

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

T3_3000

Projekt für Angewandte Informatik

Duale Hochschule Baden-Württemberg Mosbach

Studienpartner

AZO GmbH & Co. KG



von

Rico Kursidem

betreut von

Steffen Michalowski

21. März 2023

Inhaltsverzeichnis

1	Einführung	2
2	Technische Grundlagen	3
2.1	Docker	3
2.2	ACAS	3
2.3	Grafana	4
3	Funktionsanalyse	5
3.1	Einbinden von Datenquellen	5
3.2	Interaktion zwischen Grafen	6
3.2.1	Filterungen durch Variablen	6
3.2.2	Links	7
3.2.3	Annotationen	8
3.3	Berechtigungen und Rollenkonzepte	9
3.4	Erscheinungsbild und Design	9
4	Fazit	10

Abbildungsverzeichnis

3.1	Einbinden von MSSQL Datenbanken	5
3.2	Node-Red Flow mit Rückgabe einer JSON	6
3.3	Annotationen in einer Zeitreihe	8

Tabellenverzeichnis

1.1	Untersuchte Anforderungen an Grafana	2
3.1	Übergabeparameter in Links	8

Abstract

Deutsch

Grafana soll als Visualisierungslösung des intern entwickelte System ablösen. Dafür muss es mehrere Anforderungen erfüllen, die in dieser Arbeit überprüft werden sollen. Das Ziel ist es zu ermitteln, ob Grafana die benötigten Funktionen abdeckt, die die interne Lösung erfüllte. Es wurde eine Grafanaumgebung als Testumgebung aufgesetzt und mehrere Test-Dashboards angefertigt, die die Anwendung von Grafana in den Anforderungsgebieten abbildet.

Die Arbeit zeigte, dass alle Funktionen von Grafana abgedeckt werden können. Das Design ist die einzige nicht erfüllte Anforderung, da Grafana keine individuelle Farbgebung erlaubt. Durch die Analyse der Funktionen stellte sich heraus, dass Grafana dazu geeignet ist, die interne Visualisierungssoftware zu ersetzen.

English

Grafana is intended to replace the internally developed system as a visualization solution. To do this, it must meet several requirements, which are to be examined in this work. The goal is to determine whether Grafana covers the required functions that the internal solution fulfilled. A Grafana environment was set up as a test environment and several test dashboards were created that depict the use of Grafana in the requirement areas.

The work showed that all functions can be covered by Grafana. The color theme is the only requirement not met as Grafana does not allow custom coloring. By analyzing the functions, it turned out that Grafana is suitable to replace the internal visualization software.

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 essenziell zur Entscheidungsfindung. Auch das Monitoring von Live-Daten ist wichtig. Eine Möglichkeit zur Visualisierung von Daten sind Grafen, Tabellen und andere grafische Darstellungen. Diese Grafen müssen daraufhin in Dashboards angezeigt und geordnet werden, um dem menschlichen Nutzer das beste Verständnis der Daten zu versichern.

Eine eigene Lösung für diese Aufgabe zu implementieren ist umfangreich und führt oftmals zu nicht zufriedenstellenden Ergebnissen. Deshalb fiel der Blick auf die Open-Source Lösung Grafana, welche das einfache und schnelle einbinden von Datenquellen und Visualisierung von Daten ermöglicht. Grafana soll bei AZO in ein Projekt eingebunden werden und muss deshalb Anforderungen erfüllen. Das Ziel dieser Arbeit ist es die Anforderungen zu betrachten, zu beschreiben und eine erste Informationsbasis zu schaffen, auf der entschieden werden kann, ob Grafana sich als Visualisierungswerkzeug für AZO eignet. In Tabelle 1.1 wird dargestellt, in welchem Kapitel die Anforderungen beschrieben werden.

Anforderung	Kapitel
Einbinden verschiedener Datenquellen	3.1
Grafeninteraktion	3.2
Filtern von Grafen und ganzen Dashboards	3.2.1
Gruppenmanagement	3.3
Verlinkung auf andere Dashboards	3.2.2
Erscheinungsbild an das bestehende Automatisierungssystem anpassen	3.4

Tabelle 1.1: Untersuchte Anforderungen an Grafana

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 demselben Host. Docker Container 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 werden 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 System mit der Docker Engine gestartet werden und die Applikation läuft.

