Die approbierte Originalversion dieser Diplom-/Masterarbeit ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.
http://www.ub.tuwien.ac.at

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology.
http://www.ub.tuwien.ac.at/eng

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Institut für
Fertigungstechnik und
Hochleistungslasertechnik

Diplomarbeit

Konzeption und Evaluierung der Vernetzung von industriellen Anlagen im IIoT

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom Ingenieurs unter der Leitung von

a.o.Univ.-Prof. Dr. Burkhard Kittl

(Institut für Fertigungstechnik und Hochleistungslasertechnik)

DI Florian Pauker

(Austrian Center for Digital Production)

eingereicht an der Technischen Universität Wien

Fakultät für Elektrotechnik und Informationstechnik

von

Thomas Siedler, BSc

0828939 (066 506)

Vivenotstraße 44 / 4

3180 Lilienfeld

Lilienfeld, im September 2018

_____

Thomas, Siedler

Danksagung

An dieser Stelle möchte ich all jenen danken, die durch ihre fachliche und persönliche Unterstützung zum Gelingen dieser Diplomarbeit beigetragen haben.

Zuerst gebührt mein Dank den Herren a.o.Univ.-Prof. Dr. Burkhard Kittl und DI Florian Pauker, für die Vergabe und Betreuung der Diplomarbeit. Ihre fachliche und organisatorische Unterstützung, sowie ihre Anregungen waren eine große Hilfe bei der Erstellung dieser Arbeit.

Daneben gilt mein Dank der Firma Neuman Aluminium für die Möglichkeit, eine aktuelle und industrienahe Thematik im Rahmen meiner Diplomarbeit behandeln zu dürfen. Ein großes Dankeschön gilt Herrn Mag. DI Johannes Matiasch für seine Betreuung, Unterstützung und motivierende Rolle in schwierigen Phasen sowie für die zahlreichen Stunden, die er für Korrektur und Hinweise auf inhaltliche Verbesserungen investiert hat.

Mein besonderer Dank gilt meiner Familie, allen voran meinen Eltern, die mir mein Studium ermöglicht und mich stets unterstützt haben.

Meinen Freunden Stefan, Katharina und Lukas danke ich besonders für den starken emotionalen Rückhalt über die Dauer meines gesamten Studiums.

Kurzfassung

Die Vision der Smart Factory, in der Fertigungsanlagen und Logistiksysteme im Sinne des Internets der Dinge (IoT) miteinander kommunizieren und sich weitgehend selbst organisieren, erfordert den Einsatz fortschrittlicher Kommunikationstechniken in einer sicheren Umgebung. Für die horizontale und vertikale Kommunikation in Produktionsumgebungen setzt sich OPC UA zunehmend als Standard durch. Für die M2M-Kommunikation und zur Interaktion mit Cloud-Systemen hat sich zudem MQTT als eines der Standard-Protokolle des Internet der Dinge etabliert.

Die vorliegende Arbeit bietet einen detaillierten Einblick, welche Möglichkeiten, Eigenschaften und Stärken OPC UA und MQTT mit sich bringen. Dadurch wird verdeutlicht, weshalb diese Kommunikationstechniken fixer Bestandteil von intelligenten Produktionsumgebungen sind. Der ansteigende Vernetzungsgrad mit der einhergehenden Kommunikation innerhalb und über die Unternehmensgrenzen hinweg erfordert die Einführung eines Sicherheitskonzeptes in industriellen Automatisierungs- und Kontrollsystemen (IACS). Die Norm IEC 62443 liefert dafür empfohlene Vorgehensweisen. Gemäß IEC 62443 ist im Rahmen dieser Arbeit ein durchgängiger und strukturierter Leitfaden zur Umsetzung eines effizienten Security-Konzeptes angefertigt worden.

Maschinen mit festverdrahteter Steuerung, die über keine speicherprogrammierbare Steuerung (SPS) verfügen, werden im Rahmen dieser Arbeit als Legacy Systems bezeichnet. Sie sind auch heutzutage noch häufig in produzierenden Unternehmen anzutreffen. Für eine digitale Produktion müssen bewährte Legacy Systems genauso in den Digitalisierungsprozess eingebunden werden. Es werden zwei Möglichkeiten gezeigt, wie Legacy Systems in Umgebungen des Industrial Internet of Things (IIoT) integriert werden können. Der Fokus liegt dabei auf der Erfassung der Signale zur Beschreibung der Basiszustände einer Maschine. Damit lässt sich eine automatisierte und lückenlose Betriebs- bzw. Maschinendatenerfassung speziell für Legacy Systems realisieren.

Die Testumgebung des praktischen Teils dieser Arbeit dient zur Durchführung einer praxisorientierten Evaluierung unter Berücksichtigung einer ganzheitlichen Betrachtung der genannten Themen. Die Evaluierung reicht von der Anbindung von Legacy Systems unter Verwendung der Datenübertragung mit OPC UA und MQTT innerhalb eines segmentierten Netzwerks bis hin zur Visualisierung der erfassten Maschinendaten auf einem übergeordneten System.

Abstract

The vision of the Smart Factory, in which production facilities and logistics systems communicate with each other within the Internet of Things (I0T) and largely organize themselves, requires the use of advanced communication technologies within a secure environment. OPC UA is increasingly becoming the standard for horizontal and vertical communication in production environments. MQTT has also established itself as one of the standard protocols of the Internet of Things for M2M communication and interaction with cloud systems.

The present work offers a detailed insight into the possibilities, features and strengths of OPC UA and MQTT. It illustrates why these communication technologies are an essential part of intelligent production environments. The increasing degree of networking accompanied by communication within and beyond company boundaries requires the introduction of a safety concept in industrial automation and control systems (IACS). The IEC 62443 standard provides best practices for this. In this work, a consistent and structured guideline for the implementation of an efficient security concept according to IEC 62443 has been prepared.

Machines with hardwired controls that do not have a programmable logic controller (PLC) are referred to as legacy systems. They are still frequently found in manufacturing companies these days. For digital production, proven legacy systems must be integrated in the digitization process. Two ways of integrating legacy systems in the environment of the Industrial Internet of Things (IIoT) are demonstrated. The focus is on the acquisition of signals to describe the basic states of a machine. This enables the implementation of automated and complete production and machine data acquisition especially for legacy systems.

The test environment in the practical part of this work serves for carrying out a practice-oriented evaluation considering the topics mentioned. The evaluation ranges from the connection of legacy systems using data transmission with OPC UA and MQTT within a segmented network to the visualization of the collected machine data on a higher-level system.

Inhaltsverzeichnis

1 Einleitung 8
2 Datenübertragung mittels OPC UA und MQTT 9
2.1 OPC UA (OPC Unified Architecture) 9
2.1.1 Einleitung und Überblick 9
2.1.2 Kommunikationsprinzip 12
2.1.3 Entwurfsmöglichkeiten der Kommunikationsarchitektur 17
2.1.4 OPC UA AddressSpace-Modell und Informationsmodelle 19
2.1.5 Der OPC UA Server 23
2.1.6 Der OPC UA Client 28
2.1.7 Security-Konzepte in OPC UA 29
2.1.8 Redundanzkonzepte 34
2.1.9 Die Applikations-Architektur 38
2.2 Kommunikationsprotokoll MQTT 41
2.2.1 Struktur der MQTT Datenpakete 42
2.2.2 Semantik für Topics 43
2.2.3 Kommunikationsablauf 44
2.2.4 Security Mechanismen 47
2.3 Gegenüberstellung 49
2.4 Weitere Kommunikationsprotokolle im IIoT 49
3 Industrial Security 50
3.1 Einführung 50
3.2 Konzepte der IEC 62443 51
3.2.1 Defense in Depth 52
3.2.2 Netzsegmentierung – Zonen und Conduits 52
3.2.3 Risikobewertung nach VDI/VDE 2182 53
3.2.4 Anlagenlebenszyklus 54
3.2.5 PDCA-Zyklen im Anlagenlebenszyklus 54
3.2.6 Security Levels bis hin zu einem ganzheitlichen Ansatz 54
3.3 Einführung eines Cyber Security Management Systems 55
3.4 Leitfaden zur Umsetzung der IEC 62443 im Unternehmen 56

3.4.1 Ausgangspunkt und Zielsetzung 56
3.4.2 Voraussetzungen 58
3.4.3 Schritte zur Umsetzung der IEC 62443 59
3.5 Design eines segmentierten Netzwerks 73
3.5.1 Szenario 73
3.5.2 Lösungsvorschlag 74
4 IIoT – Integration von Legacy Systems 75
4.1 Einleitung 75
4.2 Raspberry Pi 3 mit UniPi v1.1 Erweiterungsboard 76
4.2.1 Raspberry PI 3 Modell B 77
4.2.2 Erweiterungsboard UniPi v1.1 77
4.2.3 Programmierung 79
4.2.4 Übertragung der erfassten Maschinendaten 79
4.3 Siemens S7-1200 79
4.3.1 Programmierung 80
4.3.2 Übertragung der Maschinendaten 80
4.4 Gegenüberstellung 80
5 Testumgebung und Evaluierung 81
5.1 Einleitung und Überblick 81
5.2 Aufbau und Setup der Komponenten 83
5.2.1 Steuerboard zur Simulation von Maschinenzuständen 83
5.2.2 Raspberry Pi 3 mit UniPi v1.1 Erweiterungsboard 84
5.2.3 Siemens S7-1200 86
5.2.4 Industrie Firewall – FL mGuard RS4004 TX/DTX VPN 87
5.2.5 Laptop – MQTT-Broker, OPC UA Server, Node-RED 87
5.3 Netzwerkumgebung 89
5.4 Programmierung für die Datenerfassung 92
5.4.1 Programmierung des Raspberry Pi mit Node-RED 92
5.4.2 SPS Programmierung im TIA Portal 93
5.5 Kommunikationsumgebung – MQTT und OPC UA 95
5.5.1 Kommunikationsumgebung für MQTT 95
5.5.2 . Kommunikationsumgebung für OPC UA 99

5.6 Visualisierung der Maschinendaten 103
5.6.1 Struktur des Dashboards in Node-RED 103
5.6.2 Dashboard mit visualisierten Maschinendaten 103
6 Conclusio 106
7 Schlussfolgerungen / Resümee / Ausblick 107
8 Anhang 108
8.1 Anhang Kapitel 1 108
8.2 Anhang Kapitel 2 108
8.2.1 Abschnitt 2.1.1 – Einleitung und Überblick 108
8.2.2 Abschnitt 2.1.4 – OPC UA Informationsmodelle 109
8.2.3 Abschnitt 2.1.5 – Der OPC UA Server 111
8.2.4 Abschnitt 2.1.7 - Security-Konzepte in OPC UA 112
8.3 Anhang Kapitel 3 113
8.3.1 Abschnitt 3.2 – Normenreihe IEC 62443 113
8.3.2 Abschnitt 3.4.3 – Schritte zur Umsetzung der IEC 62443 115
8.4 Anhang Kapitel 4 131
8.5 Anhang Kapitel 5 131
8.5.1 Abschnitt 5.2.1 – Steuerboard zur Simulation 131
8.5.2 Abschnitt 5.2.2 – Raspberry Pi 3 mit UniPi v1.1 Erweiterungsboard 131
8.5.3 Abschnitt 5.4.2 – SPS Programmierung im TIA Portal 133
8.5.4 Abschnitt 5.5.1 – Kommunikationsumgebung für MQTT 137
8.5.5 Abschnitt 5.5.2 – Kommunikationsumgebung für OPC UA 139
9 Literaturverzeichnis 140
10 Abbildungsverzeichnis 142
11 Tabellenverzeichnis 145
12 Abkürzungsverzeichnis 146

1 Einleitung

Die fortschreitende Digitalisierung in der produzierenden Industrie bringt eine Reihe neuer Herausforderungen mit sich. Aufgrund der Kommunikation innerhalb und über die Unternehmensgrenzen hinweg sowie der horizontalen und vertikalen Interaktion unterschiedlicher Systeme innerhalb eines Unternehmens ist ein ganzheitliches Sicherheitskonzept unerlässlich. Dazu zählt eine sichere und zuverlässige Kommunikation. Des Weiteren zeigt der Trend, dass neben den Sicherheitsaspekten komplexe Datenstrukturen im industriellen Umfeld rapide an Bedeutung gewinnen.

Im ersten Teil der Arbeit werden deshalb die Möglichkeiten und Stärken einer Kommunikationsumgebung mit OPC UA aufgezeigt. Neben der OPC UA Spezifikation zeigen die Werke von Rinaldi [1] und Damm [2], dass OPC UA die Modellierung komplexer Datenstrukturen ermöglicht und eine Datenübertragung unter Einhaltung der Sicherheitsziele wie Authentifizierung, Autorisierung, Vertraulichkeit, Integrität, Verfügbarkeit und Nachvollziehbarkeit auf verschiedenen Ebenen gewährleistet. Des Weiteren wird MQTT, welches als eines der Standard-Protokolle des Internet der Dinge gilt, detailliert betrachtet. MQTT eignet sich hervorragend zur M2M-Kommunikation und speziell für asynchrone one-to-many, many-to-one und many-to-many Kommunikationsstrukturen.

Der nächste Abschnitt der Arbeit behandelt die Norm IEC 62243 für die Umsetzung eines effizienten Security-Konzeptes in Automatisierungssystemen aus Sicht des Anlagenbetreibers und -integrators. Mit den Konzepten der IEC 62443 wird ein strukturierter Leitfaden zur Umsetzung der IEC 62443 im Unternehmen erarbeitet.

In weiterer Folge wird auf die Anbindung und Erfassung von Maschinendaten bei Legacy Systems, welche heutzutage noch immer häufig in produzierenden Unternehmen anzutreffen sind, eingegangen. Es werden zwei Möglichkeiten gezeigt, wie Legacy Systems in Umgebungen des Industrial Internet of Things (IIoT) integriert werden können und wodurch sich diese unterscheiden. Der Fokus liegt dabei auf der Erfassung der Signale zur Beschreibung der Basiszustände einer Maschine. Damit lässt sich eine automatisierte und lückenlose Betriebs- bzw. Maschinendatenerfassung speziell für Legacy Systems realisieren.

Der letzte Abschnitt zeigt den Aufbau der Testumgebung zur Durchführung einer praxisorientierten Evaluierung. Dabei wurden alle in dieser Arbeit behandelten Themen berücksichtigt, integriert und evaluiert. Der Abschnitt gliedert sich hierfür in: Aufbau und Setup der Komponenten, Netzwerkumgebung, Programmierung für die Datenerfassung, Kommunikationsumgebungen für MQTT und OPC UA sowie die Visualisierung der Maschinendaten.

2 Datenübertragung mittels OPC UA und MQTT

2.1 OPC UA (OPC Unified Architecture)

2.1.1 Einleitung und Überblick

OPC UA ist die nächste Generation der OPC Technologie und stellt eine Erweiterung des 1996 entwickelten und in der automatisierten Industrie weit verbreiteten OPC Classic Standards dar. Unter OPC Classic werden die vier Standards OPC DA (Data Access) für den Zugriff auf aktuelle Prozessdaten, OPC A&E (Alarm und Events) für die Zustellung eventbasierter Daten, OPC HDA (Historical Data Access) für den Zugriff auf historische, sprich archivierte Daten und OPC XML-DA, das sich jedoch nicht durchgesetzt hat, verstanden. OPC Classic wurde als Schnittstelle für Gerätetreiber entwickelt und basiert auf der Microsoft Windows COM/DCOM-Technologie (Component Object Model/Distributed Component Object Model-Technologie), die den damaligen Standard für die Kommunikation von Softwareapplikationen bei Microsoft darstellte. Aufgrund der DCOM-Technologie konnte und kann OPC Classic ausschließlich in Verbindung mit Microsoft Systemen verwendet werden. Zusätzlich zeigt OPC Classic eine starke Limitierung bezüglich der Modellierungsmöglichkeiten von Daten [1]. Der Trend zeigt jedoch, dass komplexe Datenstrukturen im industriellen Umfeld rapide an Bedeutung gewinnen. Des Weiteren fordert der steigende Bedarf an Sicherheit in Industrienetzwerken aufgrund des wachsenden Vernetzungsgrades eine Kommunikation über eine oder mehrere Firewalls, welche sich mit der COM/DCOM Technologie kaum in einem vertretbaren Rahmen umsetzen lässt. Die Anforderungen und die Struktur zur Absicherung eines in der Industrie einsetzbaren Netzwerkes werden in Kapitel 3 diskutiert.

Um die gestiegenen Anforderungen befriedigen zu können, wurde OPC UA von der OPC Foundation ins Leben gerufen. 2008 wurde die erste Spezifikationen veröffentlicht, in welcher alle OPC Classic Funktionalitäten (DA, A&E und HDA) in einer einzigen Service-orientierten Architektur (SOA) vereint wurden. Zusätzlich wurden Erweiterungen für eine sichere, zuverlässige und plattformunabhängige Interoperabilität und die Möglichkeit, komplexe Informationsmodelle nutzen zu können, umgesetzt [3]. Daten können dadurch in OPC UA inklusive zugehöriger Metadaten und Semantik abgebildet werden. Mit OPC UA besteht also die Möglichkeit, sowohl horizontalen als auch vertikalen Daten- und Informationsaustausch im Unternehmen zu realisieren [2]. Wie in ***Abbildung 1*** ersichtlich, kann der Datenverkehr innerhalb der Feldebene, aber auch von der Feldebene zu übergeordneten Systemen – wie beispielsweise Supervisory Control and Data Acquisition (SCADA), Manufacturing Execution System (MES) oder Enterprise Resource Planning (ERP) mittels OPC UA erfolgen. OPC UA nutzt zurzeit eine klassische Client/Server-Architektur und als

Transportprotokoll entweder HTTP(S), UA TCP oder WebSocket. UA TCP ist ein von der OPC Foundation definiertes und auf TCP aufbauendes Protokoll [1]. Für die zurzeit stattfindende Integration des Publish/Subscribe-Konzeptes (PubSub) sind als Transportprotokoll beispielsweise UDP, AMQP oder MQTT vorgesehen. Vor allem im Zusammenhang mit dem PubSub-Konzept muss die voranschreitende Weiterentwicklung der TSN-Technologie (Time Sensitive Network) genannt werden. TSN sind mehrere Substandards des Ethernet-Standards IEEE 802.1, welche es ermöglichen eine echtzeitfähige Datenübertragung mit konventionellem Ethernet zu realisieren. Das PubSub-Konzept in Kombination mit TSN soll damit die Erfüllung der Echtzeitanforderungen in der Industrie ermöglichen und ist somit ein wichtiger Bestandteil für die Integrationsfähigkeit von OPC UA. Für die Nachrichtencodierung in OPC UA gibt es nach [1] und [4] folgende Möglichkeiten:

- UA Binary, das sich aufgrund des geringen Overheads für Einsatzbereiche mit hoher Performanz eignet.
- XML, welches von einer Vielzahl an Applikationen unterstützt wird.
- JSON, das zur Kommunikation über Publish/Subscribe-Nachrichten mit Cloud-Applikationen oder JavaScript Clients eingesetzt werden kann.

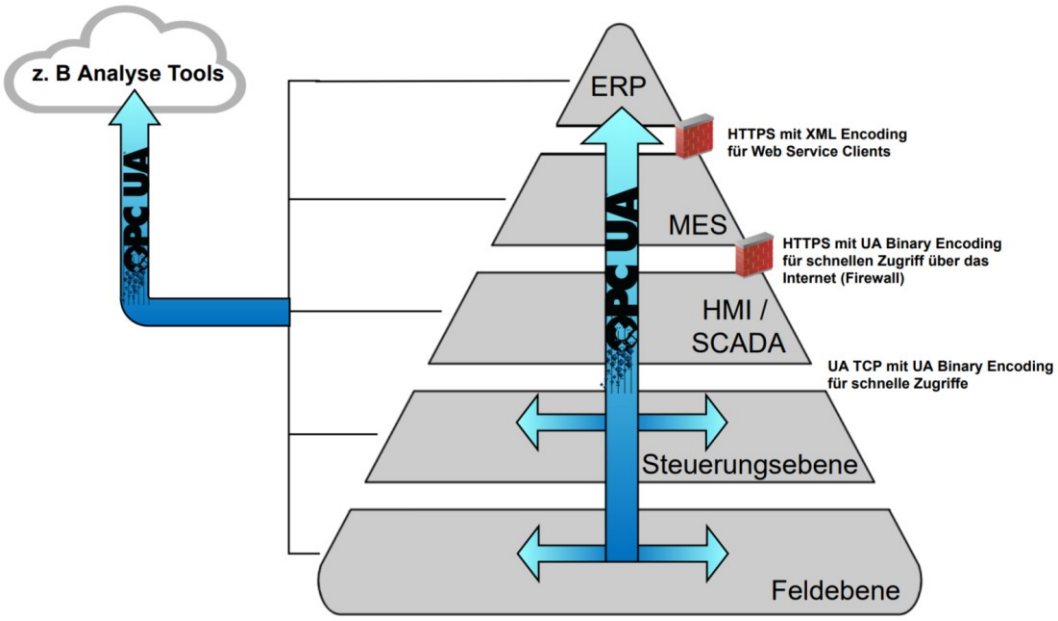


Abbildung 1: OPC UA auf allen Ebenen [2]

Die wichtigsten Charakteristika hinsichtlich Kommunikation und Datenmodellierung in OPC UA sind spaltenweise in ***Tabelle 1*** zusammengefasst.

Kommunikation	Datenmodellierung
Verlässlichkeit hinsichtlich Robustheit, Fehlertoleranz und Redundanz	Gemeinsame Modelle für alle OPC Daten
Plattformunabhängigkeit	Objektorientierung

Kommunikation	Datenmodellierung
Skalierbarkeit (Nutzung sowohl auf ressourcenbeschränkten als auch auf sehr leistungsfähigen Geräten)	Erweiterbares Typensystem
Hohe Performanz	Abbildung von Metadaten
Firewall und Internet	Komplexe Datenmodelle und Methoden
Sicherheit und Zugriffskontrolle	Skalierbarkeit von simplen bis hin zu komplexen Modellen
Interoperabilität	Abstraktes Basis-Modell, welches als Ausgangspunkt für komplexe Modelle herangezogen werden kann

Tabelle 1: Charakteristika von OPC UA [2]

Die OPC UA Spezifikation besteht zurzeit aus 14 Teilen, wobei Letzterer erst als Release Candidate im Februar 2017 erschienen ist und das PubSub-Kommunikationsmodell beschreibt. Alle Teile, ausgenommen Teil 12 und Teil 14, sind bereits im IEC Standard unter IEC 62541 zu finden. Die Übersicht der Spezifikationen ist in ***Abbildung 2*** dargestellt. Eine kurze Beschreibung der wesentlichen Inhalte der einzelnen Spezifikationen ist im Anhang 8.2.1 zu finden.

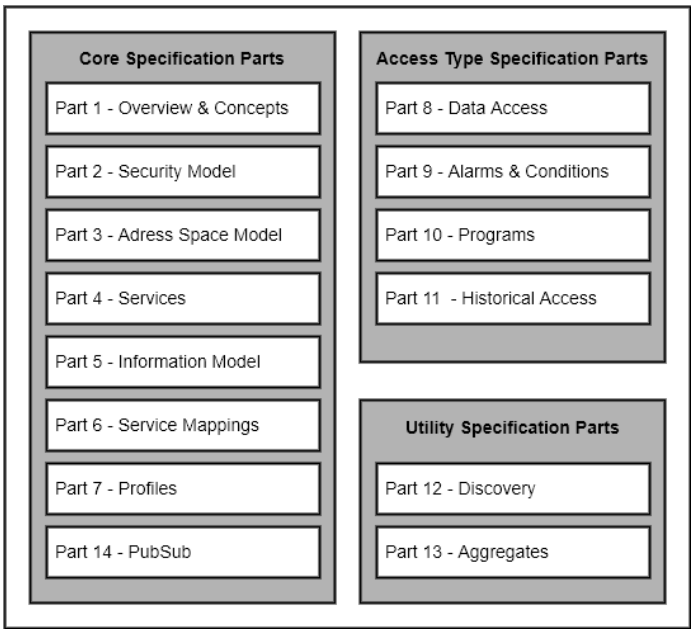


Abbildung 2: Übersicht OPC UA Spezifikationen [5]

Abbildung 3 gibt einen ersten Überblick über die hierarchischen Zusammenhänge der in der Spezifikation beschriebenen Funktionalitäten.

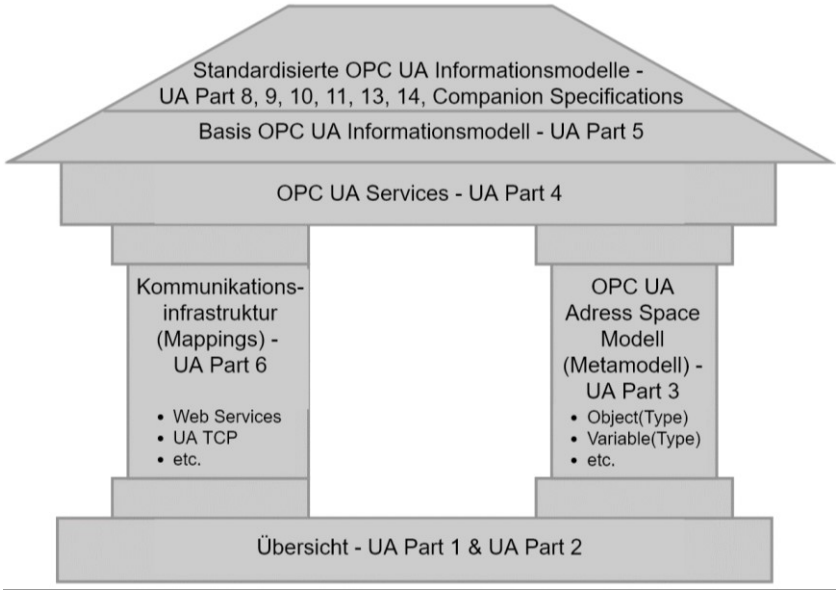


Abbildung 3: Hierarchie der Spezifikationen [6]

2.1.2 Kommunikationsprinzip

Dieser Abschnitt betrachtet das klassische Client/Server-Prinzip von OPC UA und liefert einen ersten Überblick über das nachträglich in die Spezifikation aufgenommene Publish/Subscribe-Prinzip. Letzteres kann beispielsweise zur Anbindung an Cloud-Applikationen für Themen wie Predictive Maintenance Analysen verwendet werden. Die Standardisierung in OPC UA erlaubt die Integration des Publish/Subscribe-Prinzips in klassische OPC UA Applikationen. Auf eine detaillierte Betrachtung von PubSub wird aber, aufgrund des noch fehlenden Release (Stand 27.01.2018) der OPC UA Spezifikation Part 14, in dieser Arbeit verzichtet. Deswegen soll hier darauf hingewiesen werden, dass sich alle restlichen Abschnitte dieses Kapitels auf die klassische Client/Server-Architektur beziehen. Die Fertigstellung der OPC UA Spezifikation Part 14 ist aber in naher Zukunft zu erwarten.

- Client/Server

Dieses Kommunikationsprinzip basiert auf einem klassischen Request/Response-Modell zwischen Client und Server – siehe ***Abbildung 4***. Der Informationsaustausch wird vom Client mittels Request an den Server initiiert.

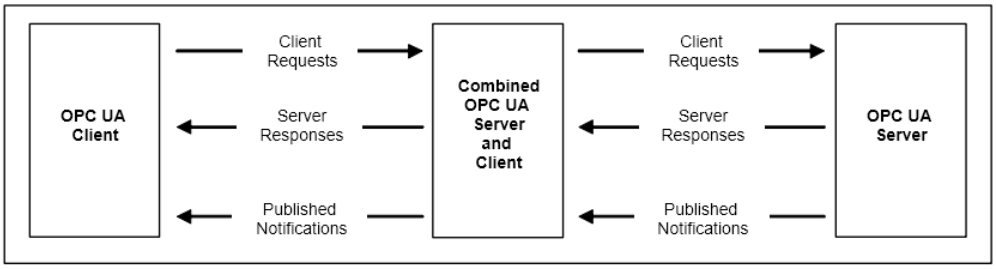


Abbildung 4: OPC UA Client/Server-Architektur [5]

Der Server stellt dem Client ein objektorientiertes und standardisiertes OPC UA Informationsmodell eines dahinterliegenden realen Prozesses – im sogenannten AddressSpace – zur Verfügung. Darüber hinaus stellt der Server den Clients bis zu 39 Services zur Verfügung, welche mittels Request vom Client genutzt werden können. Beispiele für Services sind:

- Services für den sicheren Verbindungsaufbau
- Möglichkeiten zum Browsen und zur Manipulation des AddressSpace im Server
- Management für Monitoring- und Benachrichtigungen
- Aufruf von Methoden
- Datenzugriff (Browse/Read/Write) von Echtzeit-, historischen, Alarm- und Event-Daten

Zur Kommunikation zwischen Server und Client können je nach Anwendung als Transportprotokoll UA TCP, HTTP(S), WebSocket und unterschiedliche Encodings, wie beispielsweise Binary, XML oder JSON zum Einsatz kommen. OPC UA unterstützt Datenpufferung, die Nutzung von Empfangsbestätigungen sowie eine erneute Datenübertragung für eine zuverlässige Datenübertragung, wodurch jedoch der Server für jeden verbundenen Client entsprechende Ressourcen benötigt [7].

Zu Beginn nutzt der Client eigens dafür vorgesehene Services (Discovery-Services), um den Server im Netzwerk zu finden und dessen Kommunikationscharakteristika (unterstützte Transportprotokolle, Security Policies etc.) zu identifizieren. Dieser sogenannte Discovery Prozess wird in Kapitel 2.1.5 detaillierter diskutiert. Mit diesen Informationen wird anschließend auf Transportebene ein sicherer Verbindungskanal – als Secure Channel bezeichnet – realisiert. Aufbauend auf den Secure Channel kann ein Client eine individuelle Session erzeugen, in der das Management der Kommunikationsbeziehung zwischen Client und Server erfolgt [7]. Der detaillierte Ablauf für den Aufbau einer sicheren Kommunikationsumgebung – Secure Channel und Session – wird in Kapitel 2.1.7 anhand der OPC UA TCP Secure Conversation mit Binary Encoding beschrieben. Innerhalb einer Session können mehrere Subscriptions, und in jeder Subscription wiederum mehrere MonitoredItems, erzeugt werden. Der beschriebene Aufbau der Kommunikationsumgebung für den Datenaustausch zwischen Server und Client ist in ***Abbildung 5*** dargestellt: Der Aufbau erfolgt in definierten Schritten von unten beginnend, wobei jeder einzelne dieser Schritte sequentiell – mit individuellen vom Server angebotenen Services – umgesetzt wird.

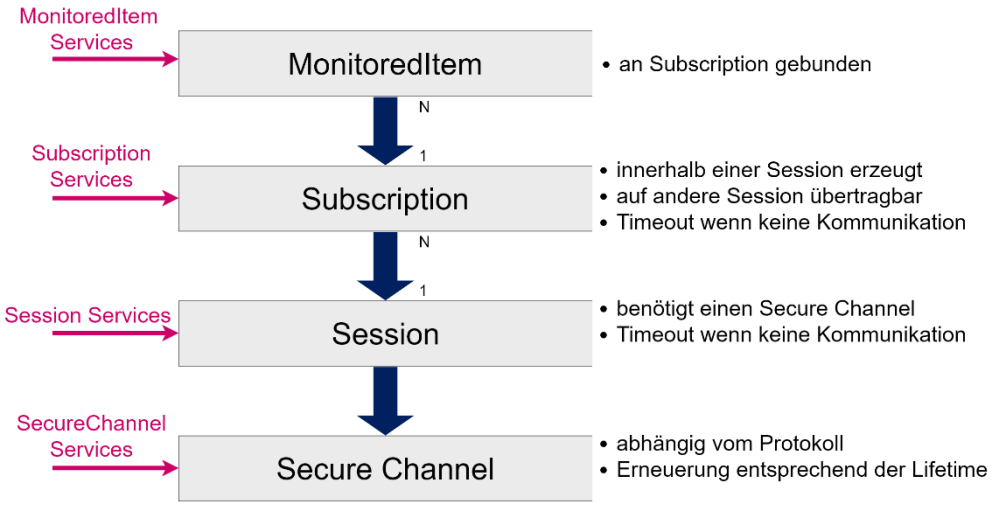


Abbildung 5: Ablauf des Aufbaus einer Kommunikationsumgebung

Unter dem Begriff MonitoredItem versteht man das Abonnieren von Events und Variablen- bzw. Statusänderungen von Objektattributen im AddressSpace. Für MonitoredItems sind vier wichtige Parameter definiert, mit denen das Abtastintervall, der Monitoring-Modus, die Filterkriterien und die Queueparameter vorgegeben werden können. Der Austausch dieser Variablenänderungs- und Eventbenachrichtigungen zwischen Server und Client findet im Rahmen einer Subscription statt [2]. Subscriptions weisen ein Publishingintervall auf, welches angibt, in welchen Abständen Benachrichtigungen an den Client gesendet werden. Die Parameter für MonitoredItems und Subscriptions sind in ***Abbildung 6*** dargestellt. Subscriptions können zur Laufzeit dynamisch angelegt und gelöscht oder auf eine andere Session übertragen werden [7]. Detaillierte Betrachtungen des MonitoredItem- und des Subscription-Modells sind in den Abschnitten 5.12.1 und 5.13.1 aus [8] zu finden.

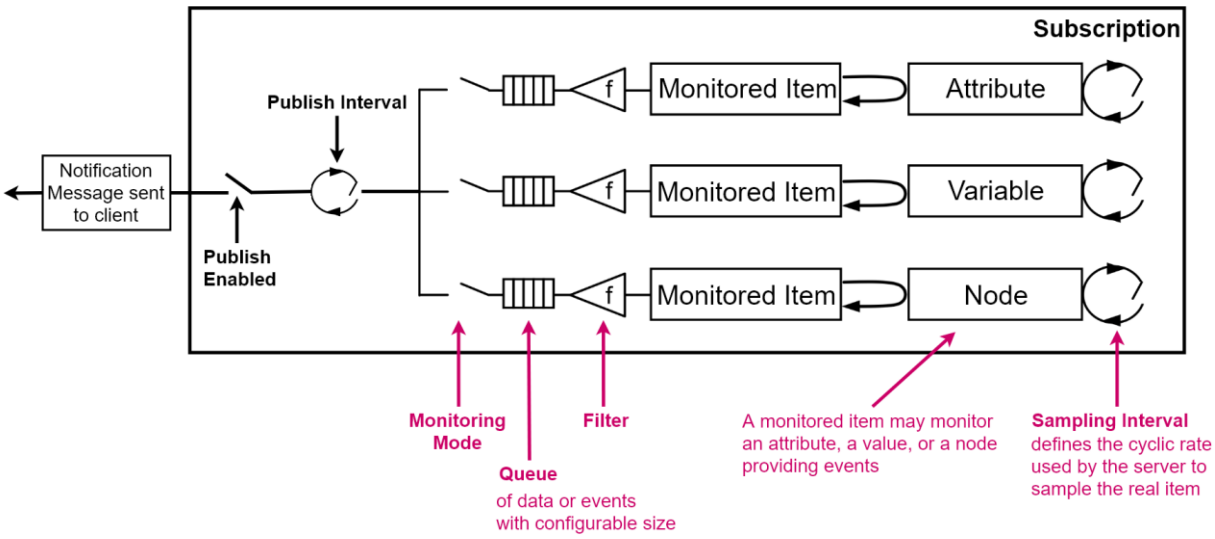


Abbildung 6: Parameter von MonitoredItems und Subscriptions [2], [8]

Abbildung 7 stellt die OPC UA Client/Server-Architektur detaillierter dar.

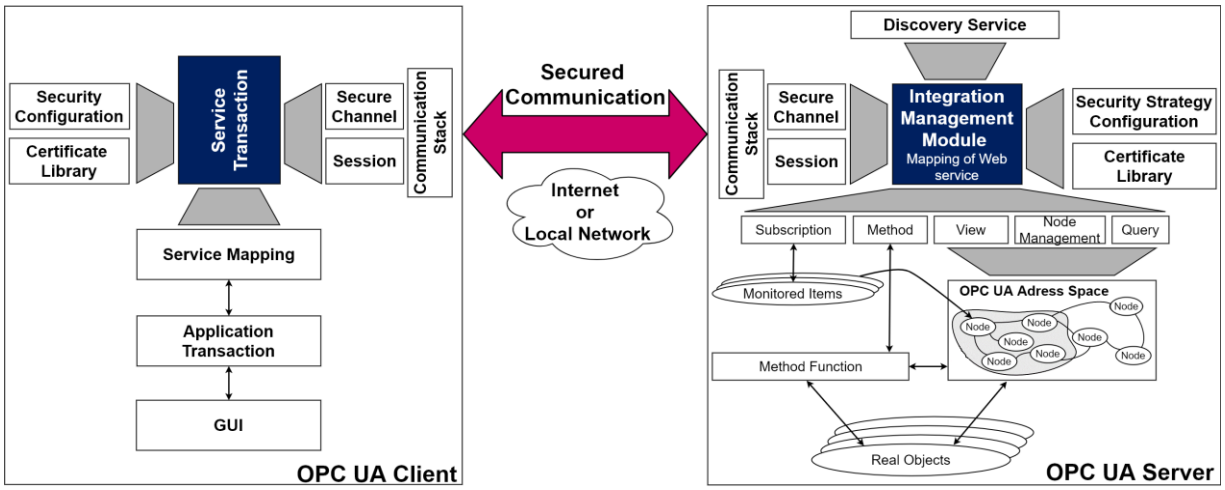


Abbildung 7: Detaillierte OPC UA Client/Server-Architektur [9]

Ein System kann mehrere Clients und Server, aber auch Applikationen, die sowohl Server- als auch Client-Komponenten kombinieren, beinhalten [5]. Das OPC UA Client/Server-Modell weist keine Einschränkungen dahingehend auf, wie viele Clients mit einem Server bzw. wie viele Server mit einem einzelnen Client in Verbindung stehen können oder wie zwei Server miteinander kommunizieren können [1].

- Publish/Subscribe

Dieses Kommunikationsprinzip basiert darauf, dass Applikationen entweder Daten an eine nachrichtenorientierte Middleware übergeben oder von dieser anfordern. Applikationen, die Informationen an die Middleware senden, werden als Publisher bezeichnet; Applikationen, die Informationen von der Middleware erhalten, als Subscriber. Applikationen sind nicht auf eine dieser Rollen eingeschränkt und können sowohl als Publisher als auch Subscriber fungieren. Das Spezielle an dem Publish/Subscribe-Modell ist, dass keine direkte Verbindung zwischen Publisher und Subscriber bestehen muss und diese nicht über die Identität des Anderen Bescheid wissen müssen. Der schematische Aufbau eines Publish/Subscribe-Modells ist in ***Abbildung 8*** dargestellt.

OPC UA PubSub unterstützt zwei unterschiedliche nachrichtenorientierte Middleware Varianten [5]. Auf der einen Seite kann eine Form ohne Broker eingesetzt werden, wobei hier die Netzwerkinfrastruktur in der Lage sein muss, das richtige Routing von UDP Nachrichten zu übernehmen [7]. Publisher und Subscriber nutzen Datagrammprotokolle wie beispielsweise UDP Mulitcast [7]. Auf der anderen Seite kann ein Broker als Middleware zum Einsatz kommen, welcher mittels AMQP oder MQTT mit den Publishern und Subscribern kommunizieren kann [7]. Bei der Broker-basierenden Form werden Nachrichten in sogenannten Topics veröffentlicht und an den Broker gesendet. Subscriber hingegen abonnieren diverse Topics und werden bei

auftretenden Änderungen vom Broker verständigt. Der Broker alleine ist für die Aufnahme und Verteilung der Nachrichten zuständig. Wenn nötig, hat dieser zusätzlich die Aufgabe, Nachrichtenprotokolle zwischen Publisher und Subscriber zu übersetzen [7].

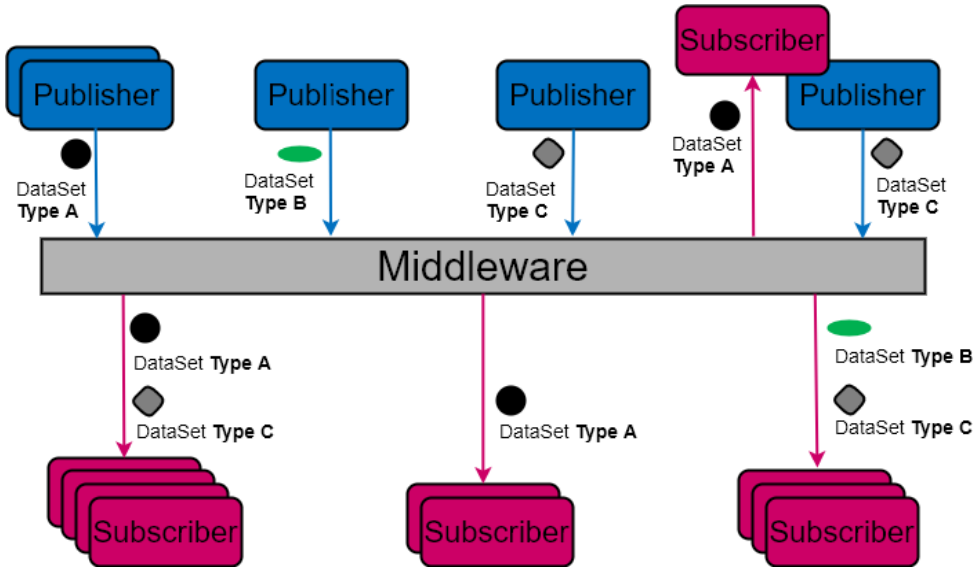


Abbildung 8: Publish/Subscribe-Modell [7]

- Synergie des Client/Server und des Publish/Subscribe-Prinzips

Da sowohl das Client/Server- als auch das Publish/Subscribe-Prinzip auf dem OPC UA Informationsmodell basieren und keine Rollenabhängigkeit besteht, kann das Publish/Subscribe-Prinzip in OPC UA Server bzw. OPC UA Clients integriert werden [5]. In der Regel wird aber ein Server als Publisher agieren, da hier die Informationen bereits vorliegen, und ein Client als Subscriber. ***Abbildung 9*** zeigt eine schematische Darstellung einer möglichen Kommunikationsumgebung. Diese Umgebung beinhaltet:

- einen klassischen OPC UA Client
- mehrere Subscriber
- einen OPC UA Server, welcher auch gleichzeitig als Publisher fungiert
- eine nachrichtenorientierte Middleware (z.B. Broker)

Eine derartige Umgebung würde somit gleichzeitig das herkömmliche OPC UA Client/Server-Modell und das OPC UA Publish/Subscribe-Modell verwenden.

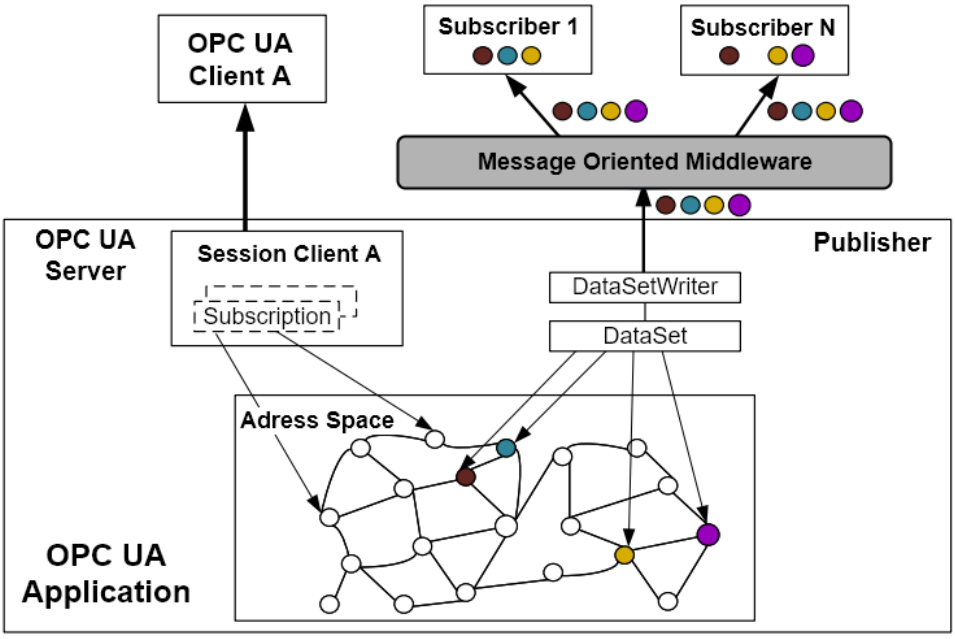


Abbildung 9: Mögliche schematische Kommunikationsstruktur mit Client/Server- und Publish/Subscribe-Modell [5]

2.1.3 Entwurfsmöglichkeiten der Kommunikationsarchitektur

- Client/Server

Die Client/Server-Architektur stellt den Basisaufbau der OPC UA Kommunikation dar. Der Server ist in vielen Fällen bereits Teil eines Gerätes z.B. einer SPS, der die Daten in einem Informationsmodell inklusive diverser Services einem Client zur Verfügung stellt. Der Client nutzt diese Services um auf die Daten zuzugreifen. Die Kommunikation, wie in ***Abbildung 10*** dargestellt, setzt auf den Austausch wohldefinierter Nachrichten mit einfachem Request/Respond-Mechanismus.

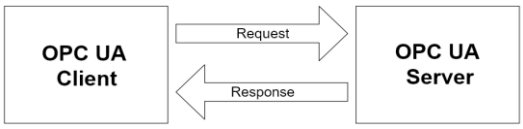


Abbildung 10: Client/Server-Architektur [2]

- Chained Server

Die Chained-Server-Architektur stellt eine Möglichkeit dar, OPC UA zur vertikalen Kommunikation innerhalb der Automatisierungspyramide des Unternehmens einzusetzen. Der Chained-Server-Ansatz kann sozusagen als Gateway angesehen werden. ***Abbildung 11*** zeigt den schematischen Aufbau und enthält neben klassischen Clients und Servern auch Komponenten, die als Client und Server fungieren können. Diese Komponenten sind notwendig um die verschiedenen Ebenen vertikal zu vernetzen und, als Folge dessen, unterschiedliche Protokolle an den verschiedenen Stellen entsprechend umzusetzen. Beispielsweise könnte mit einem im

Operation Network befindlichen Client ein Zugriff per HTTPS über eine Firewall auf einen darunterliegenden Server erfolgen. Dieser Server enthält zusätzlich einen internen Client, über den die Daten letztendlich wiederum von einem untergeordneten Server abgerufen werden. Letzterer könnte möglicherweise ein SPS-integrierter Server sein, der aufgrund eingeschränkter Hardware jedoch nur Binärprotokolle unterstützt. Der Server mit dem integrierten Client agiert also als Gateway zwischen zwei Servern und realisiert eine Umsetzung zwischen HTTPS und dem Binärprotokoll.

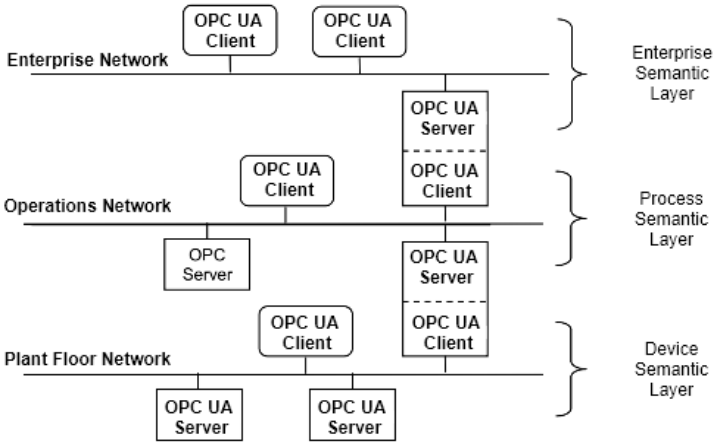


Abbildung 11: Chained-Server-Architektur für vertikale OPC UA Integration [5]

- Server/Server

Sollen Kommunikation und Datenaustausch zwischen zwei Servern in bidirektionaler Weise erfolgen, muss jeder der beiden Server – wie in ***Abbildung 12*** dargestellt – eine integrierte Client-Komponente aufweisen. Der integrierte Client ermöglicht dem Server eine Kommunikation mittels Request zu initiieren [2]. Diese Architektur spielt beispielsweise bei Redundanzkonzepten, die in Abschnitt 2.1.8 näher erläutert werden, eine entscheidende Rolle.

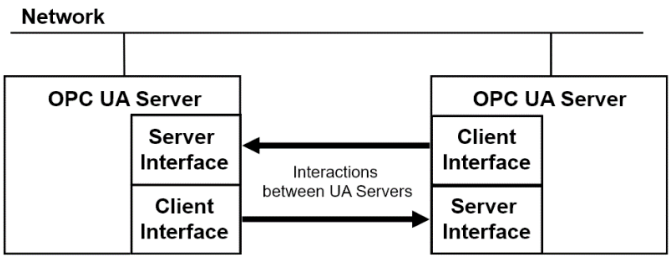


Abbildung 12: Server/Server-Architektur [5]

- Aggregating Server

Der Aggregating Server wird dazu verwendet, Daten mehrerer untergeordneter Server zu sammeln und für übergeordnete Clients gemeinsam aufzubereiten. In umgekehrter Richtung übernimmt er die Aufgabe der Distribution von Client-Anweisungen an die entsprechenden Server. ***Abbildung 13*** zeigt, dass die Realisierung eines Aggregating

Servers wiederum eine integrierte Client-Komponente für den Zugriff auf die darunterliegenden Server erfordert.

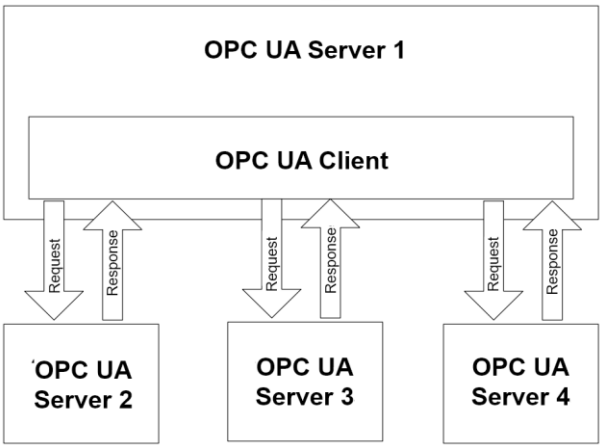


Abbildung 13: Aggregating-Server-Architektur [2]

2.1.4 OPC UA AddressSpace-Modell und Informationsmodelle

Das AddressSpace-Modell nach Part 3 der Spezifikation definiert das Metamodell für alle anderen Informationsmodelle in OPC UA [2]. Das AddressSpace-Modell kommt auf der Implementierungsebene zu tragen und beschreibt die Spezifika für das Deployment des Informationsmodells im Server [1]. Informationsmodelle haben hingegen die Aufgabe, eine logische, strukturierte und standardisierte Beschreibung von realen Systemen, Prozessen und Komponenten auf konzeptioneller Ebene zu realisieren. Informationsmodelle berücksichtigen nicht, wie Informationen letztendlich gespeichert und zur Verfügung gestellt werden. Darstellungen können einzelne Komponenten, eine Maschine oder auch eine ganze Fertigungslinie umfassen. Für eine konsistente Lösung werden sowohl Informationsmodelle als auch der AddressSpace auf die gleiche Art und Weise organisiert. Das heißt, Informationsmodelle beschreiben – unter Einhaltung der Konzepte und Einschränkungen des AddressSpace-Modells – standardisierte Nodes des Server AddressSpace. Das Basis-Informationsmodell nach Part 5 der Spezifikation definiert dabei den AddressSpace eines leeren OPC UA Servers [10]. Das Spezielle an OPC UA ist nach Rinaldi [1], dass diese Technologie die einzige ist, die Dokumentation, Implementierung und Referenzierung, sowie Zugriffe auf Informationsmodelle standardisiert und eindeutig zwischen Informationsmodellierung und deren Umsetzung im Server unterscheidet.

