusive Benutzerverwaltung und Systemeinstellungen.

Zusätzlich:

- Per-Schema-Zugang — einzelne Schemas für Nutzer, die nicht an diesem Bereich arbeiten, verbergen.
- IP-Filterung — Zugriff auf das interne Firmennetz beschränken (192.168.0.0/16 oder VPN).
- Regelmäßiger Audit — quartalsweise aktive Nutzerliste in Users prüfen, Ausgeschiedene entfernen.

## System-Updates

Korrekturer Update-Ablauf:

- Wartungsfenster planen — Update benötigt 5-15 Min Produktionsstillstand.
- Backup machen (Konfiguration + DB).
- CHANGELOG lesen — es kann API-Breaking-Changes oder Migrationen geben.
- Dry-Run ausführen — System prüft Migrationen ohne Anwendung.
- Update anwenden — über Settings → System → Updates.
- Auf Smoke-Tests warten — System führt Basischecks automatisch aus.
- Wenn etwas bricht — Rollback nutzen (verfügbar 24 h nach Update).

Niemals während einer aktiven Charge oder bei offenen kritischen Störungen aktualisieren.

## Umgang mit Störungen

Was TUN bei Alarm:

- Ruhe bewahren. Alarmbeschreibung in der Alarm Bar lesen.
- Zur Ausrüstung gehen, die ausgelöst hat.
- Zustand visuell prüfen (Leckage, Überhitzung, Geräusche).
- Ist es sicher, Acknowledge in Alarms drücken.
- Ursache beheben.
- Nach Behebung Resolve drücken.

Was NICHT TUN:

- Massenquittieren aller Alarme ungeprüft (resolve-all nur in Ausnahmefällen nach Entscheidung des Schichtleiters).
- Alarmereignisse während des Betriebs aus der Konfiguration löschen.
- Severity senken, um einen lästigen Alarm loszuwerden — stattdessen richtige Schwellen setzen.

## Aktivitätsjournal

---

Das Journal (Activity Log) ist Ihr Schutz und die Quelle der Wahrheit. Es erfasst automatisch:

- An- und Abmeldungen.
- Jeden Gerätesteuerbefehl (Start, Stop, Open, Close).
- Rezept- und Sequenzänderungen.
- Alarmauslösungen.
- Chargenstart/-stopp/-transfers.

Empfehlungen:

- Mindestens 1 Jahr aufbewahren (Anforderung vieler Qualitätsstandards).
- Regelmäßig als CSV archivieren.
- Bei Zwischenfalluntersuchung zuerst das Journal  $\pm 1$  Stunde um das Ereignis ansehen.

## Namens-Empfehlungen

---

Eine einheitliche Namenskonvention zahlt sich mit wachsendem Projekt aus.

Geräte:

- Typ-Präfix: mot\_ (motor), val\_ (valve), ai\_ (analog input), pid\_, tnk\_ (tank).
- Nummerierung mit führenden Nullen: mot\_001, mot\_002 (bis mot\_099).
- Gruppierung nach Bereich: val\_tank1\_in, val\_tank1\_out.

Sequenzen:

- Prozess-Präfix: seq\_pasteurize\_htst, seq\_cip\_alkali.

Rezepte:

- Produktname mit Parametern: Milk\_Whole\_3.5%, Cream\_30%\_HotFill.

# Glossar

Referenz der im System und in dieser Dokumentation verwendeten Begriffe.

## Allgemeine Automatisierung

SCADA (Supervisory Control And Data Acquisition) — System zur Überwachung und Steuerung industrieller Prozesse.

SPS / PLC (Programmable Logic Controller) — speicherprogrammierbare Steuerung. Industriecomputer, der Steuerlogik in Echtzeit ausführt.

HMI (Human-Machine Interface) — Mensch-Maschine-Schnittstelle. Im System ist HMI der Schema Viewer.

P&ID (Piping and Instrumentation Diagram) — Rohrleitungs- und Instrumentenfließbild.

I/O — Input/Output, Ein-/Ausgangspunkte. Physikalische oder logische Kanäle zwischen SPS und Feldgeräten.

Tag — benannter SPS-Signalname. Entspricht in SCADA der Speicheradresse, unter der ein Wert gelesen/geschrieben wird (z. B. DB10.DBDO bei Siemens).