Das Testsystem, auf welchem 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 Monitoring Tool 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 eine solide Lösung für Monitoring, Fehlererkennung und Analyse darstellt.

Grafana läuft in einem Docker Stack zusammen mit einem Node-Red und Datenbank Container. Der Container verwendet die Version 9.3.0 von Grafana.

3 Funktionsanalyse

3.1 Einbinden von Datenquellen

Grafana soll als Visualisierungswerkzeug in ACAS, dem bereits bestehenden System, integriert werden und die bereits bestehenden Datenquellen einbinden. Die Anforderungen an das System beinhalten die Anzeige von Daten aus einer Microsoft SQL Datenbank und Sensordaten, welche von einer SPS geliefert werden. Das Integrieren einer MSSQL Datenbank ist bereits nativ von Grafana unterstützt. Zunächst muss die Datenbank in Grafana als Datenquelle eingebunden werden. Wenn alle Verbindungs- und Authentifizierungsinformationen konfiguriert sind, kann die Datenquelle in den Paneleinstellungen verwendet werden. In Abbildung 3.1 ist eine exemplarische Anfrage an den SQL Server abgebildet, bei dem die Anzahl der unterschiedlichen Wiegefehler ausgegeben wird. Die Daten, die durch die Anfrage zurückgegeben werden, können dann in verschiedensten Diagrammen visualisiert werden.

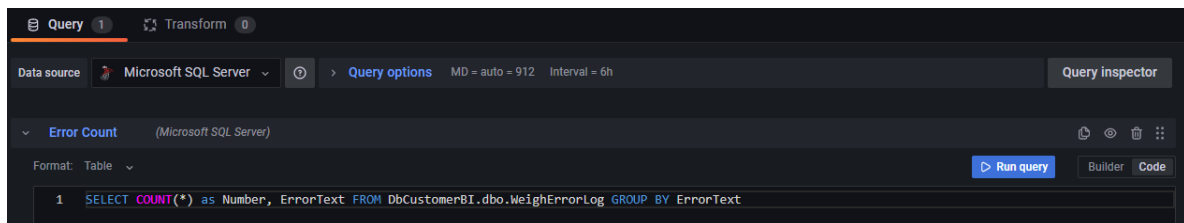


Abbildung 3.1: Einbinden von MSSQL Datenbanken

Um Daten aus einer SPS auf einem Grafanadashboard anzeigen zu lassen, müssen die Daten zunächst in eine Datenbank geschrieben und dann nach Grafana exportiert werden, da Grafana keine native Möglichkeit besitzt, MQTT Daten einzulesen. Um den Datentransport zu machen, wird ein Node-Red Server verwendet. In Abbildung 3.2 ist der Node-Red-Flow abgebildet. Zunächst werden die Daten aus der SPS gelesen. Dies wird möglich durch eine native S7-Kopplung die Daten aus einer SPS in Node-Red

3.2. INTERAKTION ZWISCHEN GRAFEN

portiert. Im Anschluss werden die Daten über einen MQTT Broker in eine InfluxDB geschrieben die in Grafana als Datenquelle eingebunden werden kann, wie die Microsoft SQL DB bevor.