Gemäß [2] sind die grundlegenden Prinzipien der Modellierung in OPC UA:

- Objektorientierte Techniken, Typhierarchien und Vererbung. Der objektorientierte Ansatz kann nicht nur auf Objekte, sondern auch auf Variablen, Referenzen und Datentypen angewandt werden [11].
- Typinformationen, auf die in gleicher Art und Weise wie auf Instanzen zugegriffen werden kann.
- Netzwerk von Knoten zur flexiblen Darstellung von Informationen je nach Anwendungsfall.
- Erweiterungsmöglichkeiten durch Definition eigener Typen und Referenzentypen.
- Anpassungsfähigkeit, da vorhandene Informationsmodelle eines Systems vom Server – ohne zusätzliches Mapping – nativ unterstützt werden.
- Informationsmodellierung immer serverseitig, das bedeutet, dass Modelle auch nur serverseitig existieren.

Genau diese Möglichkeiten und die Flexibilität in der Modellierung erlauben es, OPC UA sowohl horizontal als auch vertikal im gesamten Unternehmen zum Austausch von Daten – inklusive Metadaten und Semantik – einzusetzen.

Abbildung 14 zeigt die standardisierten OPC UA Informationsmodelle und deren hierarchischen Aufbau. Es ist zu sehen, dass das Basis-Informationsmodell mit Modellen für Data Access, Alarms und Conditions, Programs, Historical Data Access, Aggregates und dem noch nicht verabschiedeten PubSub-Modell erweitert werden kann. Daneben besteht die Möglichkeit, herstellerspezifische proprietäre Modelle und zusätzliche Modelle anderer Domänen/Organisationen zu berücksichtigen. Letztere werden als Companion Spezifikationen[1] bezeichnet. Beispielsweise bietet die OPC Foundation ein Basismodell speziell für Gerätespezifikationen an – OPC UA for Devices – damit verschiedene Hersteller ähnliche gerätespezifische Informationen auf die gleiche Weise modellieren können. Daneben sind viele weitere Spezifikationen von Informationsmodellen, beispielsweise für CNC Systeme, IEC 61131-3 (Standard für die Programmierung von SPS), sowie Common Object Models nach ISA-95 bzw. IEC 62264-2:2013 und noch einige mehr verfügbar. In [11] wird empfohlen, wenn möglich immer auf vorhandene Basisinformationsmodelle oder auf Companion Spezifikationen zurückzugreifen.

[1] https://opcfoundation.org/developer-tools/specifications-unified-architecture – 12.08.2018

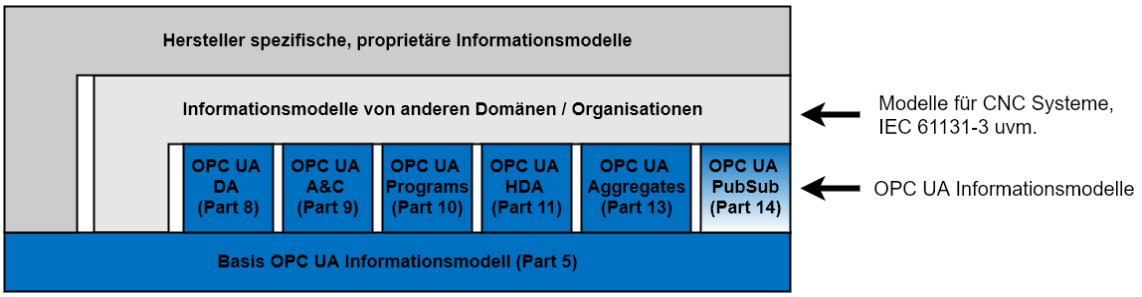


Abbildung 14: Hierarchie und Aufbau von Informationsmodellen [2]

Für die Modellierung wird eine hierarchische Struktur von Objekten verwendet. Wie in ***Abbildung 15*** gezeigt, sind Objekte in OPC UA durch Variablen und Methoden eindeutig definiert. Darüber hinaus können Objekte Beziehungen, beschrieben durch Referenzen, zu anderen Objekten aufweisen. Die Modellierung kann somit alle wichtigen Eigenschaften wie Konzepte, Beziehungen, Einschränkungen, Regeln und Operationen berücksichtigen um die Charakteristik eines Prozesses widerzuspiegeln [1]. OPC UA Informationsmodelle werden üblicherweise entweder im NodeSet-File in XML oder in einer von der OPC Foundation standardisierten grafischen Notation dargestellt.

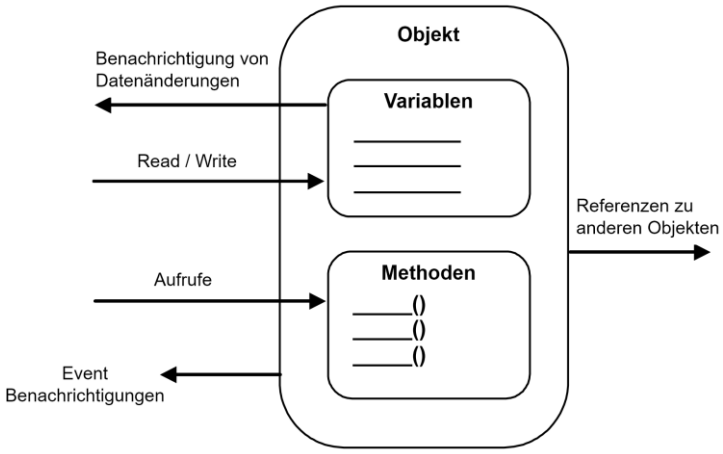


Abbildung 15: OPC UA Objekt-Modell [12]

Die einzelnen Elemente des OPC UA Objektmodells, wie in ***Abbildung 15*** gezeigt, werden im AddressSpace in Form von Nodes dargestellt. Ein Objekt besteht aus einem Set von Nodes. Wie in ***Abbildung 16*** zu sehen ist, werden Nodes in OPC UA mit Attributen beschrieben und die Beziehung zu anderen Nodes mittels Referenzen charakterisiert. Für die Beschreibung werden ein oder mehrere der 21 fix vorgegebenen Attribute verwendet. Jeder Node ist einer von acht NodeClasses (Object, ObjectType, Variable, VariableType, Method, ReferenceType, View oder DataType) zugeordnet und muss neben optionalen Attributen – je nach Zuordnung – gewisse verpflichtende Attribute aufweisen. Die verschiedenen NodeClasses repräsentieren unterschiedliche Elemente des Objektmodells [12]. Neben den Objekten aus der Modellierung enthält der AddressSpace ein Objekt zur

Charakterisierung des Servers selbst. Das Server-Objekt enthält einige Variablen wie z.B. ServerArray (Pointer zu anderen referenzierten Servern), NamespaceArray (Erklärung – siehe folgender Absatz) oder ServerStatus. Das Server-Objekt gliedert sich darüber hinaus in weitere Objekte wie beispielsweise:

- ServerCapabilities – welche Funktionalitäten unterstützt der Server
- ServerDiagnostics – diverse Informationen zur Unterstützung bei der Fehlerbehebung
- VendorServerInfo – für Zusatzinformationen des Herstellers der Server-Implementierung
- ServerRedundancy – welche Redundanzmechanismen unterstützt der Server

Detaillierte Informationen über NodeClasses, Attribute, Referenztypen und Datentypen sind in [10] und [12] zu finden.

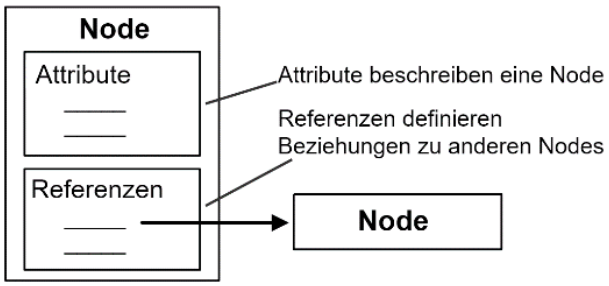


Abbildung 16: OPC UA Node-Modell [12]

Jeder Server enthält das Basis-Informationsmodell. Der Server kann aber zur selben Zeit mehrere Modelle von unterschiedlichen Organisationen oder Domänen unterstützen, z.B. diverse Companion Spezifikationen. Um dennoch jeden Knoten im gesamten Server AddressSpace eindeutig identifizieren zu können, wird der AddressSpace in verschiedene Knotenmengen, sogenannte NodeSets, unterteilt. Um in weiterer Folge einen Knoten einem NodeSet zuordnen zu können, müssen die NodeSets ebenfalls ein einmaliges Identifikationsmerkmal aufweisen. Dazu ist jedem NodeSet ein eindeutiger Namespace zugeordnet. Ein Namespace ist ein URI, aus dem ersichtlich ist, von welcher Autorität oder Organisation die Definition des NodeSets entstammt. Die unterschiedlichen Namespaces bzw. URIs werden im Server im NamespaceArray gespeichert und mit Indizes ausgestattet. Diese sogenannten NamespaceIndexes (ns=0, ns=1, ns=2 usw.) werden im Server anstelle des langen URI zur Ausführung diverser Aktionen verwendet. Index 0 ist dem Namespace von OPC UA und Index 1 dem lokalen Server vorbehalten [10]. Jedem Node kann mit Hilfe seines NodeId-Attributes eindeutig identifiziert werden, da die NodeId den NamespaceIndex bzw. den URI und zusätzlich einen eindeutigen Identifier innerhalb dieses NodeSets beinhaltet. Der NamespaceIndex 0 muss nicht angegeben werden. Der Identifier innerhalb eines NodeSets besteht aus einer Typangabe (Numeric (i),

String (s), Guid (g) oder Opac) und einem Wert [11]. Beispielweise könnten NodeIds wie folgt aussehen:

Syntax laut Spezifikation [4]: **ns=<namespaceindex>;<type>=<value>**

ns=2;s=Motor_an ns=3;i=1212 ns=0;i=84 ⇔ i=84

Jeder Node ist an sein Informationsmodell gebunden und erlaubt somit eine eindeutige Abgrenzung und Unterscheidung zu anderen Modellen.

Das NodeSet, welches die Gruppierung aller im OPC UA Standard verfügbaren und definierten Knoten enthält, ist frei zugänglich[2]. Es ist, wie bereits erwähnt, immer dem Namespace http://opcfoundation.org/UA/ bzw. Namespaceindex 0 zugeordnet [4]. Dieses NodeSet wurde zur Einhaltung einer standardisierten Syntax nach dem dafür vorgesehenen XML-basierenden Schema[3] erstellt. Alle NodeIds des eben beschriebenen NodeSets sind in einem CSV-File[4] zusammengefasst.

Für die Companion Spezifikationen der Informationsmodelle anderer Domänen und Organisationen sind die zugehörigen Namespaces und NodeSets inklusive der Auflistung aller im NodeSet verwendeten NodeIds in deren Anhang angeführt.

In Anhang 8.2.2 ist zu Beginn der Ablauf beschrieben, wie man von Daten zum Informationsmodell kommt und dieses in weiterer Folge im AddressSpace des Servers implementiert. Daran anschließend werden Tools aufgelistet, die bei der Modellierung unterstützen können. Das zurzeit am häufigsten eingesetzte Tool wird näher beschrieben.

2.1.5 Der OPC UA Server

- Beschreibung

Die Beschreibung des Servers in OPC UA erfolgt anhand der Darstellung in ***Abbildung 17***. Der Server stellt dem Client ein objektorientiertes standardisiertes OPC UA Informationsmodell in seinem AddressSpace zur Verfügung. Der AddressSpace repräsentiert das einsatzfähige Datenmodell im Server. Reale Objekte können Objekte physischer Natur oder Softwareobjekte sein, auf die ein OPC UA Server Zugriff hat oder intern verwendet [5]. Wie bereits im vorigen Abschnitt erwähnt wurde, wird jedes Element von modellierten Objekten und deren Beziehungen zueinander in Form von Nodes im AddressSpace dargestellt. Der Server bietet die Möglichkeit, seinen AddressSpace in diverse Views zu unterteilen. Eine View ist eine Teilmenge der Nodes aus dem AddressSpace, wobei per Default der komplette AddressSpace repräsentiert

[2] http://www.opcfoundation.org/UA/schemas/1.04/Opc.Ua.NodeSet2.xml – 12.08.2018
[3] http://www.opcfoundation.org/UA/schemas/1.04/UANodeSet.xsd – 12.08.2018
[4] http://www.opcfoundation.org/UA/schemas/1.04/NodeIds.csv – 12.08.2018

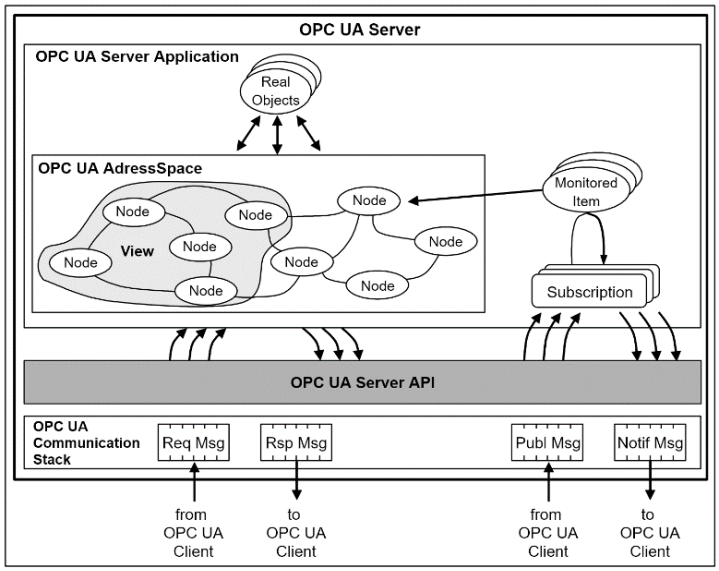


Abbildung 17: OPC UA Server-Architektur [5]

wird. Der Server kann mit Hilfe einer View die Sichtbarkeit und die Interaktionsmöglichkeit eines Clients auf gewünschte Nodes einschränken. Für das Monitoring von Node-Attributen im AddressSpace werden vom Client im Server sogenannte MonitoredItems erzeugt. Tritt eine Datenänderung oder ein Event auf, generiert das entsprechende MonitoredItem eine Benachrichtigung, welche dem Client mittels einer Subscription übermittelt wird [5]. Eine Subscription kann aus mehreren MonitoredItems bestehen. Die Rate, mit der die Informationen einer Subscription ausgelesen werden, kann vom Client über Publish-Messages geregelt werden. Ein Server kann Clients bis zu 39 Services zur Verfügung stellen, welche mittels Requests vom Client genutzt werden können.

Die OPC UA Server Applikation ist der Code zur Implementierung der beschriebenen Serverfunktionalitäten. Für den Nachrichtenaustausch mit Clients nutzt die Serverapplikation die OPC UA Server API. Die Server API ist ein internes Interface, welches eine Trennung zwischen dem Sourcecode der Applikation und des OPC UA Kommunikationsstacks vornimmt. Eine detaillierte Beschreibung der Applikations-Architektur ist in Kapitel 2.1.9 zu finden.

Zur Kommunikation kann ein Server ein oder mehrere in OPC UA verfügbare Transport- und Sicherheitsprotokolle sowie Encodings (Transportprofile) unterstützen. Derzeit definierte Transportprofiles sind:

- UA TCP mit UA Secure Conversation und Binary Encoding
- HTTPS mit Binary Encoding
- HTTPS mit XML Encoding
- HTTPS mit JSON Encoding
- WebSocket mit UA Secure Conversation und Binary Encoding
- WebSocket mit JSON Encoding

Die mögliche Skalierbarkeit in OPC UA erlaubt es, Server für verschiedenste Anwendungsbereiche zu konzipieren. Das bedeutet, dass je nach Einsatzbereich und den damit verbundenen Anforderungen, verschiedene funktionale Ausprägungen von Servern verwendet werden können. Die Art der Ausprägung wird mit sogenannten Profiles, welche ebenfalls von der OPC Foundation definiert wurden, charakterisiert[5]. Ein Profile definiert welche Funktionalitäten eine Applikation aufweisen muss um im Sinne der Profiles die Konformität nach OPC UA gewährleisten zu können. Beim Kauf einer Anwendung sollte deshalb auch immer darauf geachtet werden, welche Profiles in der Anwendung tatsächlich realisiert sind.

- Services

OPC UA bietet insgesamt 39 Services, welche in zehn logische Gruppen – sogenannte Service Sets – unterteilt und im Server implementiert sind. Fünf Services sind für Discovery Prozesse und sechs Services für das Management der Kommunikationsverbindungen zuständig. Die restlichen Services ermöglichen Datenzugriffe und die Interaktion mit dem AddressSpace des Servers. Die Services in OPC UA sind zustandsbehaftet und erfordern deshalb für den Aufruf, je nach Servicetyp, eine bestimmte und bereits aufgebaute Kommunikationsumgebung – vgl. ***Abbildung 5*** in Abschnitt 2.1.2. Die zehn Service Sets sind mit einer kurzen Beschreibung in ***Tabelle 2*** zusammengefasst.

Service Sets	Beschreibung
Discovery Service Set	Auffinden verfügbarer Server im System und Identifikation der Security Konfigurationen für den Verbindungsaufbau.
SecureChannel Service Set	Erstellung eines Kommunikationskanals unter Sicherstellung von Vertraulichkeit und Integrität des Nachrichtenaustausches. Typischerweise im UA Stack und nicht in der OPC UA Applikation implementiert [5].
Session Service Set	Verbindungsaufbau auf Applikationsebene mit der Berücksichtigung von Usern.
NodeManagement Service Set	Möglichkeiten für Clients den AddressSpace des Servers zu modifizieren.
View Service Set	Erlaubt Clients durch Views (Default View ist der komplette AddressSpace) zu browsen.
Query Service Set	Abfrage des AddressSpace.
Attribute Service Set	Lese- und Schreibzugriffe auf Attributwerte.

[5] zurzeit definierte Profiles: https://opcfoundation-onlineapplications.org/ProfileReporting/ – 12.08.2018

Service Sets	Beschreibung
Method Service Set	Aufruf von Methoden.
MonitoredItem Service Set	Definition und Verwaltung von MonitoredItems zum Monitoring von Variablen, Attributen und Eventbenachrichtigungen.
Subscription Service Set	Erstellung und Verwaltung von Subscriptions zur periodischen Übermittlung der Benachrichtigungen an den Client.

Tabelle 2: Service Sets in OPC UA [8]

Die Service Sets mit den gesamten in OPC UA verfügbaren Services sind in Anhang 8.2.3 unter dem Punkt – Services in ***Tabelle 11*** zu finden. Detaillierte Beschreibungen der Services mit den zugehörigen Parametern für die Service-Requests und Service-Responses sind in den Abschnitten 5.4 bis 5.13 von [8] nachzuschlagen.

- Discovery Prozess und Discovery Server

Im einfachsten Fall befinden sich Client und Server am selben Host. Für diesen Fall reicht es aus, dem Client die Adresse des Servers statisch vorzugeben. OPC UA ist jedoch dahingehend ausgelegt, dass eine Vielzahl an UA Applikationen in verschiedensten Ebenen und Netzwerkstrukturen im Unternehmen zum Einsatz kommen kann. Deshalb bietet OPC UA den Discovery Prozess, mit dem Clients alle verfügbaren Server in lokalen und in Remote-Netzwerken auffinden können.

Der Discovery Prozess ermöglicht es Clients, Server im Netzwerk zu finden und Informationen wie deren Applikationstyp, Applikationsnamen, Hostnamen, IP-Adresse und Portnummer in Erfahrung zu bringen. Für die weitere Beschreibung muss zuvor der Begriff eines Endpoints erläutert werden. Ein Endpoint ist eine spezielle physische Adresse im Netzwerk, welche der Client für Zugriffe auf Services im Server nutzen kann [8]. Jeder Server unterstützt im Normalfall zumindest zwei Endpoints, wobei einer davon ausschließlich für den Discovery Prozess vorgesehen ist und als Discovery Endpoint bezeichnet wird. Für den Discovery Prozess verfügt jeder Server über Discovery Services, welche ein Client über den Discovery Endpoint in Anspruch nehmen kann. Dieser Endpoint weist keinerlei Sicherheit auf, d.h. es wird für den Zugriff keine Session benötigt und keine Nachrichtensignatur und -verschlüsselung verwendet [8]. Eine Erhöhung der Sicherheit wird durch Nutzung von Sicherheit auf der Transportebene, beispielsweise durch Verwendung von HTTPS, erreicht [1]. Der Client interagiert mit einem Server über diesen speziellen ungesicherten nicht-operationalen Discovery Endpoint und eruiert dabei [1]:

- Security Policies – Algorithmen für das Signieren und Verschlüsseln
- Security Modi – No Security bzw. None, Sign oder Sign and Encrypt

- unterstützte Transportmechanismen aller verfügbaren (operationalen) Endpoints

Der Client kann damit in Erfahrung bringen, ob der vorliegende Server einen Endpoint aufweist, der den Anforderungen der Clientapplikation genügt. Ebenso wird dem Client auf diesem Wege das Applikationszertifikat des Servers übergeben. Der Client speichert alle Informationen und kann bis zur Änderung der Serverkonfiguration eine Verbindung zum Server aufbauen, ohne den Discovery Prozess erneut zu durchlaufen [13].

Betrachtet man Umgebungen mit ein paar wenigen Servern, genügt es, dass jeder einzelne Server die Discovery-Services über den Discovery Endpoint zur Verfügung stellt. Ein Client muss dafür die lokalen Hostnamen (TCP/IP Adresse) und Ports der Server kennen, um die Endpoints erreichen zu können. Adressen eines Discovery Endpoints könnten wie folgt aussehen:

- für OPC UA TCP → opc.tcp://localhostname:4840/UADiscovery
- für OPC UA WebSocket → opc.wss://localhostname:443/UADiscovery
- für OPC UA HTTPS → https://localhostname:443/UADiscovery

Für das Management einer Vielzahl an UA Servern können je nach deren Verteilung im Netzwerk eigens dafür vorgesehene Discovery Server eingesetzt werden. Diese Server sammeln sämtliche Discovery Informationen der im Netzwerk befindlichen OPC UA Server. Ein Server muss sich dafür beim Discovery Server registrieren und übergibt diesen damit das Handling von Discovery-Nachrichten. Die Registrierung erfolgt beispielsweise über den Endpoint https://localhostname:443/UADiscovery/registration. Ein Client kann durch den Zugriff auf den Discovery Server im Laufe des Discovery Prozesses die Informationen aller registrierten Server bestimmen. Bei Discovery Servern wird zwischen drei Typen unterschieden:

- Local Discovery Server (LDS): Übernehmen die Verwaltung der Discovery Informationen aller registrierten Server, welche auf derselben Plattform bzw. im gleichen lokalen Netzwerk wie der LDS selbst laufen.
- Local Discovery Server mit Multicast Extension (LDS-ME): Verwalten die Discovery Informationen aller registrierten Server, die einem Multicast Subnetz angehören.
- Global Discovery Server (GDS): Realisieren die Verwaltung der Discovery Informationen aller Server in großen Systemen bzw. Netzwerken. Ein GDS ist selbst ein OPC UA Server und bildet die im System verfügbaren Server in seinem AddressSpace ab [13]. Das bedeutet, dass ein Client nicht wie in den vorangehenden Fällen durch einfache Anfragen die Informationen über Server einholt, sondern durch den AddressSpace des GDS navigiert und auf diesem

Weg herausfinden kann, welche Arten von Servern im System verfügbar sind. Ein GDS übernimmt meist zusätzlich die Funktionen eines Zertifikatmanagers [13].

In den Abschnitten 4.3.3 bis 4.3.6 von [13] sind die verschiedenen Discovery Prozesse mit den zugehörigen Serviceaufrufen grafisch dargestellt. Alle notwendigen Services für den Discovery Prozess sind im Discovery Service Set zusammengefasst und in Abschnitt 5.4 von [8] detailliert beschrieben.

2.1.6 Der OPC UA Client

Ein OPC UA Client besitzt lediglich die Funktionalitäten, die vom Server zur Verfügung gestellten Services aufzurufen und entsprechende Service-Responses zu verarbeiten. Bei jedem Serviceaufruf erhält der Client in der Antwort vom Server servicespezifische Statuscodes, welche einerseits den Status des Serviceaufrufs selbst und andererseits den Status der einzelnen Operationen, die mit dem Serviceaufruf einhergingen, anzeigen [8]. Die vom Client am häufigsten verwendeten Services zur Interaktion mit dem AddressSpace sind die Services des Subscription Service Sets und des Attribute Service Sets [1]. ***Abbildung 18*** zeigt eine typische OPC UA Client-Architektur [5]. Eine detaillierte Beschreibung der Applikations-Architektur ist in Kapitel 2.1.9 zu finden.

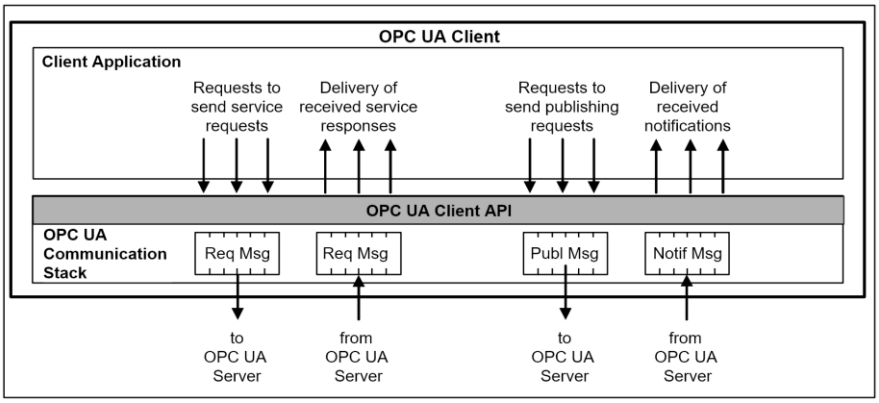


Abbildung 18: OPC UA Client-Architektur [5]

Ein Client sollte mehrere Transportprofiles unterstützen, um einen kompatiblen Kommunikationspartner für möglichst viele unterschiedliche Server darzustellen. Analog zu den Servern können auch bei Clients verschiedene funktionale Ausprägungen für Client-Applikationen verwendet werden. Wie bereits erwähnt wird die Art der Ausprägung mit Hilfe von Profiles definiert und charakterisiert[6].

Die notwendigen Informationen, um eine authentifizierte Verbindung und in weiterer Folge eine gesicherte und autorisierte logische Kommunikationssession aufbauen zu können, erhält der Client vom Server während des Discovery Prozesses. Der

[6] zurzeit definierte Profiles: https://opcfoundation-onlineapplications.org/ProfileReporting/ – 12.08.2018

Discovery Prozess wurde im Detail in Kapitel 2.1.5 diskutiert. Da dieser Prozess ohne ausgeprägter Sicherheit abläuft, müssen die empfangenen Informationen des Discovery Endpoints vom Client auf Richtigkeit überprüft werden [8]. Einige Punkte die vom Client verifiziert werden sollten, sind in Abschnitt 5.4.1 von [8] angeführt.

2.1.7 Security-Konzepte in OPC UA

Mit der steigenden Vernetzung sowie der horizontalen und vertikalen Interaktion unterschiedlicher Systeme innerhalb der Automatisierungspyramide, steigt die Anforderung hinsichtlich einer sicheren und zuverlässigen Kommunikation. Ein unerlaubter Eingriff in die Kommunikation kann von ökonomischen Schäden des Unternehmens bis hin zu Personenschäden bei Produktionsmitarbeitern führen. OPC UA realisiert, wie in ***Abbildung 19*** dargestellt, notwendige Sicherheitsziele wie Authentifizierung, Autorisierung, Vertraulichkeit, Integrität, Verfügbarkeit und Nachvollziehbarkeit auf verschiedenen Ebenen einer dreischichtigen Sicherheitsarchitektur. Eine wichtige Eigenschaft von OPC UA ist die Unterscheidung zwischen Applikationen und Usern sowie deren getrennte Authentifizierung [1].

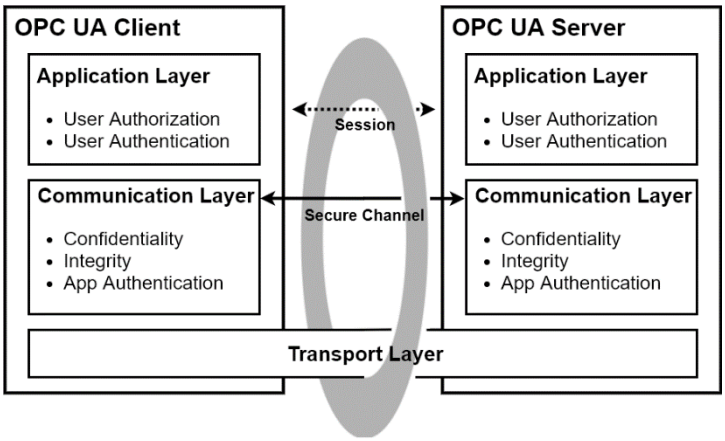


Abbildung 19: Sicherheitsarchitektur von OPC UA [14]

- Applikations-Layer – Session

Der Informationsaustausch zwischen Server und Client findet in der Applikationsebene innerhalb einer sogenannten Session statt. Eine Session, in der eine Authentifizierung und Autorisierung des Benutzers vollzogen wird, wird mit den Session-Services in zwei Schritten aufgebaut.

- Kommunikations-Layer – Secure Channel

Die Kommunikation der Session in der Applikationsebene erfolgt über einen Secure Channel in der Kommunikationsebene, welcher einen sicheren Nachrichtenaustausch zwischen Server und Client gewährleisten soll. Die

SecureChannel-Services sind mit Hilfe des Protokoll-Stacks implementiert und stellen beispielsweise folgende Sicherheitsmechanismen zur Verfügung [14]:

- Nachrichtenverschlüsselung für die Erhaltung der Vertraulichkeit
- Nachrichtensignatur für die Wahrung der Integrität
- Validierung digitaler Zertifikate für die Applikationsauthentifizierung

Letzteres wird beispielsweise beim Aufbau eines Secure Channels mit HTTPS nicht unterstützt und muss, falls dennoch erwünscht, auf Basis von Nutzerdaten durchgeführt werden [14].

Die Herausforderung beim Aufbau eines Secure Channels besteht nun darin, dass ein sicherer Austausch von kryptografischen Schlüsseln zwischen Client und Server zur Nachrichtenverschlüsselung in einer unsicheren Umgebung stattfinden muss. Dazu verwenden die beteiligten Kommunikationsteilnehmer einen speziellen Schlüsselaustauschalgorithmus (engl. Key Exchange Algorithm), der funktionell ähnlich einem SSL-Handshake ist [14]. Für den Fall, dass ein Secure Channel zusammenbricht, bleibt die zugehörige Session im Applikations-Layer bis zum Ablauf der Session-Lifetime bestehen und erlaubt es dem Client in dieser Zeit die Verbindung mit einem neuen Secure Channel wiederherzustellen [14]. Ansonsten wird auch die Session in der Applikationsebene geschlossen.

- Transport-Layer

Die Transportebene ist für die Übermittlung der von der Kommunikationsebene kommenden, verschlüsselten und signierten Nachrichten zuständig. Die Übertragung nutzt dabei eine Socket-Verbindung [2]. Im Falle eines Verbindungsabbruchs ist der übergeordnete Kommunikations-Layer für den Aufbau einer neuen Verbindung zuständig [14]. In dieser Zeit darf der Secure Channel nicht unterbrochen werden.

Im Gegensatz zu den meisten anderen Kommunikationssystemen ermöglicht der schichtweise Aufbau von OPC UA eine Entkopplung des Request/Response-Nachrichtenprotokolls von der Codierung (Binary, XML oder JSON), dem Mechanismus für Nachrichtensicherheit (OPC UA Secure Conversation) und dem Transport-Mechanismus (UA TCP, HTTPS oder WebSocket) [1], [4]. Aus diesem Grund ist OPC UA sehr flexibel und wandlungsfähig, da neu veröffentlichte Verschlüsselungs- oder Sicherheitsmechanismen und Transportprotokolle nachträglich in die Struktur integriert werden können [1]. Der Prozess der Zuordnung von Codierung, Security- und Transportmechanismen wird als Mapping bezeichnet. ***Abbildung 20*** zeigt einen grafischen Überblick derzeitiger Mapping-Möglichkeiten. ***Abbildung 21*** illustriert anhand eines Service-Requests vom Client an den Server auf welchen Ebenen und in welcher Abfolge das Mapping stattfindet.

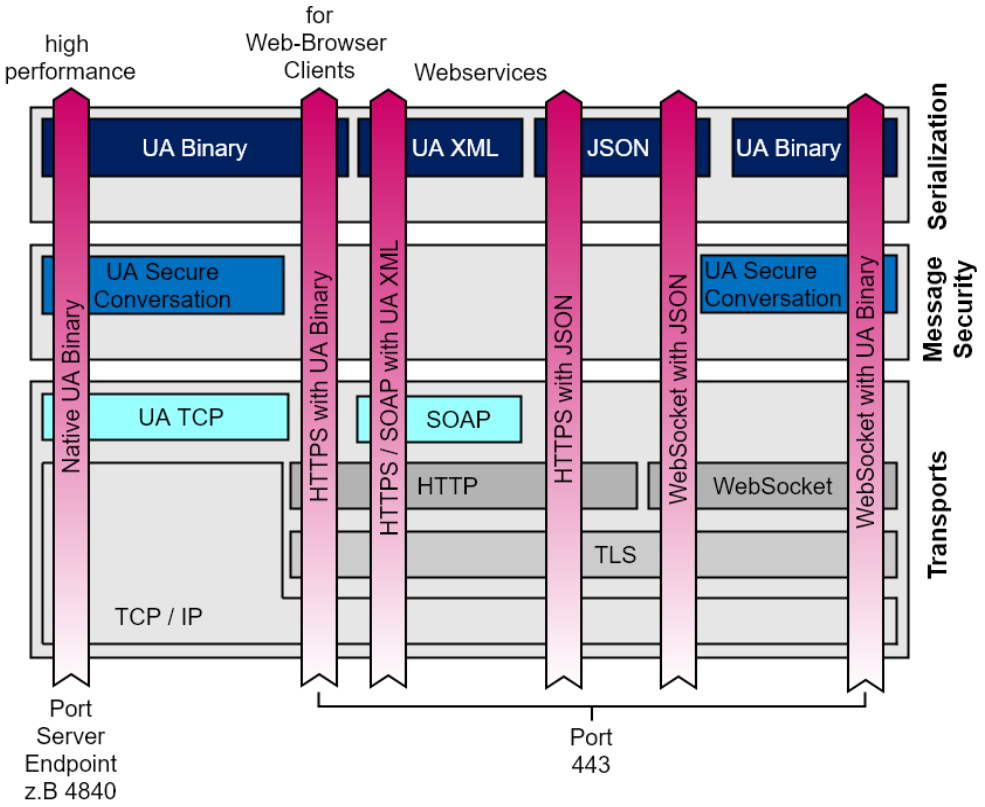


Abbildung 20: Mapping-Möglichkeiten für Client/Server-Architekturen

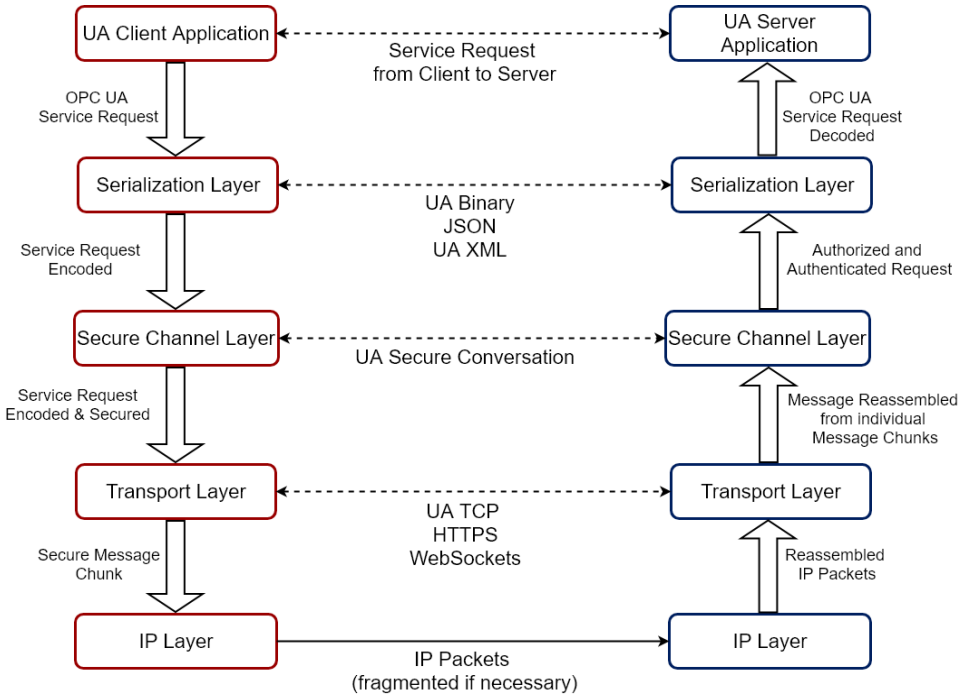


Abbildung 21: Mapping zu einem Service Request vom Client an den Server [1]

- Ablauf für den Aufbau einer sicheren Kommunikationsumgebung anhand der OPC UA TCP Secure Conversation mit Binary Encoding

Grundsätzlich basiert der Vorgang bei der OPC UA TCP Secure Conversation mit Binary Encoding auf einem asymmetrischen Schlüsselaustausch unter Verwendung des X.509 Standards und einer darauffolgenden symmetrischen Verschlüsselung der Nachrichten.

Der Client holt im Zuge des Discovery Prozesses die Informationen über die Security Policies und Security-Modi der verfügbaren Server-Endpoints sowie das digitale Applikationsinstanz-Zertifikat des Servers ein. Grundsätzlich gilt, dass jede OPC UA Applikationsinstanz ein eigenes digitales Zertifikat besitzt, mit dem sich die Applikation bei ihrem Kommunikationspartner identifiziert. Für die Zertifikate wird der X.509 Zertifikatstandard verwendet. Mit den vom Administrator zuvor festgelegten Security Policies teilt der Server dem Client mit, welche Algorithmen für das Signieren, Verschlüsseln und für die Schlüsselbestimmung unterstützt werden [14]. Als Sicherheitsmodi stehen hierfür No Security bzw. None, Sign und Sign And Encrypt zur Verfügung [2]. Neben den Security Policies wird dem Client ebenfalls mitgeteilt, in welcher Form sich ein User beim späteren Aufbau einer Session am Server identifizieren muss. Mögliche Optionen sind:

- Anonymous – keine Identifikation erforderlich
- Username und Passwort
- Zertifikat – ein eigens für den User ausgestelltes digitales Zertifikat

Kommunizieren mehrere Clients mit dem gleichen Server, kann jeder Client für sich eine der vom Server angebotenen Policies auswählen [14]. Des Weiteren erhält der Client das X.509 Zertifikat der Serverapplikation. Der Client überprüft ob das Zertifikat bereits in seiner Trust-List vorhanden ist und somit als vertrauenswürdig eingestuft wurde. Falls dies nicht der Fall ist, kann ein Administrator das Zertifikat in die Trust-List aufnehmen oder in die sogenannte Revocation-List für nicht vertrauenswürdigen Zertifikate verschieben, womit die Kommunikation zur jeweiligen Applikation unterbunden wird. Das Zertifikat der Serverapplikation enthält unter anderem den sogenannten Public Key des Servers [14]. Jedem Public Key ist ein Private Key zugeordnet und umgekehrt – sie bilden ein sogenanntes Schlüsselpaar. Der Private Key darf ausschließlich dem Besitzer zur Verfügung stehen [1]. Der Client übermittelt seinen Public Key in einem X.509 Zertifikat und zusätzliche geheime Informationen mittels der OpenSecureChannel-Servicenachricht an den Server. Diese Nachricht ist asymmetrisch mit dem Public Key des Servers verschlüsselt und ebenfalls mit einer asymmetrischen Signatur durch den Private Key des Clients versehen. Das Zertifikat selbst ist jedoch unverschlüsselt, damit der Empfänger die asymmetrische Signatur verifizieren kann. Im nächsten Schritt entschlüsselt der Server die Nachricht unter Nutzung seines Private Keys und verifiziert die asymmetrische Signatur mit dem Public

Key des Clients. Aus den geheimen Informationen des Clients bestimmt der Server nun einen neuen Satz kryptografischer Schlüssel für die Sicherung des weiteren Nachrichtenaustauschs. Die geheimen Informationen des Servers sind im Service-Response an den Client enthalten, womit dieser ebenfalls das gleiche Set an Schlüsseln bestimmen kann. Ab diesem Zeitpunkt findet eine sichere Kommunikation zwischen Server und Client unter Verwendung symmetrisch verschlüsselter und symmetrisch signierter Nachrichten statt. Ebenso ist anzumerken, dass diese symmetrischen Schlüssel je nach Lifetime des Secure Channels periodisch geändert werden, um potenziellen Angreifern nur eine geringe Anzahl an Nachrichtensequenzen für die Bestimmung der Schlüssel zu geben. Der Aufbau des Secure Channels ist damit abgeschlossen. [14]

Im letzten Schritt wird nun eine Session mit einem vertrauenswürdigen User in zwei Schritten erstellt und überprüft, ob der Client, welcher die User- Informationen übergibt, derselbe ist, der zuvor den Aufbau des Secure Channels veranlasst hat. Dazu macht der Client einen CreateSession-Request und es wird überprüft, ob das Applikationsinstanz-Zertifikat des Clients valide ist und ob dieses ebenfalls im Zusammenhang mit dem Aufbau des Secure Channel verwendet wurde [1]. Der Client validiert anhand der Signatur des Service-Response, ob die Antwort tatsächlich vom richtigen Server kommt. Zu guter Letzt nutzt der Client das ActivateSession-Service und übergibt damit dem Server die Informationen des Users. Wie bereits zuvor erwähnt kann für die Identifikation des Benutzers Anonymous, Username/Passwort und Zertifikat verwendet werden. Ist der Client derjenige, der auch zuvor die Session mit dem CreateSession-Request erstellt hat und sind die Userdaten zulässig, dann werden diese an den Applikations-Layer des Servers weitergereicht, validiert und dem Benutzer entsprechende Zugriffsrechte eingeräumt [1].

- Security Threats und Empfehlungen

In der Sicherheitsanalyse OPC UA [15] werden folgende Bedrohungen für Umgebungen mit OPC UA Implementierungen genannt:

- Denial of Service (DoS) Attacken (z.B. Message Flooding)
- Message Spoofing
- Message Alteration oder Message Replay
- Eavesdropping
- Malformed Messages
- Server Profiling
- Session Hijacking
- Rogue Server
- Kompromittierung von Nutzerdaten

Beschreibungen der einzelnen Bedrohungen können [14] und [15] entnommen werden. Eine Sicherheitsanalyse in [15] ergab, dass durch Verwendung des Security-Modes Sign And Encrypt allen Bedrohungen bis auf DoS-Angriffen ausreichend entgegengewirkt werden kann und damit die Schutzziele bis auf die Verfügbarkeit erreicht werden. Des Weiteren wird in der Analyse darauf hingewiesen, dass sich der Schutz gegen DoS-Attacken grundsätzlich als schwierig erweist und dies keineswegs eine OPC UA spezifische Problematik darstellt. Das Risiko von DoS-Angriffen kann nur durch zusätzliche Maßnahmen außerhalb der OPC UA Infrastruktur verringert werden.

Die Algorithmen zur Verschlüsselung und zum Signieren sollten natürlich bei Bedarf auf zur jeweiligen Zeit als sicher angesehene Algorithmen umgestellt werden. Nach [15] wird aktuell Basic256Sha256 empfohlen. Ein anderer Aspekt, der dabei berücksichtigt werden sollte, ist, dass sich bei steigender Komplexität der Algorithmen und gleichbleibender Rechenleistung früher oder später die Kommunikationseffizienz vermindert [9].

- Einstellungen für Firewalls

Bei Nutzung von OPC UA TCP (tcp.opc) muss der Port des jeweiligen Server-Endpoints in der Firewall geöffnet werden, wohingegen bei Verwendung von HTTPS oder WebSocket lediglich Port 443 freigegeben werden muss [16].

2.1.8 Redundanzkonzepte

OPC UA bietet, mithilfe spezieller Datenstrukturen und Services, Redundanzkonzepte zur Vervielfältigung von Client- und Server-Applikationen an [8]. Aufgrund einer redundanten Betriebsweise von systemkritischen Komponenten kann die Zuverlässigkeit des Gesamtsystems deutlich erhöht werden.

- Client-Redundanz

Es kann sinnvoll sein, eine Client-Redundanz für eine streng kontinuierliche Überwachung von Produktionsprozessen einzusetzen. Das in ***Abbildung 22*** dargestellte Konzept basiert einerseits darauf, dass vom Backup Client ein Monitoring, von den im AddressSpace (Server) abgelegten Informationen des aktiven Clients durchgeführt wird. Der Backup-Client benötigt die SessionId sowie die SubscriptionId des aktiven Clients [2]. Des Weiteren erfordert das Konzept die Nutzung des sogenannten TransferSubscription-Services. Mit diesen Services können aktive Subscriptions auf Daten in eine andere Session übertragen werden. Fällt nun der aktive Client aus, erkennt der Backup-Client die Statusänderung der Session und nutzt das TransferSubscription-Service, um die gesamten aktiven Subscriptions auf die andere Session zu übertragen [2]. Das ist möglich, da die Lifetime von Subscriptions nicht an die der Session, in welcher die Subscriptions erzeugt wurden, gebunden ist

[8]. Der Server muss im Stande sein, die an den Client zu übertragenden Daten zu puffern, um diese, falls nötig, nochmals übermitteln zu können und somit Datenverlust zu verhindern [2].

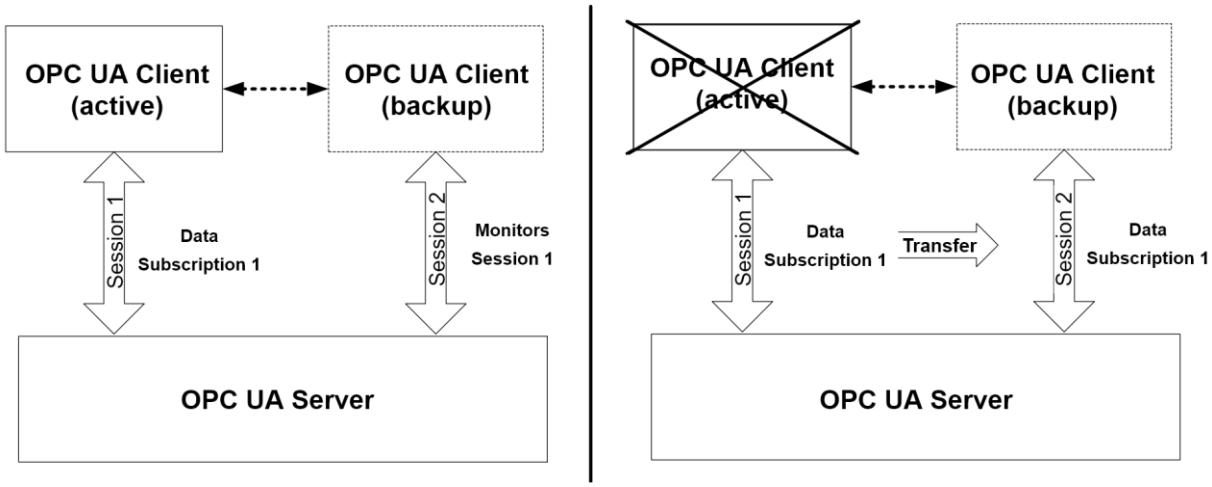


Abbildung 22: Mögliches Client-Redundanzkonzept [2]

- Server-Redundanz

Bei der Server-Redundanz wird zwischen transparenter und nicht-transparenter Server-Redundanz unterschieden. Alle Server im Redundanz-Set müssen den gleichen AddressSpace mit gleichen NodeIds, gleichem Browse-Path und gleicher Struktur sowie identer Logik für das Setzen des Service-Levels aufweisen [8]. Der Service-Level eines Servers bietet einem Client die Möglichkeit den Gesundheitsstatus des Servers zu beurteilen. Weitere Informationen sind in [8] in Abschnitt 6.6.2.4.2 zu finden. Lediglich Nodes aus dem lokalen Server-Namespace, wie beispielsweise die ServerDiagnostic-Nodes, dürfen bei den Servern aus dem Redundanz-Set voneinander abweichen [8].

Transparent Server-Redundanz

Bei dem Prinzip der transparenten Server-Redundanz merkt ein Client im Fehlerfall nicht, dass ein Server ausgefallen ist. Der Server allein ist dafür zuständig, dass der Client weiterhin ungestört Daten konsumieren kann. Aus diesem Grund ist eine Spiegelung des aktiven Servers in Form eines Backup-Servers vorzunehmen. Beide Server besitzen idente ServerUri und EndpointUrl können aber mithilfe einer eindeutigen ID unterschieden werden [8]. Das Prinzip ist in ***Abbildung 23*** gezeigt.