## Geräte

AI (Analog Input) — Analogeingang. Nimmt ein kontinuierliches Signal (z. B. 4-20 mA, 0-10 V) und rechnet es in eine Messgröße um.

AO (Analog Output) — Analogausgang. Gibt ein kontinuierliches Steuersignal aus (FU-Sollwert, Ventilstellung).

DI / DO (Digital Input / Output) — Digitaleingang/-ausgang. Logisches 0/1-Signal (Taster, Endschalter, Lampe, Schütz).

PID (Proportional-Integral-Derivative) — PID-Regler. Regelalgorithmus mit drei Komponenten: proportional (Kp), integral (Ki), differenziell (Kd).

Setpoint (SP) — Sollwert, Zielwert der Regelgröße.

Process Value (PV) — aktuell gemessener Wert der Regelgröße.

VSD (Variable Speed Drive) — Frequenzumrichter (FU). Steuert die Drehzahl eines Motors.

COS (Coordination Signal) — Koordinationssignal. Logische Variable zur prozessübergreifenden Synchronisation (z. B. «Linie 1 bereit für Chargentransfer»).

Counter — Zähler (Impulse, Ereignisse). Für Durchflusszählung, Umdrehungen, Zyklen.

Interlock — Verriegelung. Bedingung, die eine Aktion blockiert (z. B. «Pumpe nicht bei geschlossenem Ventil starten»).

## Tanks

Peiltabelle (Strapping table) — Kalibriertabelle, die Füllstandssensordaten dem Tankvolumen zuordnet. Für Tanks mit nicht standardisierter Geometrie.

HH / H / L / LL — High High / High / Low / Low Low. Vier Alarmschwellen für Tanks: kritisch hoch, hoch, niedrig, kritisch niedrig.

Hydrostatischer Füllstand — Füllstand aus dem Druckdifferenz ( $\Delta P$ ) zwischen oberem und unterem Drucksensor.

## Sequenzen und Rezepte

---

Sequenz — Zustandsautomat, der die geordneten Schritte eines Prozesses beschreibt.

Schritt — ein Zustand einer Sequenz.

Übergang — Verbindung zwischen zwei Schritten mit Auslösebedingung.

Übergangsbedingung — Auslöser: Device (Gerätewert-Vergleich), Timer (Zeit), Option (Bedienertaste).

Rezept — konkrete Gerätewerte pro Schritt einer Sequenz.

Produktionsrezept — Variablenset für ein bestimmtes Produkt.

Rezeptvariable — Parameter, dessen Wert vom Produkt abhängt.

Option — bediener- ausgelöstes Steuersignal (Start-/Pause-/Skip-Button).

## Chargen und Produktionslinien

---

Charge (Batch) — Produktionseinheit, die den Prozess durchläuft. Konzept aus ISA-88.

Produktionslinie — Gerätegruppe zur Produktfertigung.

Produktionslinien-Stufe — eine Verarbeitungsstufe der Linie (z. B. Reinigen → Trocknen → Verpacken).

Chargentransfer (Batch Transfer) — Weiterreichen der Charge von einer Stufe zur nächsten. Gesteuert durch `batch_transfer_action` am Sequenzschritt.

Batch Queue — Warteschlange der zu startenden Chargen.

ISA-88 — internationaler Standard für Chargensteuerung in rezeptgesteuerter Produktion.

## PLC-Export und Siemens

---

TIA Portal — Totally Integrated Automation Portal, Siemens IDE für SPS-Projektierung.

SCL (Structured Control Language) — Siemens PLC-Sprache (IEC 61131-3 Strukturierter Text).

FB (Function Block) — Funktionsbaustein mit Logik und eigenen Instanzdaten.

DB (Data Block) — Datenbaustein zur Variablenspeicherung.

OB (Organization Block) — Einstiegspunkt für PLC-Code.

OB1 — zyklischer Haupt-OB, jeder Zyklus aufgerufen.

SimaticML — XML-Format für TIA-Portal-Export/Import.

GSD-Datei (Generic Station Description) — Hardware-Modulbeschreibung für PROFIBUS/PROFINET.

## WebSocket und API

---

REST API — Representational State Transfer, HTTP-basiert für CRUD.

WebSocket — bidirektionales TCP-Protokoll für Server-Push zum Client.

JWT (JSON Web Token) — Authentifizierungstoken mit signierten Nutzerdaten.

SSE (Server-Sent Events) — einseitiges Streaming vom Server zum Client. Vom KI-Assistenten für Streaming-Antworten genutzt.

