

# Grafana als (alternative) Dashboardlösung für industrielle „Use Cases“

T3\_3000

Projekt für Angewandte Informatik

Dualen Hochschule Baden-Württemberg Mosbach

Studienpartner

AZO GmbH & Co. KG



von

Rico Kursidem

betreut von

Steffen Michalowski

January 30, 2023

# Contents

<b>1</b>	<b>Einführung</b>	<b>1</b>
<b>2</b>	<b>Technische Grundlagen</b>	<b>2</b>
2.1	Docker . . . . .	2
2.2	ACAS . . . . .	2
2.3	Grafana . . . . .	3
<b>3</b>	<b>Funktionsanalyse</b>	<b>4</b>
3.1	Einbinden von Datenquellen . . . . .	4
3.2	Interaktion zwischen Grafen . . . . .	6
3.2.1	Filterungen durch Variablen . . . . .	6
3.2.2	Links . . . . .	6
3.2.3	Annotationen . . . . .	7
3.3	Berechtigungen und Rollenkonzepte . . . . .	8
3.4	Erscheinungsbild und Design . . . . .	8

# List of Figures

3.1	Einbinden von MSSQL Datenbanken . . . . .	4
3.2	Node-Red Flow mit Rückgabe einer JSON . . . . .	5
3.3	Transformation der Sensordaten . . . . .	5
3.4	Übergabeparameter in Links . . . . .	7
3.5	Annotationen in einer Zeitreihe . . . . .	7

# 1 Einführung

Daten und der Einsatz von ihnen im industriellen Umfeld ist nicht mehr wegzudenken. Die Fähigkeit, Daten visuell darzustellen und so dem Administrator des Systems Mustererkennung möglich zu machen, ist essentiell zur Entscheidungsfindung. Auch das Monitoring von Live-Daten ist wichtig. Diese Aufgaben übernehmen Dashboards auf denen Daten in verschiedensten Formen angezeigt werden können. Grafana ist eine Technologie mit der einfach und schnell Daten aus verschiedensten Datenquellen zusammengetragen und visuell dargestellt werden können.

Eine Eigene Lösung für die Auswertung von Daten zu entwickeln ist Zeitaufwendig und das Resultat vermisst viele Funktionen welche Stand der Technik sind. Deshalb soll Grafana als kostengünstige Alternative in Betracht gezogen werden. Es gibt mehrere Anforderungen an das System welche erfüllt werden müssen, darunter das Einbinden der vorhandenen Datenquellen und Erstellung von bereits existierenden Dashboards, welche in einem Fragenkatalog festgehalten wurden. Das Ziel dieser Arbeit ist es diesen Katalog zu beantworten und einen ersten Überblick zu liefern, ob Grafana als mögliches Visualisierungswerkzeug eingesetzt werden kann.

## 2 Technische Grundlagen

### 2.1 Docker

Docker ist eine Open Source Virtualisierungssoftware mit welcher die Ausführung von Applikationen in Containern möglich ist. Jede Anwendung bekommt einen eigenen Container mit individueller Umgebung und läuft dann, parallel zu anderen Containern, auf dem selben Host. Dockercontainer sind leichtgewichtiger da sie sich, anders als VMs, ein gemeinsames Betriebssystem teilen. Außerdem werden redundante Dependencies von einzelnen Containern nicht dupliziert sondern als gemeinsame Ressource abgespeichert. Ein anderer Vorteil von Docker Containern ist der einfache Transport. In einem Image wird hierfür alle nötigen Informationen abgespeichert. Die Dependencies und der Code der Anwendung. Dann muss nur das Image übertragen, ein Container daraus erstellt und auf einem beliebigen Dockersystem gestartet werden und die Applikation läuft.

Das Testsystem auf welches diese Arbeit basiert nutzt Docker zur bereitstellung zahlreicher Dienste. Wichtig für diese Arbeit sind der Grafana, Node-Red und Datenbank Container.

### 2.2 ACAS

ACAS ist eine intern bei AZO entwickelte Automatisierungsplattform. Sie soll eine Schnittstelle zwischen Bediener und Anlage sein. Dabei sollen diverse Aufgaben wie Virtualisierung, Fehlererkennung und Monitoring erfüllt werden. Grafana soll als Monitoringtool eingesetzt werden um Anlagen im Betrieb zu überwachen und Fehlerursachen frühzeitig zu erkennen um Schäden zu minimieren.

### 2.3 Grafana

Grafana ist eine Open Source Software mit der dynamische Dashboards zur Überwachung und Auswerten von Daten angefertigt werden können. Grafana bietet hierzu eine zugängliche und simple Benutzeroberfläche zur Erstellung von einzelnen Diagrammen die Panels genannt werden. Grafana bietet state of the Art funktionen wie da Verschieben und individualisieren von Oberflächen und zahlreiche Datenquellen, sodass es einen solide Lösung für Monitoring, Fehlererkennung und Analyse darstellt.