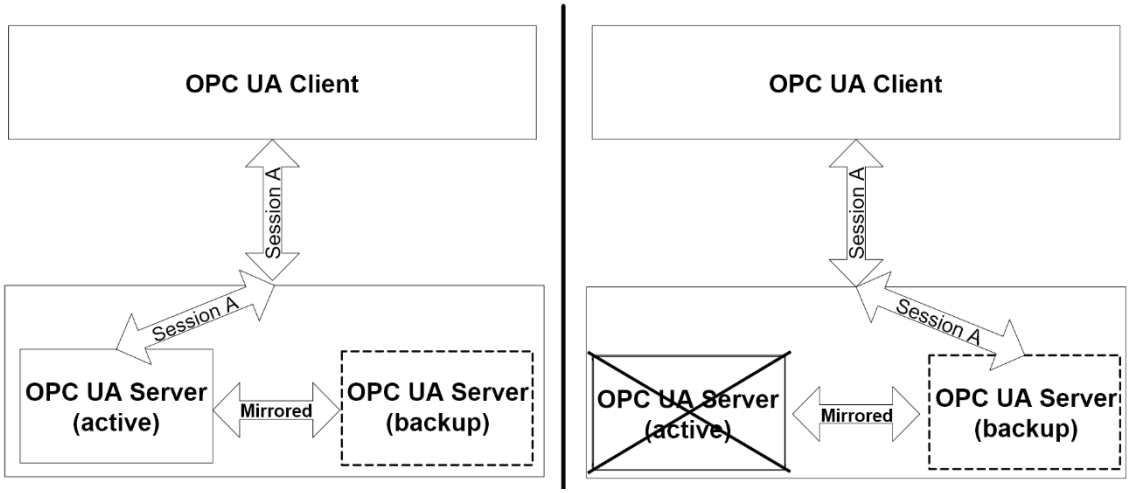
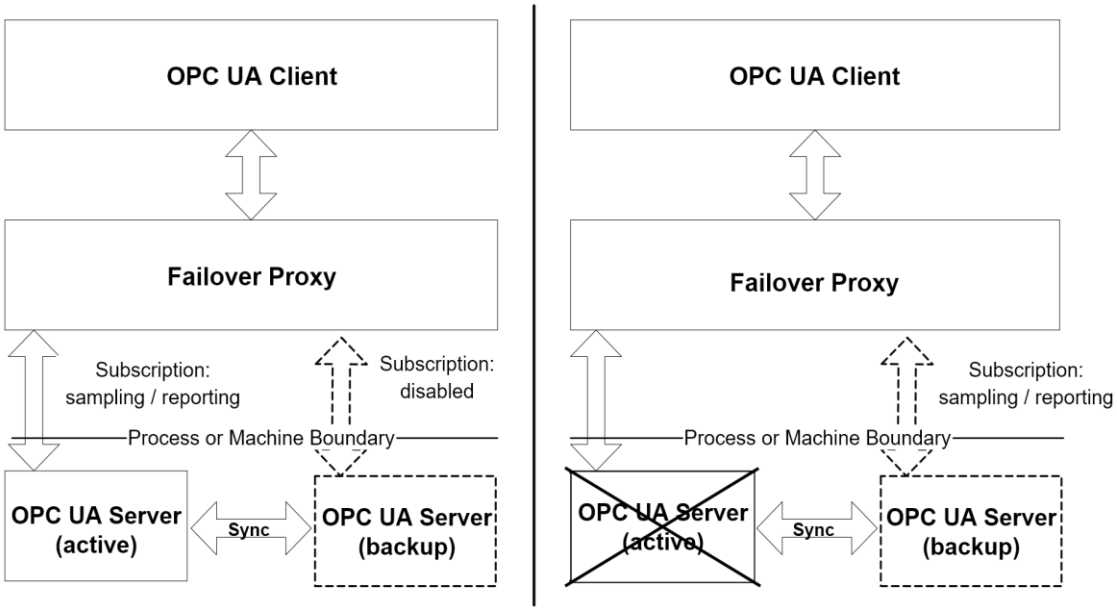


Abbildung 23: Prinzip der Transparent Server-Redundanz [2]

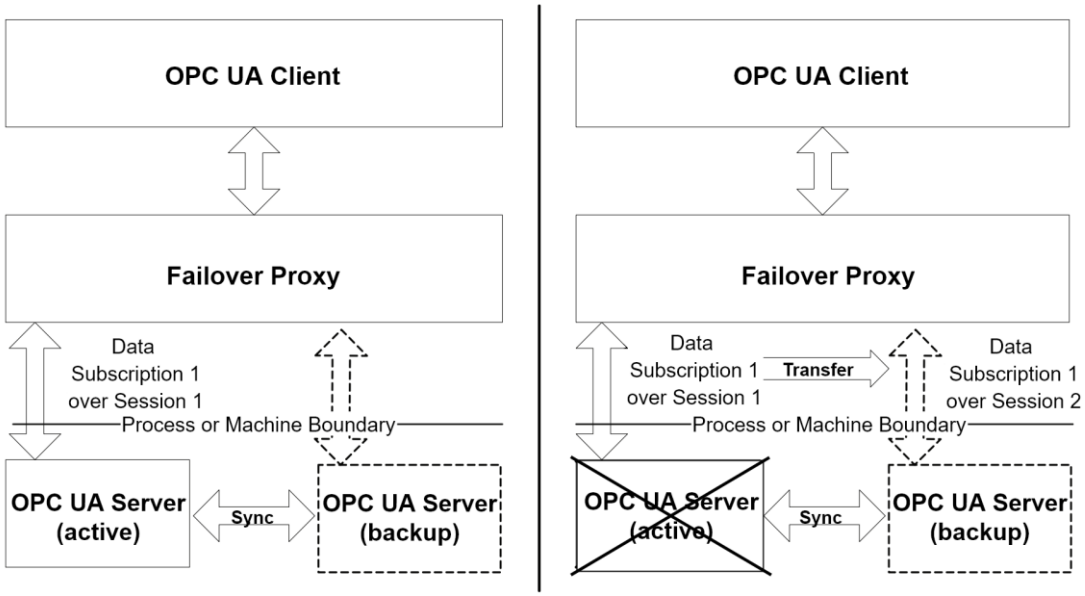
Non-Transparent Server-Redundanz

Bei diesem Konzept übernimmt der Client die entsprechenden Aktionen, um den Datenverkehr, während und nach eines Fehlerfalles des Servers, aufrechtzuerhalten und somit Datenverlust zu verhindern. Die Server weisen jeweils eine eindeutige IP-Adresse auf [8]. Eine in OPC UA bereitgestellte Datenstruktur erlaubt es dem Client die Server im Redundanz-Set, inklusive deren Modi im Fehlerfall, zu identifizieren und Informationen über notwendige Aktionen entsprechend dem Mode zu erhalten [8]. Als Modi stehen Cold, Warm, Hot und Hot And Mirrored zur Verfügung. Detaillierte Informationen über die Modi können [8] ab Abschnitt 6.6.2.4.4 entnommen werden.

Der Client muss im Fehlerfall eine neue Session zum Backup-Server erstellen und dessen Subscriptions, entweder duplizieren, oder mit dem TransferSubscription-Service transferieren. Da dieser Mechanismus von allen Clients genutzt werden kann, wird dies oftmals in einer separaten Komponente, dem sogenannten Failover Proxy, implementiert [2]. Im ersten Fall, wie in ***Abbildung 24 (oben)*** dargestellt, dupliziert der Client seine Subscriptions. Der Client muss seine Subscriptions auf beiden Servern synchron aufrechterhalten, wobei das Sampling und Reporting ausschließlich am aktiven Server stattfindet [2]. Im Fehlerfall werden die Anfragen auf den funktionierenden Server vom Failover Proxy umgeleitet. Als zweite Möglichkeit zeigt ***Abbildung 24 (unten)*** ein Konzept, welches das TransferSubscription-Service nutzt. Der Failover Proxy übernimmt das Anlegen der Abonnements am aktiven Server. Dieser Server ist aber dafür verantwortlich die Subscriptions synchron, durch Spiegelung auf den Backup-Server, aufrechtzuerhalten [2]. Tritt nun ein Fehlerfall auf, erzeugt der Proxy eine neue Session auf dem Backup-Server und nutzt das TransferSubscription-Service, um die Abonnements der zuvor aktiven Session zu transferieren.



Möglichkeit 1: *Subscriptions* duplizieren



Möglichkeit 2: *Subscriptions* transferieren

Abbildung 24: Konzepte der Non-Transparent Server-Redundanz [2]

2.1.9 Die Applikations-Architektur

Die Applikations-Architektur besteht grundsätzlich aus drei verschiedenen funktionalen Software-Layern: Applikation, SDK und Stack. Die Bestandteile der Layer sind in ***Abbildung 25*** dargestellt, wobei mehrere Abbildung aus [2] in einer Darstellung vereint wurden.

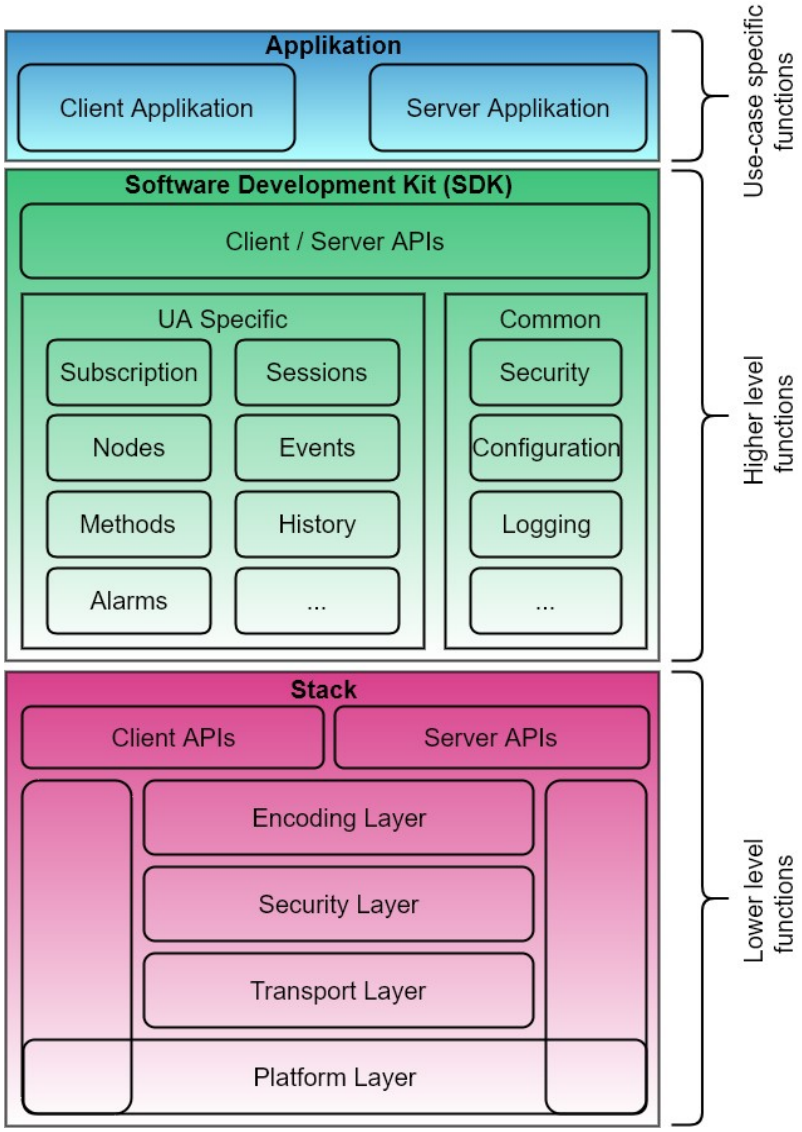


Abbildung 25: Applikations-Architektur

- Applikation

Als Applikation kommt entweder ein Client oder ein Server zum Einsatz, wobei ein Client für den Datenkonsum und ein Server zur Bereitstellung von Daten zuständig sind.

- Software Development Toolkit – SDK

Mithilfe von Client- und Server-SDKs kann der Aufwand für die Entwicklung einer Applikation mit erforderlichen OPC UA Funktionalitäten deutlich reduziert werden. Das SDK setzt auf dem OPC UA Protokoll Stack auf und besteht im Wesentlichen aus drei Teilen:

1. Die Client/Server Application Programming Interfaces (APIs) sind das Verbindungstück zur Applikationsebene und werden vom Client für das Senden bzw. Empfangen von Requests bzw. Responses benötigt. Der Server sollte Interfaces, für die Konfiguration und Initialisierung des SDK sowie für die Integration untergeordneter Systeme zum Datenaustausch bereitstellen [2].
2. Im OPC UA spezifischen Teil des SDK sind die Konzepte und Services nach der OPC UA Spezifikation implementiert [2]. Diese Implementierungen müssen im SDK des Clients und des Servers realisiert sein.
3. Im dritten Teil des SDK sind gewöhnliche Funktionen, wie beispielsweise das Zertifikatmanagement für den Aufbau sicherer Verbindungen, Konfiguration der Applikation und Protokollierung implementiert [2].

- Stack

Im OPC UA Stack sind diverse OPC UA Mappings implementiert und die Kommunikation über Prozess- und Netzwerkgrenzen hinaus realisiert. Die APIs des Stacks repräsentieren die Schnittstellen zu übergeordneten Layern. Ein Stack kann nach [2] nachfolgende Methoden zur Verfügung stellen:

- die Konfiguration des Stacks
- das Management des Verbindungsaufbaus
- das Senden von Servicenachrichten
- die Benachrichtigung der darüber liegenden Layer bei Erhalt einer Nachricht

Der Encoding-Layer ist für das Encodieren – in Richtung des Security-Layer – bzw. in umgekehrter Richtung für das Decodieren von Nachrichten zuständig. Der Security-Layer eines Empfängers überprüft eingehende Nachrichten um herauszufinden, auf welche Art und Weise die Nachricht abgesichert ist. Der Transport-Layer ist für das Senden und Empfangen von Nachrichten, sowie der Handhabung von Fehlern auf der Vermittlungsschicht (OSI Layer 3) verantwortlich [2]. Es besteht die Möglichkeit die beschriebenen Layer plattformunabhängig zu entwickeln und anschließend auf einen Platform-Layer aufzubauen [2]. Der Platform-Layer enthält den notwendigen plattformspezifischen Code, wie etwa die Integration spezieller Bibliotheken für das Socket-Management, Threads oder kryptografische Operationen [2]. Der Stack weist in diesem Fall eine hohe Wiederverwendbarkeit auf, da für eine andere Plattform lediglich der Code des Platform-Layers geändert werden muss.

In ***Abbildung 26*** ist durch Vereinigung mehrerer Abbildungen aus [2] das Zusammenspiel der Software-Layer für das Client/Server-Kommunikationsprinzip exemplarisch dargestellt.

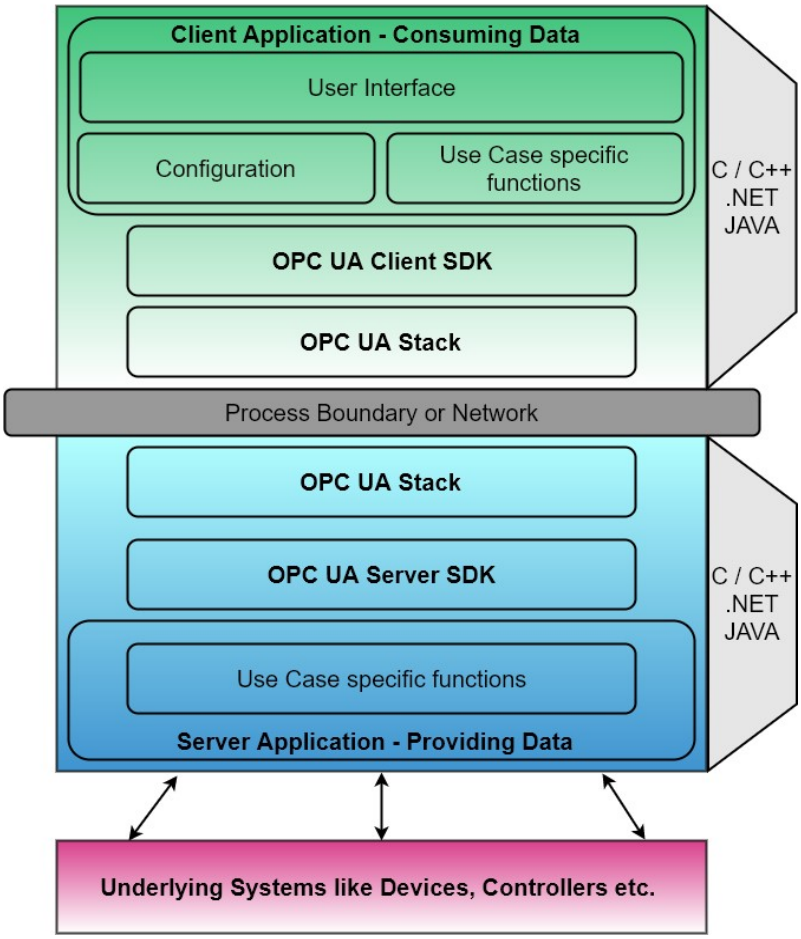


Abbildung 26: Software-Layer bei der Client/Server-Interaktion

2.2 Kommunikationsprotokoll MQTT

Das Message Queue Telemetry Protocol – MQTT – wurde 1999 von Andy Stanford-Clark von IBM und Arlen Nipper von Cirrus Link Solutions entwickelt und MQTT Version 3.1.1 2014 von der Organization for the Advancement of Structured Information Standards (OASIS) als Protokoll des Internet der Dinge (IoT) standardisiert [17]. Seit Mitte 2016 ist MQTT ein internationaler Standard ISO/IEC 20922 [18].

MQTT ist ein schlankes, offenes und leicht zu implementierendes Protokoll, welches auf den TCP- (OSI Layer 4) und IP-Stack (OSI Layer 3) aufsetzt. Der schlanke Aufbau weist im Vergleich zu anderen TCP-basierenden Protokollen einen stark reduzierten Overhead auf [19], wobei der fixe Headeranteil nur 2 Byte benötigt und alle nötigen Kontrollinformationen des Protokolls enthält. MQTT setzt auf eine asynchrone nachrichtenorientierte Kommunikation unter Verwendung eines Publish/Subscribe Interaktionsmodells zwischen Clients und Server (Broker). Das allgemeine Modell kann ***Abbildung 27*** entnommen werden. Es ist ebenso möglich, eine Hierarchie von mehreren Brokern aufzubauen (Bridging), die dazu eingesetzt werden könnte, um die Broker an mehreren Produktionsstandorten als Schnittstelle zwischen den Geräten und einem zentralen Broker fungieren zu lassen [20]. Ein Client kann sowohl die Funktion eines Publishers als auch eines Subscribers übernehmen. Die Clients als Publisher veröffentlichen Daten in Form von Topics, die dem Broker übermittelt werden. Der Broker sammelt diese Topics und verteilt diese an die entsprechenden, hier als Subscriber fungierenden Clients, welche ein oder mehrere Topics abonniert haben. Das Konzept von Topics ermöglicht eine hierarchische Nachrichtenkategorisierung, wodurch ein Client sehr gezielt Topics oder auch nur Teile davon abonnieren und empfangen kann. Mit Hilfe sogenannter Wildcards (+, #) können gleichzeitig mehrere Topics mit nur einer einzigen Subscription abonniert werden. Sobald eine Änderung eines Topics dem Broker bekannt gemacht wird, wird diese sofort an die entsprechenden Subscriber übermittelt.

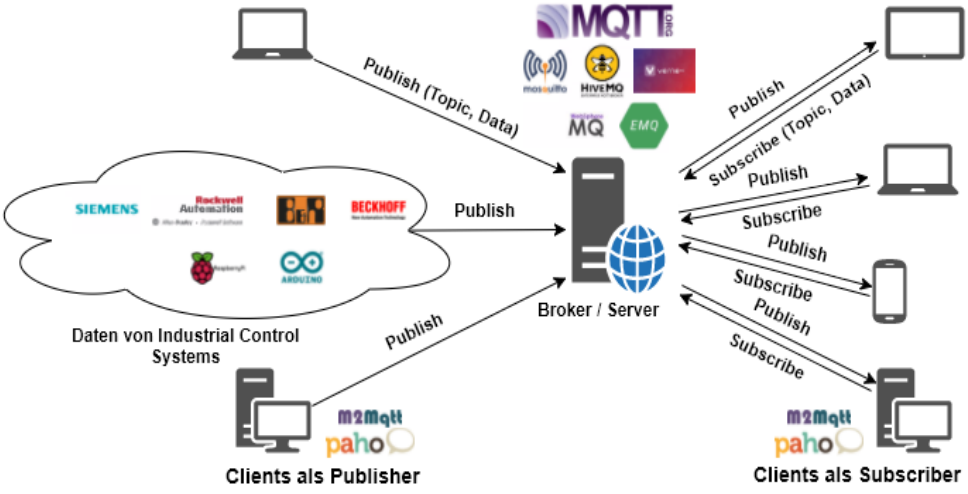


Abbildung 27: Publish/Subscribe Modell

Dies hat den Vorteil, dass ein Client nicht, wie etwa bei Request/Response Architekturen, z.B. REST-Protokoll, ständig Anfragen hinsichtlich Änderungen an den Server senden muss. Für die zuverlässige Datenübertragung bietet MQTT einerseits eine Datenpufferung bei Verbindungsabbruch und andererseits drei verschiedene Quality of Services (QoS) an [21]. Hier ist darauf zu achten, dass bei Verwendung von QoS 1 und QoS 2 sowohl der Broker als auch der Client für die Datenpufferung eine Persistent Session benötigen. In dieser werden beispielsweise vom Broker alle notwendigen Daten, wie etwa alle Client-Subscriptions und Nachrichten, die noch nicht vom Client als angekommen bestätigt wurden, gespeichert. Sobald die Verbindung zum Client wiederhergestellt ist, werden diesem die (verpassten) Nachrichten zugestellt. Darüber hinaus sollte bedacht werden, dass ein höherer QoS-Level zwar mit einer Reduktion des Datenverlusts, jedoch aufgrund eines erhöhten Paketaustauschs mit einer Erhöhung der Latenz in der Übertragung einhergeht [22]. Mehr Details zu den QoS finden sich in Abschnitt 2.2.3.

Aus all den oben genannten Aspekten eignet sich MQTT hervorragend zur Maschine-zu-Maschine (M2M) Kommunikation, für mobile Geräte, bei denen Prozessorleistung und Energiequellen (Batterien, Akkus, etc.) eine Einschränkung darstellen, sowie in Bereichen mit eingeschränkten, latenzreichen oder instabilen Netzwerken und begrenzten Bandbreiten. Speziell für asynchrone one-to-many, bzw. many-to-one und many-to-many Kommunikationsstrukturen ist MQTT hervorragend geeignet [21].

2.2.1 Struktur der MQTT Datenpakete

Die Arbeitsweise des MQTT Protokolls basiert auf dem definierten Austausch einer Serie von Datenpaketen, sogenannter Kontrollpakete (Control Packets). Es wird zwischen 14 solcher Pakete unterschieden, welche in ***Tabelle 3*** aufgelistet sind und in [17] nachgeschlagen werden können. Jedes dieser Pakete kann aus bis zu drei Teilen bestehen, welche immer in der in ***Abbildung 28*** dargestellten Reihenfolge auftreten [17].

	Packetname	Funktion
1	CONNECT	Verbindungsanfrage des Clients an den Server
2	CONNACK	Empfangsbestätigung der Verbindungsanfrage
3	PUBLISH	Veröffentlichen der Nachricht
4	PUBACK	Bestätigung für das Veröffentlichen der Nachricht
5	PUBREC	Nachricht erhalten (mit QoS 2 Teil 1)
6	PUBREL	Veröffentlichungsrelease (mit QoS 2 Nachricht erhalten Teil 2)
7	PUBCOMP	Veröffentlichung komplett (mit QoS 2 Nachricht erhalten Teil 3)

	Packetname	Funktion
8	SUBSCRIBE	Abonnieren eines Topics
9	SUBACK	Bestätigung für das Abonnieren des Topics
10	UNSUBSCRIBE	Abonnement eines Topics aufheben
11	UNSUBACK	Bestätigung für das Aufheben des Abonnement eines Topics
12	PINGREQ	PING Anfrage
13	PINGRESP	PING Antwort
14	DISCONNECT	Disconnect-Benachrichtigung

Tabelle 3: MQTT Control Packets

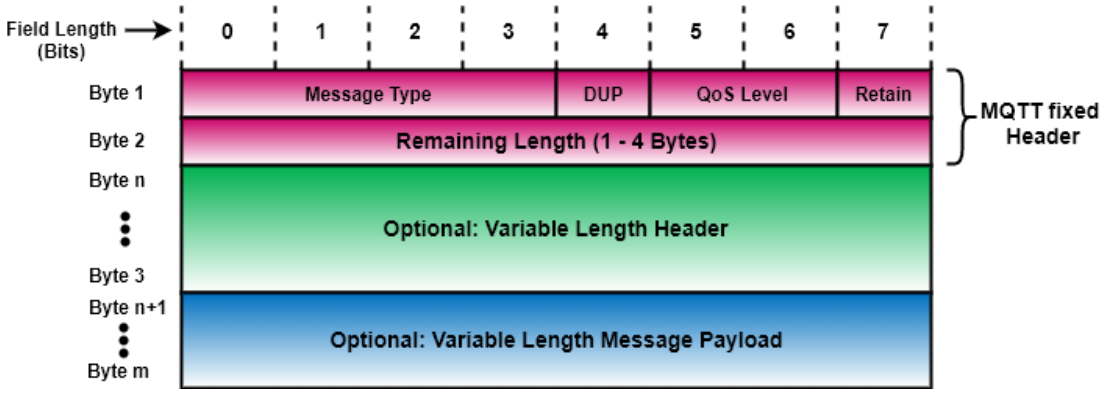


Abbildung 28: Struktur MQTT Message [23]

2.2.2 Semantik für Topics

Die folgende Auflistung ist [17] entnommen und zeigt die semantischen Richtlinien für den Namen eines Topics und dem Topic Filter. Diese unterscheiden sich nur darin, dass Ersterer bei Publish-Prozessen und Letzterer bei Subscribe-Prozessen verwendet wird.

- Alle Topic Namen und Topic Filter müssen zumindest ein Zeichen lang sein
- Topic Name und Filter sind case sensitive
- Es dürfen Leerzeichen verwendet werden
- Ein führendes oder nachfolgendes Slash erzeugt ein neues individuelles Topic
- Ein Topic Name und der Filter dürfen auch nur aus einem Slash bestehen
- Der Name und der Filter dürfen nicht das Nullzeichen (alle Bits 0) enthalten
- Topic Name und Topic Filter sind UTF-8 codiert und dürfen nicht auf mehr als 65535 Bytes codieren

2.2.3 Kommunikationsablauf

Im Folgenden ist der grundlegende Kommunikationsablauf inklusive der dabei betroffenen Control Packets gezeigt. Es sei hier darauf hingewiesen, dass die Nummern in den folgenden Abbildungen die zeitliche Abfolge der Datenpakete widerspiegeln.

- Verbindungsaufbau zwischen Client und Broker – Connect

Nach dem erfolgreichen Aufbau einer Netzwerkverbindung zwischen Client und Server startet, wie aus ***Abbildung 29*** ersichtlich, der Paketaustausch.

Der Payload des CONNECT Packet kann eine Vielzahl von codierten Feldern enthalten, wobei alle bis auf die ClientID optional verwendet werden können. Die zugehörigen Flags im variablen Header bestimmen, welche dieser Felder zur Anwendung kommen sollen. Jeder Client besitzt eine eindeutige ClientID anhand derer der Broker den Client und dessen aktuellen Status identifiziert. Der Broker antwortet mit dem CONNACK Packet und einem Statuscode. Der Client muss zum weiteren Senden von Control Packets nicht auf das CONNACK Packet vom Broker warten.

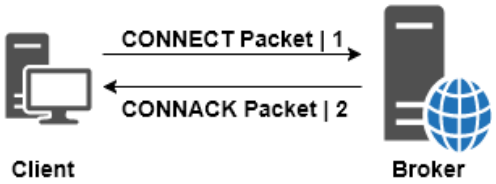


Abbildung 29: Verbindungsaufbau zwischen Client und Broker

Als zusätzliches Feature kann der Client beim Verbindungsaufbau eine Last-Will-Nachricht angeben. Der Broker speichert diese Nachricht inklusive des Bezugs zur Netzwerkverbindung und veröffentlicht diese bei einem Verbindungsabbruch zum Client. Zu möglichen Verbindungsabbrüchen gehören beispielsweise vom Server detektierte Netzwerkfehler, keine Kommunikation des Clients innerhalb der Keep-Alive-Time, der Client schließt die Verbindung ohne das zuvor ein DISCONNECT Packet gesendet wurde oder der Server beendet die Verbindung aufgrund eines Protokollfehlers [17]. Dieses angebotene Feature kann zum Beispiel für die Echtzeitüberwachung von Geräteausfällen in der Infrastruktur eingesetzt werden [20].

- Übertragung einer Nachricht – Publish

Die Übertragung einer Nachricht (Publish) kann entweder vom Client zum Broker oder vom Broker zum Client als Abonnent eines bzw. mehrerer Topics, wie in ***Abbildung 30*** gezeigt, erfolgen. Bei jeder Nachrichtenübermittlung kann eines der drei QoS genutzt werden. Deren Unterschiede bezüglich des Paketaustausches können ***Abbildung 31***, in der als Beispiel der Nachrichtentransport vom Client zum Broker herangezogen wurde, entnommen werden. ***Tabelle 4*** zeigt sowohl die Parameter des

PUBLISH Packets, als auch eine grobe Beschreibung. Für detailliertere Informationen sei auf Abschnitt 3.3 aus [17] verwiesen.

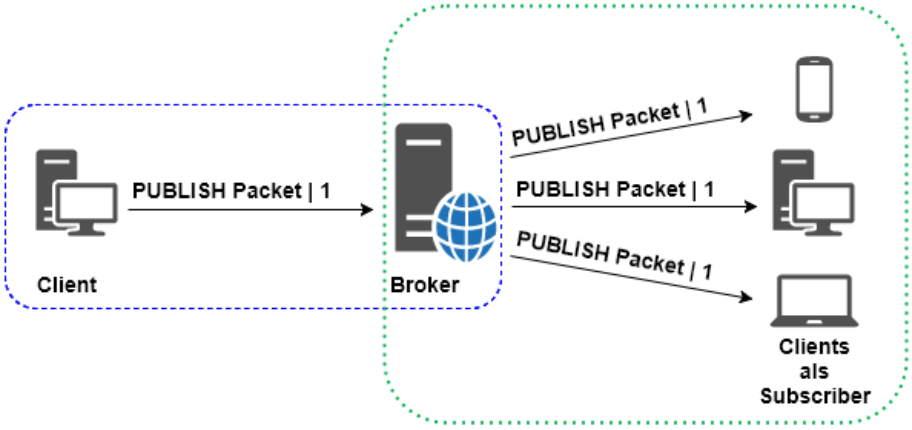


Abbildung 30: Nachrichtentransport von Client zu Broker bzw. Broker zu Client

	Parameter	Beschreibung
1	DUP Flag	Das DUP Flag zeigt an ob es sich um eine erstmalige Sendung oder ein Duplikat handelt. Bei Verwendung von QoS 0 muss das Flag auf 0 gesetzt werden [17]
2	QoS	Auswahl des QoS-Levels
3	RETAIN Flag	Das RETAIN Flag bestimmt ob die Nachricht vom Broker für das spezifische Topic als letzte bekannte gültige Nachricht gespeichert wird. Diese Nachricht wird neuen Clients unmittelbar nach dem Abonnement dieses Topics zugestellt [24].
4	Topic Name	Der Topic Name muss als erstes Feld als UTF-8 codierter String im Variablen Header vorhanden sein und es dürfen keine Wildcard Charakters (# or +) verwendet werden [17]. Die Topic-Hierarchie kann mit / durchlaufen werden. Beispiel: "siedler/example/temperature"
5	Packet Identifier	Der Packet Identifier wird zur eindeutigen Identifizierung von Nachrichten zwischen Client und Broker bzw. vice versa verwendet. Der Packet Identifier ist **nur** bei Verwendung von QoS-Level 1 oder 2 vorhanden. Der Identifier der im Publish Paket verwendet wird, muss ebenfalls bei den zusätzlich auftretenden Paketen für QoS > 0 (z.B. PUBACK, PUBREQ,…) verwendet werden.
6	Payload	Nachrichteninhalt

Tabelle 4: Parameter des PUBLISH Control Packets

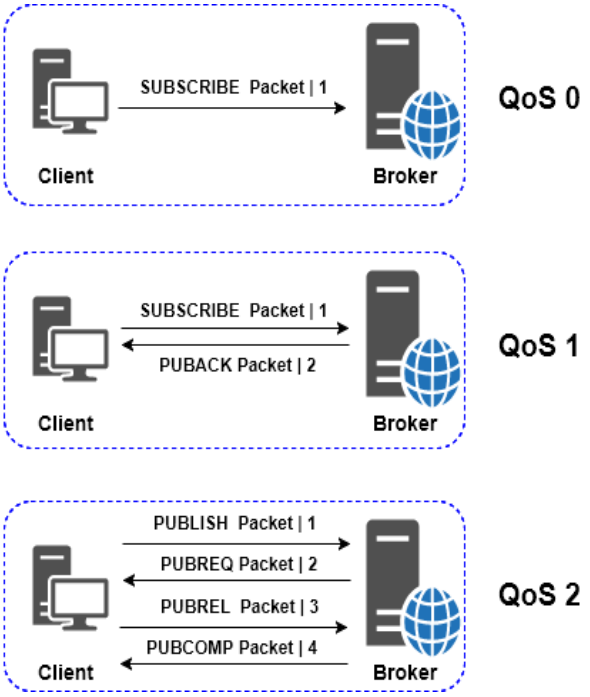


Abbildung 31: Publish von Client zu Broker mit den drei QoS Level

QoS 0 stellt die Nachricht maximal einmal zu und verwirft diese sobald sie gesendet wurde, unabhängig davon, ob die Nachricht das Ziel erreicht hat.

QoS 1 stellt die Nachricht mindestens einmal zu und garantiert, dass die Nachricht das Ziel erreicht. Hier können jedoch Nachrichtenduplikate auftreten.

QoS 2 stellt die Nachricht genau einmal zu und garantiert, dass die Nachricht das Ziel erreicht.

- Abonnement eines Topics – Subscribe

Ein Client erzeugt durch Übermittlung des SUBSCRIBE Packets an den Broker ein oder mehrere Abonnements, sogenannte Subscriptions. Darauf wird vom Broker mit einem SUBACK Packet geantwortet. Jede Subscription enthält ein oder mehrere Topics, von denen der Client künftig Publish Packets vom Broker empfangen möchte. Der Kommunikationsvorgang ist in ***Abbildung 32*** dargestellt. Beim Subscribe-Vorgang wird für jede Subscription der maximale QoS-Level, mit dem der Broker dem Client die Nachricht mittels dem Publish Packet übermitteln darf, angegeben. Für detailliertere Informationen sei auf Abschnitt 3.8 aus [17] verwiesen.

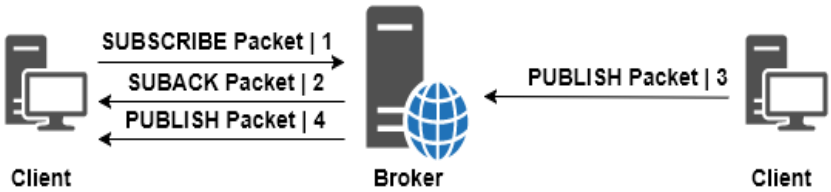


Abbildung 32: Subscription eines oder mehrerer Topics

- Abonnement eines Topics löschen – Unsubscribe

Der Client sendet ein UNSUBSCRIBE Packet an den Broker um eine bestehende Subscription auf ein oder mehrerer Topics, welche im Payload Parameter gelistet sind, zu widerrufen. Der Broker antwortet dem Client mit einem UNSUBACK Packet.

- Verbindung zwischen Client und Broker trennen – Disconnect

Der Client sendet als letztes Paket das DISCONNECT Packet an den Broker und zeigt damit eine saubere Verbindungstrennung an. Der Client muss nach dieser Sendung die Netzwerkverbindung abbauen und darf keine weiteren Pakete an den Broker übermitteln.

2.2.4 Security Mechanismen

Grundsätzlich liegt das Hauptaugenmerk von MQTT als Transportprotokoll auf der zuverlässigen Datenübertragung. Der Anwender selbst ist dafür zuständig, geeignete Security Maßnahmen zu treffen. Gewöhnlich wird TLS verwendet, wobei bei Serverimplementierungen mit Unterstützung von TLS empfohlen wird, TCP Port 8883 zu verwenden [17]. Des Weiteren ist aber darauf hinzuweisen, dass TLS bei Geräten mit stark eingeschränkten Ressourcen, wie etwa Wireless-Sensoren etc., nicht verwendet werden kann [21]. Nachfolgend sind einige Authentifizierungsmöglichkeiten aus [17] zusammengefasst.

- Authentifizierung des Clients beim Broker

  - Es kann eine Authentifizierung mittels Username und Passwort, welche im CONNECT Packet beim Verbindungsaufbau mitgeschickt werden können, implementiert werden. Dabei sollte im Hinterkopf behalten werden, dass Username und Passwort in Klartext übermittelt werden und somit die Gefahr eines Man-in-the-Middle Angriffs besteht.
  - Bei der Verwendung von TLS kann eine Authentifizierung mittels eines SSL-Zertifikates realisiert werden.
  - Als Alternative könnte eine sichere Datenübertragung zwischen Client und Broker mit einem Virtual Private Network (VPN) umgesetzt werden.
  - Es besteht ebenfalls die Möglichkeit, eigene Authentifizierungsmaßnahmen zu implementieren.

- Authentifizierung des Brokers beim Client

Dafür sind in MQTT grundsätzlich keine Mechanismen vorgesehen. Dennoch kann bei Nutzung von TLS wie zuvor eine Authentifizierung per SSL-Zertifikat durchgeführt werden. Alternativ kann auch hier zur abgesicherten Datenübermittlung ein VPN zum Einsatz kommen.

- Datenvertraulichkeit

Für die Datenvertraulichkeit wird eine Verschlüsselung des Payloads auf der Applikationsebene vorgeschlagen, wofür alle zur Verfügung stehenden Verschlüsselungsalgorithmen genutzt werden können [21].

- Absicherung eines MQTT-Systems auf drei Ebenen

Die folgenden Vorschläge zur Absicherung eines MQTT-Systems wurden [25] entnommen und zusammengefasst.

Infrastruktur – Jede Verbindung zu einem MQTT-Broker soll zumindest eine Firewall passieren. Jeder Datenstrom, der nicht von Applikationen nachgeschalteter Systeme benötigt wird, soll von der Firewall geblockt werden. Darunter fallen sowohl UDP-Datenpakete als auch eventuell ping und traceroute ICMP-Pakete. Des Weiteren sollten alle Ports, die nicht benötigt werden, geblockt werden. Die MQTT Standard Ports 1883 (MQTT über TCP) und 8883 (Secure MQTT mit TLS) müssen natürlich ausgenommen werden. Sind die IP-Adressen der MQTT Clients bekannt, sollte man nur die entsprechenden IP-Ranges zum Datenaustausch freigeben. Besteht eine Netzwerkstruktur mit einer vorhandenen Demilitarized Zone (DMZ) sollte der Broker in dieser implementiert werden. Die Definition und weiterführende Informationen zur DMZ finden sich in Abschnitt 3.

Operating System – Software und Bibliotheken müssen regelmäßigen Updates unterzogen werden. Bei Verbindungen via SSH sollte man keine Root-Rechte vergeben und für die Absicherung gegen Brute Force Attacken anstatt der Passwortauthentifizierung SSH-Schlüssel verwenden. Bei Nutzung eines Linux Servers kann auch die Freeware Fail2ban zur Abwehr von Brute Force Attacken eingesetzt werden. Des Weiteren wird auch die Verwendung der Software-Firewall Iptables empfohlen.

MQTT-Broker – Zusätzlich zur Authentifizierung können eine Autorisierung und eine Einschränkung der Nachrichtengröße zur Verbesserung der Sicherheit vorgenommen werden. Bei der Autorisierung wird mittels einer im Broker hinterlegten Einschränkung geregelt, ob ein Client ein Topic veröffentlichen bzw. lesen darf. Diese Einschränkungen müssen zur Laufzeit anpassbar und konfigurierbar sein. Die Message Size darf in MQTT maximal 256 MB betragen und sollte auf die notwendige Nachrichtengröße reduziert werden. Damit kann immerhin das Senden von großen Datenmenge von Malicious MQTT-Clients verhindert werden.

2.3 Gegenüberstellung

OPC UA ist im Gegensatz zu MQTT im Grunde kein Protokoll, sondern eine komplette Kommunikationsumgebung mit der Möglichkeit, ein geeignetes Transportprotokoll je Anwendung mit passendem Encoding auszuwählen. Eine Stärke von OPC UA ist die Möglichkeit, komplexe Datenmodelle in standardisierter Form zur Verfügung zu stellen und eine Datenübertragung unter Einhaltung der Sicherheitsziele (Authentifizierung und Autorisierung, Vertraulichkeit, Integrität und Verfügbarkeit) zu gewährleisten. MQTT beschränkt sich hingegen auf eine zuverlässige Datenübertragung ohne tiefergreifende Security-Mechanismen. MQTT ist ein schlankes Protokoll und eignet sich daher hervorragend zur M2M-Kommunikation und speziell für asynchrone one-to-many, many-to-one und many-to-many Kommunikationsstrukturen. OPC UA unterstützt aufgrund seiner Skalierbarkeit nicht nur die (horizontale) M2M-Kommunikation, sondern auch die vertikale Kommunikation innerhalb eines Unternehmens. Die derzeit stattfindende Integration des Publish/Subscribe-Konzeptes in OPC UA wird mit sehr hoher Wahrscheinlichkeit MQTT als Protokoll unterstützen. In Zukunft wird man sich weniger die Frage stellen, ob man OPC UA oder MQTT verwendet, sondern ob man bei Publish/Subscribe in OPC UA als Transportprotokoll MQTT oder ein anderes Protokoll verwendet.

2.4 Weitere Kommunikationsprotokolle im IIoT

Neben den in dieser Arbeit näher betrachteten Protokollen in OPC-UA und dem Protokoll MQTT existieren noch viele andere Protokolle, die im IIoT eingesetzt werden können. Hierzu zählen beispielsweise XMPP, DPWS, CoAP, DDS, AMQP, REST oder WebSocket [21], [22], [26].

3 Industrial Security

3.1 Einführung

Die fortschreitende Digitalisierung in der Industrie sowie die zunehmende Kommunikation innerhalb und über die Unternehmensgrenzen hinweg, machen die Einführung eines Sicherheitskonzeptes auch im Produktionsbereich unerlässlich. Ein unerlaubter Eingriff in industrielle Automatisierungs- und Kontrollsysteme (engl. Industrial Automation and Control Systems, kurz IACS) kann von ökonomischen Schäden des Unternehmens bis hin zu Personenschäden von Mitarbeitern in der Produktion führen. Ökonomische Schäden entstehen durch unautorisierte Zugriffe, Diebstahl von vertraulichen Daten, Verlust der Integrität und/oder Zuverlässigkeit von Prozess- und Produktionsdaten, Verlust der Verfügbarkeit oder Beschädigungen von Equipment. Ein Sicherheitskonzept für Produktionsanlagen muss nach [27] neben technischen Aspekten und Aspekten der Netzwerkkommunikation, ebenso organisatorische und soziale Aspekte berücksichtigen. Nur durch eine ganzheitliche Berücksichtigung dieser Aspekte kann eine optimale Absicherung von Produktionsanlagen erzielt werden. Zur Absicherung greift man dabei auf einen strukturierten und geschichteten Ansatz, einer sogenannten Defense in Depth Strategie, zurück. Defense in Depth bedeutet, dass mehrere Maßnahmen auf unterschiedlichen Ebenen vorgenommen werden, um einen umfassenden und verlässlichen Schutz gewährleisten zu können. Es ist wichtig in einem vertretbaren wirtschaftlichen Rahmen ausreichende und bedarfsgerechte Sicherheitsmaßnahmen zu realisieren. Das Thema Sicherheit verlangt kontinuierliche Aufmerksamkeit und darf nicht als Projekt mit einem Start- und einem Endpunkt angesehen werden. Vielmehr muss das Sicherheitskonzept laufend auf neue Bedrohungen/Risiken überprüft und wenn notwendig entsprechende Anpassungen vorgenommen werden. Als ersten Überblick zeigt ***Tabelle 5*** die Top 10 Bedrohungen für IACS, die vom BSI 2016 veröffentlicht wurden.

Nr.	Bedrohungsszenario
1	Social Engineering und Phishing
2	Einschleusen von Schadsoftware über Wechseldatenträger und externe Hardware
3	Infektion mit Schadsoftware über Internet und Intranet
4	Einbruch über Fernwartungszugänge
5	Menschliches Fehlverhalten und Sabotage
6	Internet-verbundene Steuerungskomponenten
7	Technisches Fehlverhalten und höhere Gewalt

Nr.	Bedrohungsszenario
8	Kompromittierung von Extranet und Cloud-Komponenten
9	(D)DoS Angriffe
10	Kompromittierung von Smartphones im Produktionsumfeld

Tabelle 5: Top 10 Bedrohungen für IACS aus dem Jahr 2016 [28]

Die Unterscheidung von herkömmlicher IT-Security zur Security für IACS ist auf die unterschiedlichen Anforderungen und der damit abgewandelten Priorisierung der Schutzziele, wie in ***Abbildung 33*** dargestellt, zurückzuführen.

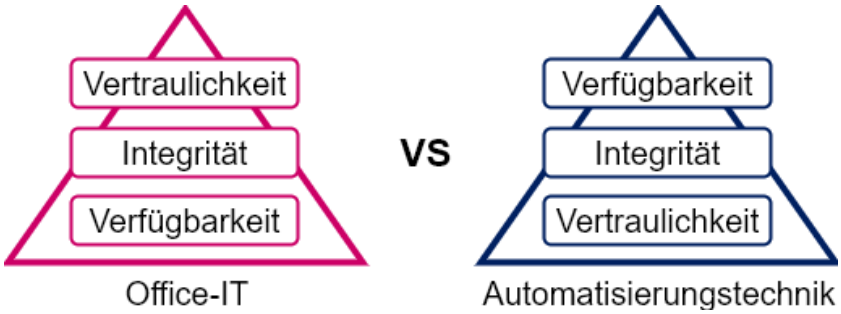


Abbildung 33: Priorisierung der Schutzziele im Office- bzw. Automatisierungstechnikbereich [27]

3.2 Konzepte der IEC 62443

Es existiert keine Standardlösung zur Absicherung von IACS, da jede Anlage unterschiedliche und individuelle Rahmenbedingungen und Anforderungen aufweist. Die IEC 62443 liefert dennoch bewährte Vorgehensweisen bzw. denkbare Maßnahmen zur Implementierung eines effizienten Security-Konzeptes. Obwohl die Norm noch nicht vollständig verabschiedet wurde, kann sie dennoch bereits heute angewendet werden, da die Inhalte der wesentlichsten Dokumente ausreichend stabil sind [29]. Die Struktur und ein Inhaltsüberblick dieser wichtigen Normenreihe, entnommen aus [27] bzw. [29], können im Anhang unter Kapitel 8.3.1 nachgeschlagen werden. Es sei hier angemerkt, dass vor allem IEC 62443-2 und IEC 62443-3 für den Anlagenbetreiber und -integrator von Bedeutung sind, weshalb auch der Fokus dieser Arbeit darauf gerichtet ist. Neben der IEC 62443/ISA 99 adressieren die Richtlinie VDI/VDE 2182 sowie Teile der Norm IEC 62351 für Sicherheit in Energiemanagementsystemen und zugehörigem Datenaustausch, Sicherheitsthemen speziell im Produktionsbereich.

Die wichtigsten Konzepte werden nun überblicksweise vorgestellt und in Abschnitt 3.4 an passender Stelle detaillierter behandelt.

3.2.1 Defense in Depth

Dieser Ansatz besagt, dass mehrere verschiedene, untereinander abgestimmte und koordinierte Maßnahmen zum Schutz von IACS vorgesehen werden müssen. Es müssen dabei sowohl funktionale als auch organisatorische Maßnahmen integriert werden [29]. Mit Defense in Depth werden einem möglichen Angreifer mehrere Verteidigungslinien auf unterschiedlichen Ebenen entgegengestellt.

3.2.2 Netzsegmentierung – Zonen und Conduits

- Zonen

In IEC 62443 wird zwischen physischen und logischen Zonen unterschieden. Eine physische Zone gruppiert Assets (Betrachtungsgegenstände) in Bezugnahme auf deren örtliche Lokation. Im Gegensatz dazu fassen logische Zonen Assets mit gleicher Funktionalität oder anderen Charakteristiken zusammen. Beispielsweise könnte es sinnvoll sein, veraltete SPS mit schwacher oder nicht vorhandener Authentifizierung in einer Zone zusammenzufassen, um diese gezielt mit zusätzlichen Sicherheitsmaßnahmen zu versehen. Natürlich kann es zur Umsetzung des Defense in Depth Konzeptes erforderlich sein, innerhalb einer bereits definierten Zone weitere Zonen festzulegen. Diese Sub-Zonen müssen zumindest alle Sicherheitsanforderungen der Hauptzone erfüllen.

- Conduits (Kommunikationskanäle)

Der Informationsfluss innerhalb einer Zone oder zwischen Zonen und den damit einhergehenden Kommunikationsanforderungen und Security Aspekten wird als spezieller Typ und Teil einer Sicherheitszone angesehen. Mehrere dieser Kommunikationsprozesse in einer Gruppe zusammengefasst, bilden eine Zone, welche speziell als Conduit bezeichnet wird. Anders ausgedrückt beschreibt ein Conduit auf der einen Seite eine Gruppe von Kommunikationsassets zwischen Komponenten innerhalb einer Zone und auf der anderen Seite Kommunikationsassets zwischen verschiedenen Zonen. Die Kommunikationsform ist hierbei keineswegs nur auf die Netzwerkkommunikation eingeschränkt, obwohl in IACS-Umgebungen die Conduits meist mit dem Netzwerk übereinstimmen [30]. In diesem Fall umfasst der Conduit die für die Kommunikation notwendigen Assets, wie Verkabelung, Switches, Router, Netzwerkmanagement Devices und einiges mehr. Trotzdem ist es nach [31] ratsam, nicht nur das Netzwerk zu betrachten, sondern auch Informationsflüsse, welche via USB-Drives etc. stattfinden, zu berücksichtigen. Eine weitere wichtige Thematik, die keinesfalls vernachlässigt werden darf, sind Fernwartungszugänge. Grundsätzlich soll jeder Conduit hinsichtlich der zu verbindenden Zonen und der dafür verwendeten Techniken, Transportprotokolle und notwendigen Sicherheitsfeatures definiert werden [31]. Die spezifischen Kommunikationsverbindungen innerhalb eines

Conduits werden als Channels bezeichnet, wobei ein Channel die Sicherheitseigenschaften des übergeordneten Conduits erbt.

Das Zonen- und Conduit-Konzept erlaubt es, Subsysteme einer kompletten IACS-Landschaft zu isolieren und zu segmentieren. Somit können Geräte innerhalb dieser Zone, vor unbefugten Zugriffen von außen – beispielsweise durch Einsatz einer Firewall – geschützt werden, ohne dabei die Echtzeitfähigkeit, Performanz oder andere Funktionen zu beeinträchtigen [32]. Die Firewall kann nun den Zugriff auf die Zelle kontrollieren, wobei festgelegt werden kann, welche Netzteilnehmer miteinander und ggf. auch mit welchen Protokollen kommunizieren dürfen. Damit können nicht nur unbefugte Zugriffe unterbunden, sondern auch die Netzlast reduziert werden, da nicht jede, sondern nur die gewollte und erforderliche Kommunikation passieren darf [32]. Betrachtet man das Zonenkonzept aus wirtschaftlicher Sicht, ist es leicht nachzuvollziehen, dass es keinen Sinn machen würde, allen Komponenten eines großen und komplexen IACS die gleichen Sicherheitsanforderungen zuzuweisen, da dadurch oftmals unnötige und teure Gegenmaßnahmen getroffen werden müssten. Ein weiterer Grund für die Verwendung von Zonen und Conduits besteht darin, dass bei einem Sicherheitsvorfall innerhalb einer Zone andere Zonen davon unberührt bleiben und so eine Ausweitung des Vorfalls eingedämmt werden kann. Conduits übernehmen dabei die Aufgabe, den Zugriff auf und zwischen Zonen zu regeln, Risiken von DoS-Attacken und die weitere Ausbreitung von z.B. Malware zu mildern sowie die Integrität und Vertraulichkeit des Netzwerkverkehrs, z.B. durch Kommunikation mittels OPC UA unter Verwendung von signierter und verschlüsselter Informationsübertragung, sicherzustellen [31].

3.2.3 Risikobewertung nach VDI/VDE 2182

Mit Hilfe der Risikobewertung werden Bedrohungen und Konsequenzen im Falle eines erfolgreichen Angriffes für einen Betrachtungsgegenstand (z.B. Device, Maschine, Maschinenzelle, Maschinenlinie, etc.) im Vorfeld beschrieben und analysiert, um daraus die notwendigen Schutzmaßnahmen festlegen zu können. Grundsätzlich kann nach [29] die Vorgehensweise in vier Phasen unterteilt werden, welche zyklisch wiederholt werden.

- Planung und Bewertung der Risiken als auch der möglichen Gegenmaßnahmen
- Festlegung der Schutzmaßnahmen
- Effizienzbewertung dieser Schutzmaßnahmen
- Umsetzung der Maßnahmen

Eine detailliertere Beschreibung für die Vorgehensweise der Risikobewertung nach VDI/VDE 2182 findet sich in Abschnitt 3.4.3. Auf zusätzliche Informationen und Dokumente wird dort entsprechend verwiesen. Die Risikobewertung muss unbedingt

in regelmäßigen Abständen oder ereignisgesteuert wiederholt werden. In der Literatur wird dies als Plan-Do-Check-Act, kurz PDCA, bezeichnet.

3.2.4 Anlagenlebenszyklus

Die IEC 62443 bietet Hilfestellungen für Integratoren und Betreiber während eines Anlagenlebenszyklus. Dieser Zyklus gliedert sich im Allgemeinen in Spezifikation (Lastenhefterstellung), Integration und Inbetriebnahme, Betrieb und Wartung sowie letztendlich die De-Kommissionierung.

3.2.5 PDCA-Zyklen im Anlagenlebenszyklus

Der Fokus der PDCA-Zyklen des Betreibers während der Spezifikationsphase des Anlagenlebenszyklus sollte sich nach [29] darauf beziehen, dass im Falle eines Angriffs die Aufrechterhaltung des Betriebes ermöglicht werden kann. Der Integrator hingegen führt PDCA-Zyklen mit dem Ziel durch, in Abstimmung mit dem Betreiber ein Schutzkonzept zur Erfüllung der definierten Schutzziele festzulegen.