Delta-Update — nur geänderte Felder eines Objekts senden.

Batching — mehrere Updates in eine Nachricht bündeln, um Netzlast zu senken.

## Industriestandards

---

HACCP (Hazard Analysis and Critical Control Points) — Gefahrenanalyse und kritische Kontrollpunkte, Methodik zur Lebensmittelsicherheit.

GMP (Good Manufacturing Practice) — Gute Herstellungspraxis, besonders in Pharma und Lebensmittel.

CIP (Cleaning In Place) — Reinigung vor Ort ohne Demontage der Prozessanlagen.

HTST (High Temperature Short Time) — Hochtemperatur-Kurzzeit-Pasteurisierung (72-75 °C, 15-20 s).

ISO 9001 — internationaler Qualitätsmanagement-Standard.

## Anhang A: WebSocket-Kanäle-Referenz

Kanal	Beschreibung	Richtung
ws/device/{id}	Gerätestatus	Server → Client
ws/device/{id}/control	Gerätebefehle	Bidirektional
ws/schema/{id}/devices	Alle Schema-Geräte	Server → Client
ws/schema/{id}/pipes	Schema-Rohrleitungen	Server → Client
ws/schema/{id}/sequences	Schema-Sequenzen	Server → Client
ws/schema/{id}/batches	Schema-Chargen	Bidirektional
ws/schema/{id}/accidents	Schema-Störungen	Server → Client
ws/sequence/{id}	Sequenzstatus	Server → Client
ws/sequences/all	Alle Sequenzen	Server → Client
ws/pipe/{id}	Rohrleitungsstatus	Server → Client
ws/system	Systemmetriken	Server → Client
ws/plc/status	SPS-Status	Server → Client
ws/device-status	Globaler Status	Server → Client
ws/cos/all	Koordinationssignale	Server → Client
ws/math-functions	Math-Funktionen	Server → Client
ws/activity-log	Aktivitätsprotokoll	Server → Client
ws/accidents	Alle Störungen	Server → Client

Rust WebSocket-Server (Port 8001, Proxy `/rws/ws/`):

- Read-only für Massen-Geräteaktualisierungen
- Automatischer Fallback auf Django-WebSocket
- Authentifizierung via JWT im URL-Query

## Anhang B: API-Referenz (Übersichtstabelle)

Ressource	Base URL	CRUD	Zusätzliche Aktionen
Auth	/api/auth/	—	login, register, me, refresh
Users	/api/users/		set_password, upload-avatar, roles
Roles	/api/roles/		permissions
Permissions	/api/permissions/		categories, actions
Devices (all)	/api/devices/	—	alldevices, batchwebsocket_update
Analog Inputs	/api/analog-inputs/		setmode, setsimulation, setinterlock, setmanual_value
Analog Outputs	/api/analog-outputs/		(analog)
Digital Inputs	/api/digital-inputs/		(analog)
Digital Outputs	/api/digital-outputs/		(analog)
Motors	/api/motors/		(analog)
Valves	/api/valves/		(analog)
PID Controllers	/api/pid-controllers/		(analog)
Counters	/api/counters/		(analog)
Timers	/api/timers/		(analog)
COS	/api/coordination-signals/		(analog)
VSD	/api/variable-speed-drives/		(analog)
Tanks	/api/tanks/		calculatevolume, import/exportstrapping
Tank Sections	/api/tank-sections/		—
Tank Strapping	/api/tank-strapping-entries/		—
PLCs	/api/plcs/		status, set-run-mode
Schemas	/api/schemas/		add/remove devices, sequences, options, production lines
Pipes	/api/pipes/		—
Sequences	/api/sequences/		steps, transitions, positions
Sequence Steps	/api/sequence-steps/		—
Transition Conditions	/api/transition-conditions/		simulate, clear_simulation
Recipes	/api/recipes/		—
Production Recipes	/api/production-recipes/		select-production-recipe
Recipe Variables	/api/production-recipe-variables/		—
Events	/api/events/		set_active
Accident Events	/api/accident-events/		—
Accidents	/api/accidents/		acknowledge, resolve, resolve-all
Sequence Options	/api/sequence-options/		—
Production Lines	/api/production-lines/		activate, deactivate, create_batch
Production Stages	/api/production-line-stages/		—
Batches	/api/batches/		start, pause, resume, cancel, transfer
Batch Queue	/api/batch-queue/		reorder, set_priority
Pulse Modules	/api/pulse-modules/		start, stop