## 3 Funktionsanalyse

### 3.1 Einbinden von Datenquellen

In dem bereits bestehenden System, in welches Grafana integriert werden soll, liegen mehrere Datenquellen vor. Sichergestellt sollte der Import von Daten in Microsoft SQL Datenbanken und aus Rest APIs über Node-Red. Grafana unterstützt Microsoft SQL nativ als Datenquelle neben zahlreichen anderen Datenbanken. Die Datenbank kann unter dem Menüpunkt Daten Sourcen eingebunden werden und es können weitere Einstellungen zur Authentifizierung gemacht werden. Um diese Datenquelle zu verwenden, können in den Panels unter "Query" die Datenquelle ausgewählt und ein SQL Befehl abgesetzt werden um Daten anzuzeigen. In Abbildung 3.1 ist der "Query" Reiter abgebildet mit einem SQL Befehl.



Figure 3.1: Einbinden von MSSQL Datenbanken

Die zweite Datenquelle welche als Anforderung an Grafana Eingebungen werden muss ist eine Node-Red Schnittstelle zu einem SPS-System. Hierbei sollen aktuelle Sensorwerte einer Anlage über eine Node-Red API abgerufen und in Grafana angezeigt werden. In Abbildung 3.2 ist der Node-Red Flow abgebildet, welcher Sensordaten aus einer SPS ausliest und als JSON an den Aufrufer der API zurückgibt. Die JSON besteht aus einem Array von Objekten mit vier Attributen. Name, Wert, Einheit und Farbe. Die Farbe kann als Statuswert genutzt werden, wobei beispielsweise ein zu kleiner Druck in Warnfarben angezeigt werden soll.

### 3.1. EINBINDEN VON DATENQUELLEN

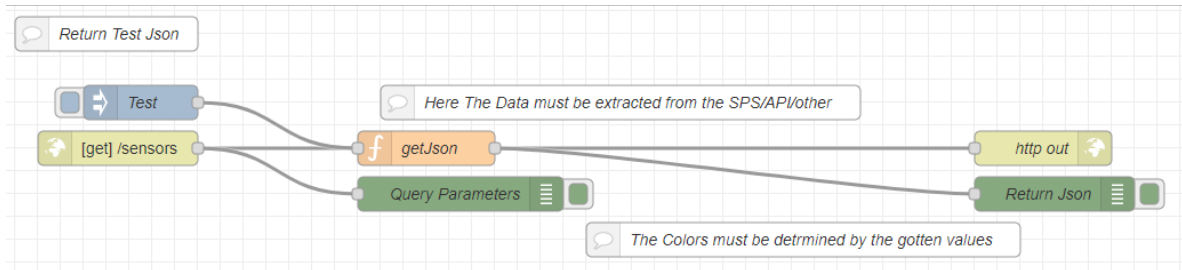


Figure 3.2: Node-Red Flow mit Rückgabe einer JSON

Um JSON Daten in Grafan verarbeiten zu können muss ein Plugin integriert werden. Hierbei fiel die Entscheidung auf das "Infinity" Plugin von Sriramajeyam Sugumaran da es neben der Integration von JSON Daten auch XML, CSV und weitere Datenformate unterstützt und regelmäßige Updates bekommt. In den Data Sourcen Einstellungen können weitere Einstellungen vorgenommen werden, die für den in dieser Arbeit nicht von Nöten sind. Wird die Data Source in einem Panel dann ausgewählt, muss die Adresse des API Aufrufs und optionale Übergabe Parameter eingetragen werden. Bei jeder Aktualisierung wird der API Abfrage getätigt und die Daten visualisiert.

Um die Sensordaten ansehnlich zu präsentieren, können Transformationen genutzt werden. Im Transformationen Reiter kann die Funktion "Rows from Fields" wie in Abbildung 3.3 ausgewählt werden, welche die Werte gewisser Spalten zu Konfigurationswerten übernimmt.

Rows to fields	
Field	Use as
color	Color (auto) ▾
name	Field name (auto) ▾
unit	Unit (auto) ▾
value	Field value (auto) ▾

Figure 3.3: Transformation der Sensordaten




## 3.2 Interaktion zwischen Grafen

### 3.2.1 Filterungen durch Variablen

Grafana bietet dashboardweite Variablen, welche zuvor festgelegte Werte annehmen können. Diese Werte sind entweder manuell eingetragen oder aus Datenquellen ausgelesen. In den Dashboard-Einstellungen können Variablen angelegt und ihre Wertauswahl konfiguriert werden. Jede Variable kann ganz oben auf einem Dashboard angezeigt und dort anhand eines Drop-Down-Menüs gesetzt werden.