PDCA-Zyklen müssen grundsätzlich vom Hersteller, Integrator und Betreiber auf die zugehörigen Betrachtungsgegenstände zur Anwendung kommen. Dabei muss beachtet werden, dass diese Zyklen ineinander verflochten sind und sich deshalb gegenseitig beeinflussen.

3.2.6 Security Levels bis hin zu einem ganzheitlichen Ansatz

Auf Basis einer Risikoanalyse werden IACS in verschiedene Zonen und Conduits aufgeteilt. Durch die Einführung sogenannter Security Levels (SL) kann der Schutz einer Zone oder eines Conduits qualitativ definiert werden. Die Norm unterscheidet dabei zwischen vier SL, womit der Kompetenz, verfügbarer Ressourcen und Motivation für die Überwindung des Schutzes Rechnung getragen wird. Darüber hinaus kann zwischen zu erreichenden SL (SL-T), erreichte SL (SL-A) und erreichbare SL (SL-C) unterschieden werden.

Damit der Schutz von IACS jedoch im laufenden Betrieb bewertet werden kann, sollten neben funktionalen auch organisatorische Maßnahmen für ein ganzheitliches Schutzkonzept Berücksichtigung finden [29]. Dabei können in Analogie zur Bewertung von funktionalen Maßnahmen mit Security-Levels, organisatorische Maßnahmen mit vier Maturity-Levels (ML) bewertet werden. Die Kombination dieser Bewertungen, beispielsweise in Form einer zweidimensionalen Matrix, führt in weiterer Folge zu so genannten Protection-Levels (PL). Die Zuordnung der PL zu den Kombinationen von SL und ML sind bis zum jetzigen Zeitpunkt noch nicht in der Norm festgehalten [29].

3.3 Einführung eines Cyber Security Management Systems

Nach Empfehlung der IEC 62443 soll ein organisationsweites Cyber Security Management System (CSMS) eingeführt werden. Das CSMS übernimmt prinzipiell alle Aufgaben zur Umsetzung/Aufrechterhaltung/Verbesserung der Konzepte der IEC 62443 für IACS. Dieses Managementsystem kann als ein Profil aus der bereits seit einigen Jahren in herkömmlichen IT-Systemen verbreiteten ISO/IEC 2700x angesehen werden. Dies zeigt die enge Verbindung zwischen der IEC 62443/ISA 99 und der ISO/IEC 2700x. In ***Abbildung 34*** sind die Hauptkategorien mit den zugehörigen Elementgruppen und den darin enthaltenen Elementen eines CSMS für den Schutz von IACS dargestellt.

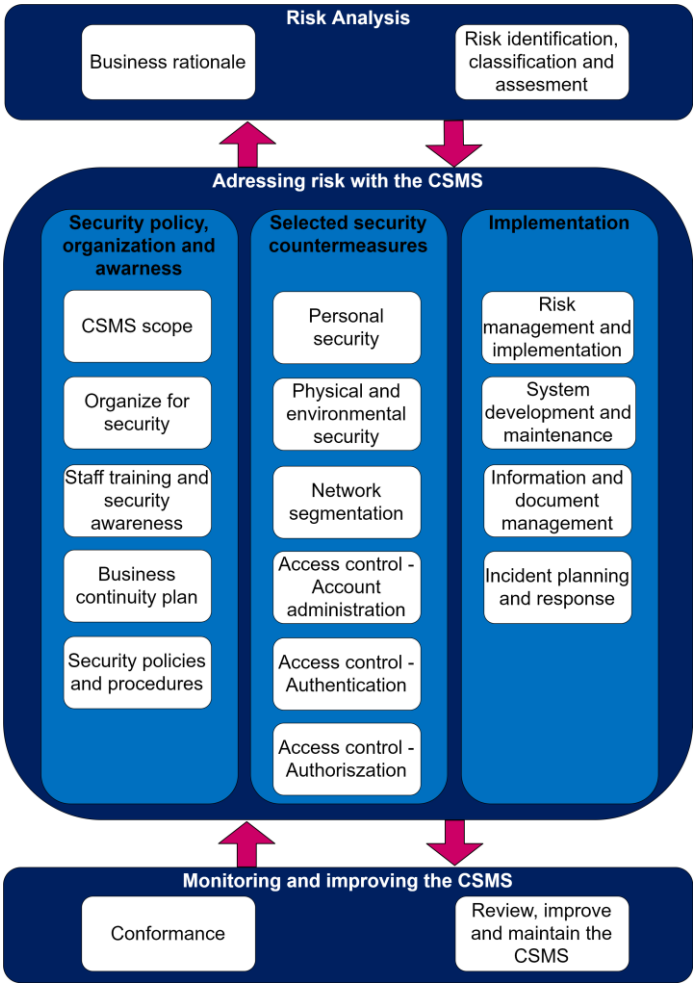


Abbildung 34: Elemente eines Cyber Security Management Systems [33]

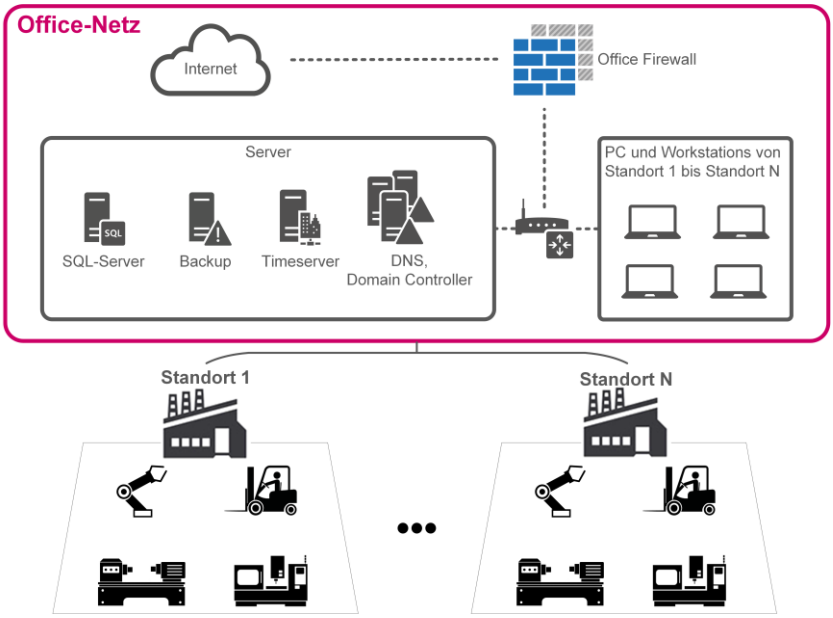
Für eine detaillierte Beschreibung eines CSMS und die Möglichkeiten zur Einführung eines solchen Systems sei auf [33], im Speziellen Annex A und Annex B verwiesen, da deren Betrachtung den Rahmen dieser Arbeit übersteigen würde. Annex A bietet eine Anleitung sowie praxisrelevante Empfehlungen, wie die einzelnen Elemente des CSMS erarbeitet werden können. Annex B hingegen zeigt den Prozess zur Entwicklung eines CSMS.

3.4 Leitfaden zur Umsetzung der IEC 62443 im Unternehmen

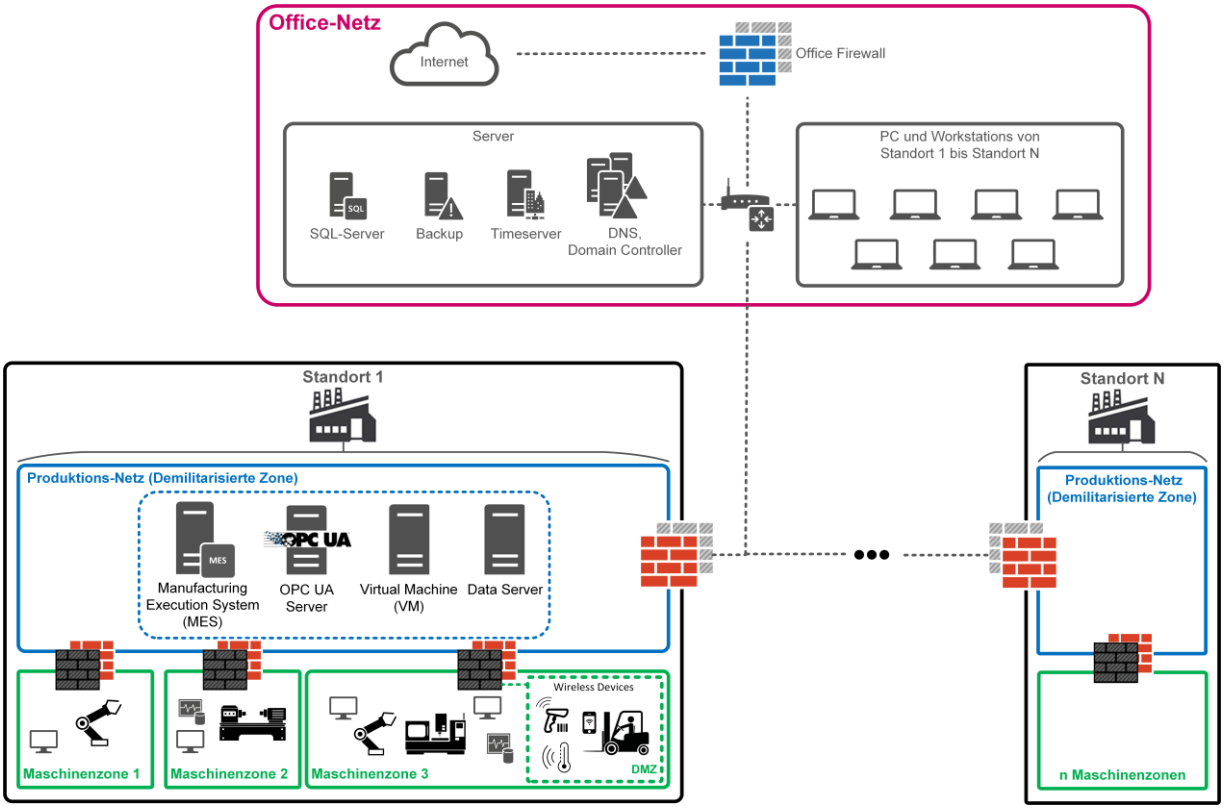
In diesem Abschnitt soll nun anhand der zuvor kurz vorgestellten Konzepte ein Leitfaden zur Umsetzung der IEC 62443 im Unternehmen gezeigt werden. Die dafür notwendigen Analysen werden durchgängig anhand einer Firewall zur Netztrennung gezeigt. Als zweites Beispiel wird eine Engineering Workstation, jedoch nur bis zum Schritt der Risikoanalyse, betrachtet. Unterstützende Abbildungen, Empfehlungen und Anregungen aus Normen etc. zu den jeweiligen Punkten aus Abschnitt 3.4.3 sind dem Anhang zu entnehmen. Die Strukturierung des Anhangs stimmt mit der von Abschnitt 3.4.3 zur besseren Nachvollziehbarkeit überein. Die Empfehlungen und vorgeschlagenen Maßnahmen im Anhang sollen darauf hinweisen welche Überlegungen getätigt werden können und dürfen keineswegs als vollständig angesehen werden.

3.4.1 Ausgangslage und Zielsetzung

In den meisten Fällen müssen Digitalisierungsmaßnahmen in der Produktion und die dafür zwingend notwendige Einführung eines Sicherheitskonzeptes in vorhandene Strukturen eingegliedert werden und können nicht von Grund weg neu aufgebaut und konzipiert werden. Viele Unternehmen besitzen beispielsweise bereits seit Jahren eine Office-IT mit dazugehörigen Sicherheitsmaßnahmen. Als Ausgangspunkt soll deswegen davon ausgegangen werden, dass ein intaktes Office-IT Netzwerk, welches in Folge als Office-Netz bezeichnet wird, im Unternehmen vorhanden ist und sich über einen oder mehrere Unternehmensstandorte erstreckt. Das Office-Netz beinhaltet alle Komponenten, die für das Management eines produzierenden Betriebes von Nöten sind. Dazu zählen beispielsweise das Enterprise Resource Planning System (ERP-System), alle Computer in den unterschiedlichen Abteilungen wie Einkauf, Verkauf, Technik, Qualitätssicherung, Produktion, aber auch Mailserver, Backupserver etc. Die Überlegungen und Maßnahmen zur Absicherung eines Office-Netzes sind nicht Gegenstand dieser Arbeit. Ziel ist es, die Konzepte der IEC 62443 in deren Gesamtheit anhand einer einzelnen Produktionsstätte in strukturierter Weise zu zeigen. Das Hauptaugenmerk liegt dabei ganz klar auf den Rollen des Betreibers und Integrators, da viele Unternehmen die Aufgaben dieser beiden Rollen selbst übernehmen. Die hier beschriebene Vorgehensweise kann natürlich auf analoge Art und Weise auf andere Produktionsniederlassungen angewandt werden. Eines der Ergebnisse wird eine segmentierte Netzwerkstruktur sein, welche an das oben erwähnte Office-Netz angebunden wird. ***Abbildung 35*** zeigt zum besseren Verständnis die Ausgangssituation (oberes Bild) mit nicht vernetzten Produktionsstätten und die erweiterte Netzwerktopologie durch Anbindung einer oder mehrerer vollständig vernetzter Produktionsniederlassungen mit segmentierten Netzwerken (unteres Bild). Beiden Fällen ist jeweils ein gemeinsames Office-Netz übergeordnet.



Ausgangssituation mit nicht vernetzten Produktionsstandorten



Erweiterte Netzwerkstruktur vollständig vernetzter Produktionsstandorte mit segmentierten Netzwerkzonen

Abbildung 35: Netzwerkstruktur der Ausgangssituation und nach Anbindung von segmentierten Netzwerken von Produktionsstandorten nach IEC 62443

3.4.2 Voraussetzungen

Im Vorfeld müssen einige organisatorische Punkte geklärt sein, um eine sinnvolle Umsetzung der weiteren Vorgehensweise gewährleisten zu können. Da diese Aufgaben zu Beginn vom Management delegiert werden müssen, werden hier folgende Punkte vorausgesetzt:

- Definition einer Geschäftsgrundlage für die Sicherheit von IACS im Unternehmen. Nach [33] besteht diese Geschäftsgrundlage aus vier Schlüsselkomponenten: Geschäftliche Konsequenzen inklusive Priorisierung, Bedrohungen mit zugehöriger Priorisierung, Abschätzung des jährlichen Geschäftseinflusses und Kosten für Gegenmaßnahmen.
- Definition und Erstellung eines formalen Security Programms inklusive der Beschreibung der strategischen Ziele, Prozesse und Zeitplanung.
- Zusammenstellung eines funktionsübergreifenden Teams, mit genauen Rollenverteilungen und Zuständigkeiten. Der richtige Mix an Personen mit unterschiedlichsten Fähigkeiten ist für den Erfolg eines Sicherheitsprogramms von entscheidender Bedeutung. Das Kernteam soll nach [33] aus Personen, die IACS implementieren und betreuen, Operatoren der Maschinen, Verantwortliche für personelle Sicherheit und Umwelt, IT-Experten für Netzwerkdesign, Netzwerkbetrieb und Administration von PCs und Servern sowie Personen mit Kenntnissen in IT-Security bestehen.
- Erstellung von Policies und Procedures speziell für IACS. Policies und entsprechende Procedures sind beispielsweise für Themen wie Risikomanagement, Zugriffsmanagement, Backup- und Recovery-Prozesse, Aufrechterhaltung des Betriebs im Falle eines erfolgreichen Cyber-Angriffs, Anlagensicherheit, Nutzung von tragbaren und WLAN-fähigen Geräten, Remotezugriffe, Trainings und Schulungen, Audits, Updates der Policies, Systemwartungen, Rundgänge in der Produktion für Kunden/Besucher und diversen Netzwerkthemen zu verfassen. Letzteres kann Netzwerkdesign, Firewall-Konfiguration, Authentifizierung und Autorisierung von Nutzern, Nutzung von WLAN, Patch-Management und Systemhärtungsmaßnahmen beinhalten.
- **WICHTIG:** Erstellung eines sogenannten Business Continuity Plans zur Aufrechterhaltung/Wiederherstellung des laufenden Betriebes nach einem signifikanten Cyber-Security Vorfall. Ein wichtiger Aspekt in diesem Zusammenhang sind diverse Backup- und Wiederherstellungsprozesse. Neben dem Backup von Informationen und Daten, sollen beispielsweise auch Maschinenprogramme, Rezepturen und technische Zeichnungen regelmäßig gesichert und versioniert werden.

3.4.3 Schritte zur Umsetzung der IEC 62443

In diesem Abschnitt wird eine denkbare Vorgehensweise im Sinne der IEC 62443 gezeigt. Diese Vorgehensweise kann sequentiell abgearbeitet werden und ist in ***Abbildung 36*** zusammenfassend dargestellt. Wichtig ist, dass alle genannten Analysen und Bewertungen in regelmäßigen Abständen bzw. ereignisgesteuert zu wiederholen sind, um bei Änderungen an der Anlage, Entstehung neuer Bedrohungen und Risiken etc., ein lückenloses Sicherheitskonzept gewährleisten zu können. Werden Risikoanalyse und Ermittlung von Schutzzielen vernachlässigt oder gar nicht durchgeführt, läuft man Gefahr, dass ungeeignete, zu teure oder wirkungslose Maßnahmen getroffen werden, oder das manche Schwachstellen gar nicht erkannt werden [32].

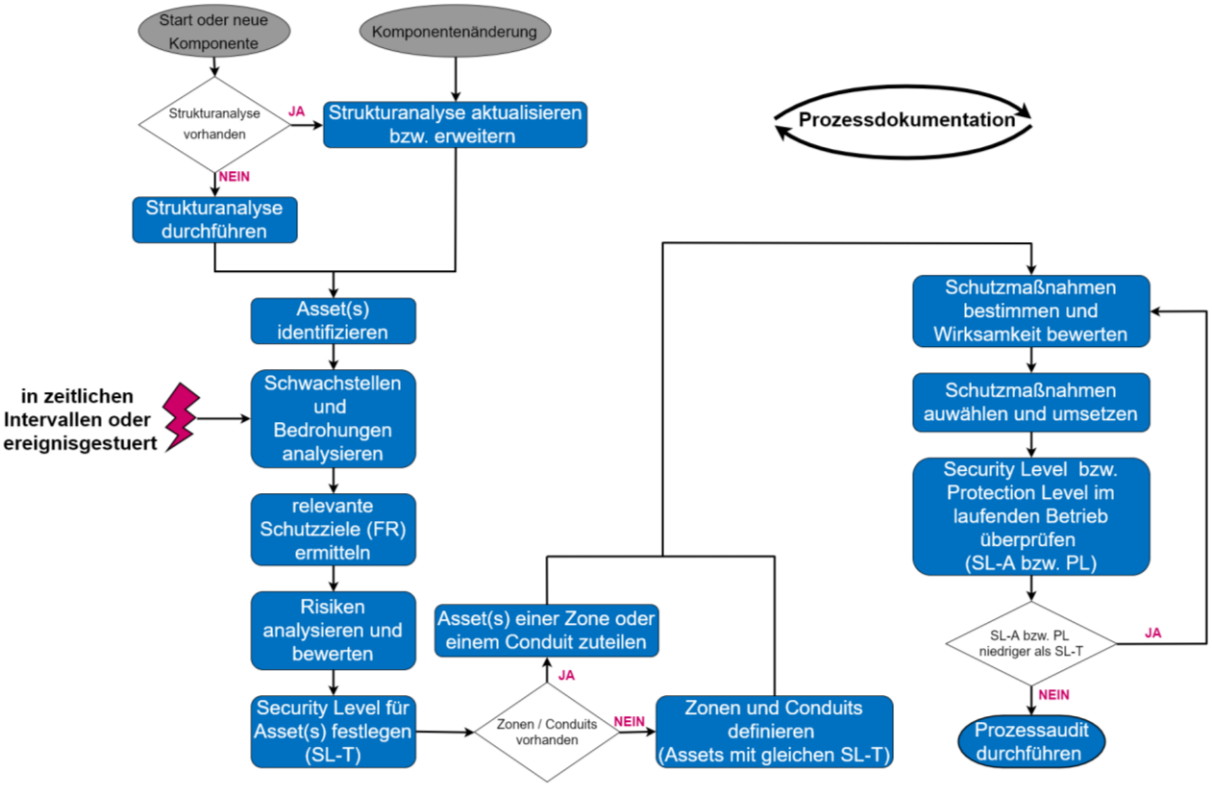


Abbildung 36: Prinzipielle Vorgehensweise im Sinne der IEC 62443

Strukturanalyse durchführen

- Eingangsinformation: keine oder bestehende Analyse
- Aktionen: Die Strukturanalyse hat das Ziel alle Komponenten, die für den reibungslosen Betrieb und Informationsfluss in der vernetzten Produktion erforderlich sind, zu erfassen und zu dokumentieren.
  Existiert bereits eine Strukturanalyse müssen bei Komponentenänderungen oder bei Aufnahme neuer Komponenten alle Dokumentationen und Diagramme entsprechend aktualisiert oder erweitert werden.
- Ergebnis: Liste und Diagramme aller Komponenten und Informationsflüsse

Für die Dokumentation müssen deren funktionale Anforderungen möglichst exakt spezifiziert und deren Einsatzumgebung beschrieben werden. Die funktionalen Anforderungen sollen den genauen Umfang an Funktionen, die Schnittstellen und Datenflüsse, sowie die Anwendung erfassen. Nach [33] soll dazu im ersten Schritt eine grafische Darstellung der Netzwerkinfrastruktur erstellt werden. Zur Analyse des Netzwerkverkehrs, der Protokolle und Services, welche in bestehenden Systemen auftreten, können kostenlose Tools, wie z.B. Wireshark, eingesetzt werden.

Im Anhang 8.3.2 unter dem Punkt – Strukturanalyse findet sich eine Liste mit einigen Beispielen, welche als Komponenten in der Strukturanalyse erfasst und entsprechend dokumentiert werden sollten. Ebenso werden dort ein beispielhaftes Formular zur Aufnahme der Daten der identifizierten IACS und ein exemplarisches Netzwerkdiagramm angeführt. Das Netzwerkdiagramm soll dabei zumindest die Komponenten, die mit Hilfe des Formulars aufgenommen wurden, beinhalten [33].

Asset(s) identifizieren

- Eingangsinformation: Ergebnis der Strukturanalyse
- Aktionen: Die Komponenten aus der Strukturanalyse, welche geschützt werden müssen – in Folge als Assets bezeichnet – werden definiert, gegliedert und wenn möglich in logischen Gruppen zusammengefasst. Als Assets gelten insbesondere Automatisierungskomponenten, Rechner, Server, etc., aber auch Software, Applikationen, geistiges Eigentum, Know-How und gewisses Personal mit speziellen Fähigkeiten.
- Ergebnis: Liste der identifizierten Assets und eventuelle Gruppierungen

Nach IEC 62443 können Assets wie folgt klassifiziert werden:

Physical Assets: jegliche physikalische Komponenten oder Gruppen davon, z.B. SPS, Sensoren, Netzwerkkomponenten, Fördersysteme und jegliche Komponenten, die von den Steuer- und Regeleinrichtungen beeinflusst werden, aber auch Wände, Räume, Gebäude etc.

Logical Assets: sind alle Informationen, die für den Betrieb des Unternehmens von Bedeutung sind und bei Verlust oder Manipulation direkte geschäftliche Auswirkungen haben. Beispiele dafür sind geistiges Eigentum, Testresultate, spezifisches Prozesswissen, Logik der Prozessautomatisierung usw. und können in Form von gespeicherten schriftlichen und digitalen Dokumenten, SPS-Programmen, Source Code von Software, technischen Zeichnungen, Datenbanken und Konfigurationsdateien auftreten oder in den Köpfen von Personen verankert sein.

Human Assets: sind Personen mit speziellen Fähigkeiten und speziellem Wissen für Prozesse, Equipment oder Notfallsituationen. Meist besitzen sie dafür erforderliche

Ausbildungen und Zertifizierungen. Ein Cyberangriff mit Personenschäden wäre ein Angriff auf Human Assets.

Damit eines der oben genannten Beispiele oder eine Komponente aus der Strukturanalyse letztendlich zu einem der beschriebenen Assets wird, muss das betrachtete Objekt natürlich einen Wert für das Unternehmen aufweisen. Dieser Wert sollte entweder in quantitativer (monetärer Verlust) oder qualitativer (abstrakte Bewertung der Auswirkung) Form ausgedrückt werden. Eine weitere Möglichkeit besteht darin, den Wert eines Assets durch direkte (z.B. Kosten für den Ersatz des Assets) oder indirekte (z.B. Betriebsstillstände durch Kontrollverlust oder Beschädigung/Zerstörung von Assets) Verluste zu klassifizieren. Dazu zählen auch all jene Komponenten, die bei Beeinträchtigungen oder Ausfall eine Gefahr für Personensicherheit, Umwelt, öffentliches Vertrauen oder für die Einhaltung rechtlicher Auflagen sind [30].

Schwachstellen und Bedrohungen analysieren

- Eingangsinformation: Liste der identifizierten Assets
- Aktionen: Es werden alle relevanten Bedrohungen der (gruppierten) Assets identifiziert bzw. aktualisiert. Es sind diverse Szenarien festzuhalten, die für verschiedene Bedrohungen relevant sind. Es ist sinnvoll, bekannte oder mögliche Schwachstellen der Assets bereits hier festzuhalten, da daraus zusätzliche Bedrohungsszenarien erkannt werden können.
- Ergebnis: Schwachstellen und Bedrohungen und Bedrohungsszenarien der Assets

Schwachstellen (engl. Vulnerabilities) sind im Grunde inhärente Schwächen von Systemen, Komponenten oder in der Organisation und treten beispielsweise durch Designfehler oder veraltete Komponenten und Systeme auf [30]. Ebenso müssen fehlende Policies und Procedures, fehlende oder mangelhafte Zutrittskontrollen, fehlende oder mangelhafte Absperrungen etc. als Schwäche angesehen werden. Damit diese Schwachstellen nicht missbräuchlich ausgenutzt werden können, müssen diese letztendlich durch Gegenmaßnahmen eliminiert werden. Dabei muss berücksichtigt werden, dass Schwachstellen mit der Zeit durch den technischen Fortschritt etc. neu entstehen oder bestehende einer Veränderung unterliegen können.

Bedrohungen (engl. Threats) beschreiben Aktionen, die gegen ein System gerichtet werden können. Bedrohungen, die in weiterer Folge aktiv auf ein System wirken, werden als Angriff oder Attacke bezeichnet. Dabei können ebenfalls versehentliche (z.B. Personen ohne Kenntnisse der Policies und Procedures) und nicht-validierte Änderungen (z.B. Updates, Korrekturen, Änderungen an Betriebssystemen, Applikationen, Konfigurationen, Equipment) als Bedrohungsform auftreten. Besonderes Augenmerk ist auf Bedrohungen zu legen, welche von innerhalb des

Systems initiiert werden, da der Angreifer hier meist bereits im Vorfeld einige Gegenmaßnahmen umgangen hat (z.B. Zutrittskontrollen etc.). Des Weiteren berücksichtigt man von wem eine Bedrohung ausgehen kann. Dafür existieren die Begriffe des Insiders (z.B. vertrauenswürdige Personen, wie Angestellte, Vertragspartner, Lieferanten) und des Outsiders (z.B. ehemalige Mitarbeiter, Hacker). Darüber hinaus können strukturelle Faktoren (z.B. fehlerhafte/veraltete Komponenten) oder auch ökologische Faktoren (z.B. Stürme, Erdbeben, Überflutungen) als Verursacher fungieren.

Die bereits identifizierten (gruppierten) Assets sollen nun gemeinsam mit deren Schwachstellen, möglichen Bedrohungen bzw. Szenarien und möglichen Verursachern, wie in ***Abbildung 37***, tabellarisch erfasst werden. Natürlich kann auch eine andere Erfassungsmethodik, wie z.B. in Form Matrizen oder Baumstrukturen zum Einsatz kommen. Im Anhang 8.3.2 unter dem Punkt – Bedrohungen analysieren sind einige Beispiele für Bedrohungen, welche für die Analyse in Betracht gezogen werden können, angeführt.

Asset	Schwachstellen	Bedrohungen	Auslöser
Engineering Workstation	- veraltetes Betriebssystem - fehlende Policy und fehlendes Training für die Nutzung tragbarer Speichergeräte	- Nutzung von Exploits zur Kompromittierung - Infektion mit Malware	- Hacker - Personal
Firewall (Netztrennung zw. Office-IT und Produktionsstandort)	- fehlende Physical Security - schwache Passwort Policy - Nichteinhaltung von Policies (Standartpasswort wurde nicht geändert) - fehlende Regelsätze	- physische Zerstörung - Konfigurationsmodifikation - Überwindung der Firewall	- ehem. Mitarbeiter - Personal - Hacker - Hacker

Abbildung 37: Schwachstellen- und Bedrohungsanalyse von Assets

Relevante Schutzziele ermitteln

- Eingangsinformation: Schwachstellen, Bedrohungen und Bedrohungsszenarien der Assets
- Aktionen: In der weiteren Vorgehensweise betrachtet man die Tabelle aus der Schwachstellen- und Bedrohungsanalyse und wählt für jedes Asset die dafür notwendigen Systemanforderungen (SR) als Schutzziele.
- Ergebnis: Ergebnis der zuvor gewählten Erfassungsmethodik um relevante Schutzziele erweitert

Die aus der herkömmlichen IT verwendeten Schutzziele Verfügbarkeit, Integrität und Vertraulichkeit reichen nicht aus, um die Sicherheitsanforderungen für IACS angemessen zu behandeln. Deshalb verwendet man speziell für IACS definierte Anforderungen – die sieben Foundational Requirements (FR). Da diese FR ausschließlich Automatisierungslösungen adressieren, sind nach [29], zwei weitere Anforderungen hinzuzufügen, welche den physischen Zugang und die Organisation

betrachten – diese werden hier als Additional Foundational Requirements (AFR) bezeichnet. Damit ergeben sich insgesamt neun Anforderungen, welche in ***Abbildung 38*** aufgelistet sind.

7 Foundational Requirements	
FR 1 - Identificationand Access Control (IAC) – Identifizierung und Authentifizierung	alle Nutzer (Personen, Systeme und Prozesse) müssen ausreichend authentifiziert werden um auf Ressourcen zugreifen zu können
FR 2 - Use Control (UC) – Nutzungskontrolle	Nachvollziehbar und jeder Zugriff auf Systeme und darauf befindliche Daten ist aufzuzeichnen
FR 3 - System Integrity (SI) – Systemintegrität	Daten, Programme und Hardware können nicht modifiziert werden oder eine Modifikation kann zumindest erkannt werden
FR 4 - Data Confidentiality (DC) – Vertraulichkeit der Daten	Sicherstellung, dass Informationen nur Befugten zugänglich ist. Unautorisierte haben keinen Zugriff auf übertragene Nachrichten oder gespeicherte Informationen
FR 5 - Restricted Data Flow (RDF) – Eingeschränkter Datenfluss	Datenkommunikation auf notwendige Netzwerkbereiche einschränken
FR 6 - Timely Response to Events (TRE) – Rechtzeitige Reaktion auf Ereignisse	Bei erkannten Systemanomalien muss die Möglichkeit einer raschen Einleitung von Gegenmaßnahmen erfolgen können
FR 7 - Resource Availability (RA) – Ressourcenverfügbarkeit	Gewährleistung der Verfügbarkeit relevanter Netzwerk- und Systemressourcen
AFR 1 - Physical Access (PA) - Physischer Zugang	
AFR 2 - Organisation (O) – Organisation der Security	

Abbildung 38: Foundational Requirements

Jede dieser Anforderungen (AFR1 und AFR2 ausgenommen) beinhaltet mehrere der insgesamt 51 Systemanforderungen (SR), von denen jede wiederum eine oder mehrere Anforderungserweiterungen (RE) zur Sicherheitssteigerung umfassen kann. Die komplette Liste der FR, SR und RE sind Anhang 8.3.2 unter dem Punkt – Relevante Schutzziele ermitteln zu entnehmen. In dieser Liste sind zusätzlich die erreichbaren Security-Levels angegeben. Das Ergebnis dieses Abschnitts kann, wie in ***Abbildung 39*** exemplarisch gezeigt, aussehen.

Asset	Schwachstellen	Bedrohungen	Auslöser	Schutzziele (FR/SR)
Engineering Workstation	- veraltetes Betriebssystem	- Nutzung von Exploits zur Kompromittierung	- Hacker	- FR 1.1/1.3/1.7 & FR 2.3 & FR 3.2 & FR 7.3
	- fehlende Policy und fehlendes Training für die Nutzung tragbarer Speichergeräte	- Infektion mit Malware	- Personal	- AFR 2
Firewall (Netztrennung zw. Office-IT und Produktionsstandort)	- fehlende Physical Security	- physische Zerstörung	- ehem. Mitarbeiter	- AFR 1
	- schwache Passwort Policy - Nichteinhaltung von Policies (Standartpasswort wurde nicht geändert)	- Konfigurationsmodifikation	- Personal - Hacker	- FR 1.1/1.3/1.7/1.12 & FR 6.1/6.2
	- fehlende Regelsätze	- Überwindung der Firewall	- Hacker	- AFR 2

Abbildung 39: Festlegung der Schutzziele

Risiken analysieren und bewerten

- Eingangsinformation: Tabelle mit relevanten Schutzzielen
- Aktionen: Aus dem Gefahrenpotenzial und der Einfachheit eines Angriffes werden die Wahrscheinlichkeiten für das Eintreten der im vorigen Schritt identifizierten Bedrohungen bewertet. Zusätzlich müssen zugehörige denkbare Konsequenzen abgeschätzt werden.
- Ergebnis: Risikobewertungen für relevante Bedrohungen der Assets

Das Risiko ist eine Funktion von Bedrohungen, Schwachstellen und möglichen Konsequenzen, welche negative Auswirkungen auf ein Unternehmen haben. Bedrohungen und Konsequenzen werden dabei meist mit abstrakten Eintrittswahrscheinlichkeiten belegt. Das bedeutet, dass eine Bedrohung, welche unter Ausnutzung von Schwachstellen im System mit einer gewissen Wahrscheinlichkeit eintritt, gewisse Konsequenzen mit sich bringt. Als typische Risiken gelten:

- Safety, z.B. Verletzung oder Tod von Personal
- Prozess-Safety, z.B. Schäden an Equipment oder Fertigungsstillstände
- Informationssicherheit
- Umweltschäden, z.B. Verletzung von gesetzlichen Auflagen

Damit sich die resultierenden Konsequenzen und die damit verbundenen Schadenshöhen abschätzen lassen, kann man leicht nachvollziehen, dass die Risikoanalyse für jedes Asset getrennt durchgeführt werden sollte. In nachstehender Abbildung ist für ein spezielles Asset und der dafür relevanten Bedrohungen ein Beispiel angeführt. Natürlich muss eine für das Unternehmen angepasste Einteilung der Schadenshöhe und Eintrittswahrscheinlichkeit definiert und spezifiziert werden. Die gestrichelte Linie zeigt den Risikotoleranz-Level des Unternehmens für den betrachteten Fall an. Damit erfolgt eine Trennung in zwei Bereiche, wobei der grüne Bereich ein akzeptables Risiko angibt und somit für den vorliegenden Fall (zurzeit) keine weiteren Aktionen erfordert. Sollte sich das Ergebnis der Analyse jedoch im roten Bereich befinden, welcher ein inakzeptables Risiko beschreibt, müssen in weiterer Folge Aktionen zur Risikoreduzierung geplant werden.

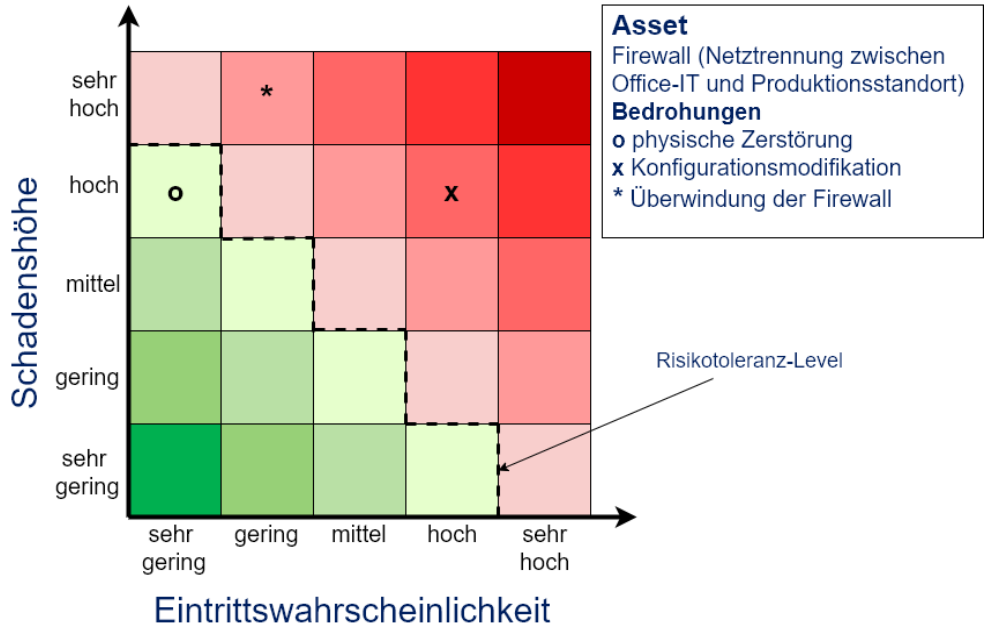


Abbildung 40: Risikomatrix für ein Asset und der dafür relevanten Bedrohungen

Security-Level für Asset(s) festlegen

- Eingangsinformation: Risikobewertungen für relevante Bedrohungen der Assets
- Aktionen: Jeder Kategorie der zuvor definierten Schutzziele des betrachteten Assets werden, anhand der Erkenntnisse aus den Risikobewertungen, zu erreichende Security-Levels (SL-T) zugeordnet.
- Ergebnis: Assets mit festgelegten Security Levels

Die IEC 62443 unterscheidet zwischen vier Security Levels. Die Charakterisierung berücksichtigt dabei, welche Mittel und Ressourcen einem Angreifer zur Verfügung stehen und welche Fähigkeiten und Motivation der Angreifer besitzt. Die Definition der SL ist in ***Abbildung 41*** dargestellt.

SL 1	Schutz gegen zufälligen/unabsichtlichen Missbrauch
SL 2	Schutz gegen gewollten Missbrauch unter Verwendung einfacher Mittel, mit niedrigem Aufwand, allgemeinen Fähigkeiten und niedriger Motivation
SL 3	Schutz gegen gewollten Missbrauch unter Verwendung technisch ausgefeilter Mittel, mit moderatem Aufwand, IACS spezifischen Fähigkeiten und moderater Motivation
SL 4	Schutz gegen gewollten Missbrauch unter Verwendung technisch ausgefeilter Mittel, mit erheblichem Aufwand, IACS spezifischen Fähigkeiten und hoher Motivation

Abbildung 41: Definition der Security-Level (SL) [29]

Die IEC 62443 unterscheidet drei Typen von Security Levels:

1. Target SL (SL-T): zu erzielendes SL für (gruppierte) Assets oder präziser für Zonen und Conduits
2. Achieved SL (SL-A): erreichtes SL für (gruppierte) Assets oder präziser für Zonen und Conduits. Dieses Maß wird zum Vergleich zwischen des aufgrund bereits umgesetzter Risikoreduzierungsmaßnahmen erreichten SL einer Zone oder Conduits und dem zugehörigen zu erzielenden SL eingesetzt. Ein SL-A kann sich aufgrund neu entstehender Schwächen, Angriffsmethoden etc. mit fortschreitender Zeit vermindern. Das SL-A einer Zone oder Conduits sollte zu jedem Zeitpunkt zumindest das SL-T erfüllen [30].
3. Capability SL (SL-C): erreichbares SL wird auf Komponenten und Produkte angewandt und stellt den Level dar, der erreicht werden kann, wenn alle Funktionen und Eigenschaften eines Produktes ausgenutzt und korrekt in einer Automatisierungslösung eingesetzt werden [29].

Eine mögliche Zuordnung von SL-T zum vorliegenden Beispiel kann ***Abbildung 42*** entnommen werden. Aus Platzgründen wurde in dieser Darstellung auf die Spalte Auslöser verzichtet.

Asset	Schwachstellen	Bedrohungen	Schutzziele (FR/SR)	SL-T
Engineering Workstation	- veraltetes Betriebssystem	- Nutzung von Exploits zur Kompromittierung	- FR 1.1/1.3/1.7 & FR 2.3 & FR 3.2 & FR 7.3	Risikoanalyse ausständig
	- fehlende Policy und fehlendes Training für die Nutzung tragbarer Speichergeräte	- Infektion mit Malware	- AFR 2	
Firewall (Netztrennung zw. Office-IT und Produktionsstandort)	- fehlende Physical Security	- physische Zerstörung	- AFR 1	- SL 1
	- schwache Passwort Policy - Nichteinhaltung von Policies (Standartpasswort wurde nicht geändert)	- Konfigurationsmodifikation	- FR 1.1/1.3/1.7/1.12 & FR 6.1/6.2	- SL 4 & SL 2
	- fehlende Regelsätze	- Überwindung der Firewall	- AFR 2	- SL 3

Abbildung 42: Zuordnung der zu erreichenden Security Levels (SL-T)

Eine weitere Hilfestellung für den Ablauf zur Festlegung der SL-T ist in [33] - Annex A unter dem Security Level Lifecycle Model zu finden.

Ein Security-Level stellt im Grunde die Eigenschaft einer Zone oder eines Conduits dar und weniger die Eigenschaft eines einzelnen Gerätes oder Systems. Deswegen werden im nächsten Abschnitt Assets mit gleichen Sicherheitsanforderungen in Zonen oder Conduits zusammengefasst.

Zonen und Conduits definieren oder neues Asset aufnehmen

- Eingangsinformation: Assets mit festgelegten SL-T
- Aktionen: Die zuvor identifizierten Assets, die denselben Sicherheitsanforderungen (SL-T) aufgrund der Faktoren Kritikalität und Schadenskonsequenzen unterliegen, werden in sogenannten Zonen oder Conduits zusammengefasst. Jede Zone oder Conduit weist aufgrund der im Schritt zuvor gewählten Zuteilung der Security Levels natürlich ebenfalls für jede Schutzzielkategorie ein SL-T auf. Es kann in manchen Fällen sinnvoll sein, dass man ein Asset, welches höhere SL-T als die Zone aufweist, dennoch dieser Zone zuteilt. Assets mit niedrigeren SL-T als die einer Zone dürfen dieser jedoch nicht zugeordnet werden.
  Falls bereits ein segmentiertes Netzwerk vorliegt können neue Assets entweder einer bestehenden Zone zugewiesen oder dafür zusätzliche Zonen definiert werden.
- Ergebnis: Segmentierte Netzwerkstruktur

Abbildung 43 soll das Konzept der Zonen und Conduits nochmals veranschaulichen. Das gesamte Unternehmensnetzwerk ist dabei in sieben eigenständige Zonen unterteilt. Die Zone Plant A Control Zone beinhaltet darüber hinaus eine Sub-Zone. Bei einer derartigen Zoneneinteilung können für jede einzelne Zone (ausgenommen der Sub-Zonen) komplett unabhängige Policies definiert werden. Wie bereits oben erwähnt, müssen Sub-Zonen dagegen zumindest die Anforderungen der Hauptzone erfüllen. Nach [30] müssen bei der Festlegung der Zonen und Conduits sowohl das Asset-Modell als auch die Referenzarchitektur berücksichtigt werden. Beispiele dafür und weitere Modelle als Hilfestellung können Anhang 8.3.2 unter dem Punkt – Zonen und Conduits entnommen werden.

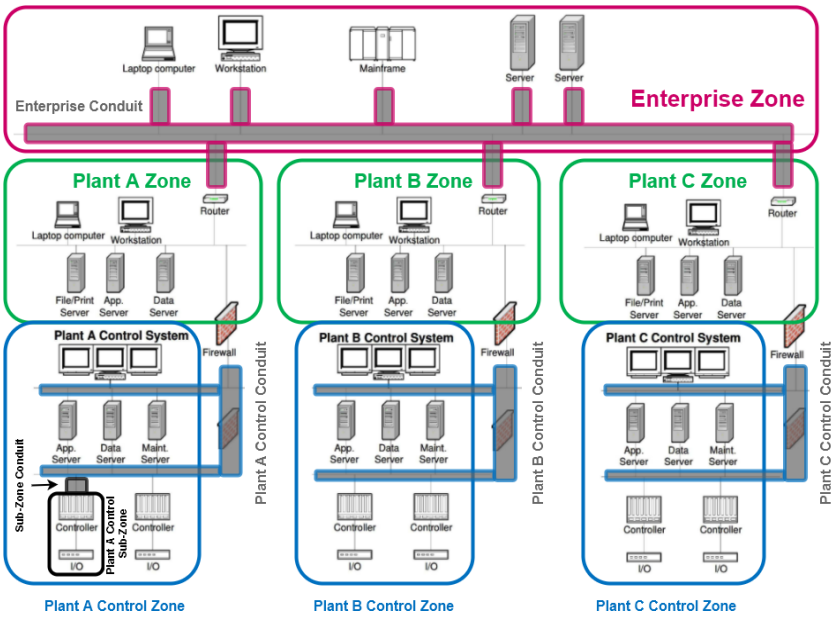


Abbildung 43 : Beispiel für ein segmentiertes Netzwerk und Conduits [30]

Demilitarisierte Zone (DMZ)

Die Einführung einer weiteren Zone – einer sogenannten DMZ – zwischen dem Office- und dem Maschinen-Netz trägt zur weiteren Risikominimierung bei. Diese Zone ist durch Firewalls von den beiden anderen Netzen getrennt und dient als eine Art Puffer zwischen der Office-Zone und den Maschinen-Zonen. Das hat zur Folge, dass jegliche Kommunikation zuerst die DMZ passieren muss und dadurch keine direkten Zugriffe vom Office-Netz auf das Maschinen-Netz (und umgekehrt) erfolgen. Die Zone mit der höheren Sicherheit, sprich die Maschinen-Zone, pusht ihre Daten und stellt diese den autorisierten Usern im Office-Netz über die DMZ zur Verfügung. Im Gegensatz dazu sollen Zugriffe von Usern aus dem Office-Netz auf die DMZ ausschließlich über Pull-Zugriffe erfolgen. Üblicherweise ist eine DMZ so ausgelegt, dass davon kein Zugriff oder Verbindungsaufbau in das Maschinen-Netz möglich ist. Selbst wenn ein Rechner in der DMZ von einem Angreifer übernommen wird, bleibt das Maschinen-Netz weiterhin geschützt [32]. Idealerweise sollten sich die verwendeten Ports und Services, die zwischen Office-Netz und DMZ genutzt werden, zu denen die zwischen DMZ und Maschinennetz verwendet werden, unterscheiden [33]. Ungenutzte Ports sollen im Sinne von Härtungsmaßnahmen deaktiviert werden. Zusätzlich müssen Zugriffe auf die DMZ auf autorisierte User, Applikationen und Services eingeschränkt werden [34]. Wenn die Möglichkeit einer Modellierung des Datenverkehrs zur und von der DMZ besteht, kann zusätzlich zu Quellen/Zielen/Servicefiltern eine Anomalie-Detektion zum Einsatz kommen. Historische Datenserver, Server für MES, Server für das Patch-Management, Access Points für Wireless LAN, Remote-Zugriffssysteme oder Anti-Virusserver sollen in der DMZ angesiedelt werden [33], [34]. Natürlich gilt auch für die in der DMZ befindlichen Server, dass diese regelmäßig mit Updates versorgt werden müssen. Neben dem beschriebenen Einsatzgebiet einer DMZ kann, wenn nötig, eine Maschinen-Zone eine eigene DMZ aufweisen, in welcher WLAN-fähige Geräte für diese Zone, wie WLAN-fähige Sensoren etc., untergebracht werden.

Die Definition der Zonen und Conduits erfolgt mittels eines iterativen Prozesses, indem die Zonengrenzen mehrmals umdefiniert werden und neue Zonen eingeführt werden. Nachdem der iterative Prozess für die Definition der Zonen und Conduits abgeschlossen ist, müssen die Charakteristiken und Sicherheitsanforderungen sowohl für Zonen als auch für Conduits dokumentiert werden. Die notwendigen Attribute, welche nach [30] festzuhalten sind, finden sich im Anhang 8.3.2 unter dem Punkt – Zonen und Conduits. Für eine detaillierte Beschreibung der einzelnen Attribute sei auf [30] Abschnitt 6.5.3 bzw. 6.5.5 verwiesen. Ebenso kann dem Anhang ein Beispieldokument entnommen werden, welches zeigt, wie eine Dokumentation für Zonen und Conduits aussehen könnte. Zur Dokumentation des Netzwerkverkehrs, der Protokolle und Services, welche in bestehenden Systemen auftreten, sind einfache Datenflussdiagramme, wie in ***Abbildung 44*** dargestellt, anzufertigen.

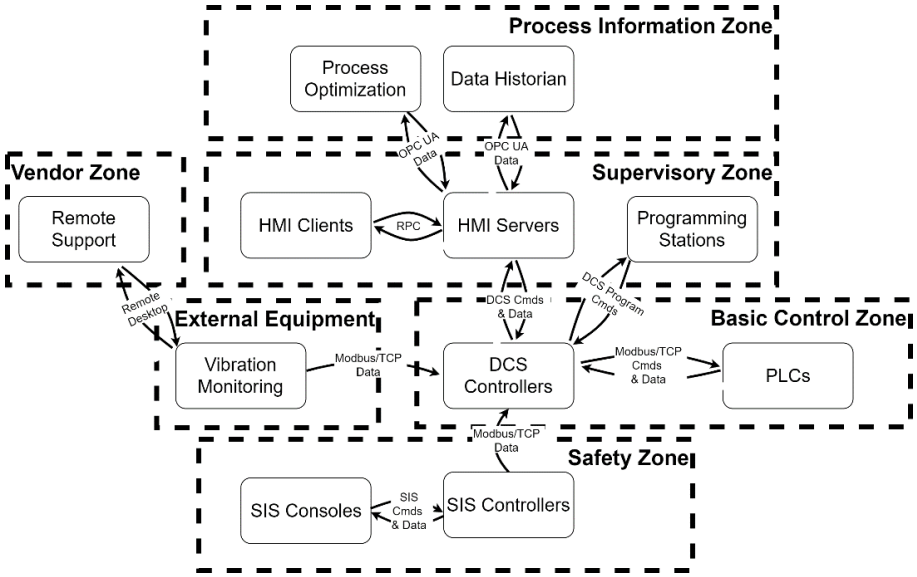


Abbildung 44: Beispiel für ein simples Datenflussdiagramm [31]

Eine mögliche Zonenzuordnung zum vorliegenden Beispiel kann ***Abbildung 45*** entnommen werden. Aus Platzgründen wurde in dieser Darstellung wiederum auf die Spalte Auslöser verzichtet.