Ressource	Base URL	CRUD	Zusätzliche Aktionen
Math Functions	/api/math-functions/		calculate, validate, test
I/O Points (AI)	/api/points/analog-inputs/		validate, test
I/O Points (AO)	/api/points/analog-outputs/		(analog)
I/O Points (DI)	/api/points/digital-inputs/		(analog)
I/O Points (DO)	/api/points/digital-outputs/		(analog)
Graphical Statuses	/api/graphical-statuses/		upload-icon, list-icons
Graphs	/api/graphs/		data, timescale-status
Data Points	/api/data-points/		—
HW Catalog	/api/hw-catalog/		import-gsd
HW Stations	/api/hw-stations/		auto-assign-addresses
HW Slots	/api/hw-slots/		bulk-update
HW Network	/api/hw-network/		—
TIA Deploy	/api/tia/	—	deploy, auto-addresses, address-mapping
TIA Agent Config	/api/tia-agent-config/		check-connection, agent-status
Activity Logs	/api/user-activity-logs/	R/D	recent, statistics, clear
IP Access	/api/ip-access/allowed-ips/		checkip, myips
License	/api/license/	—	status, activate, generate
System	/api/system/	—	info, health, updates, maintenance
Project	/api/project/	—	export, import, backup, restore
PLC Export	/api/plc-export/	—	preview, validate, generate
AI Sessions	/ai-api/v1/sessions		messages
AI Chat	/ai-api/v1/chat	—	messages (SSE)
AI Images	/ai-api/v1/images	—	analyze, apply
AI Settings	/ai-api/v1/settings	R/U	—

## Anhang C: Fehler- und Alarmcodes

Referenz der Codes und Status, denen Nutzer in Logs, WebSocket-Nachrichten und API-Antworten begegnen.

### Alarmschwere (Severity levels)

Feld `severity_level` des `AccidentType` bzw. `accident_event.severity_type`.

Code	Bezeichnung	Farbe	Empfohlene Reaktion
1 / LOW	Niedrig	Hellblau	Infomeldung. Keine Sofortaktion. Wird zur Statistik protokolliert.
2 /	Mittel	Gelb	Baldige Aufmerksamkeit nötig. Produktion kann weiterlaufen.
3 / HIGH	Hoch	Orange	Sofortige Aufmerksamkeit nötig. Qualitätsverlust oder Schädigungsgefahr möglich.
4 /	Kritisch	Rot	Sofortiger Sequenzstopp. Personen- oder Sachrisiko.

CRITICAL

Ereignis-Schweregrade (`SeverityType` für Events):

Code	Bezeichnung	Verwendung
INFO	Information	Statusmeldungen (Start, Stufenabschluss)
WARNING	Warnung	Nicht-kritische Abweichung
ERROR	Fehler	Kritisches Problem, Eingriff nötig

### Alarmstatus (Accident Status)

Feld `status` des `Accident`. Lebenszyklus:

NEW → ACKNOWLEDGED → IN\_PROGRESS → RESOLVED → CLOSED

Code	Bezeichnung	Beschreibung
NEW	Neu	Gerade aufgetreten, unquittiert. In der Alarm Bar sichtbar.
ACKNOWLEDGED	Quittiert	Bediener bestätigt (POST <code>/api/accidents/{id}/acknowledge/</code> ). Nicht mehr prioritär, bleibt aktiv.
IN_PROGRESS	In Bearbeitung	Ursachenbehebung läuft.
RESOLVE	Behoben	Ursache behoben (POST <code>/api/accidents/{id}/resolve/</code> ).
CLOSED	Geschlossen	Endstatus nach RESOLVED + Frist (für Reporting).

### Sequenzstatus

Felder des Sequence-Modells:

Feld	Typ	Wert	Beschreibung
<code>is_running</code>	bool	true	Sequenz aktiv, Schritte wechseln.
<code>is_pause</code>	bool	true	Pausiert, Übergänge eingefroren, Geräte halten Zustand.
<code>is_stop</code>	bool	true	Gestoppt, Geräte in sicheren Zustand gebracht.

Kombinationen:

- is\_running=true, is\_pause=false, is\_stop=false — läuft.
- is\_running=true, is\_pause=true, is\_stop=false — pausiert (fortsetzbar).
- is\_running=false, is\_pause=false, is\_stop=true — vollständig gestoppt.