### 3.2.2 Links

Bei Verlinkungen werden zwischen drei verschiedenen Links unterschieden. Dashboard Links, Panel Links und Data Links. Dashboardlinks können in den Dashboard-Einstellungen erstellt werden und erscheinen dann in der oberen rechten Ecke des Dashboards. Um einen Link des Zieldashboards zu erstellen muss auf das Teilensymbol  neben dem Namen geklickt werden und der Link kopiert werden. Panel Links sind Verlinkungen, welche an einem Panel angehängt sind. Sie erscheinen in der oberen linken Ecke eines Panels. Data Links können innerhalb eines Panels angelegt werden und können auf Informationen zugreifen, welche Daten in der Grafik gedruckt wurden.

Um Informationen von einem Dashboard zum nächsten zu übermitteln können die Dashboardweiten Variablen verwendet werden. In Tabelle 3.4 ist der Syntax der verschiedenen Übergabemöglichkeiten aufgeführt. Es können konstante Werte, Dashboardvariablen oder Datenpunkte übergeben werden wobei bei der Eingabe von einem Dollarzeichen Grafana bereits alle möglichen Felder in einem Drop-Down-Menü vorschlägt und bei Auswahl den korrekten Syntax einträgt. `varName` ist dabei die Variable des Zieldashboards welche den gewählten Wert annehmen soll.

### 3.2. INTERAKTION ZWISCHEN GRAFEN

Variable	Übergabe im Link
Konstante	<code>var-varName=value</code>
Variable	<code>var-varName=\$DashBoardVar</code>
Datenpunkt	<code>var-varNname=\$__data.fields.FieldName</code>
Zeitintervall als url	<code>\$__url_time_range</code>
Intervall Start	<code>\$__timeTo()</code>
Intervall Ende	<code>\$__timeFrom()</code>

Figure 3.4: Übergabeparameter in Links

es kann auch auf das selbe Dashboard verlinkt und die Variablen gesetzt werden, wodurch die manipulation von Panelen über den Klick auf ein anderes Panel ermöglicht wird.

#### 3.2.3 Annotationen

Annotationen können genutzt werden um bestimmte Ereignisse aus Zeitreihen auf dem gesamten Dashboard hervorzuheben. Diese Ereignisse können durch eine Datenquelle als Zeitabhängige Datenpunkte eingelesen werden. Sie werden dann auf jedem Zeitreihen Grafen markiert wie in Abbildung 3.5 die Fehlermeldungen in rot.

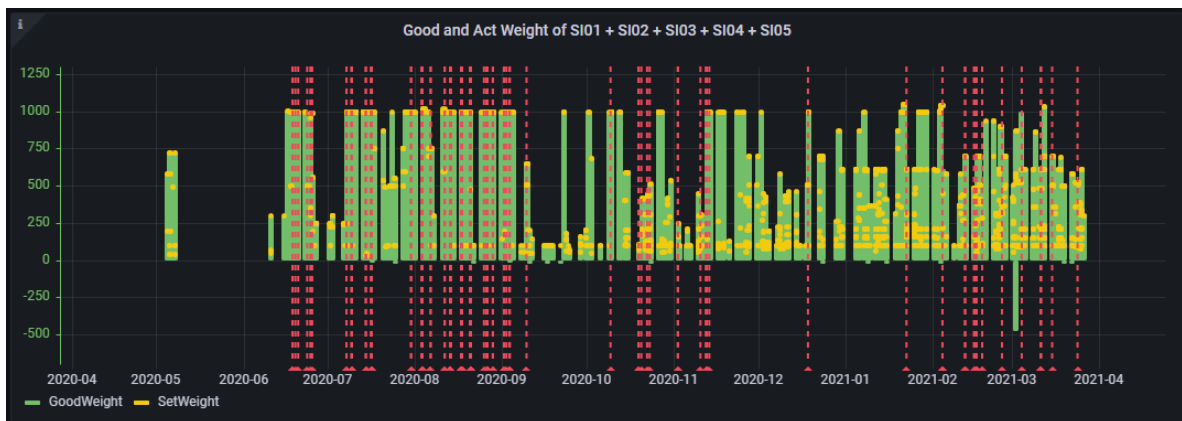


Figure 3.5: Annotationen in einer Zeitreihe

### **3.3 Berechtigungen und Rollenkonzepte**

### **3.4 Erscheinungsbild und Design**

# Bibliography

- [1] .
- [2] In: ().
- [3] .