

Überführung der Industrie 4.0 Verwaltungsschale in die Praxis anhand zweier industrieller Anwendungen

M.Sc. **Christian Schelter**, Fraunhofer ESK, München;

Kurzfassung

Durch das Abbilden von physischen Komponenten auf virtuelle Verwaltungsschalen können Unternehmen sowohl einzelne Komponenten, als auch gesamte Maschinen und Fertigungsstraßen dezentral verwalten.

Die Umsetzung der Verwaltungsschalen wurde hierbei vom ZVEI spezifiziert und in der Theorie festgeschrieben. Allerdings beschreibt die Spezifikation nicht alle Aspekte, die bei der Erstellung einer Verwaltungsschale beachtet werden müssen. Die Spezifikation enthält beispielsweise nicht die Kommunikation zwischen Verwaltungsschale und Komponente oder welche Daten in das Manifest des Headers oder in das Manifest eines Teilmodells gespeichert werden sollten. Daher kann die Spezifikation alleine nicht als Anleitung zur Umsetzung verwendet werden. Aus diesem Grund hat das Fraunhofer ESK durch zwei Projekte die praktische Umsetzung von Verwaltungsschalen durchgeführt und die dabei nicht spezifizierten Aspekte analysiert und mögliche praktikable Lösungswege erarbeitet.

Ferner wurde ein Konzept entwickelt und umgesetzt, welches Verwaltungsschalen aufgrund von Modellen generieren kann und so einzelne Teile oder gesamte Schalen wiederverwendbar eingesetzt werden können.

1. Verwaltungsschale in der Theorie

Durch den Einsatz von Verwaltungsschalen im industriellen Umfeld, werden physische Komponenten in virtuelle abgebildet und verwaltet [1]. Die Verwaltungsschale speichert hierbei nicht nur veränderliche, sondern auch statische Daten über eine Komponente, wie beispielsweise eine Komponenten-ID, CAD-Zeichnungen oder Dokumente, die zur Verwaltung der Schale benötigt werden.

Die Struktur einer Verwaltungsschale ist für jede Komponente stets identisch aufgebaut und besitzt einen *Header* mit Manifest, in der für die gesamte Komponente relevante Daten vorgehalten werden, sowie einen *Body*, der durch den Komponentenmanager dynamisch an die Komponente mit ihrem Verhalten und Kommunikationsstrukturen angepasst wird (vgl. Bild. 1). Wichtig ist, dass zwischen Komponente und Verwaltungsschale keine Eins-zu-eins-Beziehung besteht, sondern eine Komponente diverse Verwaltungsschalen für

unterschiedliche Daten und Modelle besitzen kann. Diese verschiedenen Verwaltungsschalen können wiederum in anderen Verwaltungsschalen zusammengefasst und so hierarchisch aufgebaut sowie gegenseitig referenziert werden. Hierdurch kann sowohl ein einzelner Sensor, als auch eine gesamte Maschine oder Fertigungsstraße durch eine Verwaltungsschale abgebildet und verwaltet werden.

Für den konkreten Einsatz von Verwaltungsschalen muss die zu verwaltende Komponente stets über eigene Kommunikationsmöglichkeiten verfügen. Aus diesem Grund ist es zwingend erforderlich, dass die verwaltende Komponente immer eine Industrie 4.0-Komponente, nach der Definition des ZVEI, darstellt [2].