## Chargenstatus (Batch Status)

Code	Bezeichnung	Beschreibung
PENDING	Wartend	In der Queue, nicht gestartet
RUNNING	Läuft	Aktive Charge
PAUSED	Pausiert	Vom Bediener zeitweise gestoppt
COMPLETED	Abgeschlossen	Alle Stufen erfolgreich durchlaufen
CANCELLED	Abgebrochen	Manuell abgebrochen
FAILED	Fehlgeschlagen	Wegen kritischer Störung abgebrochen

Prioritäten (Priority):

Code	Queue-Reihenfolge
URGENT	Läuft zuerst
HIGH	2
NORMAL	3 (Standard)
LOW	4

## Gerätemodi (Device Mode)

Gilt für Motoren, Ventile, PIDs, AI/AO/DI/DO:

Code	Beschreibung
AUTO	Steuerung über Sequenz / Systemlogik
MANUAL	Direkte Bedienersteuerung
SIMULATION	Simulierter Wert (nur zum Test)
INTERLOCK	Verriegelt (Wert nicht änderbar)

## API-Fehlercodes

Typische HTTP-Codes und ihre Bedeutung im System:

Code	Bedeutung	Typische Ursache
200 OK	Erfolg	Anfrage ausgeführt
201 Created	Erstellt	Neues Objekt angelegt
204 No Content	Erfolg ohne Body	Löschung
400 Bad Request	Fehlerhaft	Validierung fehlgeschlagen. Body ist {"field": ["error msg"]}
401 Unauthorized	Nicht authentifiziert	JWT fehlt oder ungültig
403 Forbidden	Keine Berechtigung	Authentifiziert, aber nicht autorisiert
404 Not Found	Nicht gefunden	ID existiert nicht
409 Conflict	Konflikt	Z. B. Gerät mit existierender ID anlegen

Code	Bedeutung	Typische Ursache
423 Locked	Gesperrt	Objekt von anderem Prozess gesperrt (z. B. Schema speichern bei laufender Sequenz)
500 Internal Error	Serverfehler	Bug oder Exception. docker-compose logs backend prüfen
502 Bad Gateway	Proxyfehler	Nginx erreicht Backend / Rust-Worker nicht
503 Service Unavailable	Nicht verfügbar	Wartungsmodus oder Überlast

## WebSocket-Close-Codes

Beim Schließen einer WebSocket-Verbindung können folgende Codes auftreten:

Code	Beschreibung
1000	Normaler Schluss
1001	Server geht (Neustart)
1006	Anormaler Schluss (Verbindungsverlust). Frontend verbindet automatisch neu.
1008	Richtlinienvorstoß (z. B. ungültiger JWT in URL-Query)
1011	Interner Serverfehler
4001	Custom: Authentication failed
4003	Custom: Forbidden (keine Kanalberechtigung)
4004	Custom: Channel not found

## Typen von Übergangsbedingungen (Transition Condition Type)

Code	Beschreibung
device	Nach Gerätewert (Vergleich mit Schwelle)
timer	Nach Dauer (nach Schritteintritt)
option	Nach Bedientastendruck

Vergleichsoperatoren (für type=`device`):

Symbol	Bedeutung
==	Gleich
!=	Ungleich
>	Größer
<	Kleiner
>=	Größer oder gleich
<=	Kleiner oder gleich

## Chargenaktionen am Schritt (batch\_transfer\_action)

Code	Beschreibung
NONE	Keine Aktion, Charge verbleibt auf aktueller Stufe
TRANSFER	Charge auf nächste Stufe weitergeben
COMPLETE	Charge abschließen (Endstatus)

## Tankdeckel-Formen

Für Feld `SchemaDevice.configuration.tank_visual`:

Boden (``bottom_type``):

Code	Beschreibung
FLAT	Flach
DOME	Sphärisch (nach unten gewölbt)
DISH	Korbbogen (elliptisch)
CONE	Konisch

Deckel (``top_type``):

Code	Beschreibung
FLAT	Flach
DOME	Sphärisch
DISH	Korbbogen

Dokument automatisch auf Basis der Quellcodeanalyse erstellt. Für Screenshots: Anwendung öffnen, zur angegebenen Ansicht wechseln und den Screenshot mit den beschriebenen Elementen aufnehmen.