Zone	Asset	Schwachstellen	Bedrohungen	Schutzziele (FR/SR)	SL-T
	Engineering Workstation	- veraltetes Betriebssystem	- Nutzung von Exploits zur Kompromittierung	- FR 1.1/1.3/1.7 & FR 2.3 & FR 3.2 & FR 7.3	Risikoanalyse ausständig
		- fehlende Policy und fehlendes Training für die Nutzung tragbarer Speichergeräte	- Infektion mit Malware	- AFR 2	
Conduit 1	**Firewall** (Netztrennung zw. Office-IT und Produktionsstandort)	- fehlende Physical Security	- physische Zerstörung	- AFR 1	- SL 1
		- schwache Passwort Policy - Nichteinhaltung von Policies (Standartpasswort wurde nicht geändert)	- Konfigurationsmodifikation	- FR 1.1/1.3/1.7/1.12 & FR 6.1/6.2	- SL 4 & SL 2
		- fehlende Regelsätze	- Überwindung der Firewall	- AFR 2	- SL 3

Abbildung 45: Zuordnung der Assets zu Zonen

Mögliche Schutzmaßnahmen bestimmen und deren Wirksamkeit bewerten

- Eingangsinformation: Zonen mit deren SL-T und den darin beinhalteten Assets inklusive deren Schwächen und Bedrohungen
- Aktionen: Das Ergebnis der Risikobewertung hat Felder identifiziert, in denen das bestehende Risiko nicht akzeptabel ist. Zur Erfüllung der SL-T, der in Zonen und Conduits befindlichen Assets, sind somit Maßnahmen zur Risikoreduzierung, so genannten Schutzmaßnahmen, vorzusehen und in Bezug auf ihre Wirksamkeit und Wirtschaftlichkeit zu bewerten.
- Ergebnis: mögliche Schutzmaßnahmen, deren Wirksamkeit und Kosten

Es muss darauf geachtet werden, dass sich die Gegenmaßnahmen stets an die möglichen Bedrohungen richten, z.B. können Viren nicht effektiv mit Firewalls bekämpft und umgekehrt Zugriffsverletzungen nicht mit Virenscannern verhindert werden. Oft wird einer Bedrohungen mit mehreren verschiedenen Schutzmaßnahmen entgegengewirkt. Des Weiteren liefert die Liste der FR, SR und RE im Anhang 8.3.2 unter dem Punkt – Relevante Schutzziele ermitteln Hinweise, welche und wie viele der Aspekte aus den SR und RE zur Erfüllung der SL-T umgesetzt werden müssen.

Im Anhang 8.3.2 unter dem Punkt – Mögliche Schutzmaßnahmen bestimmen und deren Wirksamkeit bewerten befindet sich eine Sammlung möglicher Gegenmaßnahmen.

In [35] befinden sich exemplarisch formulierte Umsetzungsvorschläge für 16 der 51 SR.

Schutzmaßnahmen auswählen und umsetzen

- Eingangsinformation: Liste möglicher Schutzmaßnahmen, deren Wirksamkeit und Kosten
- Aktionen: Aus der Vielzahl an möglichen Schutzmaßnahmen erfolgt nun die Auswahl, um in einem wirtschaftlich vertretbaren Rahmen diejenigen Schutzmaßnahmen zu implementieren, die die erforderliche Risikoreduzierung erreichen. Dabei müssen natürlich auch die Ziele des Unternehmens und die Konformität mit der Unternehmenspolitik, insbesondere in Bezug auf die IT-Security, berücksichtigt werden [29].
- Ergebnis: Umsetzung der ausgewählten Schutzmaßnahmen

Eine mögliche Auswahl von Schutzmaßnahmen zum vorliegenden Beispiel könnte, wie in ***Abbildung 46*** dargestellt, vorgenommen werden. Aus Platzgründen wurde in dieser Darstellung auf die Spalten Auslöser und Schwachstellen verzichtet.

Zone	Asset	Bedrohungen	Schutzziele (FR/SR)	SL-T	Gegenmaßnahmen
	Engineering Workstation	- Nutzung von Exploits zur Kompromittierung - Infektion mit Malware	- FR 1.1/1.3/1.7 & FR 2.3 & FR 3.2 & FR 7.3 - AFR 2	Risikoanalyse ausständig	
Conduit 1	**Firewall** (Netztrennung zw. Office-IT und Produktionsstandort)	- physische Zerstörung - Konfigurationsmodifikation - Überwindung der Firewall	- AFR 1 - FR 1.1/1.3/1.7/1.12 & FR 6.1/6.2 - AFR 2	- SL 1 - SL 4 & SL 2 - SL 3	- High Availabilty Cluster, Raum absperren - Überarbeitung der Policies und Schulungen für das Personal - Überarbeitung und wenn nötig Kompetenzausbau

Abbildung 46: gewählte Schutzmaßnahmen

Security-Level oder Protection-Level im laufenden Betrieb überprüfen

- Eingangsinformation: ausgewählte Schutzmaßnahmen
- Aktionen: Überprüfung der SL-A im laufenden Betrieb um eine nachhaltige Umsetzung der Lösung gewährleisten zu können.
- Ergebnis: Protection Levels

Für den laufenden Betrieb müssen zusätzlich zu den funktionalen Maßnahmen auch organisatorische Aspekte Berücksichtigung finden – z.B. Dokumentation und Schulung von Personal, für die korrekte Handhabung von Sicherheitsmaßnahmen, die korrekte Arbeitsweise im Sinne der Schutzmaßnahmen etc. Dabei können in Analogie zur Bewertung von funktionalen Maßnahmen mit Security-Levels, organisatorische Maßnahmen mit vier Maturity-Levels (ML), sprich Reifegraden, bewertet werden. Die Kombination dieser Bewertungen, beispielsweise in Form einer zweidimensionalen Matrix, führt in weiterer Folge zu sogenannten Protection-Levels (PL) und zu einer ganzheitlichen Betrachtung. Der PL ersetzt also in Folge die SL-A. Ein PL kann grundsätzlich nicht höher sein als der SL-A, aber aufgrund eines niedrigen ML (fehlende oder schwache organisatorische Maßnahmen) durchaus kleiner als dieser. Ist das PL niedriger als das SL-T einer Zone oder Conduits, müssen möglicherweise im Falle eines niedrigen ML weitere organisatorische Maßnahmen vorgenommen werden. In ***Abbildung 47*** sind die Definitionen für SL, ML und dem daraus resultierenden PL gezeigt. Es muss hier jedoch angemerkt werden, dass sich eine Arbeitsgruppe bereits mit einer Zuordnung der Protection Levels zu der Kombination Reifegrad zu Security-Level beschäftigt, dies aber noch nicht in einem Normdokument festgehalten wird [29]. Das bedeutet, dass die Zuordnung in Form der Matrix in ***Abbildung 47*** lediglich eine Möglichkeit darstellt und nicht einer Empfehlung der IEC 62443 entstammt. Die Ergebnisse der Überprüfung im laufenden Betrieb zum vorliegenden Beispiel können, wie in ***Abbildung 48*** dargestellt, dokumentiert werden. Aus Platzgründen wurde in dieser Darstellung wiederrum auf die Spalte Auslöser und Schwachstellen verzichtet. Zusätzlich wäre es durchaus sinnvoll, zusätzliche Spalten für SL-A und ML anzulegen, um nachvollziehen zu können, ob bei Nicht-Erreichung der SL-T funktionale oder organisatorische Maßnahmen getroffen werden müssen. Das in der Darstellung rot hervorgehobene PL zeigt an, dass hier weitere Maßnahmen vorzunehmen sind um zumindest das SL-T zu erreichen.

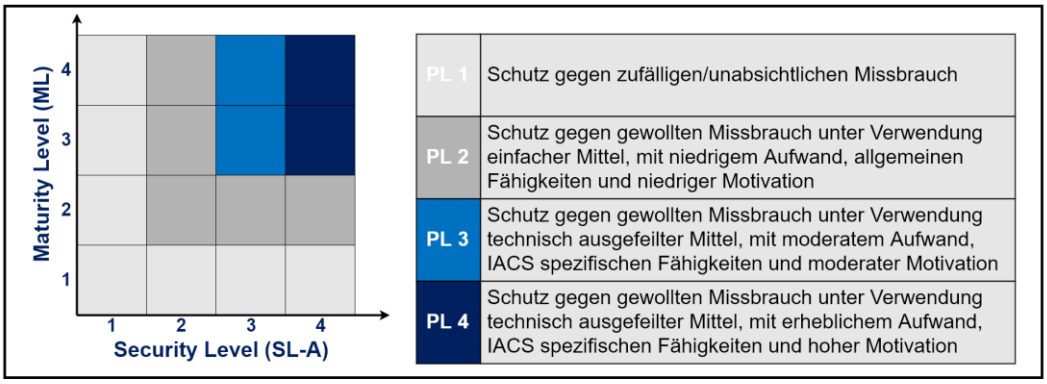
Bewertung der funktionalen Maßnahmen (SL-A)

SL 1	potentieller Schutz gegen zufälligen/unabsichtlichen Missbrauch
SL 2	potentieller Schutz gegen gewollten Missbrauch unter Verwendung einfacher Mittel, mit niedrigem Aufwand, allgemeinen Fähigkeiten und niedriger Motivation
SL 3	potentieller Schutz gegen gewollten Missbrauch unter Verwendung technisch ausgefeilter Mittel, mit moderatem Aufwand, IACS spezifischen Fähigkeiten und moderater Motivation
SL 4	potentieller Schutz gegen gewollten Missbrauch unter Verwendung technisch ausgefeilter Mittel, mit erheblichem Aufwand, IACS spezifischen Fähigkeiten und hoher Motivation

Bewertung der organisatorischen Maßnahmen und Prozesse (ML)

ML 1	Initial: Prozesse sind ad-hoc, schwach kontrolliert und nicht vorraussehbar
ML 2	Managed: Prozesse werden reaktiv gelebt
ML 3	Defined: Prozesse sind beschrieben und werden proaktiv umgesetzt
ML 4	Optimized: Prozesse werden bewertet, kontrolliert und kontinuierlich verbessert

Protection Levels für ganzheitliche Betrachtung



PL 1	Schutz gegen zufälligen/unabsichtlichen Missbrauch
PL 2	Schutz gegen gewollten Missbrauch unter Verwendung einfacher Mittel, mit niedrigem Aufwand, allgemeinen Fähigkeiten und niedriger Motivation
PL 3	Schutz gegen gewollten Missbrauch unter Verwendung technisch ausgefeilter Mittel, mit moderatem Aufwand, IACS spezifischen Fähigkeiten und moderater Motivation
PL 4	Schutz gegen gewollten Missbrauch unter Verwendung technisch ausgefeilter Mittel, mit erheblichem Aufwand, IACS spezifischen Fähigkeiten und hoher Motivation

Abbildung 47: Bewertung funktionaler und organisatorischer Maßnahmen mit Protection Level (PL) [29]

Zone	Asset	Bedrohungen	Schutzziele (FR/SR)	SL-T	Gegenmaßnahmen	SL-A bzw. PL
	Engineering Workstation	- Nutzung von Exploits zur Kompromittierung	- FR 1.1/1.3/1.7 & FR 2.3 & FR 3.2 & FR 7.3	Risikoanalyse ausständig		
		- Infektion mit Malware	- AFR 2			
Conduit 1	**Firewall** (Netztrennung zw. Office-IT und Produktionsstandort)	- physische Zerstörung	- AFR 1	- SL 1	- High Availabilty Cluster, Raum absperren	- PL 2
		- Konfigurationsmodifikation	- FR 1.1/1.3/1.7/1.12 & FR 6.1/6.2	- SL 4 & SL 2	- Überarbeitung der Policies und Schulungen für das Personal	- PL 3 & PL 2
		- Überwindung der Firewall	- AFR 2	- SL 3	- Überarbeitung und wenn nötig Kompetenzausbau	- PL 3

Abbildung 48: Überprüfung im laufenden Betrieb mit Protection Levels

Der detaillierte Ablauf für den gesamten Security Level Lifecycle ist in Figure A.14, A.17 und A.18 in [33] – Annex A zu finden. Abschnitt 5.6.1 – Maturity Phases aus [30] beschreibt hingegen den Lifecyle für Maturity Phasen.

- **Prozessaudit durchführen**
  - Eingangsinformation: Schutzmaßnahmen umgesetzt und Betriebskonzept erstellt
  - Aktionen: In den Audits wird überprüft, ob alle definierten Maßnahmen zur Erreichung der festgelegten Schutzziele sowie zum Schutz gegen die relevanten Bedrohungen ausreichend sind und erfolgreich implementiert wurden. Ebenso werden alle erstellten Dokumente geprüft. Die Audits müssen von Personen, die nicht Teil des Prozesses waren, durchgeführt werden.
  - Ergebnisse: Auditbericht

Zur Unterstützung des beschriebenen Prozesses stehen Software-Werkzeuge zur Verfügung, welche die Erfassung und Bewertung der Assets unterstützen. Das Bundesamt für Sicherheit in der Informationstechnik stellt dazu das Werkzeug LARS[7] inklusive Handbuch kostenlos zur Verfügung. Die Homeland Security bietet das Werkzeug CSET[8] nach Registrierung ebenfalls kostenlos an. Für erste Einschätzungen kann unter www.vds-quick-check.de, sowohl für den Office-Bereich als auch den Produktionsbereich ein Check zur Einstufung des Unternehmens hinsichtlich IT-Security durchgeführt werden.

3.5 Design eines segmentierten Netzwerks

3.5.1 Ausgangslage und Zielsetzung

Die Aufgabe besteht nun darin, dass für einen Produktionsstandort, welcher aus mehreren einzelnen Bearbeitungsmaschinen besteht, ein segmentiertes Netzwerk gemäß IEC 62443 zu designen. Für die digitale Erfassung und Speicherung von Maschinen- und Produktionsdaten kommt ein MES zum Einsatz. Das MES unterstützt OPC UA, welches zur Kommunikation zwischen den, im Office-Netz angesiedelten, Clients zur Datenvisualisierung und dem MES verwendet wird. Zur sicheren Kommunikation soll OPC UA signiert und verschlüsselt betrieben werden. Die Bearbeitungsmaschinen sind mit Ethernet-fähigen Siemens Steuerungen ausgerüstet, besitzen aber noch keinen integrierten OPC UA Server. Aus diesem Grund erfolgt die Datenübertragung zwischen den Maschinen und dem MES direkt über das S7-Protokoll von Siemens. Eine M2M-Kommunikation wird hier nicht betrachtet. Es wird angenommen, dass am betrachteten Produktionsstandort eine Vielzahl an verschiedenen Maschinen betrieben wird. Aus diesem Grund finden SPS-Programmänderungen nicht in-house statt, sondern werden von den jeweiligen

[7] https://www.bsi.bund.de/DE/Themen/Industrie_KRITIS/ICS/Tools/LarsICS/LarsICS_node.html – 12.08.2018
[8] https://ics-cert.us-cert.gov/Downloading-and-Installing-CSET – 12.08.2018

Maschinenherstellern über Fernzugriffe vorgenommen, weshalb diese Thematik im Designprozess berücksichtigt werden muss.

3.5.2 Lösungsvorschlag

Zu Beginn wird das komplette Unternehmensnetzwerk basierend auf operationalen Funktionen, Prozesslevels und Sicherheitsanforderungen grob in gewünschte Zonen eingeteilt (Office-Zone, DMZ und Maschinen-Zone). Die Firewall zur Netztrennung wird als High Availability Cluster ausgeführt. Damit kann bei Ausfall einer Firewall oder für Patch-Prozesse in der Zwischenzeit auf eine zweite Firewall umgeschwenkt werden. Aufgrund der häufig auftretenden Fernwartungen ist jede Anlage einer eigenen Zone zugewiesen, welche durch Firewalls von anderen Zonen getrennt ist. Die Firewalls weisen einen zusätzlichen Port auf, mit dem sich, falls es in Zukunft notwendig werden sollte, eine zur Maschinenzone gehörige DMZ realisieren lässt. Das segmentierte Netzwerk mit Zonen und Conduits für das betrachtete Szenario ist in ***Abbildung 49*** dargestellt.

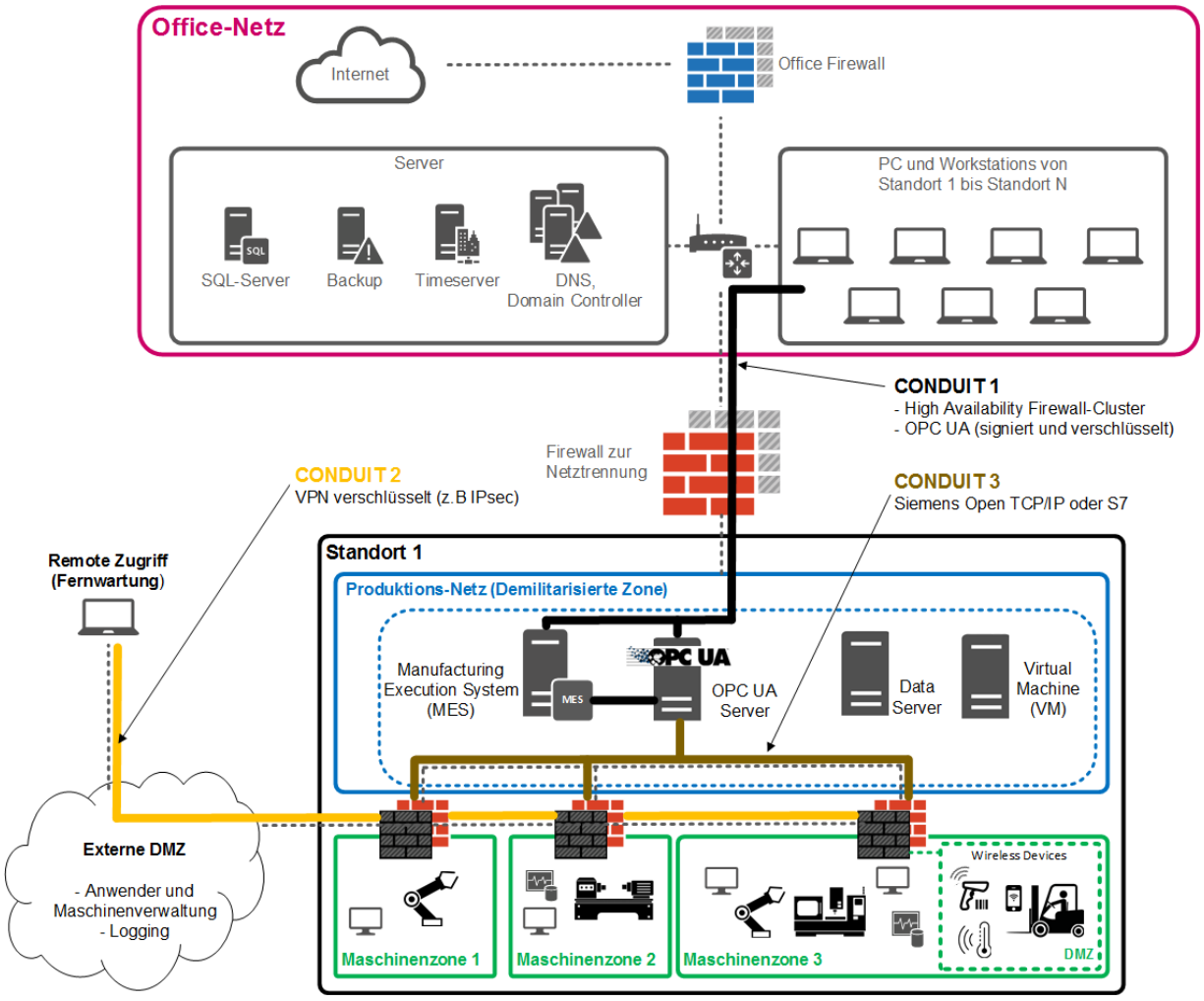


Abbildung 49: Segmentiertes Netzwerk gemäß der Anforderungen von IEC 62443

4 IIoT – Integration von Legacy Systems

4.1 Einleitung

Maschinen mit festverdrahteter Steuerung, die über keine speicherprogrammierbare Steuerung (SPS) verfügen, werden im Rahmen dieser Arbeit als Legacy Systems bezeichnet. Die Logik für die korrekte Arbeitsweise derartiger Maschinen ist ausschließlich durch die Verdrahtung von elektrischen bzw. elektronischen Bauelementen, wie z.B. Schütze, Relais, Taster, Schalter und Leiterplatten, gegeben. Legacy Systems sind meist Anlagen welche aus Zeiten stammen in denen speicherprogrammierbare Steuerungen noch nicht zur Verfügung standen. Trotzdem sind solche Maschinen auch heutzutage noch häufig in produzierenden Unternehmen anzutreffen. Die Anbindung von Legacy Systems und die Erfassung von Maschinendaten sind einerseits notwendig um eine objektive Beurteilung zu ermöglichen, ob es sinnvoll ist, ein Legacy System durch eine neue Anlage zu ersetzen. Diese Beurteilung kann im Rahmen einer Produktivitätsanalyse basierend auf den Maschinendaten durchgeführt werden. Andererseits müssen für eine digitale Produktion bewährte Legacy Systems in den Digitalisierungsprozess integriert werden. Nur damit lässt sich eine automatisierte und lückenlose Betriebs- bzw. Maschinedatenerfassung für eine transparente Produktion und die Identifikation von Optimierungspotenzialen realisieren.

Im Folgenden werden zwei Möglichkeiten gezeigt wie Legacy Systems in Umgebungen des Industrial Internet of Things (IIoT) integriert werden können. Der Fokus liegt dabei auf der Erfassung der Signale zur Beschreibung der Basiszustände der Maschine. Die dafür zu erfassenden Signale sind:

- Start/Stopp
- Hand-/Automatikbetrieb
- Störung aktiv
- Stückzähler

Darüber hinaus wird direkt am Edge eine Leistung in Form von produzierten Stück pro Stunde errechnet. Eine Berechnung am Edge bedeutet, dass die Berechnung direkt am maschinennahen Datenerfassungssystem erfolgt und anschließend das Ergebnis an ein übergeordnetes System übertragen wird.

Meistens verfügen die festverdrahteten Steuerungen über Hilfskontakte mit Arbeitsspannungen zwischen 0 und 24 V. Diese Kontakte werden für die Anbindung und Erfassung der vorhandenen Maschinensignale herangezogen. Des Weiteren besteht die Möglichkeit, die Maschine mit zusätzlichen Sensoren auszustatten und diese ausschließlich in das System zur Datenerfassung zu integrieren. Der Großteil

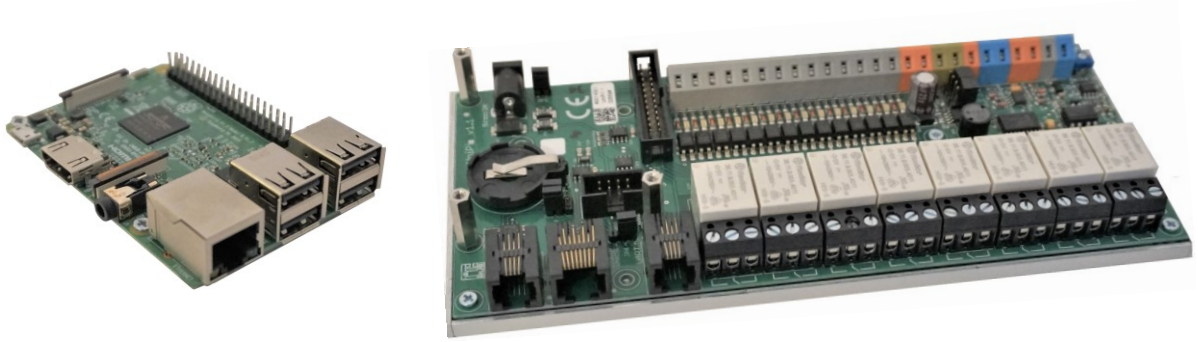
der am Markt angebotenen Sensoren wird mit Spannungen zwischen 0 und 24 V betrieben und kann deshalb ebenfalls ohne Weiteres an die in Folge vorgestellten Systeme angebunden werden.

4.2 Raspberry Pi 3 mit UniPi v1.1 Erweiterungsboard

Das in diesem Abschnitt betrachtete System zur Erfassung von Maschinendaten von Legacy Systems besteht grundsätzlich aus zwei Komponenten:

- Raspberry PI 3 Modell B
- Erweiterungsboard UniPi v1.1

Die Hauptkomponenten und das resultierende Gesamtsystem sind in ***Abbildung 50*** dargestellt.



Raspberry PI 3 Modell B Erweiterungsboard UniPi v1.1

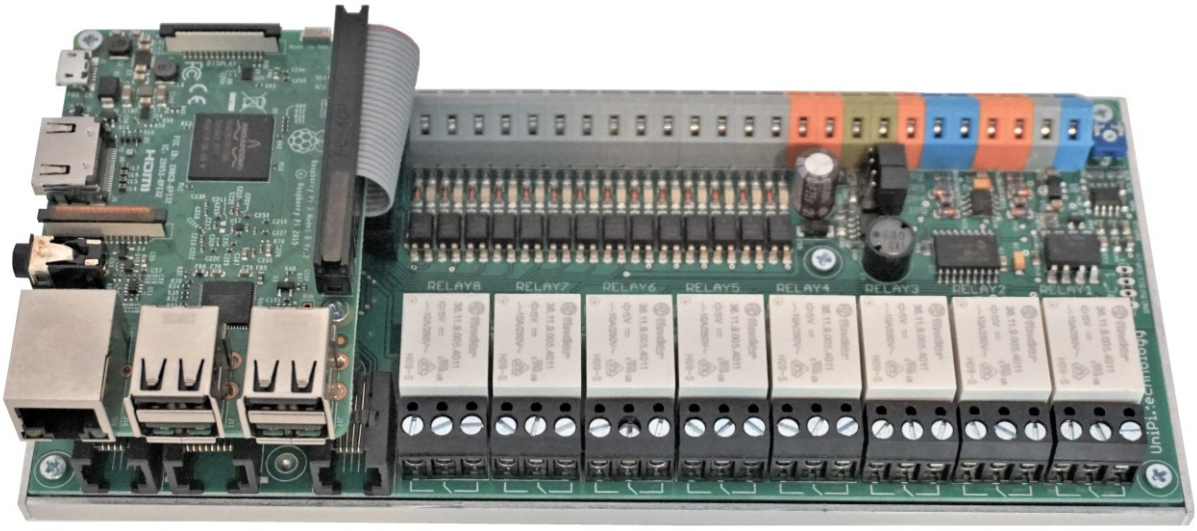


Abbildung 50: Hauptkomponenten und das Gesamtsystem

4.2.1 Raspberry PI 3 Modell B

Der Raspberry Pi ist ein preiswerter Einplatinen-Computer, der hauptsächlich im Unterhaltungs- und Hobbybereich weit verbreitet ist. In den letzten Jahren konnte man jedoch eine Zunahme von robusteren Systemen mit dem Raspberry Pi, welche den Einsatz in industriellen Umgebungen ermöglichen, beobachten. Die komplette Software die zur Programmierung notwendig ist, kann direkt am Raspberry Pi installiert werden. Python und Node-RED sind beispielsweise bei einigen Betriebssystemen als fixer Bestandteil integriert. Als Betriebssystem kommen hauptsächlich spezielle Linux-Distributionen mit grafischer Benutzeroberfläche zum Einsatz. Das Betriebssystem befindet sich auf einer MicroSD Speicherkarte, welche gleichzeitig zur Datenspeicherung verwendet wird. Weitere Spezifikationsdaten und Schnittstellen des Raspberry Pi sind:

- CPU mit ARMv8-Architektur und 1.2 GHz Quad Core
- Broadcom BCM2837 Chip-Set
- 1GB LPDDR2 SDRAM
- Schnittstellen: 10/100 Ethernet, HDMI, vier USB 2.0 Ports, CSI (Camera Serial Interface), DSI (Display Serial Interface) und ein I²C (Inter-Integrated Circuit) Datenbus

Es werden 40 Pins des integrierten Schaltkreises herausgeführt. Davon sind 26 Pins als GPIO (General Purpose Input/Output) frei programmierbar und zur Laufzeit steuerbar. Das System-on-a-Chip (SoC) des Raspberry Pi und somit die GPIOs arbeiten auf Basis von 3.3 V. Die Spannungsverträglichkeit liegt bei maximal 5 V. Die Strombelastbarkeit der GPIOs von wenigen mA stellt ebenfalls eine starke Einschränkung für den Einsatz im industriellen Umfeld dar. Aus diesen Gründen erweitert man den Raspberry Pi um eine Erweiterungsplatine, welche in Folge beschrieben wird.

4.2.2 Erweiterungsboard UniPi v1.1

Durch Einsatz des UniPi v1.1 können die in der Industrie üblichen Gleichspannungssignale – z.B. von Sensoren – mit geeigneten Schnittstellen abgegriffen werden. Der interne Schaltungsaufbau ermöglicht es, diese Signale trotz der zuvor beschriebenen Einschränkungen an das Raspberry Pi zur Weiterverarbeitung zu übertragen. Das UniPi nutzt Optokoppler um eine galvanische Trennung der digitalen Eingänge zu realisieren. Dadurch wird sichergestellt, dass eine Potentialtrennung zwischen dem Signal am Eingangs-Port des UniPi und dem Signal, welches in Folge an den Raspberry Pi übertragen wird, vorliegt. Dadurch können die digitalen Eingänge des UniPi von externen Geräten mit Gleichspannungen von 5 bis 24 V beschalten werden, ohne dabei den Raspberry Pi zu beschädigen. Die kleinste

detektierbare Impulsdauer der Spannung beträgt 5 ms. Nachfolgend sind die wichtigsten Schnittstellen des UniPi v1.1 zusammengefasst:

- Schnittstelle zum Raspberry Pi
- 14 digitale Eingänge bis 24 V-DC – durch entsprechendes Setzen der Konfigurationspins (Jumpers) und Nutzung der Konfigurations-Ports kann eine galvanische Trennung der Eingänge erreicht werden.
- 8 Relais mit 250 V-AC bei 5 A bzw. 24 V-DC bei 5 A
- UART Port – standardisierte serielle Schnittstelle
- 1 Wire Port – 1 Wire Businterface
- I²C Port – Anschluss weiterer Erweiterungsplatinen
- Real Time Clock (RTC) Modul
- 12 V bis 200 mA – kann zur Versorgung von Peripheriegeräten verwendet werden
- 2 analoge Eingänge für 0 bis 10 V
- Analoger Ausgang für 0 bis 10 V

In ***Abbildung 51*** ist das UniPi v1.1 inklusive Kennzeichnung der Schnittstellen dargestellt. Für weiterführende Informationen zum technischen Aufbau des UniPi sei auf das Manual[9] verwiesen.

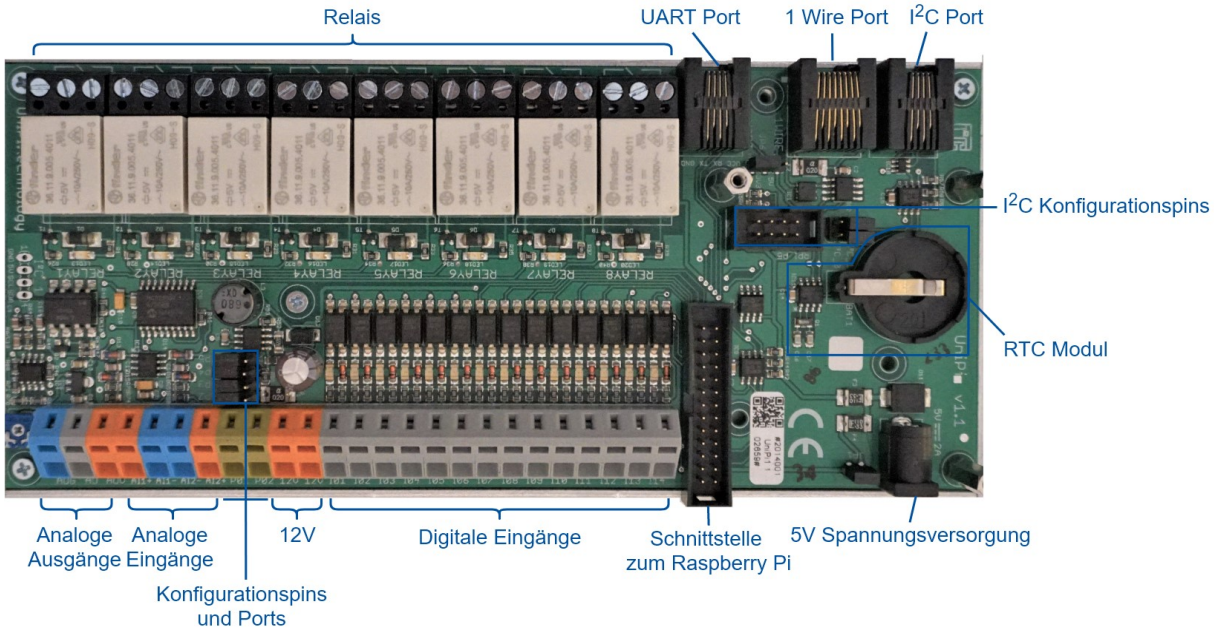


Abbildung 51: Schnittstellen des UniPi v1.1

[9] https://files.unipi.technology/s/public?path=%2FProducts%2FUniPi%201 — 12.08.2018

4.2.3 Programmierung

Die Programmierung kann und wird bei der Evaluierung in Kapitel 5 in Node-RED[10] und dem browserbasierten Editor mit graphischem User-Interface durchgeführt. Node-RED wurde von IBM entwickelt und basiert auf der Open-Source JavaScript Runtime-Umgebung Node.js. Die Programmierung setzt auf das Konzept der Ereignis- bzw. Datenstromorientierung. Es ist eine Vielzahl an einsatzfähigen Funktionen, sogenannten Nodes, verfügbar. Individuelle Funktionen können als JavaScript-Funktionen implementiert werden.

4.2.4 Übertragung der erfassten Maschinendaten

Zur Übertragung der Maschinendaten wird das in Kapitel 2.2 beschriebene asynchrone nachrichtenorientierte Kommunikationsprotokoll MQTT eingesetzt. Das in diesem Abschnitt betrachtete System fungiert als MQTT-Publisher und überträgt die Daten an einen MQTT-Broker. Von diesem Broker können die Daten über MQTT von Systemen mit MQTT-Subscriber Funktionalität für die Visualisierung oder Weiterverarbeitung bezogen werden. Es sei angemerkt, dass dieses System nicht auf die Verwendung von MQTT eingeschränkt ist, sondern durchaus die Möglichkeit besteht, die Datenübertragung mit OPC UA zu realisieren.

4.3 Siemens S7-1200

Die Siemens S7-1200 ist eine konventionelle SPS mit integrierter Ethernet-Schnittstelle. ***Abbildung 52*** zeigt eine Darstellung der Steuerung. Die Kommunikation erfolgt über den industriellen Ethernet-Standard Profinet. Als Kommunikationsprotokoll wird das auf TCP/IP basierende S7-Protokoll von Siemens verwendet. Weitere Spezifikationsdaten der Steuerung sind:

- Kompakt-CPU 1212C DC/DC/DC
- 8 digitale Eingänge 24 V-DC
- 6 digitale Ausgänge 24 V-DC
- 2 analoge Eingänge 0-10 V-DC
- IP20 nach EN 60529 und Umgebungstemperaturen im Betrieb von -20° bis +60°

[10] https://nodered.org/ – 12.08.2018

Abbildung 52: Siemens S7-1200 – CPU 1212C DC/DC/DC

4.3.1 Programmierung

Die Programmierung kann mit der Software TIA Portal von Siemens unter Verwendung der IEC Programmiersprachen KOP (Kontaktplan), FUP (Funktionsplan) oder ST (Strukturierter Text) umgesetzt werden.

4.3.2 Übertragung der Maschinendaten

Die Maschinendaten werden über das S7-Kommunikationsprotokoll an einen OPC UA Server übertragen. Dazu wird die SPS über Ethernet an den OPC UA Server angebunden, wobei dieser für den Datenaustausch entsprechende Treiber zur S7-Kommunikation unterstützen muss. Von dort können die Daten über OPC UA von Systemen mit OPC UA Client-Funktionalität für die Visualisierung oder Weiterverarbeitung abgeholt werden.

4.4 Gegenüberstellung

Einer der Hauptunterschiede der beiden vorgestellten Systeme liegt in den unterschiedlichen Domänen, in denen die Programmierung erfolgt. Die Programmierung am Raspberry Pi wird mit Programmiersprachen aus der Softwareentwicklung durchgeführt und bietet deshalb eine Vielzahl an Möglichkeiten sowie eine hohe Flexibilität. Die Programmierung der SPS muss hingegen mit den standardisierten IEC 61131 Programmiersprachen erfolgen. Für den Einsatz in raueren Umgebungen ist die SPS aufgrund seiner höheren Robustheit und Lebensdauer vorzuziehen. Vergleicht man die Kommunikationsmöglichkeiten der beiden Systeme, weist die S7-1200 SPS den Nachteil auf, dass weder eine direkte Kommunikation mit MQTT noch mit OPC UA zu übergeordneten Systemen unterstützt wird. Im Gegensatz dazu kann für die Datenübertragung mit dem Raspberry Pi sowohl MQTT als auch OPC UA verwendet werden.

5 Testumgebung und Evaluierung

5.1 Einleitung und Überblick

Die Testumgebung umfasst ein Steuerboard mit Taster und Schalter um die Zustände eines Legacy Systems zu simulieren. Um eine praxistaugliche Simulation von Legacy Systems zu erzielen, wurde mit Spannungssignalen zwischen 0 bis 24 V gearbeitet. Die Signale werden an die in Kapitel 4 beschriebenen Systeme angebunden und weiterverarbeitet.

Das System bestehend aus einem Raspberry Pi mit der UniPi v1.1 Erweiterungsplatine nutzt zur Datenübertragung der Maschinensignale an das übergeordnete System das in Kapitel 2.2 beschriebene asynchrone nachrichtenorientierte Kommunikationsprotokoll MQTT. Zur Erhöhung der Security wurde MQTT um das Verschlüsselungsprotokoll TLS zur sicheren Datenübertragung erweitert. Zusätzlich müssen sich Publisher und Subscriber mittels Username und Passwort am Broker authentifizieren. Als zweites System wurde eine konventionelle SPS von Siemens zur Signalerfassung verwendet. Die Maschinendaten werden mit dem S7-Kommunikationsprotokoll an einen übergeordneten OPC UA Server übertragen.

Jedes dieser Systeme kann als eigenständige Anbindung eines Legacy Systems angesehen werden. Beide Systeme wurden zusammen in einer Maschinenzone zusammengefasst und mit Hilfe einer industriellen Firewall von den übergeordneten Systemen getrennt. Dies ist eine mögliche Herangehensweise um die Konformität zur Norm IEC 62443, welche in Kapitel 3 beschrieben wurde, einzuhalten. MQTT-Broker und OPC UA Server wurden auf einem Laptop – als übergeordnetes System – außerhalb der Maschinenzone betrieben. Jegliche Kommunikation zwischen dem MQTT-Broker bzw. dem OPC UA Server und den Systemen innerhalb der Maschinenzone passiert die Firewall. Der Laptop ersetzt in der Testumgebung Server und Virtuelle Maschinen, welche entsprechend IEC 62443 in einer DMZ angesiedelt wären und MQTT-Broker bzw. OPC UA Server hosten. Am selben Laptop kam darüber hinaus Node-RED zur Visualisierung der Maschinendaten zur Anwendung. Die Kommunikation zwischen der Visualisierungsapplikation und dem MQTT-Broker erfolgt weiterhin über MQTT mit TLS. Zur Datenübertragung der Maschinensignale vom OPC UA Server an die Visualisierungsapplikation wird OPC UA, welches in Kapitel 2.1 ausführlich behandelt wurde, verwendet. Die im Rahmen der Testumgebung verwendeten Security Features von OPC UA erfordern eine gegenseitige Authentifizierung der OPC UA Applikationen und eine Authentifizierung des Users basierend auf Username und Passwort. Die Datenübertragung ist signiert und mit Basic 256 verschlüsselt.

Die resultierende Evaluierungsumgebung ist in ***Abbildung 53*** dargestellt und besteht zusammengefasst aus folgenden Komponenten:

- Steuerboard mit Taster und Schalter zur Simulation von Maschinenzuständen
- Raspberry Pi 3 Modell B mit Erweiterungsboard UniPi v1.1
- Siemens S7-1200 SPS
- Phoenix Contact Industrie Firewall zur Bildung der Maschinenzone
- Laptop mit Mosquitto MQTT Broker und KEPServerEX OPC UA Server sowie Node-RED zur Visualisierung

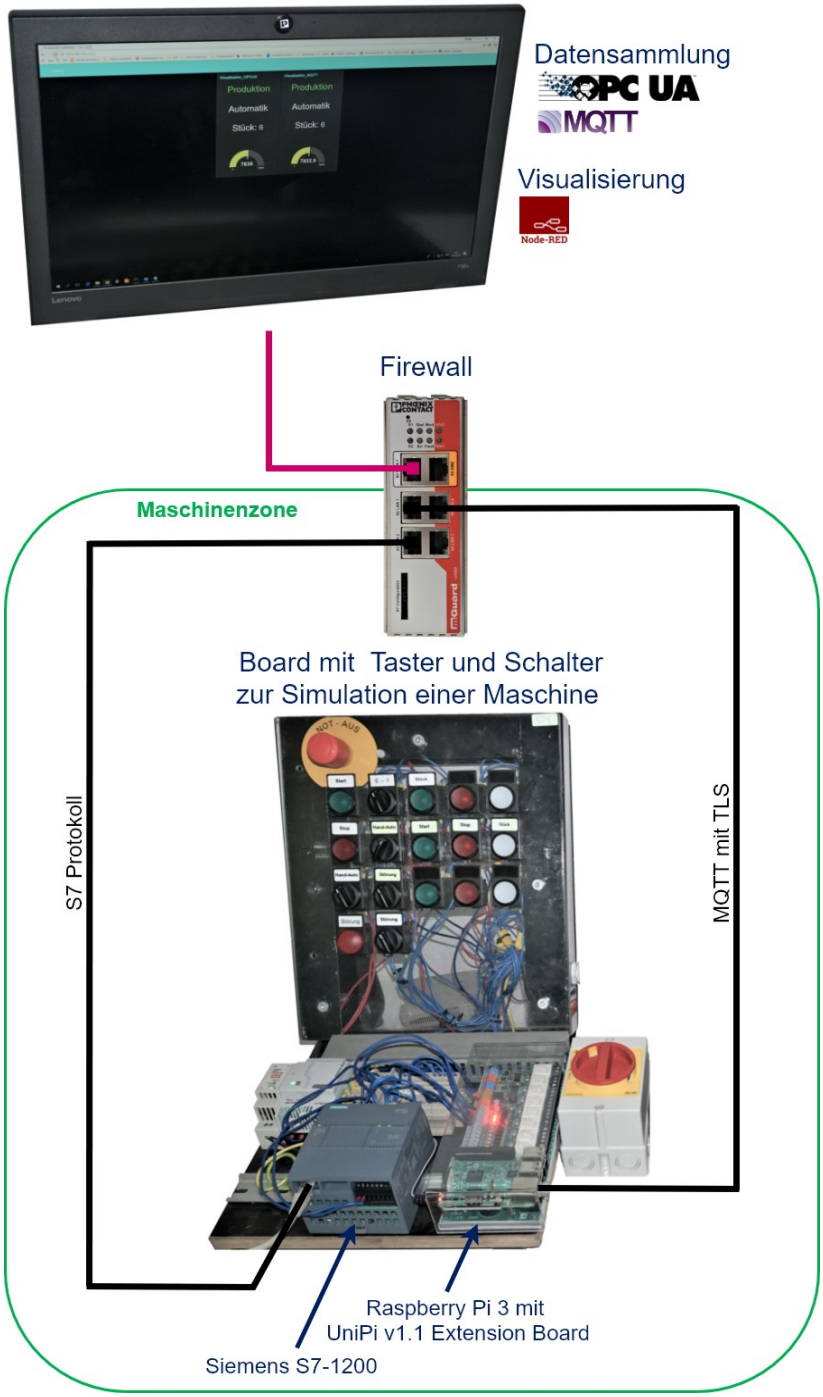


Abbildung 53: Evaluierungsumgebung

5.2 Aufbau und Setup der Komponenten

In diesem Abschnitt werden die wichtigsten Aspekte für den Aufbau, wie beispielsweise die Verkabelung für einen reibungslosen Betrieb und um Beschädigungen an den Komponenten zu vermeiden, beschrieben. Neben den Signaldefinitionen werden an den entsprechenden Stellen die Beziehungen zwischen Signalen und etwaigen Anschlussklemmen angeführt. Die Software wird anhand einer kurzen Aufgabenbeschreibung ebenfalls erläutert. Darüber hinaus werden während des Setups der Komponenten speziell vorgenommene Konfigurationen und zusätzliche Maßnahmen zur Minderung von Sicherheitslücken des Systems beschrieben.

5.2.1 Steuerboard zur Simulation von Maschinenzuständen

Das Steuerboard wurde genutzt um die Basissteuerelemente einer Maschine nachzubilden und damit die Simulation einer Maschine zu ermöglichen. Dafür wurden insgesamt drei Taster und zwei Schalter verwendet. Die Komponenten wurden gemeinsam mit +24 V DC versorgt und die Enden auf eine Klemmleiste geführt. Das Steuerboard ist in ***Abbildung 54*** dargestellt. Die verwendeten Komponenten sind farblich eingerahmt. Der Schaltplan des Steuerboards inklusive der Klemmenzuordnung ist im Anhang 8.5.1 zu finden. Start- und Stückzahl-Taster wurden als Schließkontakte (normally open – NO) ausgeführt. Der Stopp-Taster wurde hingegen als Öffner-Kontakt (normally closed – NC) ausgeführt. Die beiden Schalter signalisieren im Grundzustand (Schalter offen) die Zustände Handbetrieb bzw. keine Störung. Geschlossene Schalter signalisieren Automatikbetrieb bzw. Störung.

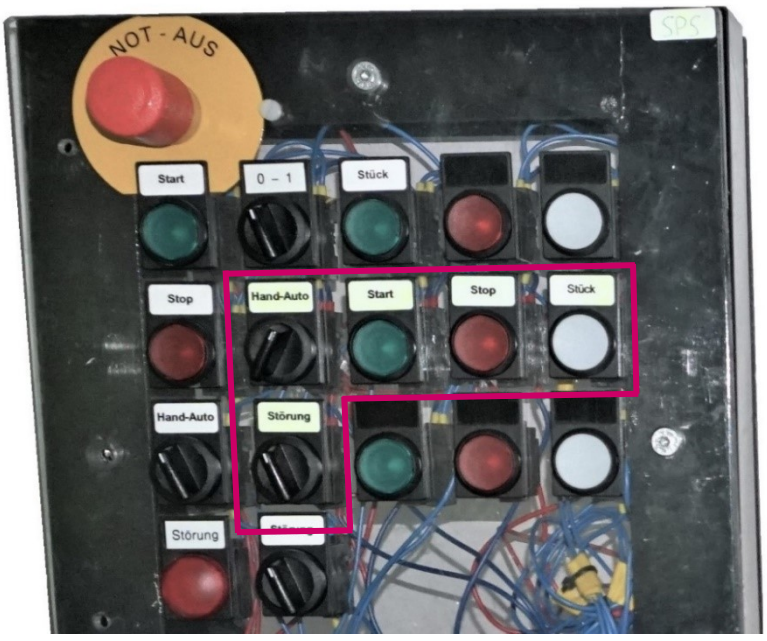


Abbildung 54: Das Steuerboard mit den verwendeten Komponenten

5.2.2 Raspberry Pi 3 mit UniPi v1.1 Erweiterungsboard

- Setup des Raspberry Pi

Die Systeminformationen und die Software, die im Rahmen dieser Arbeit am Raspberry Pi zum Einsatz kam, sind in ***Tabelle 6*** zusammengefasst. Zu Beginn wurde ein neuer User inklusive Passwort angelegt und der Standard-User am Raspberry Pi entfernt. Die Einstellungen der verwendeten Grundkonfiguration sind im Anhang 8.5.2 zu finden. Node.js und Node-RED wurde zur Programmierung verwendet[11]. Die browserbasierte Programmierumgebung von Node-RED wurde mit Username und Passwort abgesichert[12]. EVOK ist ein API und Web-Interface für das UniPi v1.1 Erweiterungsboard und liefert REST-, JSON- sowie ein WebSocket-Interface um die Kommunikation mit den Relays, den analogen bzw. digitalen Eingängen sowie den analogen Ausgängen zu ermöglichen.

Damit per Computer Einstellungen am Raspberry Pi vorgenommen werden können, wurde eine SSH-Verbindung konfiguriert und eingerichtet. Um die Konfiguration des am Raspberry Pi vorhandenen SSH-Servers zu optimieren, wurden die im Anhang 8.5.2 zusammengefassten Änderungen vorgenommen. Unter anderem wurde die verwendete Port-Einstellung von Port 22 auf Port 3180 geändert um einen potenziellen Angreifer nicht auf den ersten Blick zu zeigen, dass ein Zugriff per SSH auf das Raspberry Pi möglich ist. Zusätzlich wurde zur Absicherung des SSH-Zugriffs Fail2Ban installiert. Fail2Ban sperrt für eine gewisse Zeit alle IP-Adressen, welche eine voreingestellte Anzahl von erfolglosen Anmeldeversuchen per SSH durchführen. Damit wurde eine Schutzmaßnahme gegen Brute-Force-Angriffe realisiert. Die Dauer der Sperre und die Anzahl erfolgloser Anmeldeversuche kann in der Konfigurationsdatei gewählt werden. Darüber hinaus können gesperrte IP-Adressen eingesehen und Sperren, falls erforderlich, wieder aufgehoben werden.

Systeminformationen	Bezeichnung
Betriebssystem	Raspbian GNU/Linux 9 – Stretch
Veröffentlichungsdatum	2018-03-13
Kernel Version	4.9
Software	**Version**
Node.js	8.11.1
Node-RED	0.18.5
EVOK (pigpiod Library enthalten)	2.0.5a

[11] https://nodered.org/docs/hardware/raspberrypi – 12.08.2018
[12] https://nodered.org/docs/security – 12.08.2018

Fail2Ban	0.9.6

Tabelle 6: Systeminformationen des Raspberry Pi und verwendete Software

- Setup des UniPi v1.1

Das UniPi v1.1 wurde mit dem mitgelieferten 5 V DC Netzteil versorgt. Durch Verwendung von Jumper JP1 auf der Platine des UniPi v1.1 wird das Raspberry Pi ohne zusätzliche Kabel mitversorgt. Um die galvanische Trennung an den digitalen Eingängen sicherzustellen und damit die Anbindung von 24 V DC Signalen zu ermöglichen, wurden JP2 bis JP5 gemäß dem Manual gesetzt. Der Anschluss der Signalleitungen wurde wie in ***Abbildung 55*** gezeigt vorgenommen. Der negative Pol der externen Quelle, welcher im vorliegenden Fall dem negativen Pol der Spannungsversorgung von den Tastern und Schaltern entspricht, muss an den beiden PO-Ports des UniPi v1.1 (grüne Klemmen) angeschlossen werden. Die Signalleitungen von den Tastern und Schaltern wurden von den entsprechenden Klemmen der Klemmleiste mit den digitalen Eingängen des UniPi v1.1 (hellgraue Klemmen) verbunden. Die Zuordnung zwischen den Signalen, Klemmen und den digitalen Eingängen am UniPi v1.1 kann ***Tabelle 7*** entnommen werden.

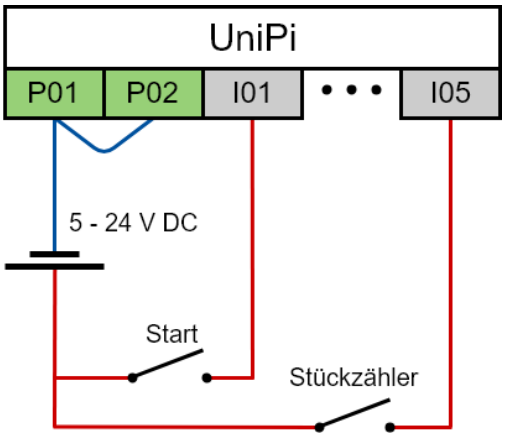


Abbildung 55: Anschluss der Signalleitungen an die digitalen Eingänge des UniPi v1.1

Signalbezeichnung	Klemme	UniPi – Digital Inputs
Start	11	I01
Stopp	14	I02
Hand-/Automatikbetrieb	9	I03
Störung	10	I04
Stückzähler	17	I05

Tabelle 7: Signalzuordnung für das UniPi v1.1

5.2.3 Siemens S7-1200

Abbildung 56 zeigt wie die SPS versorgt und die Signalleitungen an der SPS angeschlossen wurden. Ein detaillierteres Diagramm zur Verkabelung und die zugehörige Pin-Belegung sind im Anhang des Manuals[13] zu finden. Die Signalleitungen von den Tastern und Schaltern wurden von den entsprechenden Klemmen der Klemmleiste mit den digitalen Eingängen der SPS verbunden. Die Zuordnung zwischen den Signalen, Klemmen und den digitalen Eingängen der SPS können ***Tabelle 8*** entnommen werden.