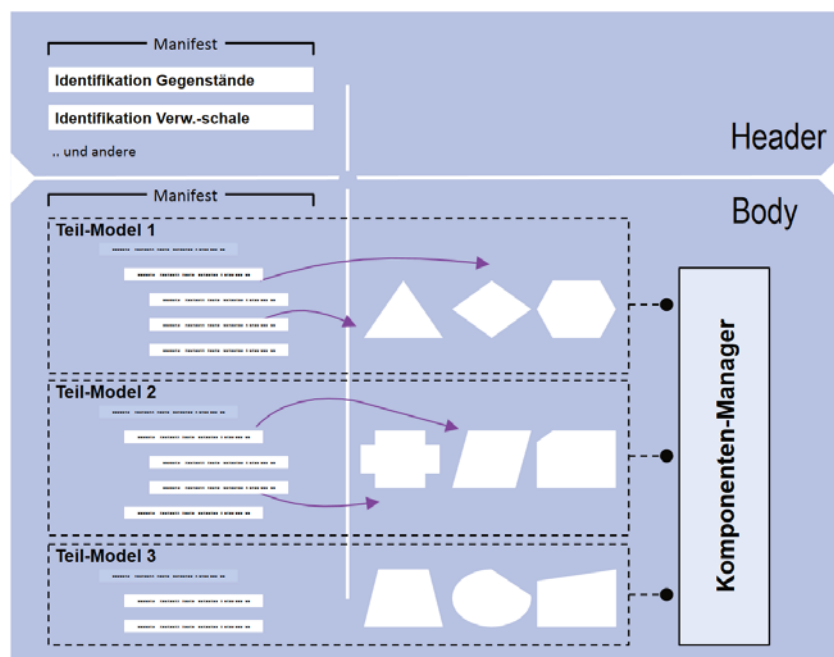


Bild 1: Grobstruktur der Verwaltungsschale. Quelle: [4]

2. Praxisrelevante Erweiterungen der Theorie

Die Spezifikation RAMI 4.0 beschreibt eine Vielzahl von Funktionalitäten und Konzepten der Verwaltungsschale für eine Industrie 4.0-Komponente [1][2]. Neben diesen standardisierten Funktionen existieren explizit nicht standardisierte Teilaspekte, die bei einem Entwurf einer Verwaltungsschale für jede Komponente, in Abhängigkeit ihrer Umgebung, erwogen und spezifiziert werden müssen.

Im Folgenden werden einige dieser zu spezifizierenden Aspekte vorgestellt.

2.1 Kommunikation mit der Komponente

Ein wesentlicher nicht genau spezifizierter Aspekt ist die Kommunikation zwischen Verwaltungsschale und Industrie 4.0-Komponente.

Eine Standardisierung der Kommunikation mit Festlegung eines spezifischen Protokolls kann aufgrund der Festlegung der Protokolle durch die Komponente und nicht durch die Verwaltungsschale ausgeschlossen werden. Verbindungen können beispielsweise durch Feldbusse oder TCP-Verbindungen mit komplett unterschiedlichem Aufbau umgesetzt werden. Wird ein Protokoll verwendet, welches starr auf das Master-Slave Verfahren setzt (z. B. Profibus), so kann die Komponente nicht direkt angesprochen werden, sondern die Daten ausschließlich am Master abgegriffen werden.

Für die Verbindung zwischen Komponente und Verwaltungsschale muss darauf geachtet werden, dass diese möglichst einfach gehalten und nicht durch performance-intensive Protokolle und Zwischenstationen unnötig verlangsamt wird. So kann zum Beispiel der Einsatz von OPC-UA zwischen Komponente und Verwaltungsschale aufgrund des benötigten OPC-UA Servers im Sensor und einer Implementierung in der Verwaltungsschale einen hohen Overhead bedeuten.

2.2 Auswahl von Werten

Die Auswahl der zu speichernden Werte ist ein wesentlicher Aspekt bei der Erstellung von Verwaltungsschalen, da diese von der Komponente, der Maschine, der Umgebung, als auch von den gewünschten Erkenntnissen abhängen.

Die Speicherung möglichst vieler Werte ist, für eine nachträgliche Analyse, unter Umständen sinnvoll, allerdings ist die Speicherung aller Werte nicht Aufgabe der Verwaltungsschale. Ferner sollten wichtige Daten explizit ausgewählt und zu sinnvollen und aussagekräftigen Werten aggregiert und anschließend in der Verwaltungsschale gespeichert werden. Hierbei ist zu beachten, dass Live-Daten, aber keine Prozessdaten in einer Verwaltungsschale gespeichert werden.

Neben der Speicherung von Live-Daten dient die Verwaltungsschale zum Speichern von Verwaltungsdaten einer Komponente. Hierzu gehören unter anderem Daten wie Seriennummern, CAD-Zeichnungen und Datenblättern.

Aufgrund der unterschiedlichen Daten und Aktualisierungsraten kann es Sinn machen, mehrere Verwaltungsschalen für eine einzelne Komponente zu erstellen. Jede Verwaltungsschale speichert in diesem Fall nur einen Teil der anfallenden Daten und referenziert im Gegenzug allerdings die anderen Verwaltungsschalen der gleichen Komponente. Aufgrund der unterschiedlichen Speicherorte der Verwaltungsschalen, können

so Live-Daten näher an der Komponente und statische Daten in der Cloud gespeichert werden.