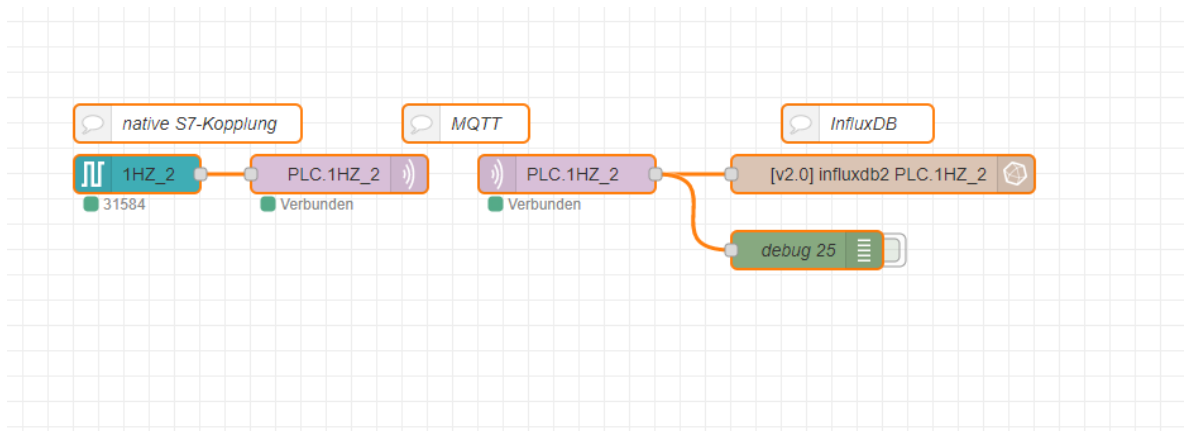


Abbildung 3.2: Node-Red Flow mit Rückgabe einer JSON

Um einzelne Splatens aus der Datenbank als Konfigurationswert wie Farbe oder Einheit für den Messwert Wert zu verwenden, können Transformationen verwendet werden. Um Schwellenwerte zu laden um beispielsweise kritische Sensorwerte in einer anderen Farbe darzustellen können Thresholds festgelegt werden. Werden diese in Grafana statisch erstellt, können multiple untere Schwellenwerte und Bereiche mit verschiedenen Farben erstellt werden. Mit Daten aus einer Anfrage kann bis jetzt mit der „Config from Rows“ oder „Rows from Fields“ nur ein Schwellenwert festgelegt werden. Allerdings befinden sich diese Funktionen in einem Beta Stadium und eine erweiterte Funktionalität ist zu erwarten. [1]

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 Dashboardeinstellungen 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. INTERAKTION ZWISCHEN GRAFEN

Um die erstellten Variablen in beispielsweise SQL Anfragen zu verwenden, kann ein Dollarzeichen und der Variablenname eingegeben werden (Beispiel: `$variableName`)[2]. Soll die Variable innerhalb eines durchgehenden Strings verwendet werden, können geschweifte Klammern genutzt werden, um die Variable vom Text zu trennen (Beispiel: `${variableName}`)[2]. Eine Variable kann auch mit einer Auswahlfunktion generiert werden, um mehrere Parameter in dieser Variable zu speichern. So können die Daten mehrerer Sensoren oder Anlagen gezeigt werden. In einer SQL Abfrage können diese mit dem IN Schlüsselwort genutzt werden (Beispiel: `${variableName}`)[2]. Außerdem kann ein Panel oder eine ganze Row als „Repeating by variable“ konfiguriert werden, um einmal pro Auswahlparameter der Variable erstellt zu werden. Sind beispielsweise mehrere Sensoren ausgewählt, wird für jeden Sensor ein Panel oder eine Row erstellt.

3.2.2 Links

Bei Verlinkungen werden zwischen drei verschiedenen Links unterschieden. Dashboard Links, Panel Links und Data Links. Dashboard Links können in den Einstellungen des Dashboards erstellt werden und erscheinen dann in der oberen rechten Ecke des . Um einen Link des Zieldashboards zu erstellen, muss auf das Teilen Symbol  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 gedrückt wurden.

Um Informationen von einem Dashboard zum nächsten zu übermitteln können Dashboardweite Variablen verwendet werden. In Tabelle 3.1 ist der Syntax der verschiedenen Übergabemöglichkeiten aufgeführt. Es können konstante Werte, Dashboard Variablen 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>

Tabelle 3.1: 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 3.3 die Fehlermeldungen in rot.