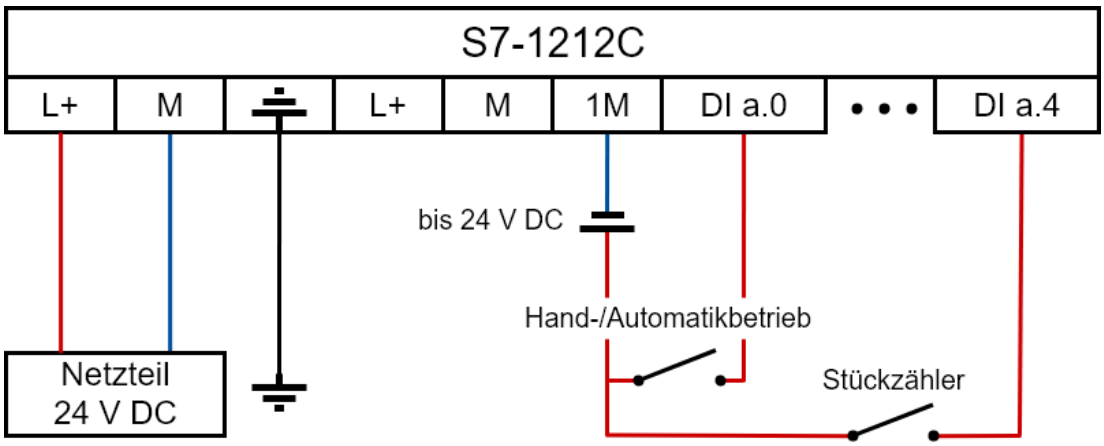


Abbildung 56: Anschluss der Signalleitungen an die digitalen Eingänge der SPS

Signalbezeichnung	Klemme	SPS – Digital Input
Start	11	DI a.2
Stopp	14	DI a.3
Hand-/Automatikbetrieb	9	DI a.0
Störung	10	DI a.1
Stückzähler	17	DI a.4

Tabelle 8: Signalzuordnung für die SPS

Die Systeminformationen und die Software zur Programmierung der SPS, welche im Rahmen dieser Arbeit zum Einsatz kam, sind in ***Tabelle 9*** angeführt.

13 https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf – 12.08.2018

Systeminformationen	Bezeichnung
CPU	1212C DC/DC/DC
Firmware	4.1
Software zur Programmierung	**Version**
Siemens TIA Portal (Testlizenz)	V13 SP2
STEP 7 Basic-Version (Testlizenz)	V13 SP2

Tabelle 9: Informationen der S7-1200 SPS und der verwendeten Software zur Programmierung

5.2.4 Industrie Firewall – FL mGuard RS4004 TX/DTX VPN

Die Industrie Firewall wurde separat über die Power-Klemme, die sich an der Oberseite des Gerätes befindet, mit einem 24 V DC Netzteil versorgt. Nachfolgend sind die wichtigsten Informationen und Schnittstellen sowie die Funktionen der Firewall, die im Rahmen dieser Arbeit zum Einsatz kamen, zusammengefasst:

- Firmware: 8.6.1.default
- 6 Ethernet-Ports: 1x WAN, 1x DMZ, 4x LAN
- Routing: Standard Routing, NAT (Network Address Translation), 1:1 NAT
- Filter: MAC- und IP-Adressen, Protokolle, Ports

Weiterführende Informationen sowie technische Details sind dem Handbuch bzw. dem Datenblatt zu entnehmen und auf der Homepage zu finden[14].

5.2.5 Laptop – MQTT-Broker, OPC UA Server, Node-RED

Der Laptop übernahm in der Evaluierungsumgebung mehrere Aufgaben, welche in einer realen IT-Infrastruktur normalerweise mit Servern oder VMs realisiert werden. Sowohl der MQTT-Broker als Datenziel des Erfassungssystems, bestehend aus dem Raspberry Pi und der Erweiterungsplatine UniPi v1.1, als auch der OPC UA Server zur Erfassung der Daten von der S7-1200 SPS liefen direkt am Laptop. Die Systeminformationen und die Software, die im Rahmen dieser Arbeit am Laptop zum Einsatz kam, sind in ***Tabelle 10*** zusammengefasst. Als MQTT-Broker wurde Mosquitto eingesetzt und als Docker-Container betrieben. Für eine sichere Datenübertragung wurde MQTT um TLS erweitert. Letzteres erforderte die Erstellung von Zertifikaten und Schlüssel, wofür das OpenSSL-Toolkit herangezogen wurde. Als OPC UA Server wurde die Demoversion des KEPServerEX von PTC verwendet. Die kostenfreie Demoversion des Servers erfordert alle zwei Stunden die Durchführung eines Neustarts. Die Visualisierung der Maschinendaten und die dafür erforderliche

[14] https://www.phoenixcontact.com/online/portal/at – 12.08.2018

Datenübertragung zwischen dem MQTT-Broker bzw. dem OPC UA Server und der Visualisierungsapplikation wurde ebenfalls am Laptop realisiert. Zur Programmierung wurden Node.js und Node-RED genutzt. Für den Zugriff auf die browserbasierte Programmierumgebung und Anzeige der Dashboards in Node-RED wurde eine Authentifizierung mit Username und Passwort eingerichtet. Die zusätzlich verwendeten Nodes in Node-RED sind ***Tabelle 10*** aufgelistet und übernahmen folgende Funktionen:

- OPC UA Client – für den Abgriff der Maschinensignale vom OPC UA Server
- Datawarehouse – bei Änderung eines einzelnen Maschinensignals werden dennoch alle Maschinensignale für eine Statusabfrage vor der Visualisierung bereitgestellt
- Dashboard – verschiedene Nodes für die Erstellung der Dashboards

Probleme während der Konfiguration von MQTT mit TLS in Node-RED erforderten den Einsatz weiterer Software. Unter anderem wurde Wireshark zur Analyse des Netzwerkverkehrs verwendet. Damit eine Fehlfunktion des MQTT-Brokers ausgeschlossen werden konnte, wurde MQTT mit TLS unabhängig von Node-RED mit dem Client MQTT.fx getestet. Die Problematik bei der Konfiguration von MQTT mit TLS in Node-RED wird in Abschnitt 5.5.1 näher betrachtet.

Systeminformationen	**Bezeichnung**
Modell	Lenovo P50s
Betriebssystem	64-Bit Windows 10 Enterprise
Prozessor	Intel(R) Core(TM) i7-6500U CPU @ 2.5 GHz
Software	**Version**
Docker	18.03.1-ce
Mosquitto MQTT Broker (Docker image ansi/mosquito)	1.4.15
OpenSSL-Toolkit	1.0.2j
OPC UA Server KEPServerEX	6.3.279.0
Node.js	8.11.1
Node-RED	0.18.5
Zusätzliche Nodes in Node-RED	**Version**
node-red-contrib-opcua	0.2.28
node-red-contrib-datawarehouse	1.1.0
node-red-contrib-dashboard	2.9.3

Software für Test- und Analysezwecke	Version
Wireshark zur Analyse des Datenverkehrs im Netzwerk	2.4.1
MQTT Client MQTT.fx	1.7.0

Tabelle 10: Systeminformationen des Laptops und die eingesetzte Software zur Datenerfassung und Visualisierung

5.3 Netzwerkumgebung

Der verwendete Netzwerkaufbau für die Evaluierungsumgebung ist in ***Abbildung 57*** dargestellt. Neben dem schematischen Aufbau sind die IP-Adressen der Komponenten bzw. die Interfaces der Firewall sowie die notwendigen Ports für die Kommunikation angegeben.

In der Evaluierungsumgebung wurden MQTT-Broker und OPC UA Server sowie Node-RED zur Visualisierung der Maschinendaten auf einem Laptop gehostet. Der Laptop befand sich außerhalb der Maschinenzonen und war deshalb über die Ethernet-Schnittstelle mit dem WAN-Port der Firewall verbunden. Dem Laptop wurde die IP-Adresse 10.0.0.100/24 aus dem privaten Netzadressbereich 10.0.0.0/24 zugewiesen. Die Konfiguration des WAN-Interfaces der Firewall – am Webserver der Firewall als *Extern* bezeichnet – ist in ***Abbildung 58*** gezeigt. Dem Interface wurde die IP-Adresse 10.0.0.10/24 zugeteilt, da sich dieses ebenfalls im Netzadressbereich 10.0.0.0/24 befinden muss.

Die Maschinenzone liegt im privaten Netzadressbereich 192.168.0.0/24 und fasst beide Systeme zur Anbindung von Legacy Systemen zusammen. Auf der einen Seite wurde das System über die Ethernet-Schnittstelle des Raspberry Pi mit der statischen IP-Adresse 192.168.0.139/24 und LAN-Port 1 der Firewall verbunden. Auf der anderen Seite wurde das System über die Ethernet-Schnittstelle der SPS mit LAN-Port 2 der Firewall verbunden. Der SPS wurde die statische IP-Adresse 192.168.0.25/24 zugewiesen. Die Konfiguration des LAN-Interfaces der Firewall – am Webserver der Firewall als *Intern* bezeichnet – ist in ***Abbildung 59*** gezeigt.

Im nächsten Schritt wurde die NAT-Funktionalität der Firewall genutzt. Jeder Komponente aus dem Maschinennetz konnte dadurch, wie in ***Abbildung 60*** gezeigt, eine eindeutige IP-Adresse des Netzwerks außerhalb der Maschinenzone zugeordnet werden. Diese IP-Adressen wurden in Folge für den Zugriff von außen auf die Komponenten der Maschinezone verwendet. Grund dafür ist, dass häufig weitere Maschinenzonen aus Komponenten mit identen IP-Konfigurationen gebildet werden. Die NAT-Funktionalität ermöglicht dennoch einen eindeutigen zentralen Zugriff auf die Komponenten der verschiedenen Maschinenzonen ohne zeitintensive Änderungen in den Konfigurationen der Komponenten durchführen zu müssen – häufig wären

derartige Änderungen mit Programmierarbeiten auf SPS-Ebene verbunden. Innerhalb einer Maschinenzone darf es natürlich dennoch zu keinen IP-Adresskonflikten kommen. Für die Evaluierungsumgebung bedeutet dies, dass mit geringem Aufwand eine exakte Kopie der bereits vorhandenen Maschinenzone in das Netzwerk integriert werden kann.

Im letzten Schritt wurde der Datenverkehr zwischen den Komponenten mit Hilfe entsprechender Regelsätze in der Firewall weitgehend eingeschränkt. Die definierten Eingangs- und Ausgangsregelsätze in der Firewall sind in ***Abbildung 61*** gezeigt. Für das Testen der Erreichbarkeit von den Netzwerkkomponenten wurde das ICMP-Protokoll für das Pingen der Komponenten der Maschinenzone ausgehend vom Laptop erlaubt. In umgekehrter Richtung wurde ein Ping des Laptops ausgehend vom Raspberry Pi ermöglicht. Damit die Programmierung in Node-RED auf dem Raspberry Pi mit dem Laptop durchgeführt werden konnte, wurde der Zugriff auf die Node-RED Programmierumgebung über Port 1880 freigegeben. Für Remote-Zugriffe auf das Raspberry Pi über SSH wurde der konfigurierte Port 3180 in der Firewall freigegeben. Dadurch können beispielsweise Änderungen in der Konfiguration des Raspberry Pi und eine Fehleranalyse bzw. -behebung sowie ein Herunterfahren bzw. Neustart mit dem Laptop vorgenommen werden. Des Weiteren wurden Port 1883 für die MQTT-Kommunikation sowie Port 102 für die S7-Kommunikation freigegeben. Die zuvor genannten Regelsätze wurden, je nachdem welche Komponente als Initiator für den betrachteten Kommunikationsaufbau gilt, entweder als Eingangsregel oder als Ausgangsregel definiert.

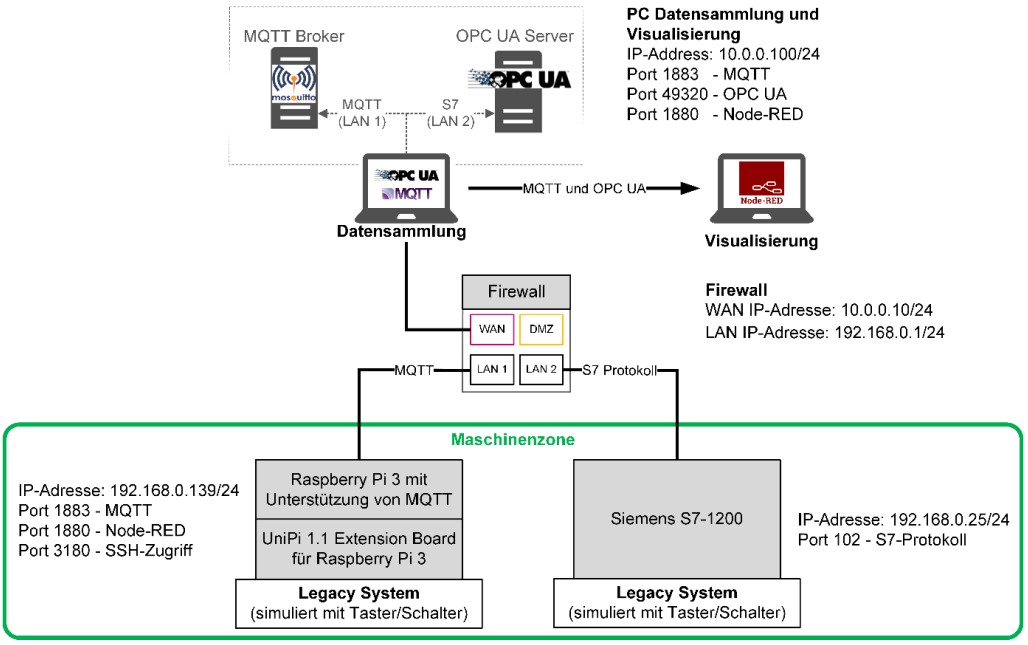


Abbildung 57: Schematische Evaluierungsumgebung – separate Integration von MQTT- und OPC UA-Daten in der Visualisierung

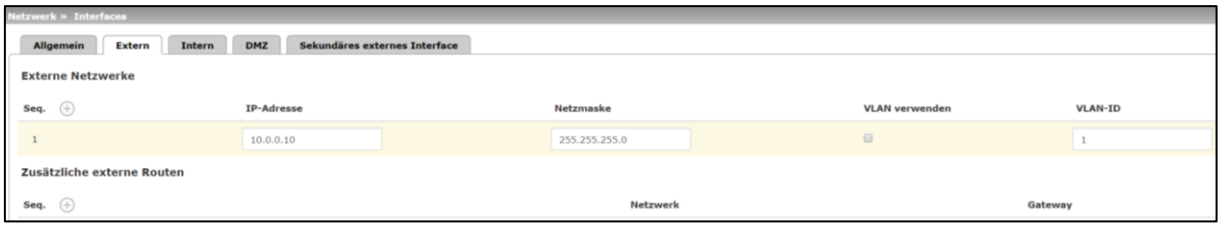


Abbildung 58: Konfiguration Firewall-Interface – Extern (WAN)

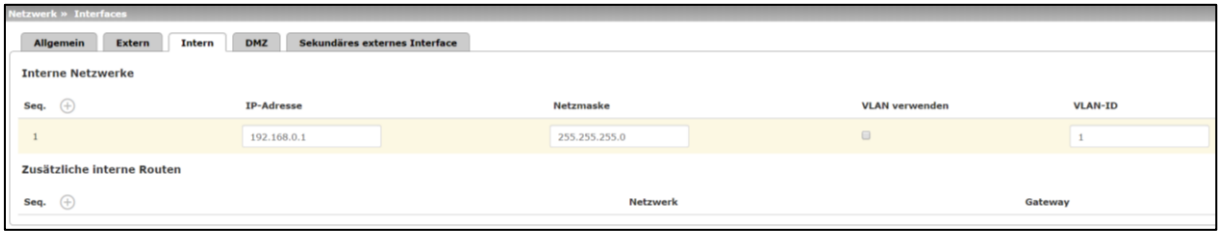


Abbildung 59: Konfiguration Firewall-Interface – Intern (LAN)

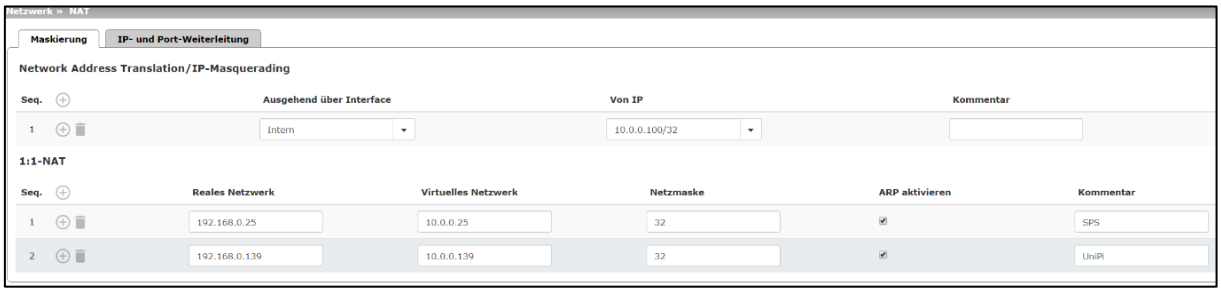
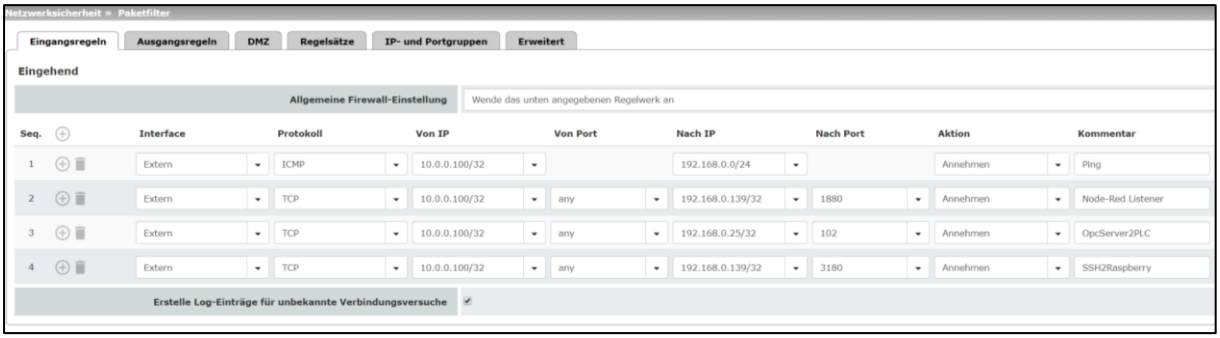


Abbildung 60: IP-Masquerading und 1:1 NAT



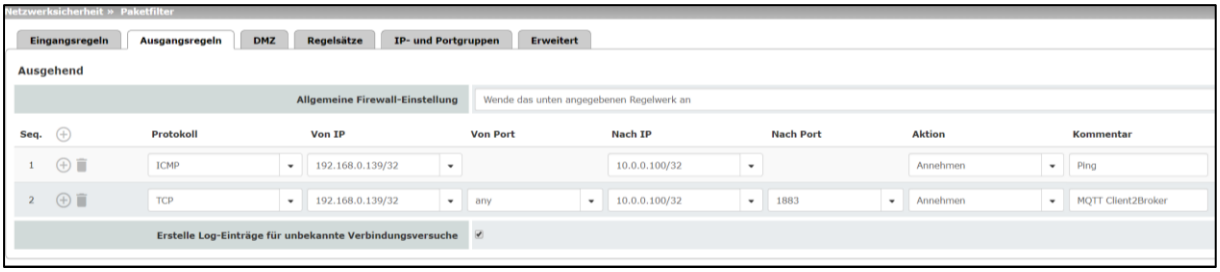


Abbildung 61: Regelsätze der Firewall

5.4 Programmierung für die Datenerfassung

In diesem Abschnitt wird die Programmierung der beiden Systeme zur Anbindung von Legacy Systemen detaillierter betrachtet. Für die Programmierung zur Datenerfassung mit dem Raspberry Pi wurde Node-RED verwendet. Die Programmierung der SPS wurde im TIA Portal mit FUP realisiert.

5.4.1 Programmierung des Raspberry Pi mit Node-RED

Der obere Teil des Node-RED Flows aus ***Abbildung 62*** zeigt, dass über die interne WebSocket-Verbindung zwischen dem Raspberry Pi und dem UniPi v1.1 die Signale an den Eingängen gesammelt ausgelesen werden. Mit Hilfe des Switch-Node erfolgte anschließend eine Separierung der Signale. Mit den darauffolgenden Funktion-Nodes wurden kleine Änderungen der Datenstruktur vorgenommen. Danach wurden die Daten bis auf den Stückzahl-Impuls an die MQTT-Nodes zur Übertragung an den MQTT-Broker weitergereicht. Die Konfiguration der MQTT-Nodes wird in Abschnitt 5.5.1 gesondert betrachtet.

Der Stückzahl-Impuls wurde zur Weiterverarbeitung an den unteren Teil des Node-RED Flows übergeben. In dem Funktion-Node „Return Counter“ werden Impulse gezählt, in eine Stückzahl umgewandelt und anschließend an den MQTT-Node übergeben. Bei jedem Stopp-Signal wird der Stückzähler auf den Wert Null rückgesetzt. Des Weiteren wurde mit der Stückzahl die Berechnung einer Leistung in Form von produzierten Stück pro Stunde als einfache Edge-Operation durchgeführt. Dazu sind im vorliegenden Fall die Zeitspanne der letzten fünf Stück herangezogen und die Stückzahl auf eine Stunde hochgerechnet worden. Die Leistung wurde somit in Intervallen von fünf Stück berechnet und aktualisiert, wobei das Intervall in der Praxis entsprechend dem vorherrschenden Maschinentakt anzupassen ist. Die resultierende Leistung wurde ebenfalls an einen MQTT-Node zur Übertragung an den Broker weitergeleitet.

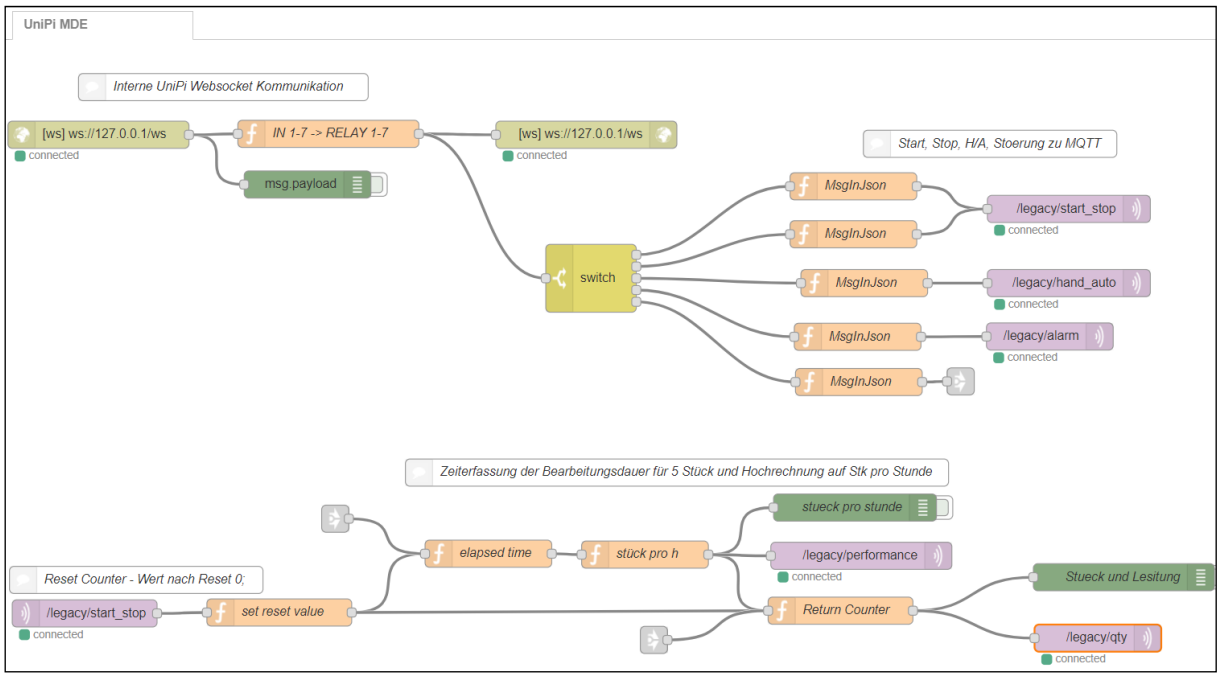


Abbildung 62: Programmierung des Raspberry Pi – Node-RED Flow

5.4.2 SPS Programmierung im TIA Portal

Der entwickelte Funktionsbaustein mit seinen Eingängen ist in ***Abbildung 63*** dargestellt. Der Baustein beinhaltet sieben Netzwerke:

1. Netzwerk 1 – Start/Stopp: Verwendung eines Flip-Flops, welches mit dem Start-Signal gesetzt und mit dem Stopp-Signal rückgesetzt wird. Das Flip-Flop ist im vorliegenden Fall notwendig um den Start-Zustand auf dem Signallevel High zu halten, da der Startzustand lediglich mit einem Signalimpuls simuliert wurde.
2. Netzwerk 2 – Hand-/Automatikbetrieb: Verwendung eines Flip-Flops, welches mit dem Signal für den Automatikbetrieb gesetzt und mit dem Signal für den Handbetrieb rückgesetzt wird.
3. Netzwerk 3 – Störung: Zuweisung des Eingangssignals zu einer Variable
4. Netzwerk 4 – Stückzähler: Die Stückzahl wurde ohne Zählerbaustein, sondern über eine Addition der positiven Signalflanken realisiert. Des Weiteren wurde die Programmierung derart ausgeführt, dass die Stückzahl ausschließlich bei aktivem Startzustand erhöht werden kann.
5. Netzwerk 5 – Leistung: Die Zeitspanne zwischen zwei aufeinander folgenden Stückzahlimpulsen wird mit Hilfe von zwei sich abwechselnden Timer-Bausteinen bestimmt. Die Zeitspannen werden in ein Schieberegister mit der Länge fünf geschrieben. Nachdem das Schiebregister erstmals vollständig mit Werten befüllt ist erfolgt die Berechnung der Leistung in Form von Stückzahl pro Stunde. Das bedeutet im vorliegenden Fall, dass die Berechnung das erste Mal beim sechsten Stückzahlimpuls ausgeführt wird. Bei jedem weiteren Stückzahlimpuls fällt die älteste Zeitspanne aus dem Schieberegister heraus

und die aktuell erfasste Zeitspanne wird stattdessen am Anfang angefügt. Bei jedem eintreffenden Stückzahlimpuls erfolgt somit auf Basis der Zeitspannen der letzten fünf Stück eine neuerliche Berechnung der Leistung. Bei jedem Stopp-Signal wird diese Leistung auf den Wert Null gesetzt und das Schieberegister entleert. In der Praxis ist die Länge des Schieberegisters entsprechend dem vorliegendem Maschinentakt und den Anforderungen anzupassen.

6. Netzwerk 6 – Reset des Stückzählers: Die Stückzahl wird bei jedem Stopp-Signal auf den Wert Null rückgesetzt.
7. Netzwerk 7 – Reset der entsprechenden Variablen für die Berechnung der Leistung aus Netzwerk 5. Der Reset wird analog zur Stückzahl bei jedem Stopp-Signal durchgeführt.

Die detaillierte Programmierung der sieben Netzwerke des Funktionsbausteins ist im Anhang 8.5.3 zu finden.

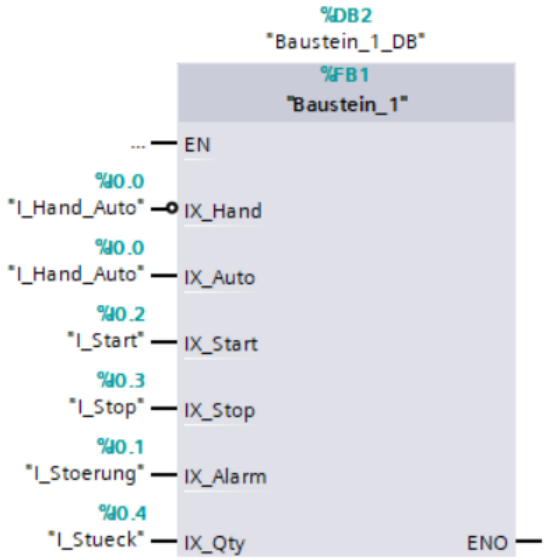


Abbildung 63: SPS Funktionsbaustein

Damit der Zugriff vom OPC UA Server auf den zum Funktionsbaustein zugehörigen Datenbaustein erfolgen kann, müssen im SPS-Projekt unbedingt zwei Einstellungen vorgenommen werden:

1. Im Datenbaustein muss in den Eigenschaften unter „Attribute“ der Haken beim optimierten Baustein-Zugriff entfernt werden.
2. In der Geräteeinstellung der SPS muss für die S7-1200 ab Firmware Version 4 unter „Schutz“ die PUT/GET Kommunikation erlaubt werden.

5.5 Kommunikationsumgebung – MQTT und OPC UA

In diesem Abschnitt werden der Aufbau und die notwendigen Konfigurationen der funktionsfähigen Kommunikationsumgebung für die Datenerfassung und anschließende Übertragung an das System zur Visualisierung mit MQTT und OPC UA beschrieben.

5.5.1 Kommunikationsumgebung für MQTT

- Mosquitto MQTT-Broker

Der Mosquitto MQTT-Broker wurde als Docker-Container am Laptop betrieben. Da MQTT ohne zusätzliche Maßnahmen keine Verschlüsselung verwendet, wird neben der Datenübertragung auch die Authentifizierung des MQTT-Clients mittels Username und Passwort in Klartext an den Broker übermittelt. Aus diesem Grund wurde zur Erhöhung der Sicherheit MQTT um TLS erweitert. Die Erstellung der notwendigen Zertifikate und Schlüssel wurde anhand einer Beschreibung[15] mit dem OpenSSL-Toolkit durchgeführt. Folgende Schritte waren notwendig:

1. Generierung eines CA-Schlüsselpaars
2. Erstellung eines CA-Zertifikates und seine Signierung mit dem privaten Schlüssel von Schritt 1
3. Generierung eines Schlüsselpaars für den Broker
4. Erstellung eines Requests zur Signierung des privaten Schlüssels von Schritt 3 mit dem CA-Zertifikat
5. Durchführung der Signierung laut Request aus Schritt 4 mit dem CA-Zertifikat von Schritt 2

Anschließend wurde im Docker-Container ein Zertifikatsverzeichnis erstellt, in dem die generierten Files abgelegt wurden. Der Broker benötigt folgende Files:

- Das Zertifikat der CA – m2mqtt_ca.crt
- Das durch die CA signierte Zertifikat des Servers – m2mqtt_srv.crt
- Den privaten Schlüssel des Servers – m2mqtt_srv.key

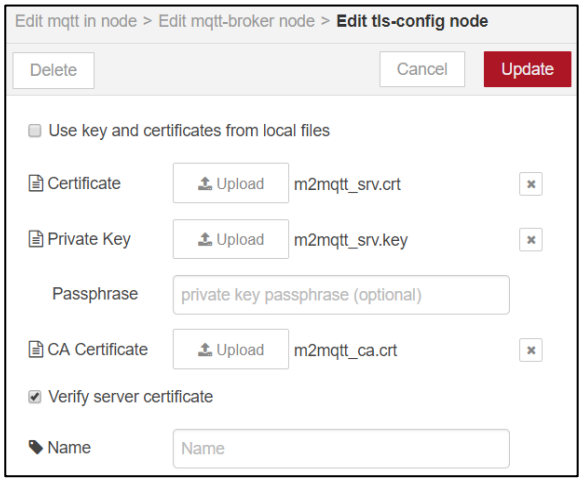
Im Anschluss daran wurden im Konfigurationsfile des Brokers – mosquitto.conf – die entsprechenden Pfade zu den Zertifikaten und dem Schlüssel eingetragen. Zusätzlich wurde zur Authentifizierung ein User mit Passwort eingetragen.

[15] https://mcuoneclipse.com/2017/04/14/enable-secure-communication-with-tls-and-the-mosquitto-broker/ – 12.08.2018

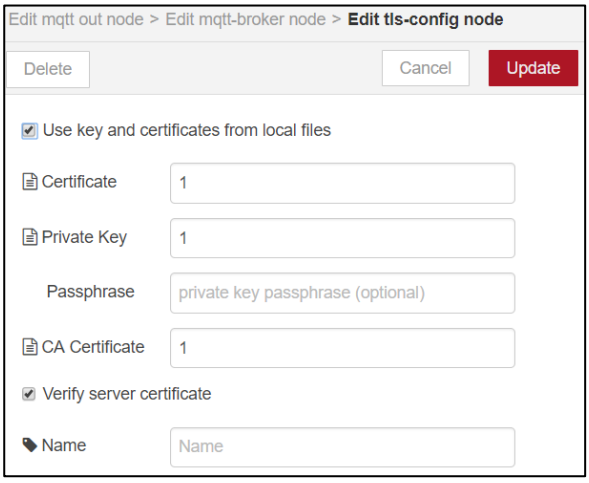
- MQTT-Client in Node-RED

Node-RED stellt zur Kommunikation mit einem MQTT-Broker zwei Nodes zur Verfügung. Der MQTT In-Node fungiert als Subscriber und der MQTT Out-Node als Publisher. Trotz der Trennung in zwei Nodes stellen diese zusammen einen Client dar, weshalb beide Nodes dieselbe Konfiguration nutzen und sich ausschließlich zum selben MQTT-Broker verbinden können. In den MQTT-Nodes wurde ein neuer MQTT-Broker angelegt und konfiguriert. Unter dem Reiter „Connection“ wurden die IP-Adresse und der Port des am Laptop gehosteten Mosquitto Brokers angegeben. Der Broker besitzt die IP-Adresse 10.0.0.100 und horcht auf Port 1883. Im nächsten Schritt wurden der Haken zur Aktivierung der sicheren Verbindung mit SSL/TLS gesetzt sowie eine neue TLS-Konfiguration angelegt und konfiguriert. Das Problem hierbei war, dass keine erfolgreiche sichere Verbindung zum Broker durch Angabe der erstellten Zertifikate und des Schlüssels, wie in ***Abbildung 64*** dargestellt, hergestellt werden konnte. Für Testzwecke wurde deshalb am Laptop unabhängig von Node-RED der MQTT Client MQTT.fx eingesetzt. Die sichere Verbindung konnte damit ohne jegliche Probleme hergestellt werden.

Es wurde ein Fehler in den MQTT-Nodes von Node-RED vermutet. Der Grund zu dieser Annahme verstärkte sich, als trotz Fehlermeldung in Node-RED plötzlich eine sichere Verbindung hergestellt werden konnte (vgl. ***Abbildung 64*** rechts). Eine Lösung zur Behebung des Fehlers konnte nicht gefunden werden. Die weiterführende Analyse des Netzwerkverkehrs mit Wireshark zeigte, dass zu Beginn des Verbindungsaufbaus ein TLS-Handshake durchgeführt und sowohl die Authentifizierung mit User und Passwort, als auch die Übertragung der Maschinendaten verschlüsselt wurden. Die Ergebnisse der Analyse des Netzwerkverkehrs werden in den Abbildungen von Anhang 8.5.4 gezeigt.



nicht zielführende TLS-Konfiguration in Node-RED



Fragwürdige, aber zielführende TLS-Konfiguration in Node-RED

Abbildung 64: TLS-Konfiguration der MQTT-Nodes in Node-RED

Im nächsten Schritt der Node-Konfiguration wurden unter dem Reiter „Security“ der Username und das zugehörige Passwort zur Authentifizierung am Broker angegeben. Die Konfiguration war damit abgeschlossen und ist für alle MQTT-Nodes gültig. Im letzten Schritt wurde in jedem Node das entsprechende Topic für das Abonnieren bzw. Veröffentlichen der verschiedenen Signalwerte eingetragen. Die Topics für die in dieser Arbeit verwendeten Signale sind:

- /legacy/start_stop
- /legacy/hand_auto
- /legacy/alarm
- /legacy/qty
- /legacy/performance

Die beschriebene Konfiguration der MQTT-Nodes wurde für die im Rahmen der Programmierung des Raspberry Pi verwendeten und in ***Abbildung 62*** von Abschnitt 5.4.1 dargestellten MQTT-Nodes durchgeführt. Die MQTT-Nodes zur Erfassung der Maschinensignale am Laptop nutzen dieselbe Konfiguration, da die Verbindung auf die gleiche Weise und zum selben Broker hergestellt wird. Die Realisierung für die Erfassung der Maschinendaten vom MQTT-Broker und die anschließende Weitergabe an die Visualisierung in Node-RED am Laptop sind in ***Abbildung 65*** dargestellt.

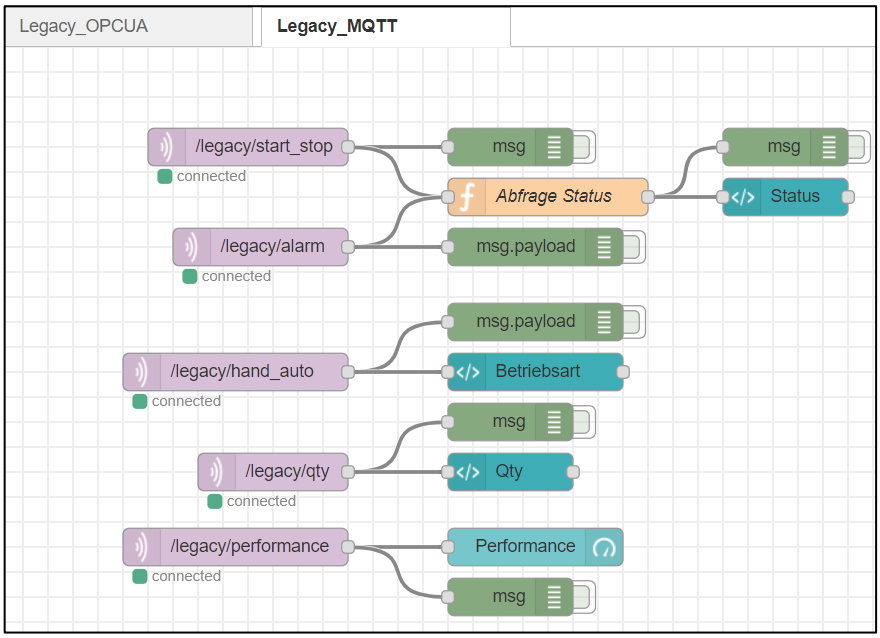


Abbildung 65: Erfassung der Maschinendaten vom MQTT-Broker und Weitergabe an die Visualisierung in Node-RED am Laptop

Innerhalb des Function-Node „Abfrage Status“ wurden je nach auftretenden Werten von Start, Stopp und Störung repräsentative Betriebszustände festgelegt. Die Abfrage und Definition der Maschinenstati ist in ***Abbildung 66*** dargestellt.

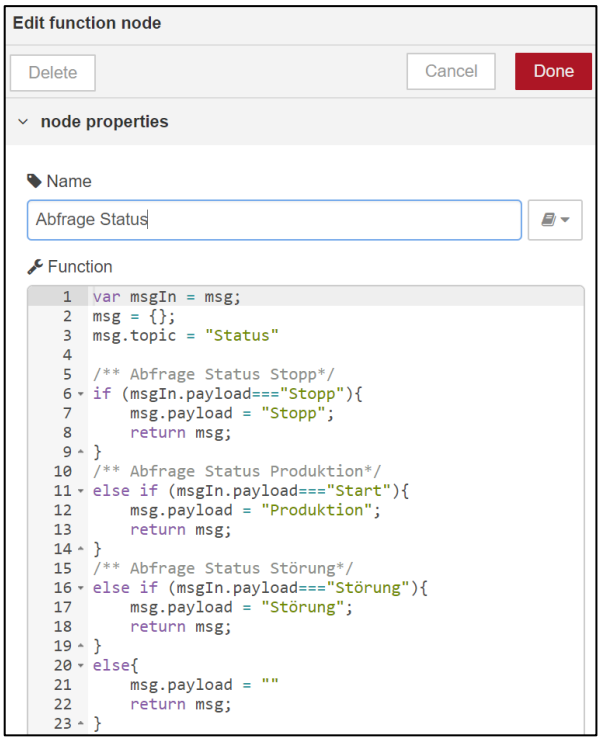


Abbildung 66: Abfrage und Definition der Betriebszustände der Maschine

Im letzten Schritt wurden die resultierenden Betriebsarten der Maschine sowie die Stückzahl und Leistung an die Dashboard-Nodes zur Visualisierung weitergereicht. Mit Hilfe des Template-Node aus der Gruppe der Dashboard-Nodes wurde die Visualisierung mittels HTML und Angular individuell angepasst. Die Anpassungen im Template-Node sind anhand der Maschinenbetriebszustände in ***Abbildung 67*** gezeigt. Für die Visualisierung der Leistung wurde ein Gauge-Node herangezogen.

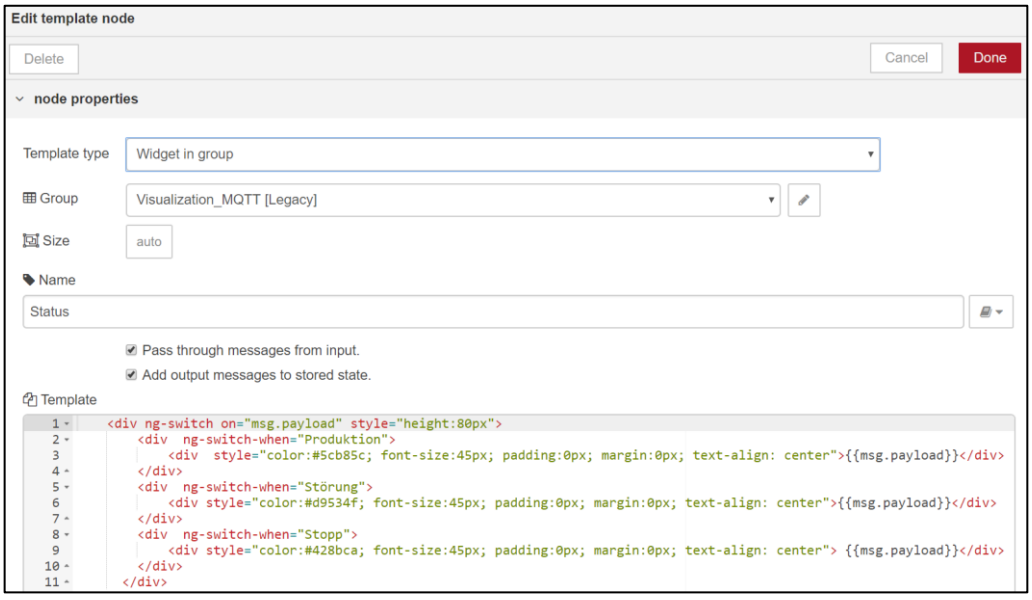


Abbildung 67: Anpassung der Visualisierung von den Maschinenbetriebszuständen im Template-Node

5.5.2 Kommunikationsumgebung für OPC UA

- KEPServerEX OPC UA Server

Der KEPServerEX OPC UA Server wurde direkt am Laptop betrieben. Zur Konfiguration des KEPServerEX wurde zu Beginn unter dem Punkt „Konnektivität“ ein neuer Kanal mit der Bezeichnung Legacy OPC UA vom Typ Siemens TCP/IP Ethernet angelegt. Im nächsten Schritt wurde dem Kanal die S7-1200 SPS mit der IP-Adresse 10.0.0.25 (NAT) als Gerät hinzugefügt. Für dieses Gerät wurden die Tag-Gruppe mit der Bezeichnung „MDE_Data“ erstellt und darin die Tags für die zu erfassenden Signale generiert. Die Adressen der Tags wurden entsprechend der Adresszuweisung im SPS-Programm eingetragen und die zugehörigen Datentypen ausgewählt. Die resultierende Struktur am OPC UA Server ist in ***Abbildung 68*** dargestellt.

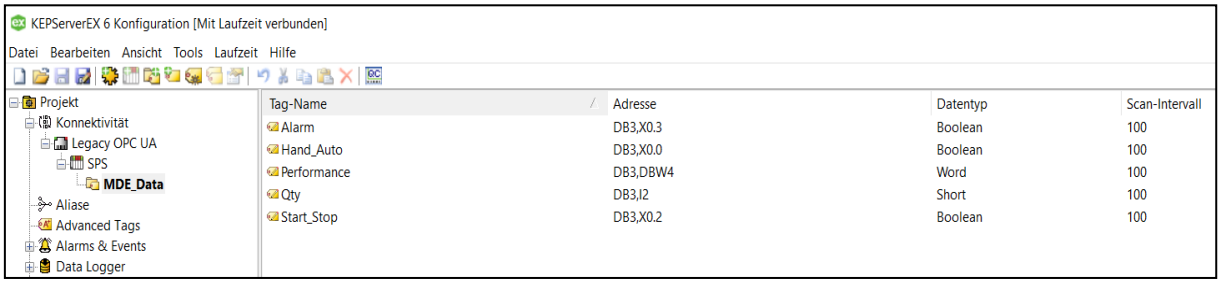


Abbildung 68: Struktur am OPC UA Server

Im OPC UA Konfigurationsmanager des Servers wurde für den verwendeten Endpoint als Sicherheitsrichtlinie „Basic256“ mit der Option „Signieren und Verschlüsseln“ eingestellt. Im Benutzermanager der Servereinstellungen wurde ein zusätzlicher User mit Passwort erstellt. Für diesen User könnte jegliche Aktion am Server mit den Security Policies in den Servereinstellungen festgelegt werden. Beispielsweise könnten die Zugriffsrechte für jeden einzelnen Tag im Server beliebig eingeschränkt werden. Für den Nachweis der Funktionalität wurde dafür testweise der lesende Zugriff des Clients und des angemeldeten Users für die Stückzahl gesperrt. Das Ergebnis ist für diesen Fall in ***Abbildung 76*** dargestellt und zeigt, dass die Stückzahl tatsächlich nicht gelesen und visualisiert werden konnte.

- OPC UA Client in Node-RED

Der OPC UA Client greift die Maschinendaten vom OPC UA Server ab und gibt diese zur Visualisierung weiter. Die Realisierung des OPC UA Clients am Laptop in Node-RED ist in ***Abbildung 69*** gezeigt.

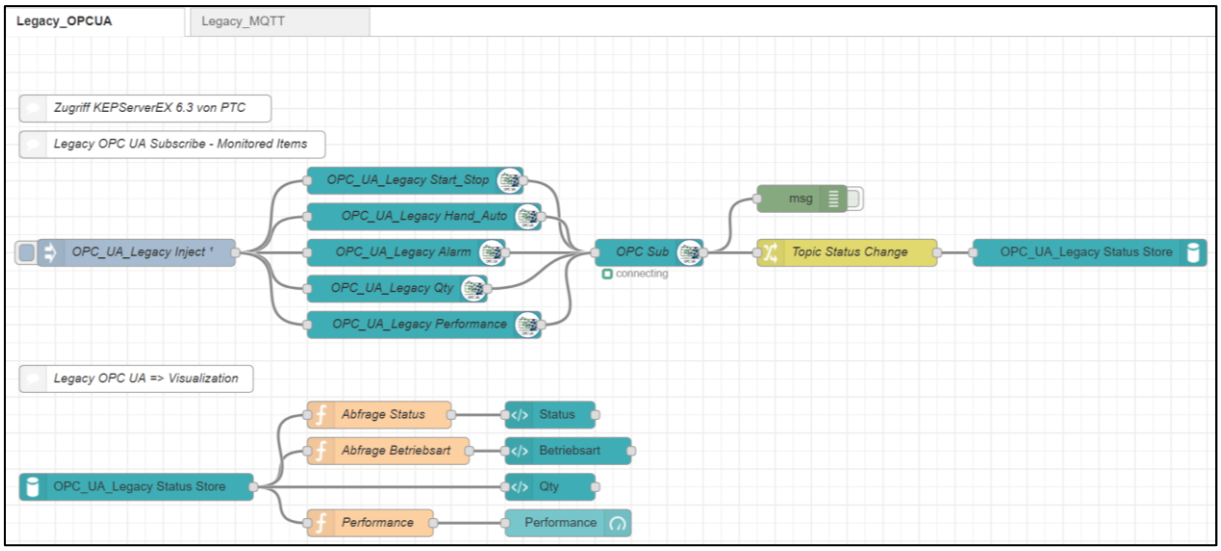
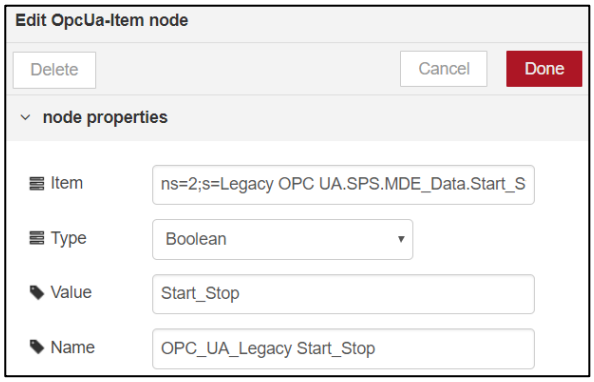
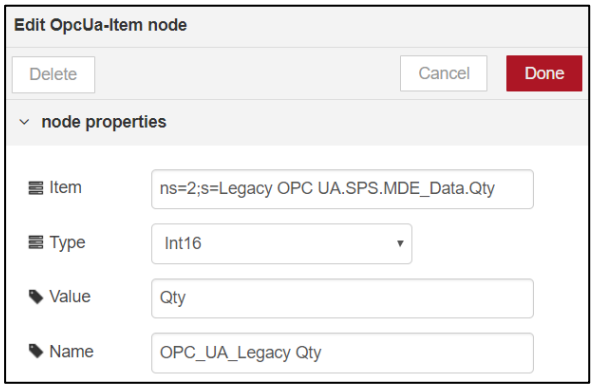


Abbildung 69: OPC UA Client zum Datenabgriff vom OPC UA Server und zur Weitergabe an die Visualisierung in Node-RED

Mit den fünf untereinander angeordneten OpcUa-Item-Nodes wurden die Tags des Servers angegeben. Die Items in den OpcUa-Item-Nodes werden durch Angabe des Namespace (ns=2) und dem Pfad der Items im Server definiert. Zusätzlich muss der Datentyp der Variable angegeben werden. Die Einstellungen der OpcUa-Item-Nodes sind beispielhaft anhand der Tags für Start/Stopp und der Stückzahl in ***Abbildung 70*** dargestellt. Die übrigen Items wurden in analoger Weise definiert.



OpcUa-Item für den Start_Stop-Tag



OpcUa-Item für den Qty-Tag

Abbildung 70: Definition der OpcUa-Item-Nodes in Node-RED für Start_Stop-Tag und Qty-Tag

Mit dem OpcUa-Client-Node wird eine Subscription auf die Items zur Erfassung von Wertänderungen realisiert. Der in ***Abbildung 69*** links oben befindliche Inject-Node hat die Aufgabe, nach einem Deployment in Node-RED die OPC UA Verbindung zu initiieren und somit die Subscription auf die Items zu aktivieren. Die Konfiguration des Client-Node ist in ***Abbildung 71*** dargestellt. Der Endpoint inklusive seiner Sicherheitsrichtlinien muss exakt mit einem der im Server konfigurierten Endpoints übereinstimmen. Der rechte Teil der Abbildung zeigt die Angaben für den Zugriff auf

den im KEPServerEX OPC UA Server konfigurierten Endpoint. Zusätzlich wurde eingestellt, dass der Serverzugriff ausschließlich mit dem am Server hinterlegten User „opcUA_DA“ erfolgen darf.