2.3 Komponentenmanager

Der Komponentenmanager dient der Initialisierung der Verwaltungsschale, indem er die Submodelle, sowie die Kommunikationsschnittstellen zur Verfügung stellt. Jedes Submodell muss vor der Verwendung spezifiziert und definiert werden. Hierfür bieten sich die eCI@ss Klassifikationen an, die auch bei fehlenden Klassifizierungen erweitert werden können [3].

Neben den Submodellen stellt der Komponentenmanager die Kommunikationsschnittstellen zwischen Verwaltungsschalen, sowie Verwaltungsschale und Komponente bzw. ERP-System zur Verfügung. Hierfür müssen sowohl die zur Verfügung gestellten Dienste nach dem SOA-Prinzip (Service-oriented architecture), als auch die Schnittstellen und die verwendeten Sprachen für die Kommunikation definiert werden.

2.4 Werte im Manifest des Headers

Jede Verwaltungsschale besitzt sowohl ein Manifest im Header, als auch ein Manifest pro Submodell im Body.

Bevor eine Verwaltungsschale für eine Maschine umgesetzt werden kann, müssen die Daten jedes Manifests in der Verwaltungsschale feststehen. Hierbei ist besonders darauf zu achten, dass bestimmte Parameter aggregiert werden. Gelten Parameter für mehrere Submodelle, so sollten diese ebenfalls im Manifest des Headers der Verwaltungsschale abgelegt werden.

2.5 Discovery Service

Neben den reinen Verwaltungsschalen wird im Netzwerk des Betreibers ein eigener *Discovery Service* benötigt, der die bestehenden Verwaltungsschalen verwaltet und über Suchfunktionen auffindbar macht. Hierbei ist es wichtig, Verwaltungsschalen aufgrund von Parametern, als auch aufgrund von bereitgestellten Services, nach dem SOA-Prinzip, suchen zu können.

Eine Standardisierung des Discovery Service ist aktuell für Verwaltungsschalen nicht vorgesehen, da hierbei oft firmenspezifische Aspekte wie beispielsweise bestehende Services verwendet oder angebunden werden müssen.

2.6 Kommunikation zwischen Verwaltungsschalen

Neben dem Auffinden der Verwaltungsschalen (Discovery Service) und der Zurverfügungstellung der API zum Ansprechen der Schalen (Komponentenmanager) muss zusätzlich die Kommunikation zwischen den Verwaltungsschalen bei der Umsetzung mit betrachtet werden.

Hierbei sind drei verschiedene Szenarien zu unterscheiden:

Das erste Szenario ist der Austausch von Daten zwischen Verwaltungsschalen auf gleicher hierarchischer Ebene mit unterschiedlichen Komponenten. Das zweite Szenario ist der Austausch von Daten zwischen hierarchisch aufgebauten Verwaltungsschalen, indem eine Verwaltungsschale beispielsweise die komplette Maschine verwaltet, untergeordnete Verwaltungsschalen hingegen spezifische Sensoren innerhalb dieser Maschine. Das dritte Szenario ist der Austausch zwischen zwei Verwaltungsschalen auf gleicher hierarchischer Ebene, die allerdings die gleiche Komponente verwalten, um unterschiedliche Daten an unterschiedlichen Orten zu speichern.

2.7 Erstellung von Verwaltungsschalen

Ein weiterer wesentlicher Aspekt für die Erstellung von Verwaltungsschalen ist eine möglichst automatisierte Erstellung, um identische Teile nicht aufwendig manuell konfigurieren zu müssen. Hierfür muss ein Ansatz entwickelt werden, der Verwaltungsschalen ohne Einschränkungen, aber möglichst automatisiert, generiert, indem es beispielsweise Attribute, Submodelle, Kommunikationsschnittstellen, Bereiche oder komplette Verwaltungsschalen aus bestehenden Verwaltungsschalen oder Modellen dieser übernehmen kann. Das manuelle Anlegen und Verwalten von Parametern der Verwaltungsschalen stellt hier keinen praktikablen und wartbaren Weg dar.