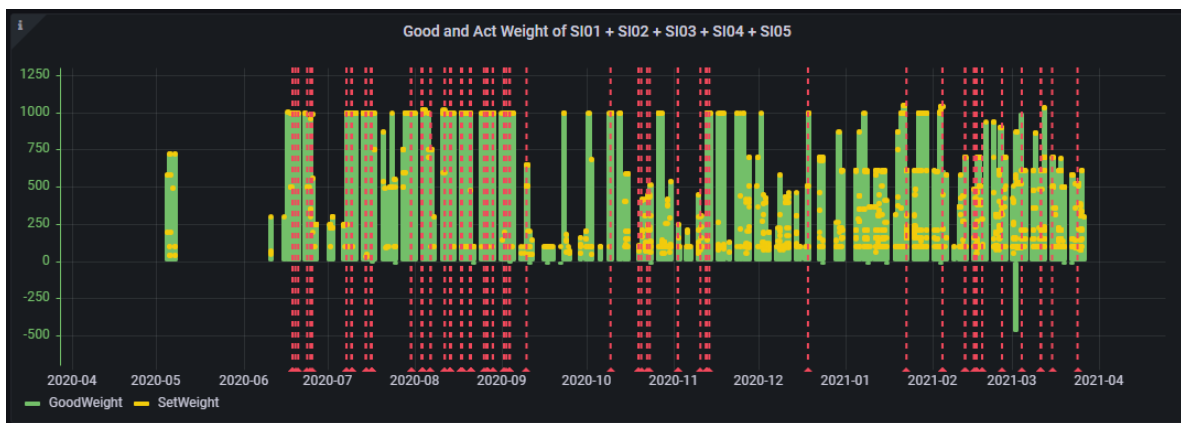


Abbildung 3.3: Annotationen in einer Zeitreihe

3.3 Berechtigungen und Rollenkonzepte

Grafana stellt drei Rollen zu Verfügung. Admin, Editor und Viewer. Viewer haben nur die Möglichkeit, Dashboards zu besuchen während Editoren sie bearbeiten können. Administratoren haben darüber hinaus Berechtigungen zur Serverweiten Einstellungen wie Nutzereinstellungen, Plugins und Datenquellen.

Jeder Nutzer hat eine default Rolle, welche Serverweit gilt, jedoch können die Berechtigungen auch von Dashboard zu Dashboard überschrieben werden um Rechte individuell festzulegen. Um Rollenkonzepte anzulegen können Nutzer zu Teams zugewiesen werden welche auch auf jedem Dashboard Rollen zugeteilt bekommen können.

3.4 Erscheinungsbild und Design

Grafana erlaubt die Auswahl zwischen einem Dark und Light Theme, allerdings gibt es keine offiziell unterstützte Funktion, Hintergrund, Schriftfarbe und weitere Konfigurationen zu verändern. Es gibt die Möglichkeit, die Veränderungen direkt in den TypeScript-Dateien zu verändern. Das bringt jedoch mehrere Probleme mit sich. Zum einen könnte nach jedem Update eine Veränderung der Struktur dazu führen, dass die Änderungen der Dateien unwirksam werden. Zum anderen müssen nach jedem Update alle Änderungen neu eingespielt werden da die Datei vom Update überschrieben wird. Das größte Problem ist jedoch, dass der Grafana Container von Lokalen Quellen erstellt werden muss, wodurch die Grafana Anwendung nicht immer auf dem neusten Stand bleibt.

Die Änderungen des Erscheinungsbildes kann jedoch mit dem „Boom Theme“ angepasst werden. Allerdings sind diese Anpassungen nur für ein Dashboard einstellbar und können von Endnutzern ebenfalls abgeändert werden. Aufgrund dessen ist die Anpassung von Grafana an den firmeninternen Designstil nicht möglich.

4 Fazit

Grafana verspricht eine nutzerfreundliche und schnelle Erstellung von Dashboards zur Visualisierung von industriellen Daten. Dieses Versprechen wird durch die zahlreichen Datenquellen, Visualisierungsformen und Erstellungsmenüs gehalten. Es erfüllt alle Anforderungen für die Implementation der vorhandenen Datenquellen und enthält genug Wege zur dynamische Interaktion von einzelnen Paneelen. Außerdem ist ein Rollenkonzept bereits mitgeliefert. Das Design der Oberfläche lässt sich nicht in die bereits bestehende Erscheinungsbild von ACAS integrieren.

Grafana soll in den folgenden Monaten in die ACAS Umgebung eingeführt und dort als Visualisierungstool verwendet werden. Trotz der fehlenden Designkonfiguration bietet Grafana viele Funktionen die in ACAS hilfreich sein werden.

Literatur

- [1] szymonzy1. *Config from query result - multiple thresholds*. <https://community.grafana.com/t/config-from-query-result-multiple-thresholds/74798>. Online; accessed 02.02.2023.
- [2] Grafana. *Variable syntax*. <https://grafana.com/docs/grafana/latest/dashboards/variables/variable-syntax/>. Online; accessed 02.02.2023.
- [3] Grafana. *Grafana Labs Documetation*. <https://grafana.com/docs/grafana/latest/>. Online; accessed 07.03.2023.
- [4] Grafana. *Grafana GitHub*. <https://github.com/grafana/grafana/issues>. Online; accessed 07.03.2023.