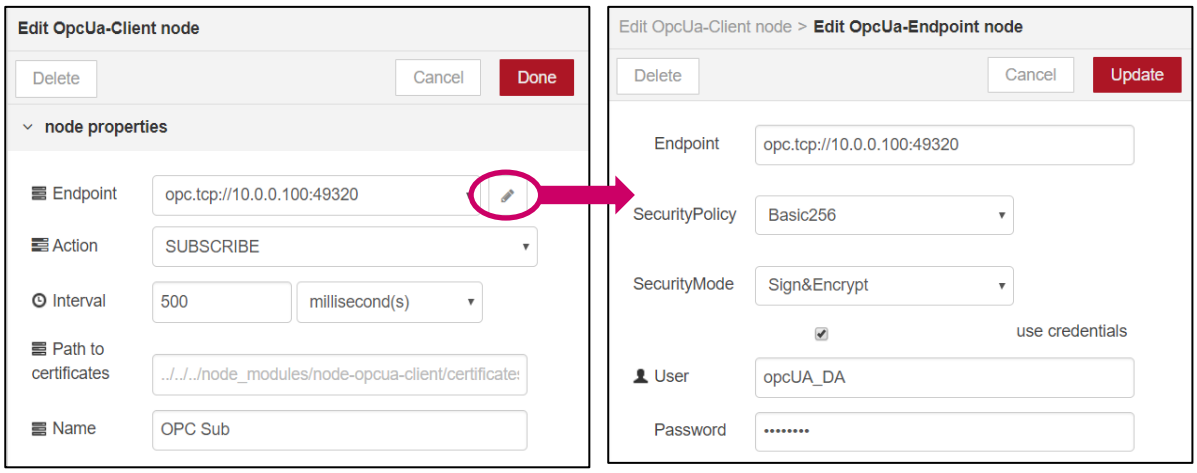


Abbildung 71: Konfiguration des OpcUa-Client-Node in Node-RED

Mit dem Change-Node wurde lediglich ein Mapping der OPC UA Item-Bezeichnungen durchgeführt. Im darauf folgenden Datawarehouse-Save-Node werden alle Items gespeichert und der gespeicherte Inhalt bei Wertänderungen eines oder mehrerer Items zur Gänze an den Datawarehouse-Listen-Node übergeben. Im Anschluss daran wurden innerhalb der Function-Nodes je nach auftretenden Item-Werten repräsentative Betriebszustände festgelegt. Die Abfrage und Definition wird anhand der Maschinenstati in ***Abbildung 72*** gezeigt. In analoger Weise wurde die Abfrage und Definition der Betriebsarten durchgeführt.

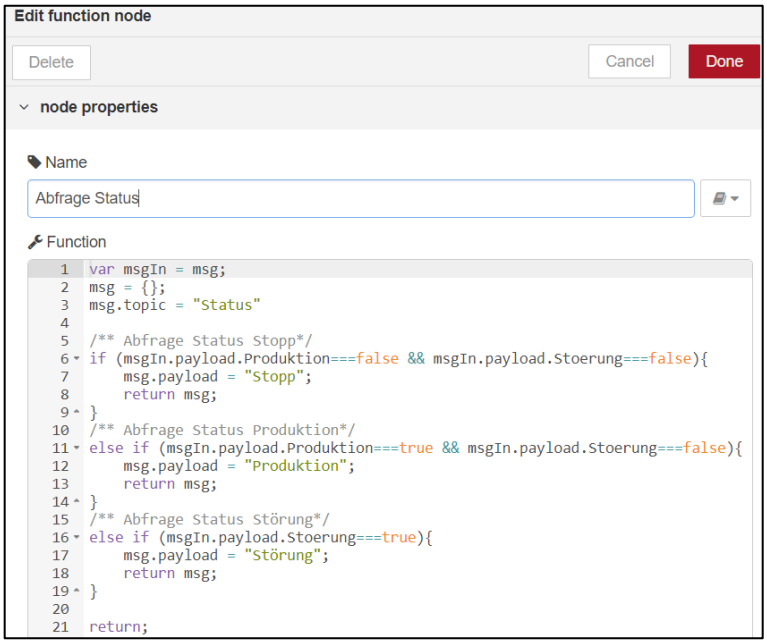


Abbildung 72: Abfrage und Definition der Betriebszustände der Maschine

Im letzten Schritt wurden die resultierenden Betriebszustände und Betriebsarten der Maschine sowie die Stückzahl und Leistung an die Dashboard-Nodes zur Visualisierung weitergereicht. Für eine idente Darstellung wurden die Anpassungen der Visualisierung im Template-Node auf die gleiche Weise wie in ***Abbildung 67*** von Abschnitt 5.5.1 durchgeführt.

- OPC UA Verbindungsaufbau

Im Rahmen des Aufbaus eines Secure Channels erfolgt der Austausch der Applikationsinstanz-Zertifikate zwischen Client und Server. Damit ein Secure Channel hergestellt werden kann, muss das Applikations-Zertifikat des Servers in die Trust-List des Clients verschoben werden und vice versa.

Der OPC UA Konfigurationsmanager des KEPServerEX OPC UA Servers bietet, wie in ***Abbildung 73*** dargestellt, eine Ansicht in der das Client-Zertifikat beim erstmaligen Verbindungsversuch erscheint und dann als vertrauenswürdig eingestuft werden kann. Danach muss zur Herstellung der Verbindung ein weiterer Verbindungsaufbau initiiert werden. Eine andere Möglichkeit wäre, das Client-Zertifikat bereits vor dem ersten Verbindungsaufbau manuell zu importieren und als vertrauenswürdig einzustufen. Die Notwendigkeit eines zweiten Verbindungsaufbaus würde hierbei entfallen.

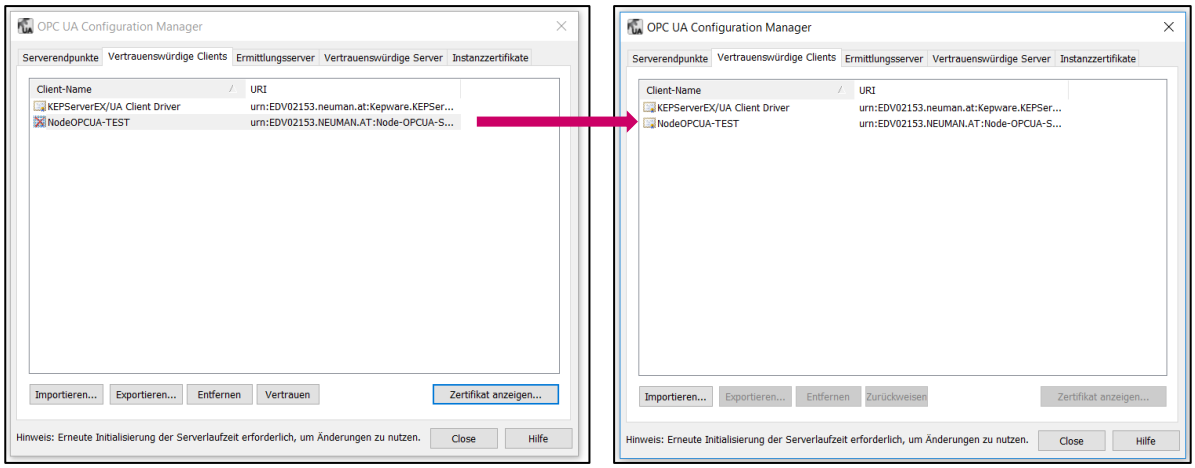


Abbildung 73: Client-Zertifikat am Server als vertrauenswürdig einstufen

Das Server-Zertifikat wurde manuell in das Unterverzeichnis des Node-RED OPC UA Clients für vertrauenswürdige Zertifikate eingefügt. Jedoch wurde auch hier ein Fehler in Node-RED beobachtet. Beim Verbindungsaufbau wurde außerhalb des Node-RED Verzeichnisses eine verschachtelte Ordnerstruktur erzeugt, in der das Applikationsinstanz-Zertifikat des Servers abgelegt wurde. Es schien als ob die Zertifikate im dafür vorgesehenen Verzeichnis des Node-RED OPC UA Clients nicht abgerufen wurden. Sobald das Zertifikat aus der automatisch erzeugten Ordnerstruktur entfernt wurde, wurde beim erneuten Verbindungsaufbau wieder ein Applikations-Zertifikat des Servers mit aktuellem Zeitstempel erzeugt und darin

abgelegt. Dennoch konnte die Verbindung erfolgreich hergestellt werden. Eine Lösung zur Behebung des Fehlers konnte nicht gefunden werden. Eine darüber hinausgehende Beeinträchtigung der Funktionalität während der Datenübertragung und Verschlüsselung konnte aber nicht festgestellt werden. Zur Überprüfung wurde der Netzwerkverkehr mit Wireshark analysiert. Das Ergebnis der Analyse des Netzwerkverkehrs ist in der Abbildung von Anhang 8.5.5 gezeigt.

5.6 Visualisierung der Maschinendaten

5.6.1 Struktur des Dashboards in Node-RED

Das Dashboard zur Visualisierung der Maschinendaten beider Erfassungssysteme wurde unter einem gemeinsamen Tab „Legacy“ angelegt. Dieser Tab beinhaltet zwei nebeneinander angeordnete und gleich strukturierte Bereiche. Der linke Bereich „Visualization_OPCUA“ stellt die per OPC UA übertragenen Maschinendaten dar, der rechte Bereich „Visualization_MQTT“ jene per MQTT. Die in Node-RED erstellte Struktur des Dashboards ist in ***Abbildung 74*** dargestellt.

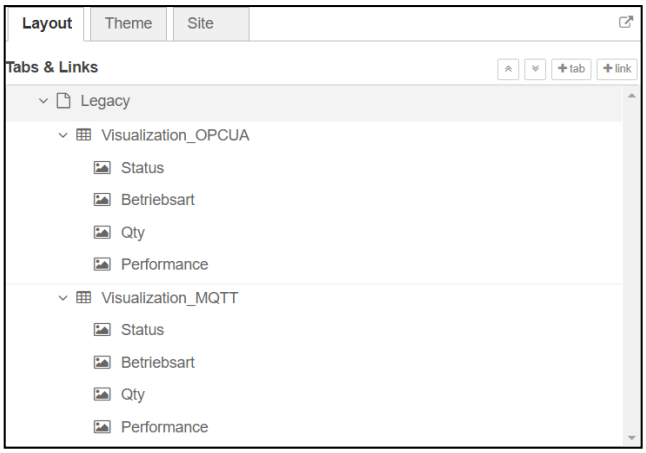


Abbildung 74: Struktur des Dashboards in Node-RED

Das Dashboard kann entweder durch Klick auf das rechts oben angeordnete Icon direkt in der Programmierumgebung von Node-RED oder per Aufruf der zugehörigen lokalen URL[16] im Browser geöffnet werden. Letzteres erfordert ebenfalls eine Authentifizierung.

5.6.2 Dashboard mit visualisierten Maschinendaten

Abbildung 75 zeigt Ausschnitte der erfassten und visualisierten Maschinendaten. Die kleineren Abweichungen bei den Leistungen können auf die unterschiedliche Umsetzung der Berechnung zurückgeführt werden.

[16] http://10.0.0.100:1880/ui/#

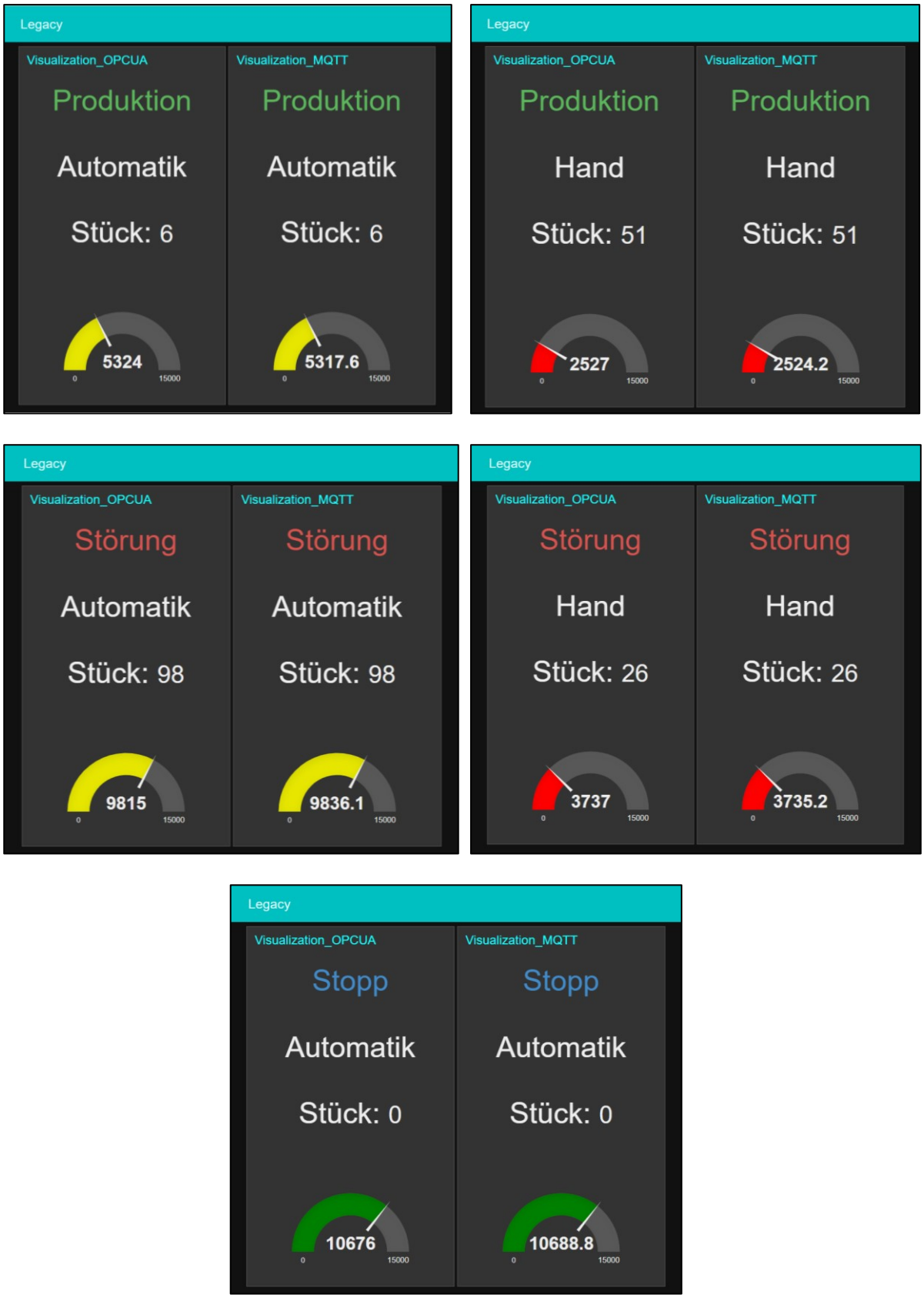


Abbildung 75: Ergebnisse der Live-Anzeige von den Maschinendaten

Abbildung 76 zeigt, dass lesende Zugriffe des Clients und des angemeldeten Users für die Stückzahl am OPC UA Server in den Security Policies gesperrt werden können. Derartige Sperren könnten individuell für alle Tags am Server und je nach User eingestellt werden.

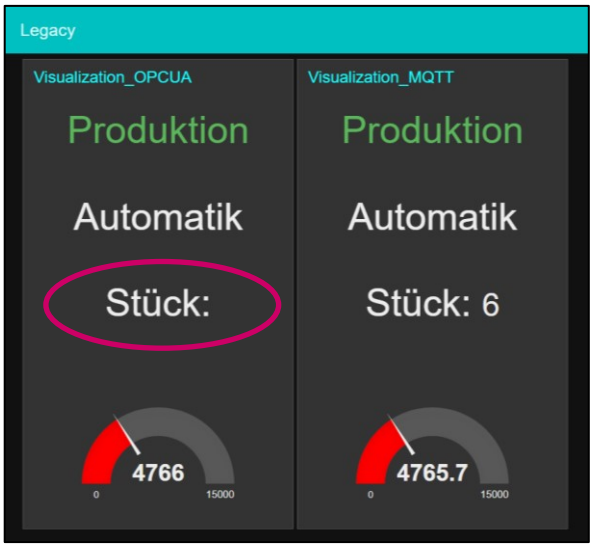


Abbildung 76: Lesezugriff auf die Stückzahl für den authentifizierten User am OPC UA Server gesperrt

6 Conclusio

Das Studium der Spezifikationen von OPC UA hat gezeigt, dass OPC UA im Grunde kein Protokoll, sondern eine komplette Kommunikationsumgebung beschreibt. OPC UA ist aufgrund des schichtweisen Aufbaus sehr flexibel und wandlungsfähig und steht für eine sichere, zuverlässige und plattformunabhängige Interoperabilität. In OPC UA besteht die Möglichkeit, neu veröffentlichte Verschlüsselungs- oder Sicherheitsmechanismen und Transportprotokolle (Mappings) nachträglich zu integrieren, wodurch ein zukunftssicherer Einsatz gewährleistet werden kann. Eine weitere Stärke von OPC UA besteht darin, komplexe Informationsmodelle nutzen zu können um Daten inklusive zugehöriger Metadaten und Semantik abzubilden. In Kombination mit der Skalierbarkeit von OPC UA kann sowohl horizontaler als auch vertikaler Daten- und Informationsaustausch im Unternehmen realisiert werden. Dies erfolgt unter Einhaltung notwendiger Sicherheitsziele wie Authentifizierung, Autorisierung, Vertraulichkeit, Integrität, Verfügbarkeit und Nachvollziehbarkeit.

Im Gegensatz dazu beschränkt sich MQTT auf eine zuverlässige Datenübertragung ohne tiefergreifende Security-Mechanismen. Zur Erhöhung der Sicherheit kann MQTT um TLS erweitert werden. Der schlanke Aufbau des Protokolls eignet sich hervorragend zur M2M-Kommunikation und speziell für asynchrone one-to-many, many-to-one und many-to-many Kommunikationsstrukturen.

Die fortschreitende Digitalisierung und der rapide ansteigende Vernetzungsgrad in der Industrie erfordern die Einführung eines Sicherheitskonzeptes in industriellen Automatisierungs- und Kontrollsystemen (IACS). Obwohl es keine Standardlösung für ein derartiges Sicherheitskonzept gibt und die Norm IEC 62443 noch nicht vollständig verabschiedet wurde, liefert sie trotz allem bewährte Vorgehensweisen anhand derer ein Leitfaden zur Umsetzung eines effizienten Security-Konzeptes erarbeitet wurde. Das Hauptaugenmerk lag dabei auf den Rollen des Betreibers und des Integrators. Darüber hinaus wurde für ein spezielles Szenario ein Lösungsvorschlag für das Design eines segmentierten Netzwerks gemäß IEC 62443 erstellt.

Für die Anbindung und zur Erfassung der Maschinendaten von Legacy Systems wurden zwei eigenständige Systeme ausgewählt und vorgestellt. Das erste System besteht aus einem Raspberry Pi und der UniPi v1.1 Erweiterungsplatine, um die in der Industrie üblichen Gleichspannungssignale verarbeiten zu können. Das zweite System ist eine konventionelle S7-1200 SPS von Siemens.

Mit Hilfe der Testumgebung wurden alle in der Arbeit beschriebenen Themen in einem gemeinsamen Kontext betrachtet und evaluiert. Die Evaluierung hat gezeigt, dass sich grundsätzlich beide betrachteten Systeme zur Anbindung von Legacy Systems mit den unterschiedlichen Kommunikationsmöglichkeiten eignen.

7 Schlussfolgerungen / Resümee / Ausblick

Das System bestehend aus Raspberry Pi und UniPi v1.1 Erweiterungsplatine besticht durch seine hohe Flexibilität. Es kann jedoch aufgrund ausständiger Langzeittests keine Aussage über die Lebensdauer dieses Systems in einer realen Industrieumgebung getätigt werden. Für eine dauerhafte und über einen längeren Zeitraum geplante Anbindung eines Legacy Systems in rauen Umgebungen ist deshalb das System bestehend aus der konventionellen S7-1200 SPS vorzuziehen.

Eine ähnliche Problematik betrifft den Einsatz der Open Source Software Node-RED. Obwohl trotz der Probleme bei der Konfiguration der Kommunikationsumgebung mit OPC UA und MQTT mit TLS in Node-RED keine Beeinträchtigung der Funktionalität festgestellt werden konnte, drängt sich die Frage auf, ob die Software für den Einsatz im industriellen Umfeld geeignet ist. Darüber hinaus müssten zur weiteren Beurteilung der Einsatzfähigkeit Langzeittests hinsichtlich Performanz und Stabilität für größere Implementierungen durchgeführt werden. Für überschaubare und unkritische Anwendungsfälle kann über die angesprochenen Probleme dennoch hinweg gesehen werden und eine sehr rasche und komfortable Umsetzung mit Node-RED erfolgen.

OPC UA und MQTT mit TLS lassen sich bestens zur sicheren Übertragung von Maschinendaten im betrachteten Ausmaß der Arbeit einsetzen. Während der praktischen Umsetzung hat sich jedoch gezeigt, dass einige Möglichkeiten, die OPC UA in der Theorie bietet, noch nicht in den erhältlichen Tools wie beispielsweise dem KEPServerEX OPC UA Server integriert sind. Der Server bietet etwa ausschließlich UA TCP als Transportprotokoll, keine Basic256Sha256 Verschlüsselung, keine Abbildungsmöglichkeit von Metadaten wie beispielsweise SI-Einheiten und keine Unterstützung der Redundanzkonzepte. Der Umstand von fehlenden Features betrifft aber nicht nur den KEPServerEX OPC UA Server sondern auch vergleichbare Server.

Der Trend am Markt zeigt, dass bereits in den einzelnen Komponenten OPC UA Server integriert sind. Zukünftig sollen diese ein vom Hersteller zur Verfügung gestelltes standardisiertes Informationsmodell beinhalten. Anstatt eines zentralen OPC UA Servers wird dadurch eine Dezentralisierung stattfinden und als Folge davon Aggregating-Server-Strukturen entstehen lassen. Die Kombination der zurzeit stattfindenden Integration des PubSub-Konzeptes in OPC UA, beispielsweise mit MQTT als Transportprotokoll, und der voranschreitenden Weiterentwicklung der TSN-Technologie sollen die Erfüllung der Echtzeitanforderungen in der Industrie ermöglichen. Dadurch könnte die Integrationsfähigkeit von OPC UA weiter erhöht und die Standardisierung mit OPC UA auf weitere Bereiche ausgedehnt werden. Mit Hilfe dieser Technologie könnte es möglich sein, Strukturen von Automatisierungssystemen zu revolutionieren und bis zu diesem Zeitpunkt noch nie dagewesene Strukturen zu realisieren.

8 Anhang

8.1 Anhang Kapitel 1

Kein Anhang

8.2 Anhang Kapitel 2

8.2.1 Abschnitt 2.1.1 – Einleitung und Überblick

- Inhalte der Spezifikationen

Core Spezifikationen

Part 1 (IEC/TR 62541-1) – gibt einen ersten Überblick.

Part 2 (IEC/TR 62541-2) – beschreibt die möglichen Bedrohungen und das Sicherheitsmodell für die Absicherung der Server/Client Interaktionen in OPC UA.

Part 3 (IEC 62541-3) – zeigt den Aufbau des Metamodells und dessen Objekte, mit welchem der Server einen Client modellierte Prozesse in standardisierter Form bereitstellen kann. Auf diesem Modell setzen alle weiteren Informationsmodelle auf – Schlüsseldokument für das Design und der Entwicklung von OPC UA Applikationen [2].

Part 4 (IEC 62541-4) – definiert (abstrakte) Services, die der Server dem Client für Interaktionen zur Verfügung stellen kann. Es gibt insgesamt 39 unterschiedliche Services - Schlüsseldokument für das Design und der Entwicklung von OPC UA Applikationen [2].

Part 5 (IEC 62541-5) – beschreibt standardisierte Nodes zur Informationsrepräsentation für das Modell aus Part 3. Inhalt sind dabei Definitionen von Einstiegspunkten in den Server AddressSpace, Objekte und Datentypen sowie die Navigation durch den Server AddressSpace.

Part 6 (IEC 62541-6) – definiert wie UA Services, Sicherheitsmechanismen und ein von UA unterstützter Transportmechanismus den Nachrichten zugeordnet werden – für Entwickler von UA Stacks von Interesse [2].

Part 7 (IEC 62541-7) – zeigt unterschiedliche Profile, welche jeweils nur eine bestimmte Gruppe von OPC UA Features (Services und eine Variation des Informationsmodells) nutzen. Eine UA Applikation muss diese Gruppe an Features komplett implementiert haben, um Interoperabilität gewährleisten zu können. Die Gruppen sind auf zwei Ebenen definiert, den Conformance Units (kleines Set an

Funktionalität) und den Profiles, welche aus mehreren Conformance Units bestehen. Während dem Verbindungsaufbau wird die Liste an unterstützten Profiles zwischen Server und Client ausgetauscht. Damit kann die jeweilige Applikation bestimmen ob der Kommunikationspartner die nötigen Features unterstützt.

Part 14 – definiert anstelle des Server/Client Modells ein OPC UA Publish/Subscribe-Modell. Beinhaltet ist eine Einführung in das Publish/Subscribe-Konzept, eine Definition der Kommunikationsparameter und ein PubSub Konfigurationsinformationsmodell sowie Mapping-Möglichkeiten.

Access Type Spezifikationen

Part 8 (IEC 62541-8) – zeigt die Definition des Informationsmodells in Verbindung mit Data Access (OPC DA).

Part 9 (IEC 62541-9) – beschreibt das Informationsmodell für Alarm und Conditions und Typen von Events.

Part 10 (IEC 62541-10) – nimmt Bezug auf die Informationsmodelldefinition hinsichtlich Programs. Ein Program modelliert eine komplexe und zustandsorientierte Funktionalität im System.

Part 11 (IEC 62541-10) – zeigt die Definition des Informationsmodells in Verbindung mit Historical Data Access (OPC HDA). Ebenso sind Funktionalitäten für die Berechnung und Rückgabe von Aggregat-Daten, wie beispielsweise Minimum, Maximum, Durchschnitt etc. erklärt.

Utility Spezifikationen

Part 12 – zeigt wie Server im Netzwerk gefunden werden können und wie ein Client die für den Verbindungsaufbau notwendigen Informationen von einem bestimmten Server erhält.

Part 13 (IEC 62541-13) – beschreibt das Informationsmodell und die Arbeitsweise von Aggregates.

8.2.2 Abschnitt 2.1.4 – OPC UA Informationsmodelle

- Der Weg von Daten zum Informationsmodell und zum AddressSpace des Servers

Schritt 1 – Identifizieren von Daten

Im ersten Schritt müssen die Daten, die später im Server abgebildet werden sollen für das gewünschte Einsatzszenario identifiziert werden. Anschließend wird jeder identifizierte Eintrag charakterisiert und die Relationen zwischen den Einträgen bestimmt. Alle Geräte einer Maschine/Anlage stellen Objekte im Modell dar, wobei

jedes Objekt wiederum aus mehreren Objekten, zur detaillierteren Beschreibung, besteht. Diese entstandene Objekthierarchie repräsentiert das Informationsmodell.

Das Informationsmodell muss am Ende als XML-File (OPC UA XML-Schema) vorliegen.

Schritt 2 – Erstellung des User NodeSet XML-File

In diesem Schritt werden alle Objekte, Variablen und Referenzen des Informationsmodells in eine Liste von Nodes, welche die Basis des resultierenden AddressSpace darstellen, umgewandelt. In diesem Vorgang werden die Attribute der Nodes definiert, jedem Knoten ein Identifier zugeordnet und die Referenzen vom Quell- zum Zielknoten erstellt [1]. Ein Compiler der aus dem XML-File des Modells ein User NodeSet XML-File erstellt ist unter https://github.com/OPCFoundation/UA-ModelCompiler zu finden.

Schritt 3 – Erstellung des Application NodeSet XML-File

Üblicherweise enthält das Modell keine Standard UA Nodes, welche aber in jeder OPC UA Serverapplikation enthalten sein müssen. Deshalb muss das User NodeSet, um diese Nodes erweitert werden [1]. Resultat dieser Vereinigung ist das Application NodeSet XML-File.

Schritt 4 – AddressSpace erstellen oder in kommerziellen Server importieren

Die erste Möglichkeit ist aus dem Application NodeSet XML-File nun den AddressSpace Code für das Zielsystem (C, C++, C#) zu erzeugen und im Server-Projekt des Zielsystems zu kompilieren. Dem OPC UA Source Code Stack wird auf diese Weise die Definition des AddressSpace zur Verfügung gestellt [1]. Wird hingegen ein kommerzieller Server eingesetzt, muss das Application NodeSet, entsprechend der Server-Software, in ein kompatibles Format gebracht werden [1].

- Werkzeuge zur Modellierung in OPC UA

UaModeler von Unified Automation (am häufigsten verwendet)

Der UaModeler bietet die Möglichkeit den AddressSpace in grafischer Form zu erzeugen. Das Modell kann in hierarchischer als auch in grafischer Form dargestellt werden. Die grafische Darstellungsweise basiert auf der Notation und Syntax von OPC UA. Der UaModeler kann XML-Files im UaNodeSet Format importieren und exportieren. Der Import kann dafür verwendet werden, um verfügbare Modelle (Companion Specifications oder andere Modelle) in den Designprozess aufzunehmen. Das Tool erlaubt den Source Code für die Implementierung im Server zu generieren, wobei der erzeugte Code natürlich auf die SDK von Unified Automation abgestimmt

ist. Es können Source Codes für C++, ANSI C und .NET basierende OPC UA SDK generiert werden.

Die Evaluierungsversion der Software beschränkt den Export eines XML-NodeSets auf zehn Knoten. Die Vollversion kann um 1200€ erworben werden. Beim Kauf einer SDK ist meist eine Lizenz für den UaModeler enthalten.

Weitere Tools

- uaModeldesigner von Softing (seit 2017 erhältlich)
- OPC UA Address Space Model Designer von CommServer – hier ist nicht ganz klar ob das Tool noch weiterentwickelt wird
- OPC UA Modeler von Fraunhofer IOSB

8.2.3 Abschnitt 2.1.5 – Der OPC UA Server

- Services in OPC UA

Service Set	Services
Discovery Service Set	• FindServers • FindServersOnNetwork • GetEndpoints • RegisterServer • RegisterServer2 (wenn möglich, ist dieses Service dem RegisterServer Service vorzuziehen)
SecureChannel Service Set	• OpenSecureChannel • CloseSecureChannel
Session Service Set	• CreateSession • ActivateSession • CloseSession • Cancel
NodeManagement Service Set	• AddNodes • AddReferences • DeleteNodes • DeleteReferences
View Service Set	• Browse • BrowseNext • TranslateBrowsePathToNodeIds • RegisterNodes • UnregisterNodes
Query Service Set	• QueryFirst • QueryNext
Attribute Service Set	• Read • HistoryRead • Write • HistoryUpdate

Service Set	Services
Method Service Set	• Call
MonitoredItem Service Set	• CreateMonitoredItems • ModifyMonitoredItems • SetMonitoringMode • SetTriggering • DeleteMonitoredItems
Subscription Service Set	• CreateSubscription • ModifySubscription • SetPublishingMode • Publish • Republish • TransferSubscriptions • DeleteSubscriptions

Tabelle 11: Services in OPC UA [8]

8.2.4 Abschnitt 2.1.7 - Security-Konzepte in OPC UA

- Sicherheitshinweise für Entwicklung und Implementierung von OPC UA Applikationen

Nachfolgende Punkte wurden Unterpunkt 6 aus [14] entnommen und sollen eine kurze Zusammenfassung der wichtigsten Sicherheitshinweise geben:

- Auswahl geeigneter Timeouts auf jeder Verbindungsebene.
- Strikte Überprüfung der Nachrichten auf valides Format laut Spezifikation. Für invalide Nachrichten soll die Ausgabe von entsprechenden Fehlercodes, welche in Teil 4 der Spezifikation zu finden sind oder ein Verwerfen der Nachricht erfolgen.
- Kryptografische Schlüssel dürfen nicht während des ersten Bootvorgangs generiert werden. Kryptografische Funktionen beispielsweise zur Zertifikaterstellung müssen ausreichende Entropie aufweisen.
- Das Management des Zertifikatstores sollte ausschließlich einem Administrator unterliegen.
- Schlüssellängen kleiner 2048 gelten bereits als unsicher.
- Eine saubere Implementierung des in OPC UA vorhandenen Alarm- und Condition-Modells ist empfohlen.
- Programs, die Teil eines Servers sind sollen auf jeden Fall auf Benutzer mit entsprechenden Zugriffsrechten beschränkt werden. Des Weiteren darf auf keinen Fall eine unlimitierte Anzahl von Aufrufen von Programs erlaubt werden.

8.3 Anhang Kapitel 3

8.3.1 Abschnitt 3.2 – Normenreihe IEC 62443

- Struktur und Inhalte der IEC 62443

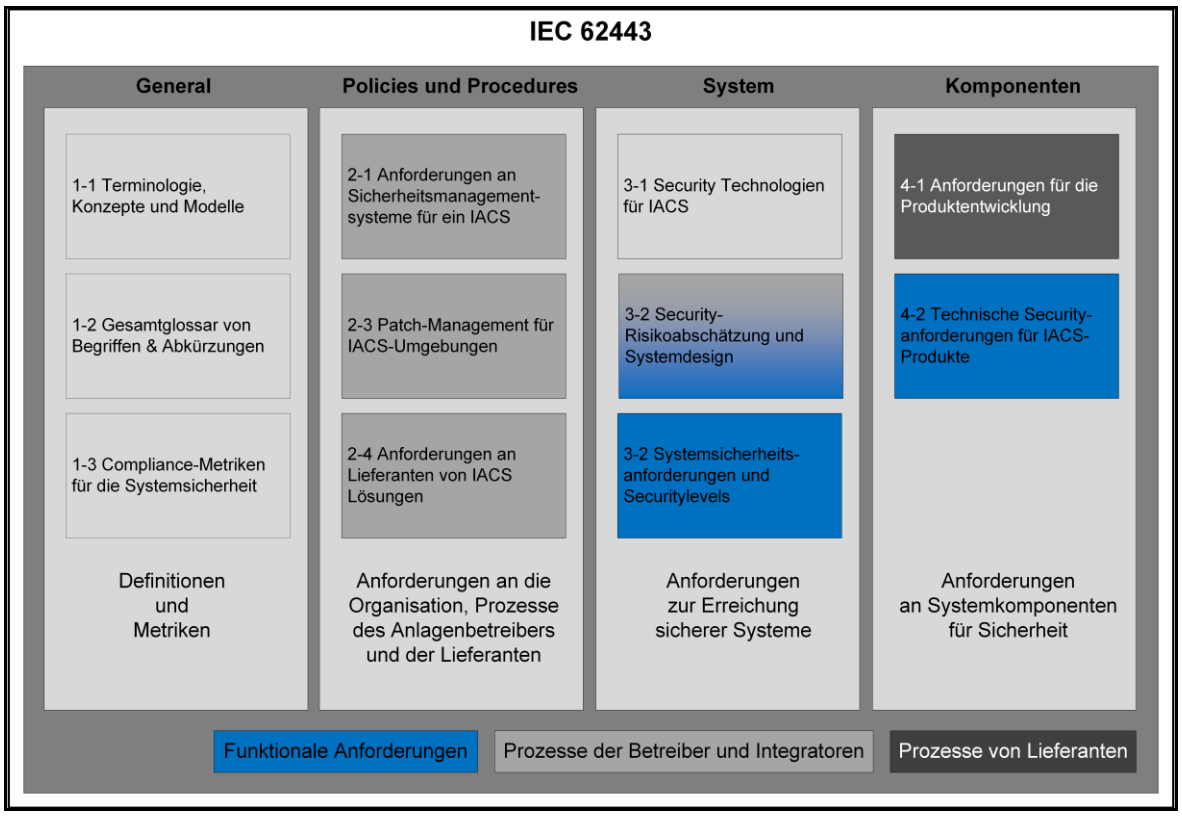


Abbildung 77: Struktur und Inhalte der IEC 62443 [29]

Die Norm-Reihe IEC 62443 besteht aus vier Hauptbereichen, welche sich in weitere Teilbereiche gliedern.

- **General** – in IEC 62443-1-1 sind zunächst Begriffe, Konzepte und Modelle definiert. In IEC 62443-1-2 sind alle Begriffe, die in den Normen verwendet werden, zusammengefasst. IEC 62443-1-3 definiert Metriken zur Bewertung der IT-Sicherheit. Der letzte Teil IEC 62443-1-4 erklärt den Sicherheitslebenszyklus und entsprechende Anwendungsfälle.
- **Policies und Procedures** – beschreibt das IT-Sicherheits-Management-System und definiert somit die Organisation der IT-Sicherheit. Darüber hinaus sind Implementierungshilfen aufgezeigt. IEC 62443-2-1 beschreibt Anforderungen an ein IT-Sicherheits-Managementsystem. IEC 62443-2-2 gibt Aufschluss darüber, wie und in welchen Bereichen diese Procedures umzusetzen sind. IEC 62443-2-3 widmet sich dem bedeutsamen Patch-Management. Zuletzt befasst sich IEC 62443-2-4 mit dem Einsatz von Dienstleistern für Inbetriebnahme und Service.

- **System** – Beschreibung der technischen Vorgehensweise. IEC 62443-3-1 beschreibt zunächst grundlegende Technologien wie z.B. Authentifizierung, Verschlüsselung etc. IEC 62443-3-2 beschreibt die Segmentierung von Anlagen. Schlagworte dieses Konzeptes sind Zonen und Conduits. IEC 62443-3-3 definiert eine Metrik um eine Bewertung des erreichten Sicherungszustandes zu ermöglichen.
- **Komponenten** – richtet sich an die Hersteller von Automatisierungssystemen und legt die Anforderungen an Komponenten und den zugehörigen Entwicklungszyklus fest. IEC 62443-4-1 definiert den Entwicklungsprozess von Komponenten, die in der Automatisierungstechnik zum Einsatz kommen sollen. IEC 62443-4-2 beschreibt die technischen Anforderungen für die Komponenten von Automatisierungssystemen, Applikationen und Funktionen.

Die Norm ist bisher nur zum Teil veröffentlicht. Alle Teile der Norm liegen jedoch zumindest als Entwurf vor. Der aktuelle Stand der Arbeiten und der Freigabestatus der Normteile kann unter http://isa99.isa.org eingesehen werden. Nachfolgende Abbildung zeigt welche Teile der Norm für die Rollen des Betreibers, Integrators und Herstellers von größter Bedeutung sind.

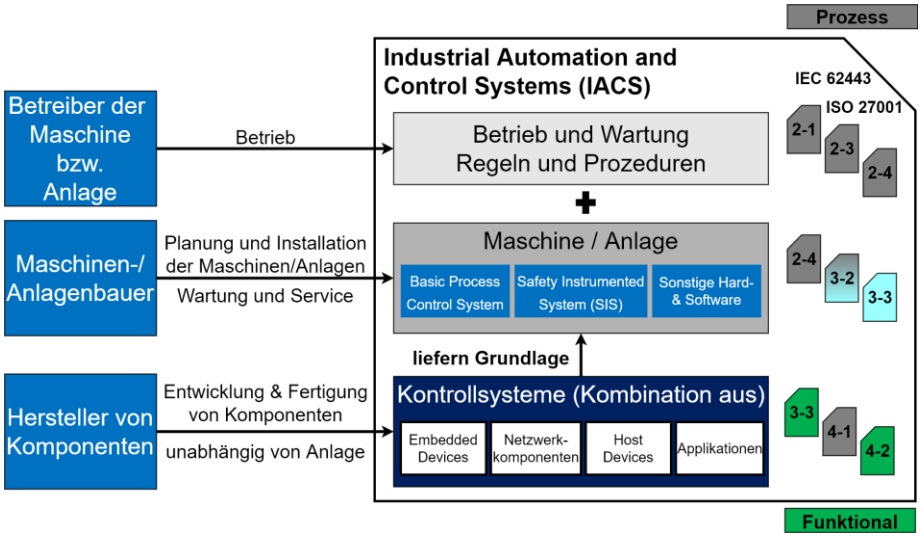


Abbildung 78: Zuordnung der Teile der Norm zu den jeweiligen Rollen [35]

8.3.2 Abschnitt 3.4.3 – Schritte zur Umsetzung der IEC 62443

■ Strukturanalyse durchführen

Komponenten
Netzwerkkomponenten, wie Hubs, Switches, Router etc. und auch die Verkabelung selbst
Komponenten für Netzwerkschutz, wie Firewalls, Intrusion Detectionsystems etc.
Speicherprogrammierbare Steuerungen (SPS) und Feldgeräte wie Sensoren, Aktoren etc.
Sicherheitssysteme und Zubehör an den Maschinenanlagen
HMI Stationen, Programmiergeräte, Computer am Shopfloor und alle CPU-basierenden Systeme, welche für die Aufrechterhaltung der Produktion essentiell sind.
Geräte zur Handhabung von Material und Tracking Systeme
Peripheriegeräte, wie Barcode Scanner etc.
Inspektionssysteme, Messsysteme und Analysegeräte, wie Kamerasysteme, Frequenzprüfanlagen, etc.
Prozessleitsysteme
SCADA, MES und Prozessinformationssysteme
Netzwerkkomponenten, wie diverse Server (Datenbankserver, OPC UA Server, Backup Server, FTP Server usw.) und virtuelle Maschinen (VM)
Eingesetzte Betriebssysteme, Firmware, Software
Kommunikationssysteme für die Kommunikation zu Remote-Geräten
Kommunikationsprotokolle
Modellierungssysteme für die industrielle Automatisierung
Dokument- und Dateiverwaltungssysteme
Dokumente und Dateien wie, Konfigurationsdaten, Rezepturen, Produktionsdaten, Prozessdaten, Qualitätsdaten, Kundenlisten, Businesspläne, F&E Daten, Usernamen, Passwörter etc.
Personen mit speziellen Fähigkeiten und spezifischem Know-How
Energiemanagementsysteme
Air Conditioning-, Heiz- und Belüftungsregelungsanlagen

Tabelle 12: Vorschläge für die Strukturanalyse

Industrial automation and control system network characterization

Business _____
Site _____
Operating unit _____
Site IT contact _____ Phone # _____
Site process control contact _____ Phone # _____
Last updated _____

PLEASE ANSWER THE FOLLOWING QUESTIONS :

_____ **Are manufacturing and control systems currently interfaced to site or corporate LANs?**
_____ **Are manufacturing and control systems remotely accessed from outside the IACS domain?**

Process control domain

_____ Total number of IP addressable nodes
_____ Number of IP addressable nodes to be accessed from outside process control domain
_____ Number of concurrent users inside IACS domain
_____ Number of concurrent users inside IACS domain requiring access to external resources
_____ Number of total users outside IACS domain requiring access to process control resources
_____ Number of concurrent users outside IACS domain requiring access to process control resources
IP addressing (check all that apply)
_____ DHCP _____ Public addresses used (i.e. x.x.x.x)
_____ Static _____ Private addresses used (192.168.x.x)

Control platforms

_____ Number of control platforms
_____ Control platform type (PLC, DCS, PC)
Control platform vendor(s) _____ _____
Control platform model(s) _____ _____

Operator consoles and HMI devices

_____ Number of operator consoles
Operator console vendor(s) _____ _____
Operator console model(s) _____ _____
Operator console operating system(s) _____ _____

Application nodes (check all that apply)

_____ Proccess management and control server
_____ SCADA
_____ OPC server
_____ Engineering workstation
_____ Batch server
_____ Other _____

Network security barriers in-use

Type (firewalls, routers, VLANS, etc.) _____

Anticipated network security support (check all that apply)

_____ Site resources
_____ External (3rd party)

Site network (answer yes / no)

_____ Current site network topology diagrams available and up-to-date?
_____ Are process control nodes on isolated LAN segment?
_____ Site information security policy in place?
_____ Security office audit completed (if yes, date completed _____)
_____ Does site use two-factor authentication?
_____ Security office risk assessment completed (if yes, date completed _____)

Remote access requirements (check all that apply)

_____ Via site / corporate LAN
_____ Via dial-up modem
_____ Via internet
_____ Via local dial-up modem directly tied to manufacturing and control node(s)

Local egress requirements (check all that apply)

_____ To site applications and resources (document management systems, quality systems, business systems)
_____ To corporate applications and resources (document management systems, quality systems, business syste
_____ To internet sites

Abbildung 79: Beispiel eines Datenerhebungsformulars für IACS [33]

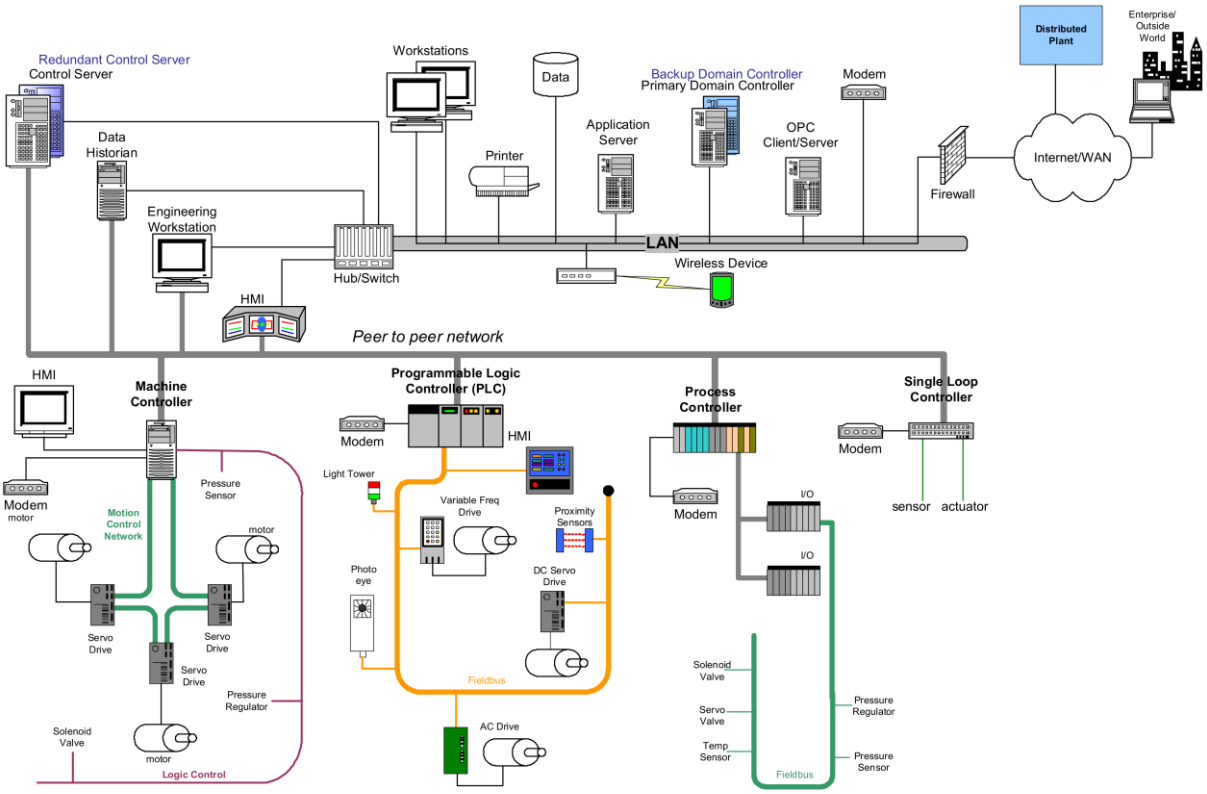


Abbildung 80: Beispiel eines Netzwerkdiagramms [33]

Schwachstellen und Bedrohungen analysieren

Bedrohungen entnommen aus [30]

Bedrohung	Beschreibung
Informationsbeschaffung	Informationsbeschaffung durch Gespräche mit Angestellten und Auftragspartnern oder Beobachtungen/Fotografie während Werksführungen.
Sniffing	Beobachtung des Netzwerkverkehrs, um Informationen und Daten aufzuzeichnen oder User-Identifikationen und Passwörter zu bestimmen. Sniffing ist nur sehr schwer zu detektieren da kein zusätzlicher Datenverkehr im Netzwerk verursacht wird.
Kommunikationsattacke	Störung der Kommunikation von IACS. Diese kann intern auf verschiedenen Ebenen als auch von außen als Denial-of-Service (DoS) Attacke auftreten.
Database Injection	Attacke auf Datenbank-getriebene Webseiten durch Ausführung unautorisierter Befehle unter Ausnutzung von Sicherheitslücken in Software. Meist zum Diebstahl von Daten aus Datenbanken oder zum Zugriff auf andere Hosts über den Computer der die Datenbank hostet.
Replay	Bei diesem Angriff werden Signale an Kommunikationspfaden von IACS aufgenommen und zu einem späteren Zeitpunkt wiederholt gesendet. Damit könnte man Zugriffe auf abgesicherte Systeme und Bereiche durch die Wiederverwendung von Zugriffskontrollsignalen oder biometrischen Signalen erhalten oder Abläufe in IACS manipulieren.
Spoofing und Impersionation	Täuschung von Hard- und/oder Software durch Imitation anderer Identitäten, z.B. ARP-Spoofing, DNS-Spoofing, IP-Spoofing in Form einer Man in the Middle Attacke.
Social Engineering	Diverse Taktiken um von Unternehmenspersonal auf direktem Weg (z.B. per Telefon) sicherheitsrelevante Informationen zu beschaffen.
Phishing	z.B. Phishing Mails

Bedrohung	Beschreibung
Malicious Code / Malware	Für das Sammeln von Personen-, Unternehmens- oder Systeminformationen, Zerstörung von Daten, Manipulation und Verfälschung von Daten und Reports. Malicious Code kann in Form von Viren, Würmern, Trojanern oder automatischer Exploits auftreten.
DoS / DDoS	Zur Überlastung und somit zur Störung der Verfügbarkeit des Netzwerks ab. Beim Distributed-Denial-of-Service (DDoS) werden dafür mehrere kompromittierte Geräte benutzt.
Escalation of Privileges	Steigerung der Berechtigungen um die Handlungsfähigkeit im System zu erhöhen, um beispielsweise diverse Modifikationen an Konfigurationen etc. vorzunehmen.
Physische Zerstörung	Zerstörung von Hardware, Speicher-Medien, Verbindungen (Kabeln), Sensoren und Steuerungen. Diese Zerstörung kann auf direktem Wege durch menschliches Handeln oder auf indirektem Wege aufgrund eines Cyberangriffes erfolgen. Letzterer führt Aktionen herbei, um Beschädigungen oder eine Zerstörung zu verursachen.