3. Überführung der Theorie in die Praxis: Protoypische Umsetzung

Am Beispiel von zwei Projekten, die bei Fraunhofer ESK zum Erstellen von Verwaltungsschalen für bestehende Maschinen bearbeitet werden, wird die prototypische Umsetzung genauer erläutert. Aus den Erfahrungen durch die Konzeptionierung und Umsetzung der Verwaltungsschalen, sind Lösungsansätze für die in Kapitel 2 beschriebenen Erweiterungen des Standards entstanden.

Als Programmiersprache für die Umsetzung von Verwaltungsschalen wurde *JavaEE* gewählt, da diese mit ihren Konzepten und angebotenen Servern bereits viele wichtige Aspekte abdeckt, oder diese einfach umsetzen kann. Da der Server in den behandelten

Fällen zum Verwalten der Verwaltungsschalen nicht in einer Maschine oder auf einer Komponente selbst ausgeführt wird, sondern im Rechenzentrum des Betreibers der Maschine oder in der Cloud steht, sind für diese Umsetzung genügend Ressourcen vorhanden. Im Folgenden werden die verwendeten Lösungsansätze, für die in Kapitel 2 vorgestellten Erweiterungen, aufgezeigt.

3.1 Kommunikation mit der Komponente

Für die Kommunikation zwischen Verwaltungsschale und Komponente ist ein einheitlicher Standard oder eine einheitliche Vorgehensweise nicht zielführend, da diese Interaktionen in der Regel sehr heterogen und spezifisch für einzelne Komponenten sind. Für die Umsetzung in beiden Projekten wurde eine REST-Schnittstelle umgesetzt. Diese ist als initiale Implementierung auf beiden Seiten einfach umsetzbar und ermöglicht so eine schnelle Anpassung an beliebige Interfaces. Innerhalb der Verwaltungsschale wird die Schnittstelle im Komponentenmanager abstrahiert und standardisiert. Dies hat den Vorteil, dass nicht ausschließlich REST Implementierungen, sondern beliebige weitere nachträglich entwickelt und für die jeweilige Komponente verwendet werden können.

Besitzt eine zu verwaltende Industrie 4.0-Komponente keine REST-Schnittstelle, so wird auf Serverseite die Software *NodeRed* verwendet, die zwischen einer Vielzahl von Protokollen übersetzt und so Komponenten ohne REST-Interface an das System anbindet [5].

3.2 Auswahl von Werten

Ein allgemeingültiges Konzept aufzustellen, dass ein Auswählen der zu speichernden Werte ermöglicht, wird als zu komplex für den Nutzen angesehen. Allerdings sollte auch beachtet werden, dass keine Prozessdaten gespeichert werden, eine Konzentration auf wichtige Daten stattfindet und diese nach Möglichkeit aggregiert werden. Eine genaue Analyse der Daten, sowie die späteren Verwendungsmöglichkeiten und Einsatzszenarien sollten vor dem Erstellen einer Verwaltungsschale geprüft werden.

3.3 Komponentenmanager

Der Komponentenmanager stellt für die Verwaltungsschale sowohl die Submodelle, als auch das Interface der Verwaltungsschale nach außen zur Verfügung. Aufgrund der unterschiedlichen Attribute im Manifest, als auch der unterschiedlichen Submodelle muss der Komponentenmanager die Schnittstelle stets an die Gegebenheiten der Verwaltungsschale anpassen. Hierfür gibt es ein generisches REST-Interface, welches die Kommunikation für alle Parameter und Komponenten erfüllt, die jede Verwaltungsschale besitzt. Der

Komponentenmanager kann dieses Interface während der Initialisierungsphase der Verwaltungsschale an die zusätzlichen Attribute und Submodelle anpassen und erweitern.

3.4 Werte im Manifest des Headers

Welche Werte im Manifest des Headers der Verwaltungsschale und welche in ein Manifest eines Submodells gespeichert werden sollen, muss von Fall zu Fall entschieden werden. Allerdings können gewisse Parameter, wie URL, ID, CAD-Zeichnungen, Typenbezeichnungen und Datenblätter stets dem Manifest der Verwaltungsschale zugeordnet werden. Ein guter Ansatz ist hier beispielsweise alle Parameter, die für eine Vielzahl von Verwaltungsschalen vorhanden sind, in das Manifest einzutragen und bei der weiteren Analyse identifizierte Ausnahmen anschließend in ein Submodell zu verschieben.

3.5 Discovery Service

Bei der Umsetzung des Discovery Services am Fraunhofer ESK wurde kein eigens entwickeltes proprietäres System verwendet, sondern ein standardisiertes, bewährtes und flexibles System zum Auffinden von Verwaltungsschalen verwendet. Hierbei handelt es sich um *JNDI (Java Naming and Directory Interface)*. Dies bietet von sich aus die Möglichkeit beliebige Verzeichnisdienste zu realisieren oder bereits bestehende zu verwenden, wie beispielsweise DNS oder LDAP.

3.6 Kommunikation zwischen Verwaltungsschalen

Für die Kommunikation zwischen Verwaltungsschalen wurde die gleiche Umsetzung wie für die Kommunikation der Verwaltungsschale mit ihrer Komponente gewählt. Das REST-Interface bietet hier ebenfalls standardisierte und einfach änderbare Schnittstellen (siehe Kapitel 3.1 und 3.3). Zudem können aufgrund der Abstraktion Änderungen der Protokolle jederzeit umgesetzt und bei der Initialisierung beliebig verwendet werden.

3.7 (Semi-)Automatisierte Erstellung von Verwaltungsschalen

Um Verwaltungsschalen effizient und mit geringem Aufwand erstellen zu können, verfolgt das Fraunhofer ESK den Ansatz einer Modellierung. Hierbei werden einzelne Sensoren, Aktoren oder die gesamte Maschine modelliert, verknüpft und mit den entsprechenden eCI@ss Klassifikationen hinterlegt. Anschließend ist es möglich, einen Großteil der Verwaltungsschaleninstanz generieren zu lassen.

Wird eine Generierung über Modelle verwendet, so ist es insbesondere möglich, einzelne Teile der Modellierung für andere Komponenten zu übernehmen. Durch diesen Ansatz kann

eine Erstellung aller benötigten Verwaltungsschalen einer Maschine oder einer Fertigungsstraße effizient und ohne spezifische Programmierkenntnisse erfolgen.

4. Beispiel ProBeam

Für die Firma ProBeam sollen die bestehenden Konzepte der Verwaltungsschale angewendet und so eine erste Schale für eine im Haus entwickelte Elektronenstrahlmaschine entworfen und umgesetzt werden. Als Ergebnis erwartet ProBeam eine bessere Verwaltung von gesamten Maschinen sowie eine Standardisierung der Überwachung und des Speicherns anfallender Daten.

4.1 Kommunikation mit Komponente

Aufgrund der Verwendung von Profibus zur Kommunikation zwischen den Komponenten innerhalb der Anlage ist ein direktes Abfragen der Live-Daten an den Komponenten für die Verwaltungsschale nicht ohne weiteres möglich. Um die benötigten Daten abzugreifen, besitzt die verwendete Maschinensteuerung eine spezielle Schnittstelle, über die sie zuvor konfigurierte Live-Daten periodisch zur Verfügung stellt. Diese Schnittstelle verwendet in diesem Fall allerdings kein Netzwerkprotokoll, wie beispielsweise TCP, über das die Daten über das Netzwerk abgerufen werden können, sondern bietet ausschließlich eine C++ und .NET Programmierschnittstelle an.

Für eigene Logging-Zwecke programmierte die Firma ProBeam bereits eine Software zur Verwendung der C++ Schnittstelle, welche die Daten über das Netzwerk an einen separaten PC zum Sammeln und Speichern aller anfallender Daten einsetzt.

Da die hier übertragenen Daten nicht über REST übertragen werden, wird zwischen der Verwaltungsschale und der Datenschnittstelle von ProBeam das Programm NodeRed eingesetzt, welches zwischen den verschiedenen Protokollen übersetzt.

4.2 Auswahl von Werten

Die richtige Auswahl der Daten ist bei jeder Anlage aufgrund ihrer Fülle eine Herausforderung und schwierige Entscheidung. Die verwendete Elektronenstrahlmaschine stellt hierbei ca. 10.000 verschiedene Daten zur Verfügung, welche teilweise im 2 ms Takt aktualisiert werden. Diese Fülle an Daten ist für die Speicherung in einer Verwaltungsschale nicht praktikabel bzw. vorgesehen. Daher wurde der Ansatz verfolgt, die wichtigsten 30 – 40 Daten der Maschine auszuwählen und diese nach Möglichkeit zu aggregieren und anschließend in der Verwaltungsschale zu speichern.

5. Beispiel Pumpenhersteller

Der zweite Anwendungsfall wird mit einem Pumpenhersteller für Vakuumpumpen für Reinräume, in denen Halbleiter gefertigt werden, umgesetzt. Für eine solche Produktionsstätte kommen bis zu 1000+ Pumpen zum Einsatz. Aktuell werden Informationen zur Verwaltung der Geräte entweder auf den Pumpen selbst oder über eine proprietäre Serverlösung gehandhabt. Mit dem Einsatz des Konzepts der Verwaltungsschale erhofft sich der Hersteller einen Vorstoß in Richtung predictive maintenance, aber auch in der flexibleren, dezentralen und herstellerübergreifende Verwaltung der Pumpen.

5.1 Kommunikation mit Komponente

Jede Pumpe besitzt eine Ethernet-Schnittstelle, über die verschiedene Protokolle übertragen werden können. Für die prototypische Anbindung der Pumpe an die Verwaltungsschale wurde das Modbus TCP Protokoll bzw. das in der Halbleiterproduktion übliche E54 Protokoll gewählt. Um die von Fraunhofer ESK entwickelte Laufzeitumgebung für die Verwaltungsschale einsetzen zu können, wird auch hier über NodeRed eine Kopplung von Modbus TCP zu REST umgesetzt.

5.2 Auswahl von Werten

Bzgl. der Auswahl der statischen und dynamischen Informationen für die Verwaltungsschale wurden einzelne Werte nach übergeordneten Bereichen gruppiert, um mögliche Submodelle definieren zu können. So wurde ein eigenes Submodell für das Alarmmanagement sowie ein weiteres für die Konfiguration der Pumpe vorgesehen. Für das Alarmmanagement wurden darüber hinaus Verhaltensweisen definiert wie z.B. das Verhalten bei Verletzung eines Wertebereichs für einen bestimmten Parameter.

5.3 Prototyp

Für die prototypische Evaluierung wurden 3 Controller der Pumpen zusammen mit einer Simulation des Verhaltens der realen Pumpe aufgebaut. Diese werden über NodeRed mit der von Fraunhofer ESK entwickelten Laufzeitumgebung der Verwaltungsschale verbunden. Da sich der Prototyp aktuell im Aufbau befindet und die Evaluierungsphase noch aussteht sind erste Ergebnisse Ende Juni zu erwarten.

6. Fazit

Die Umsetzung von Verwaltungsschalen innerhalb zwei unterschiedlicher Unternehmen hat ergeben, dass die reine Standardisierung von Verwaltungsschalen nach RAMI 4.0 mit

Industrie 4.0-Komponenten nicht ausreicht, um eine allgemeingültige Anleitung für die Erstellung von Verwaltungsschalen in unterschiedlichen Unternehmen zu besitzen. Zu den in den Standards festgelegten Punkten müssen weitere, wesentliche Punkte betrachtet und umgesetzt werden. Eine Standardisierung des gesamten Prozesses ist aufgrund der unterschiedlichen Protokolle innerhalb eines Unternehmens, sowie unterschiedlicher Komponenten, Protokolle und Parametern innerhalb einer Maschine nicht so einfach möglich.

Dennoch konnten durch die erworbenen Kenntnisse spezifische Vorgehensweisen entwickelt werden. Hierdurch konnten wiederum Teile, wie das Erstellen von Verwaltungsschalen durch Modelle und eCI@ss Klassifikationen oder die dynamische Schnittstelle mit erweiterbaren Protokollen, vereinfacht und teilweise automatisiert werden.

Literaturverzeichnis

- [1] Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0); ZVEI; <https://www.zvei.org/themen/industrie-40/das-referenzarchitekturmodell-rami-40-und-die-industrie-40-komponente/>
- [2] Die Industrie 4.0-Komponente; ZVEI; <https://www.zvei.org/presse-medien/publikationen/die-industrie-40-komponente/>
- [3] eCI@ss Klassifikationsstandard; <https://www.eclass.eu/>
- [4] Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0); DIN SPEC 91345:2016-04; April 2016; Seite 31
- [5] <https://nodered.org/>