Tabelle 13: Bedrohungen [30]

Relevante Schutzziele ermitteln

Foundational Requirements, System Requirements und Requirement Enhancements

FR 1 – Identification and Access Control (IAC) – Identifizierung und Authentifizierung	SL 1	SL 2	SL 3	SL 4
SR 1 - Identifizierung und Authentifizierung von menschlichen Nutzern	✓	✓	✓	✓
RE 1 - Eindeutige Identifizierung und Authentifizierung		✓	✓	✓
RE 2 - Multifaktor-Authentifizierung über nicht vertrauenswürdige Netze			✓	✓
RE 3 - Multifaktor-Authentifizierung über alle Netze				✓
SR 2 - Identifizierung und Authentifizierung von Softwareprozessen und Geräten		✓	✓	✓
RE 1 - Eindeutige Identifizierung und Authentifizierung			✓	✓
SR 3 - Nutzerkontenverwaltung	✓	✓	✓	✓
RE 1 - Einheitliche Nutzerkontenverwaltung			✓	✓
SR 4 - Verwaltung der Kennungen	✓	✓	✓	✓
SR 5 - Verwaltung der Authentifizierer	✓	✓	✓	✓
RE 1 - Beglaubigung der Identität von Softwareprozessen durch Hardwaremaßnahmen			✓	✓
SR 6 - Management drahtloser Zugriffsverfahren	✓	✓	✓	✓
RE 1 - Eindeutige Identifizierung und Authentifizierung		✓	✓	✓
SR 7 - Stärke der Authentifizierung durch Passwörter	✓	✓	✓	✓
RE 1 - Erzeugung und Lebensdauerbeschränkungen von Passwörtern für menschliche Nutzer			✓	✓
RE 2 - Lebensdauerbeschränkungen von Passwörtern für alle Nutzer				✓
SR 8 - PKI-Zertifikate		✓	✓	✓
SR 9 - Stärke der Authentifizierung durch öffentliche Schlüssel		✓	✓	✓
RE 1 - Beglaubigung öffentlicher Schlüssel durch Hardwaremaßnahmen			✓	✓
SR 10 - Rückmeldung vom Authentifikator	✓	✓	✓	✓
SR 11 - Erfolglose Anmeldeversuche	✓	✓	✓	✓
SR 12 - Nutzungshinweis	✓	✓	✓	✓
SR 13 - Zugriff über nicht vertrauenswürdige Netze	✓	✓	✓	✓
RE 1 - Genehmigung ausdrücklicher Anmeldebegehren		✓	✓	✓

FR 2 – Use Control (UC) – Nutzungskontrolle	SL 1	SL 2	SL 3	SL 4
SR 1 - Durchsetzung der Autorisierung	✓	✓	✓	✓
RE 1 - Durchsetzung der Autorisierung für alle Nutzer		✓	✓	✓
RE 2 - Abbildung der Berechtigungen auf Rollen		✓	✓	✓
RE 3 - Eingriffe des Aufsichtspersonals			✓	✓
RE 4 - Doppelte Zustimmung				✓
SR 2 - Nutzungskontrolle von Funkverbindungen	✓	✓	✓	✓
RE 1 - Nicht genehmigte drahtlose Geräte erkennen und anzeigen			✓	✓
SR 3 - Nutzungskontrolle von tragbaren und mobilen Geräten	✓	✓	✓	✓
RE 1 - Security-Status tragbarer und mobiler Geräte durchsetzen			✓	✓
SR 4 - Plattformübergreifender (mobiler) Code	✓	✓	✓	✓
RE 1 - Prüfung der Integrität mobilen Codes			✓	✓
SR 5 - Sitzungssperrung	✓	✓	✓	✓
SR 6 - Beendigung einer Fernzugriffssitzung		✓	✓	✓
SR 7 - Begrenzung der Anzahl gleichzeitiger Sitzungen			✓	✓
SR 8 - Prüfbare Ereignisse und deren Aufzeichnung	✓	✓	✓	✓
RE 1 - Zentral verwaltete systemweite Ereignisaufzeichnung			✓	✓
SR 9 - Speicherkapazität für Aufzeichnungen	✓	✓	✓	✓
RE 1 - Warnung, wenn die Kapazitätsgrenze zur Speicherung von Ereignisdatensätzen erreicht ist			✓	✓
SR 10 - Reaktion auf ausgefallene Verarbeitung von Ereignisdaten	✓	✓	✓	✓
SR 11 - Zeitstempel		✓	✓	✓
RE 1 - Interne Systemtakte			✓	✓
RE 2 - Schutz und Integrität der Zeitquelle				✓
SR 12 - Nicht-Abstreitbarkeit			✓	✓
RE 1 - Nicht-Abstreitbarkeit für alle Nutzer				✓
FR 3 – System Integrity (SI) – Systemintegrität	**SL 1**	**SL 2**	**SL 3**	**SL 4**
SR 1 - Kommunikationsintegrität	✓	✓	✓	✓
RE 1 - Kryptographische Schutzmaßnahmen zur Bewahrung der Integrität			✓	✓
SR 2 - Schutz vor Schadcode	✓	✓	✓	✓

RE 1 - Schutz vor Schadcode an Eingangs- und Ausgangspunkten		✓	✓	✓
RE 2 - Zentrales Management und Meldewesen zum Schutz vor Schadcode			✓	✓
SR 3 - Verifikation der IT-Sicherheitsfunktionalität	✓	✓	✓	✓
RE 1 - Automatisierte Mechanismen zur Verifikation der IT-Sicherheitsfunktionalität			✓	✓
RE 2 - Verifikation der IT-Sicherheitsfunktionalität im laufenden Betrieb				✓
SR 4 - Software- und Informationsintegrität	✓	✓	✓	✓
RE 1 - Automatisierte Hinweise auf IT-Sicherheitsverstöße			✓	✓
SR 5 - Eingabevalidierung	✓	✓	✓	✓
SR 6 - Vorbestimmte Zustände der Ausgänge	✓	✓	✓	✓
SR 7 - Fehlerbehandlung		✓	✓	✓
SR 8 - Sitzungsintegrität		✓	✓	✓
RE 1 - Annullierung der Sitzungskennungen nach Sitzungsbeendigung			✓	✓
RE 2 - Erzeugung einer eindeutigen Sitzungskennung			✓	✓
RE 3 - Zufälligkeit der Sitzungskennungen				✓
SR 9 - Schutz von Prüfinformationen		✓	✓	✓
RE 1 - Ereignisdatensätze auf nur einmal beschreibbaren Speichermedien				
FR 4 – Data Confidentiality (DC) – Vertraulichkeit der Daten	**SL 1**	**SL 2**	**SL 3**	**SL 4**
SR 1 - Vertraulichkeit von Informationen	✓	✓	✓	✓
RE 1 - Schutz der Vertraulichkeit bei der Speicherung oder Übertragung über nicht vertrauenswürdige Netze		✓	✓	✓
RE 2 - Schutz der Vertraulichkeit über Zonengrenzen hinweg				✓
SR 2 - Dauerhaftigkeit von Informationen		✓	✓	✓
RE 1 - Säuberung gemeinsam genutzter Speicher			✓	✓
SR 3 - Verwendung von Verschlüsselung	✓	✓	✓	✓
FR 5 – Restricted Data Flow (RDF) – Eingeschränkter Datenfluss	**SL 1**	**SL 2**	**SL 3**	**SL 4**
SR 1 - Netzaufteilung	✓	✓	✓	✓
RE 1 - Physikalische Netzaufteilung		✓	✓	✓
RE 2 - Unabhängigkeit von nicht automatisierungstechnischen Netzen			✓	✓

RE 3 - Logische und physikalische Isolierung kritischer Netze				✓
SR 2 - Schutz der Zonengrenze	✓	✓	✓	✓
RE 1 - Deny by default, allow by exception		✓	✓	✓
RE 2 - Inselmodus			✓	✓
RE 3 - Fail close			✓	✓
SR 3 - Beschränkung der persönlichen Kommunikation	✓	✓	✓	✓
RE 1 - Verbot der persönlichen Kommunikation			✓	✓
SR 4 - Partitionierung von Anwendungen	✓	✓	✓	✓
FR 6 – Timely Response to Events (TRE) – Rechtzeitige Reaktion auf Ereignisse	**SL 1**	**SL 2**	**SL 3**	**SL 4**
SR 1 - Zugriffsmöglichkeit auf Ereignisprotokolle	✓	✓	✓	✓
RE 1 - Programmgesteuerter Zugriff auf Ereignisprotokolle			✓	✓
SR 2 - Kontinuierliche Überwachung		✓	✓	✓
FR 7 – Ressource Availability (RA) – Ressourcenverfügbarkeit	**SL 1**	**SL 2**	**SL 3**	**SL 4**
SR 1 - Schutz gegen DoS-Ereignisse	✓	✓	✓	✓
RE 1 - Netzbelastung steuern		✓	✓	✓
RE 2 - DoS-Auswirkungen auf andere Systeme oder Netze begrenzen			✓	✓
SR 2 - Ressourcenmanagement	✓	✓	✓	✓
SR 3 - Datensicherung im Automatisierungssystem	✓	✓	✓	✓
RE 1 - Verifikation der Datensicherung		✓	✓	✓
RE 2 - Automatisierung der Datensicherung			✓	✓
SR 4 - Wiederherstellung des Automatisierungssystems	✓	✓	✓	✓
SR 5 - Notstromversorgung	✓	✓	✓	✓
SR 6 - Netz- und IT-Sicherheitseinstellungen	✓	✓	✓	✓
RE 1 - Maschinen-lesbare Meldungen der momentanen IT-Sicherheitseinstellungen			✓	✓
SR 7 - Geringste Funktionalität	✓	✓	✓	✓
SR 8 - Verzeichnis der Komponenten eines Automatisierungssystems		✓	✓	✓
AFR 1 – Physical Access (PA) – Physischer Zugang				
AFR 2 – Organisation (O) – Organisation der Security				

Tabelle 14: FR, SR und RE [29], [35]

Zonen und Conduits definieren oder neues Asset aufnehmen

Referenz-Modell – liefert die konzeptionelle Basis für die nachfolgenden detaillierteren Modelle

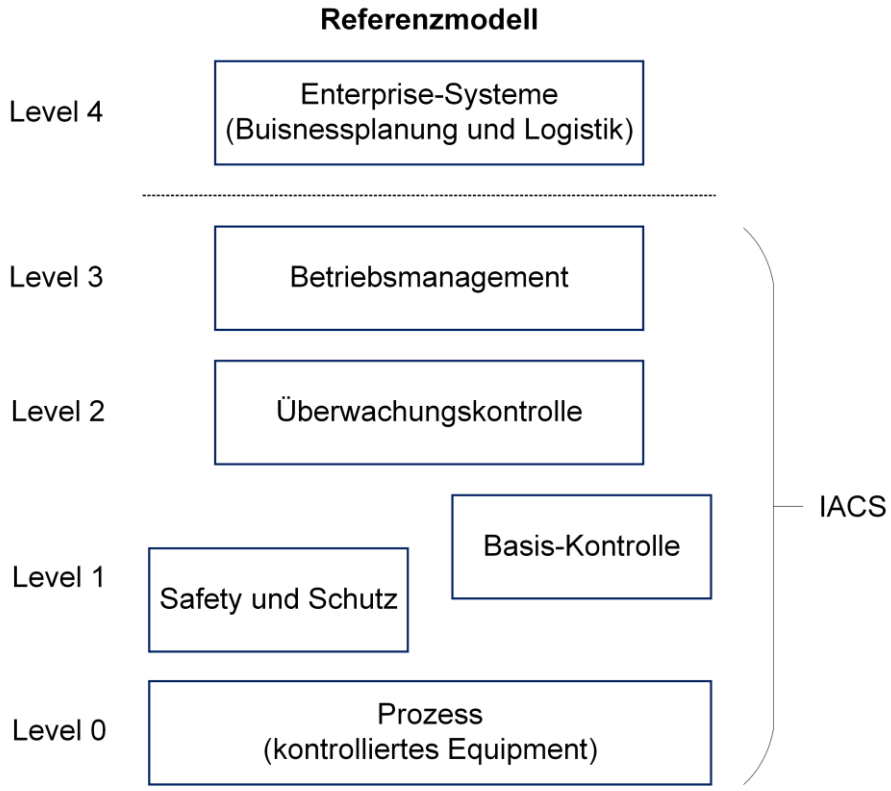


Abbildung 81: Referenz-Modell [30]

Asset-Modell – beschreibt die Zusammenhänge zwischen Assets in IACS unter Berücksichtigung zusätzlicher Netzwerkelemente

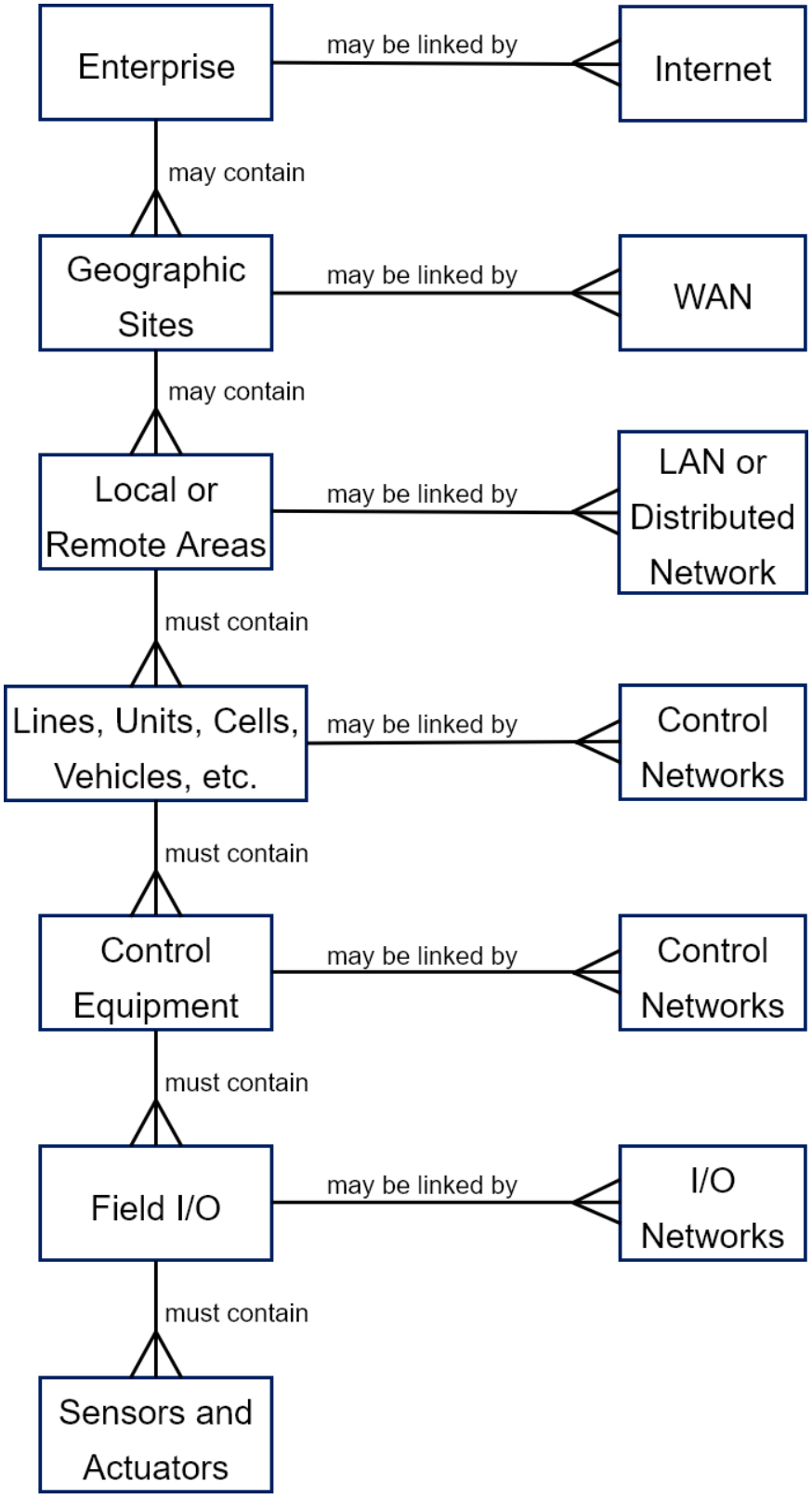


Abbildung 82: Beispiel eines Asset-Modells [30]

Referenzarchitektur – bildet die Grundlage für die Erstellung des Zonenmodells. Die Architektur enthält alle Komponenten, die als Assets festgelegt wurden

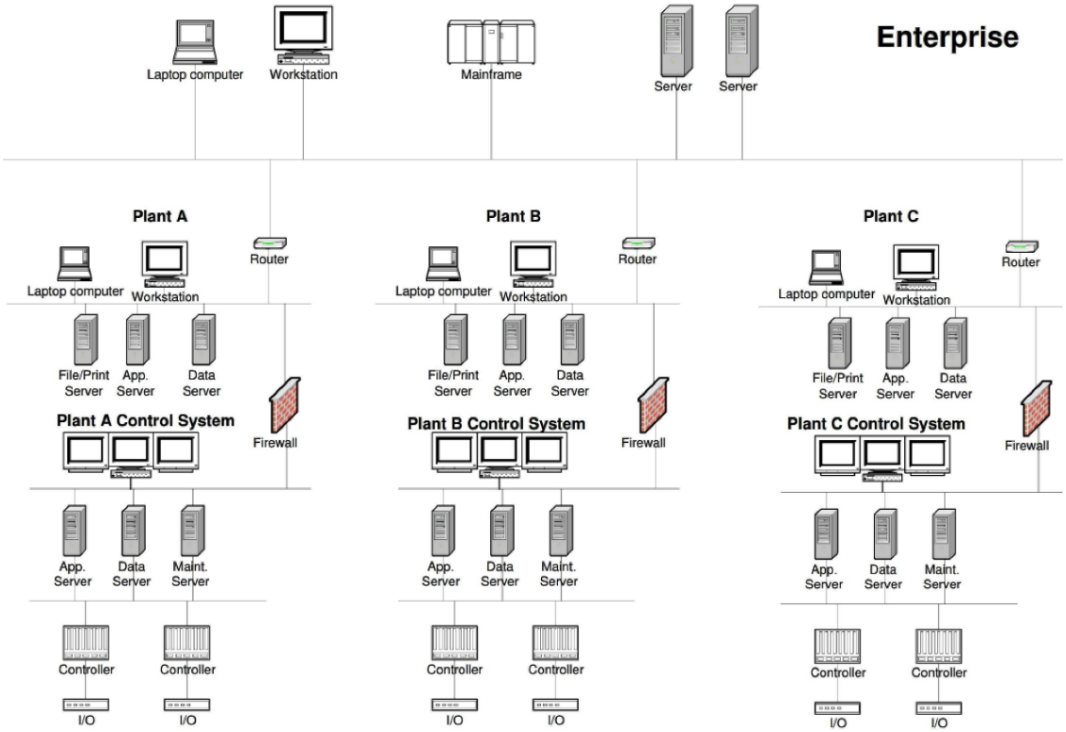


Abbildung 83: Beispiel einer Referenzarchitektur [30]

Zonen-Modelle – Möglichkeiten zur Segmentierung der zuvor dargestellten Referenzarchitektur

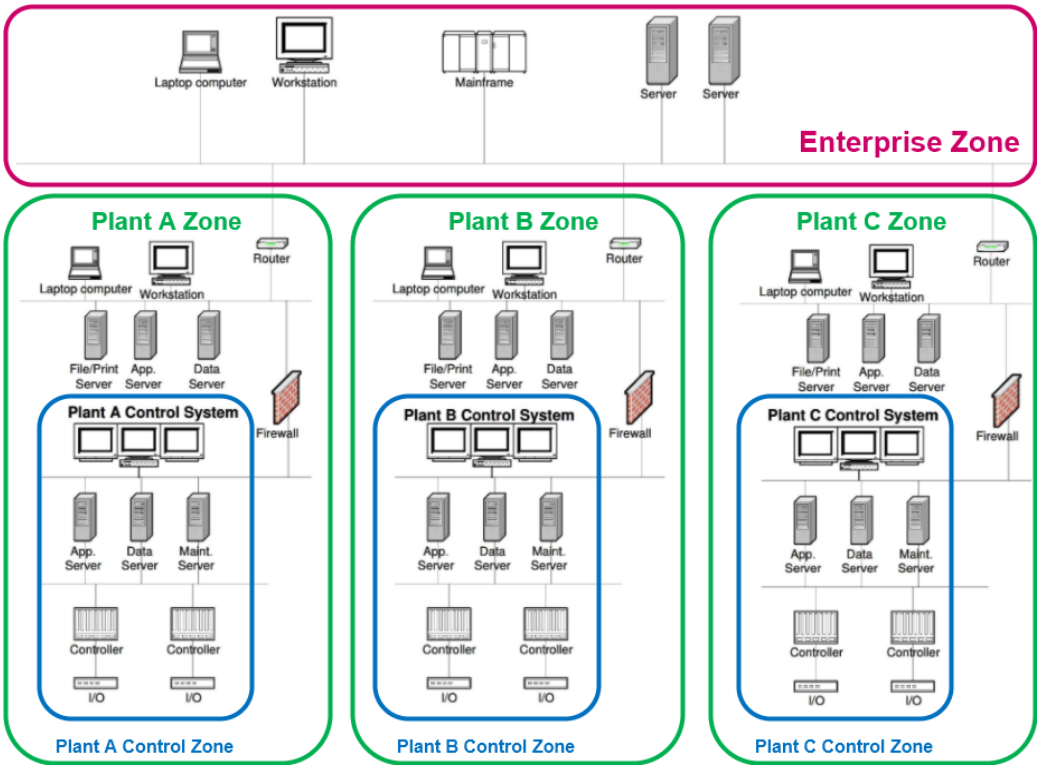


Abbildung 84: Beispiel eines Zonen-Modells mit Sub-Zonen [30]

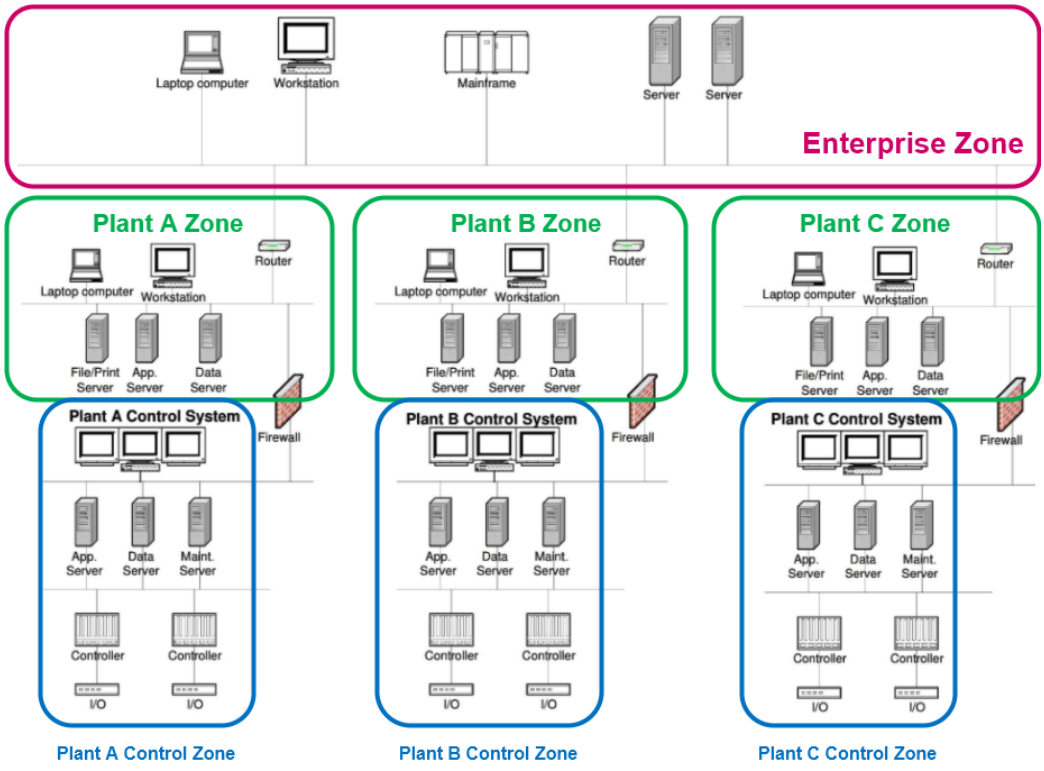


Abbildung 85: Beispiel eines separierten Zonen-Modells [30]

Inhalte gemäß [30] für die Dokumentation der Attribute von Zonen und Conduits

- Security Polcies
- Inventar der Assets
- Zugangsanforderungen und Kontrolle
- Bedrohungen und Schwächen
- Konsequenzen bei Sicherheitslücken
- Autorisierungstechniken
- Change-Managementprozess
- Verbundene Zonen (ausschließlich bei Conduits)

Beispiel für die Dokumentation einer Zone (analoges gilt für Conduits)

ZONE S_1

Zone Name: Unit 1 Hydrocracker Safety System

Definition of this Zone: This zone includes all safety integrity systems for the Unit 1 Hydrocracker.

Controlling Agency: Process Automation Department, SIS Team.

Zone Function: The systems in this zone provide safety functions to the Unit 1 Hydrocracker.

Zone Boundaries: The Safety Integrated System as defined by the Unit 1 Hydrocracker HAZOP.

Typical Assets: The Safety Integrated System controller, engineering station and communications hardware

Inheritance: This zone inherits attributes from Zone C_1 (Unit 1 Hydrocracker Basic Control System)

Zone Risk Assessment: This is a low to moderately secure zone with extreme consequences if breached.

a) **Security Capabilities of Zone Assets:** All assets are assumed to be incapable of withstanding low level attacks (i.e. those launched by unsophisticated attackers or malware) on their availability or confidentiality. This is a result of the protocols in use and system design. Assets are assumed to be capable of withstanding medium level attacks (i.e. those launched by moderately sophisticated attackers or malware) on their integrity.

b) **Threats and Vulnerabilities**: The vulnerabilities of this zone are assumed to be typical of legacy industrial control devices using MODBUS for communications. The principal threats are:
   a. Network-based Denial of Service to SIS communications.
   b. Internal or External unauthorized access to the SIS engineering station.
   c. Spoofing of MODBUS/TCP control commands.
   d. Spoofing of MODBUS/TCP responses to the process system.
   e. Reprogramming of safety functions.

c) **Consequences of a Security Breach**:
   a. Loss of production >6 hrs from false trip of emergency shutdown system.
   b. Loss of production <6 hrs due to loss of visibility to safety system.
   c. Disabling/manipulation of emergency shutdown resulting in fatality or major community incident.

d) **Business Criticality:** Extreme

Security Objective: To protect the integrity and availability of the Unit 1 Hydrocracker Safety System.

Acceptable Use Policy: I/O and Fieldbus communications is allowed to Zone P_1 (Unit 1 Hydrocracker Process). Read access to published data is allowed to approved systems in the Zone C_1 (Unit 1 Hydrocracker Basic Control System). All Write access to this zone is forbidden. All system management and programming functions shall be internal to this zone.

Inter-zone Connections: Conduits to this zone may be established from Zone C_1 (Unit 1 Hydrocracker Basic Control System) and from Zone P_1 (Unit 1 Hydrocracker Process).

Security Strategy: All connections to this zone must be controlled using type S conduits. Access to these systems must be approved by the Controlling Agency.

Change Management Process: All changes to this zone or any of its connecting conduits must follow the approved change management process of its corresponding Controlling Agency (see above). This includes, but is not limited to, the installation or replacement of equipment, modification of security policy, and exceptions to security policy or existing practices.

Abbildung 86: Beispiel für die Dokumentation der Attribute einer Zone [31]

Mögliche Schutzmaßnahmen bestimmen und deren Wirksamkeit bewerte

Schutznahmen
Technische Schutzmaßnahmen
Isolation oder Trennung von Ressourcen
Zugriffskontrollen
Netzwerksegmentierung und Schutz einzelner Zellen mit Firewalls. Firewalls sollten alle nicht-notwendigen Kommunikationen unterbinden.
DMZ
VPN (Virtual Private Networks)
Authentifizierung von Nutzern, Computern und Applikationen (Passwort, Token, Smartcard, Zertifikat, Biometrik, digitale Signatur)
Intrusion Detection und Prevention
Encryption gegen Datenmanipulation, Digitale Signaturen und Zertifikate
Anti-Malware und Anti-Virus Software
Schwachstellen Scanner
Honeypots
Whitelisting-Software: Whitelisting arbeitet mit Positivlisten, in denen der Benutzer festlegen kann, welche Prozesse bzw. Programme auf dem Rechner laufen dürfen. Versucht ein Benutzer oder eine Schadsoftware ein neues Programm, welches sich nicht auf der Liste befindet zu installieren, so wird dies unterbunden [29].
Überwachung der Systemaktivitäten
Intelligentes Kommunikationsüberwachungsequipment
Datenvalidierung
Systemhärtung durch Deaktivierung ungenutzter Ports
Unidirektionale Kommunikation mit Datendioden
Backup- und Wiederherstellungssysteme (vor allem für kritische Systeme und kritisches Equipment)
Versionierungstool für jegliche Art von Dokumenten
Monitoring von Prozessdaten und Alarmsystemen kritischer Parameter
Fail-Safe von Equipment
Detektion von Feuer und Löschanlagen
Nutzung von Funktionen wie Passwortschutz, Bausteinschutz oder Kopierschutz für SPS-Programme. Hierbei können einzelne Funktionsbausteine geschützt werden, sodass Unbefugte keinen Zugriff auf deren Inhalt haben und somit Algorithmen nicht kopieren oder verändern können [31].

Ersetzen von unsicheren Services und Kommunikationsprotokollen
Update- bzw. Patch-Management
Administrative bzw. organisatorische Schutzmaßnahmen
Erstellung von Policies und Procedures
Definierte Prozesse zum Betrieb von Anlagen
Klare Verantwortlichkeitsstrukturen innerhalb der Organisation z.B. Rollen und Privilegien aller Anwender einer Automatisierungslösung definieren und auf das Notwendigste eingrenzen – Least Privileges Ansatz.
Passwortmanagement (Dazu zählt auch nach Designprozessen und Implementierungen jegliche temporären Accounts zu löschen und alle Default-Passwörter zu ändern.)
Berücksichtigung von Security und Safety beim Prozessdesign
Bei Nutzung externer Software ist zu überprüfen ob der Hersteller Risikoanalysen, Programmierrichtlinien, Codeanalysen und Penetrationstests durchgeführt hat.
Awareness Schulungen und Security-Trainings für das Personal. Ein Beispiel wäre, dass eine Policy für Werksbesichtigungen, die Abgabe elektronischer Geräte wie Smartphones, USB-Sticks vor dem Betreten der Produktionsumgebung erfordert. Diese Policy muss einerseits dem Personal bekannt sein und andererseits von diesem auf Einhaltung kontrolliert werden.
Schulungen für Kompetenzaufbau
Periodischer Wechsel von digitalen Zutrittsschlüsseln, Codes und Passwörter
Rollen-/Nutzer- und Rechteverwaltung inklusive Wartung und Pflege
Wenn ein Mitarbeiter das Unternehmen verlässt: Account, Rechte, Passwörter löschen. Zutrittskarte,-schlüssel etc. einheben. Notwendige Passwörter auf allgemein genutzten Systemen ändern.
Duplikate essentieller Dokumente in feuerfesten Tresor aufbewahren etc.
Veraltete essentielle Dokumente vernichten.
Physische Schutzmaßnahmen
Physische Barrieren für Personen oder Fahrzeuge: versperrte Türen, Schränke, Zäune, Tore etc.
Kennzeichnung von Zutrittsbeschränkung (Zutritt verboten, unautorisierter Zugang verboten etc.)
Zutrittskontrollen und Überwachung: Identifizierung, Portier, Eskorten etc.
Verschiedene Identifikationskennzeichen für Angestellte, Kunden, Besucher etc.
Anlieferungsannahme getrennt vom Produktionsbereich
Absperrung oder Zutrittsbeschränkungen kritischer Bereiche oder Räumlichkeiten
Sicherheitskleidung

Tabelle 15: Sammlung von Gegenmaßnahmen

8.4 Anhang Kapitel 4

Kein Anhang

8.5 Anhang Kapitel 5

8.5.1 Abschnitt 5.2.1 – Steuerboard zur Simulation

- Schaltplan und Klemmenbezeichnungen des Steuerboards

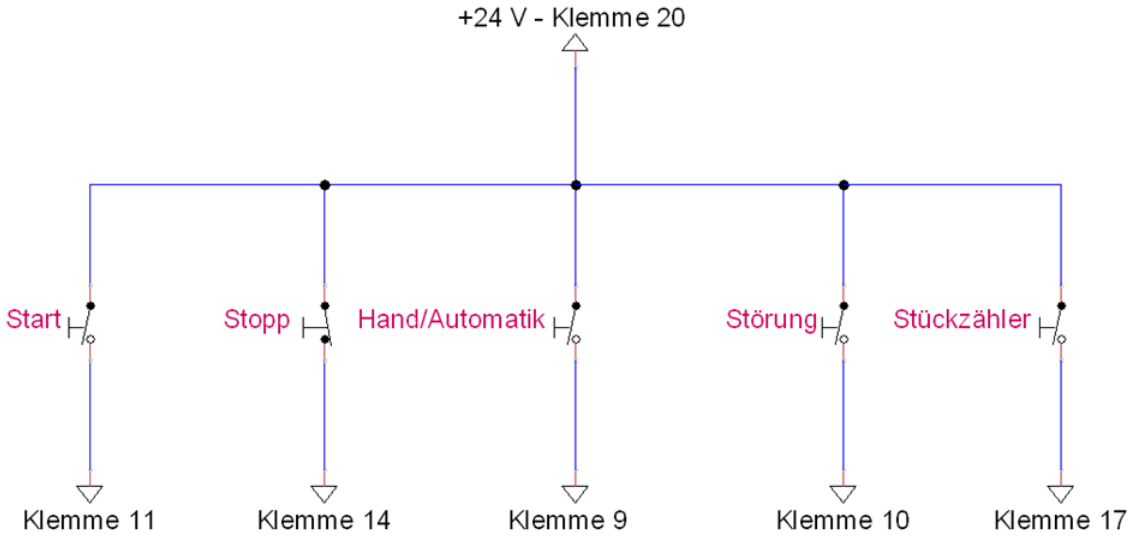


Abbildung 87: Schaltplan und Klemmenbezeichnungen des Steuerboards

8.5.2 Abschnitt 5.2.2 – Raspberry Pi 3 mit UniPi v1.1 Erweiterungsboard

- Konfigurationseinstellungen des Raspberry Pi

1 Change User Password – setzen eines neuen Passwortes für den User „pi“

2 Hostname – der standardmäßige Hostname wurde nicht verändert

3 Enable Boot to Desktop/Scratch – keine Änderungen

4 Localisation Options

- **I1 Change Locale** – keine Änderungen
- **I2 Change Timezone** – „Europe“ und „Vienna“ ausgewählt
- **I3 Change Keyboard Layout** – Generische PC-Tastatur mit 105 Tasten, „Deutsch (Österreich)“; keine Compose-Taste und Tastenkombination Strg+Alt+Zurück eingestellt
- **I4 Change Wi-fi Country** – “Austria” ausgewählt

5 Interfacing Options

- **P1 Camera** – Kamerainterface deaktiviert
- **P2 SSH** – SSH wird aktiviert, damit der SSH-Server beim Bootvorgang automatisch gestartet wird. Mit SSH können Änderungen am Raspberry Pi jederzeit mit einem Computer über das Netzwerk vorgenommen werden
- **P3 VNC** – deaktiviert
- **P4 SPI** – Damit die Input-Ports 7-14 der UniPi Erweiterungsplatine verwendet werden können, muss das Serial Peripheral Interface aktiviert werden
- **P5 I2C** – Der Inter-Integrated Circuit Bus muss für das UniPi-Board aktiviert werden
- **P6 Serial** – Der serielle Port wurde aktiviert
- **P7 1-Wire** – deaktiviert
- **P8 Remote GPIO** – Wurde aktiviert, um Fernzugriffe auf die GPIO nutzen zu können

6 Overclock – keine Änderungen

7 Advanced Options

- **A1 Expand Filesystem** – Wurde ausgeführt, damit die gesamte Speicherkapazität der SD-Karte genutzt werden kann
- **A2 Overscan** – keine Änderungen
- **A3 Memory Split** – keine Änderungen
- **A4 Audio** – „Auto“ ausgewählt
- **A5 Resolution** – keine Änderungen
- **A6 GL Driver** – keine Änderungen

8 Update – Das RPI-Tool wurde aktualisiert

■ Einstellungen in der SSH-Konfigurationsdatei überprüfen/ändern

Konfigurationsdatei sshd_config öffnen und nachfolgende Einträge überprüfen bzw. anpassen:

Port 3180 – Den SSH-Port von 22 (Standard) auf einen Port größer 1023 ändern

Protocol 2 – Für die Verwendung von Version 2 des SSH-Protokolls. Version 1 gilt als veraltet und unsicher und existiert nur noch aus Kompatibilitätsgründen

PermitEmptyPasswords no – Nutzern ohne Passwort wird kein Zugriff per SSH gewährt

PermitRootLogin no – Kein Root-Zugang per SSH. Der Administrator soll sich immer als normaler User einloggen und sich anschließend bedarfsweise Root-Rechte verschaffen

Danach müssen die Änderungen gespeichert und das SSH-Service neu gestartet werden.

8.5.3 Abschnitt 5.4.2 – SPS Programmierung im TIA Portal

- Baustein 1 – FB1

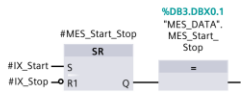
Name	Datentyp	Offset	Defaultwert	Erreichbar aus HMI	Sichtbar in HMI	Einstellwert	Kommentar
Input							
IX_Hand	Bool	0.0	true	True	True	False	FB-Eingang der am Baustein angeschlossen wird
IX_Auto	Bool	0.1	false	True	True	False	FB-Eingang der am Baustein angeschlossen wird
IX_Start	Bool	0.2	false	True	True	False	FB-Eingang der am Baustein angeschlossen wird
IX_Stop	Bool	0.3	true	True	True	False	FB-Eingang der am Baustein angeschlossen wird
IX_Alarm	Bool	0.4	false	True	True	False	FB-Eingang der am Baustein angeschlossen wird
IX_Qty	Bool	0.5	false	True	True	False	FB-Eingang der am Baustein angeschlossen wird
Output							
InOut							
Static							
MES_Hand_Auto	Bool	2.0	false	True	True	False	vom OPC-UA Server abgegriffen
MES_Dryrun	Bool	2.1	false	True	True	False	vom OPC-UA Server abgegriffen
MES_Start_Stop	Bool	2.2	false	True	True	False	vom OPC-UA Server abgegriffen
MES_Alarm	Bool	2.3	false	True	True	False	vom OPC-UA Server abgegriffen
MES_Qty	Int	4.0	0	True	True	False	vom OPC-UA Server abgegriffen
MES_Performance	Int	6.0	0	True	True	False	Gleitender Mittelwert der Stueck / Stunde - vom OPC-UA Server abgegriffen
Performance_tmp	Real	8.0	0.0	True	True	False	
Pos_Flanke	Bool	12.0	false	True	True	False	Netzwerk 4 - Flankenmerker Stueckzaehler
Time_measure1	Time	14.0	T#0ms	True	True	False	Netzwerk 5 - Zeitspanne1 zwischen 2 produzierten Stueck (Produktionsdauer eines Werkstuecks)
Time_measure2	Time	18.0	T#0ms	True	True	False	Netzwerk 5 - Zeitspanne2 zwischen 2 produzierten Stueck (Produktionsdauer eines Werkstuecks)
T_1_Real	Real	22.0	0.0	False	False	False	Netzwerk 5 - Zeitspanne1 zwischen 2 produzierten Stueck in REAL Datentyp
T_2_Real	Real	26.0	0.0	False	False	False	Netzwerk 5 - Zeitspanne1 zwischen 2 produzierten Stueck in REAL Datentyp
FIFO_IN	Real	30.0	0.0	False	False	False	Netzwerk 5 - Zeitspanne 1 / 2 die dann in Register geschrieben wird
Qty_Pos_Fl_InitReg	Bool	34.0	false	True	True	False	Netzwerk 5 - Flnakenmerker fuer Registerinitialisierung mit erster gemessener Zeitspanne
modulo_QtyRest	Int	36.0	0	True	True	False	Netzwerk 5 - Ergebnis nach Modulo-2 Operation - um die 2 Timer abwechselnd zu starten
Qty_Pos_Fl	Bool	38.0	false	True	True	False	Netzwerk 5 - Flnakenmerker fuer aktuelle Zeitspanne in Register schreiben
Flanke_tmp	Bool	38.1	false	True	True	False	Netzwerk 5 - Operand zu Flnakenmerker fuer Registerinitialisierung mit erster gemessener Zeitspanne
Flanke_tmp_M	Bool	38.2	false	True	True	False	Netzwerk 5 - Flnakenmerker fuer Registerinitialisierung mit erster gemessener Zeitspanne
Index	Int	40.0	0	True	True	False	Netzwerk 5 - Index für Register Array
Reg_Time	Array[0..4] of Real	42.0		True	True	False	Netzwerk 5 - Register fuer die aktuellsten Zeitspannen (Produktionsdauer eines Werkstuecks)
Reg_Time[0]	Real	0.0	0.0	True	True	False	
Reg_Time[1]	Real	4.0	0.0	True	True	False	
Reg_Time[2]	Real	8.0	0.0	True	True	False	
Reg_Time[3]	Real	12.0	0.0	True	True	False	
Reg_Time[4]	Real	16.0	0.0	True	True	False	
Qty_Neg_Fl	Bool	62.0	false	True	True	False	Netzwerk 5 - Flnakenmerker fuer Registerinitialisierung mit erster gemessener Zeitspanne
modulo_Rest	Int	64.0	0	True	True	False	Netzwerk 5 - Ergebnis nach Modulo-5 (Registergroesse) Operation - um nach 5 Werten Register von vorne wieder ueberschreiben
Temp							

Constant							
Time_max	Time		T#900s				maximale Zeitspanne die Timer zaehlt - wenn dieser Wert erreicht wird kann davon ausgegangen werden das es Probleme bei der Produktion gibt. REFERENZWERT
Init_Counter	DInt		0				Startwert des Counters; auch nachdem Reset

Netzwerke des Bausteins

Netzwerk 1: Start/Stop

Über FlipFlop (dominantes Setzen) da niemals Start und Stop gleichzeitig auftreten können



Symbol	Adresse	Typ	Kommentar
"MES_DATA".MES_Start_Stop	%DB3.DBX0.1	Bool	vom OPC-UA Server abgegriffen
#IX_Start		Bool	FB-Eingang der am Baustein angeschlossen wird
#IX_Stop		Bool	FB-Eingang der am Baustein angeschlossen wird
#MES_Start_Stop		Bool	vom OPC-UA Server abgegriffen

Netzwerk 2: Hand-/Automatikbetrieb

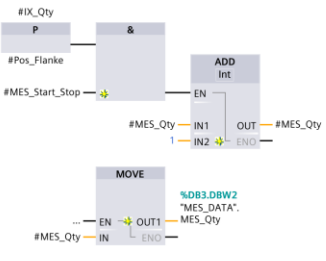
Über FlipFlop (dominantes Setzen) da niemals Hand und Auto gleichzeitig auftreten können

Symbol	Adresse	Typ	Kommentar
"MES_DATA".MES_Hand_Auto	%DB3.DBX0.0	Bool	vom OPC-UA Server abgegriffen
#IX_Auto		Bool	FB-Eingang der am Baustein angeschlossen wird
#IX_Hand		Bool	FB-Eingang der am Baustein angeschlossen wird
#MES_Hand_Auto		Bool	vom OPC-UA Server abgegriffen

Netzwerk 3: Störung

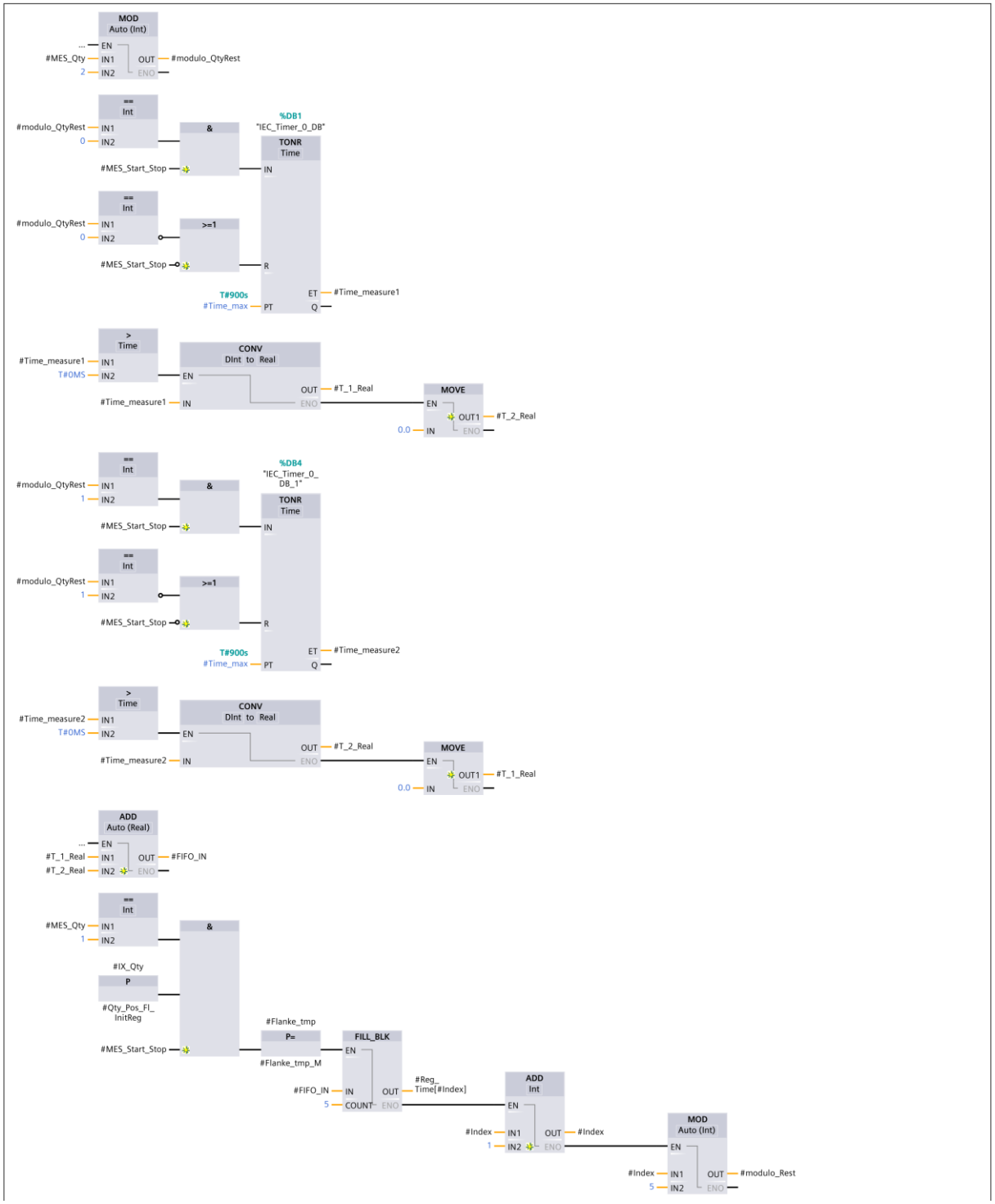
Symbol	Adresse	Typ	Kommentar
"MES_DATA".MES_Alarm	%DB3.DBX0.2	Bool	vom OPC-UA Server abgegriffen
#IX_Alarm		Bool	FB-Eingang der am Baustein angeschlossen wird
#MES_Alarm		Bool	vom OPC-UA Server abgegriffen

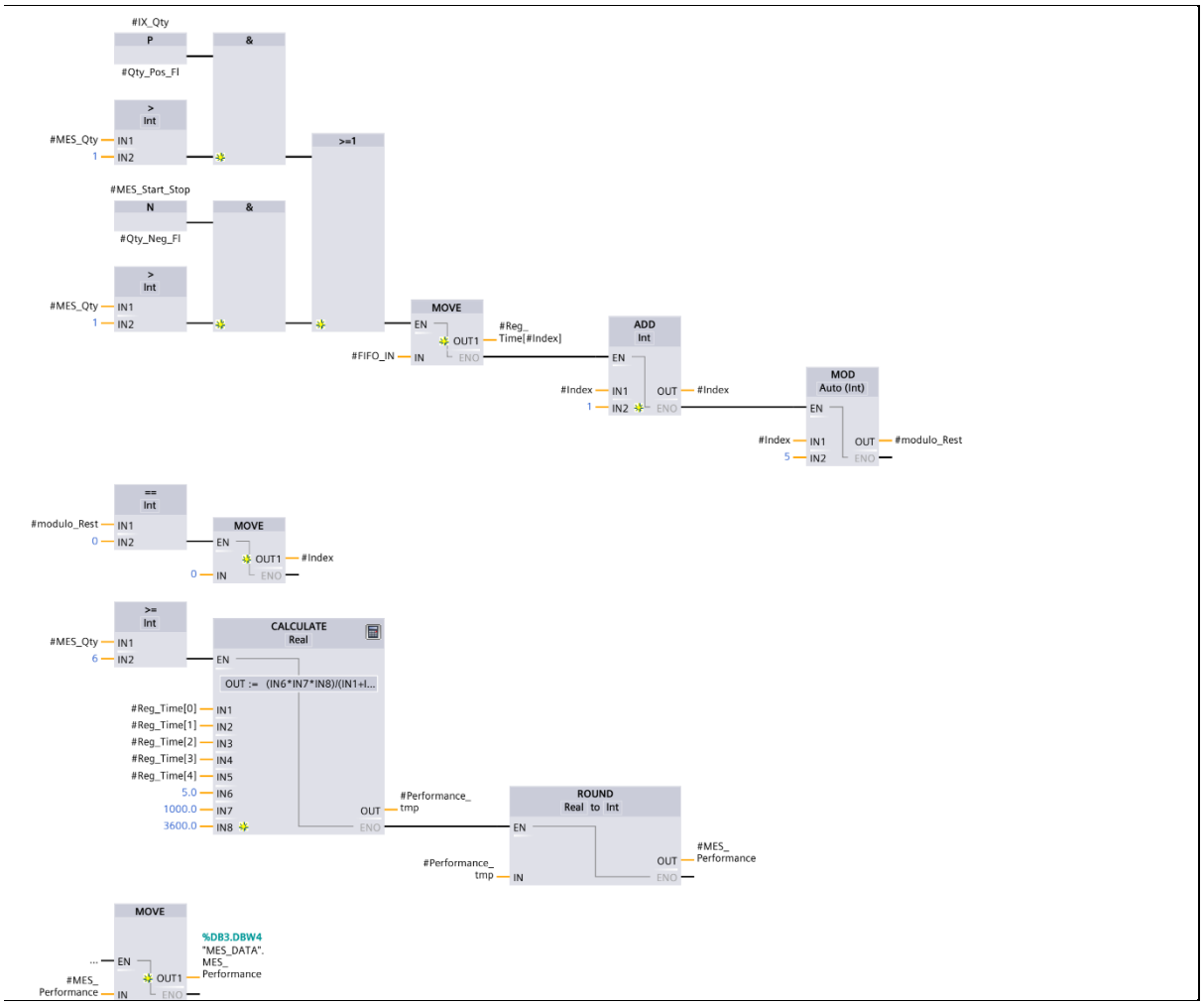
Netzwerk 4: Stückzähler ohne Counter sondern über Flankenaddition



Symbol	Adresse	Typ	Kommentar
"MES_DATA".MES_Qty	%DB3.DBW2	Int	vom OPC-UA Server abgegriffen
#IX_Qty		Bool	FB-Eingang der am Baustein angeschlossen wird
#MES_Qty		Int	vom OPC-UA Server abgegriffen
#MES_Start_Stop		Bool	vom OPC-UA Server abgegriffen
#Pos_Flanke		Bool	Netzwerk 4 - Flankenmerker Stueckzaehler

Netzwerk 5: Performance Stück pro Stunde





Symbol	Adresse	Typ	Kommentar
"MES_DATA".MES_Performance	%DB3.DBW4	Int	Gleitender Mittelwert der Stueck / Stunde - vom OPC-UA Server abgegriffen
#FIFO_IN		Real	Netzwerk 5 - Zeitspanne 1 / 2 die dann in Register geschrieben wird
#Flanke_tmp		Bool	Netzwerk 5 - Operand zu FInakenmerker fuer Registerinitialisierung mit erster gemessener Zeitspanne
#Flanke_tmp_M		Bool	Netzwerk 5 - FInakenmerker fuer Registerinitialisierung mit erster gemessener Zeitspanne
#Index		Int	Netzwerk 5 - Index für Register Array
#IX_Qty		Bool	FB-Eingang der am Baustein angeschlossen wird
#MES_Performance		Int	Gleitender Mittelwert der Stueck / Stunde - vom OPC-UA Server abgegriffen
#MES_Qty		Int	vom OPC-UA Server abgegriffen
#MES_Start_Stop		Bool	vom OPC-UA Server abgegriffen
#modulo_QtyRest		Int	Netzwerk 5 - Ergebnis nach Modulo-2 Operation - um die 2 Timer abwechselnd zu starten
#modulo_Rest		Int	Netzwerk 5 - Ergebnis nach Modulo-5 (Registergroesse) Operation - um nach 5 Werten Register von vorne wieder ueberschreiben
#Performance_tmp		Real	
#Qty_Neg_FI		Bool	Netzwerk 5 - FInakenmerker fuer Registerinitialisierung mit erster gemessener Zeitspanne
#Qty_Pos_FI		Bool	Netzwerk 5 - FInakenmerker fuer aktuelle Zeitspanne in Register schreiben
#Qty_Pos_FI_InitReg		Bool	Netzwerk 5 - FInakenmerker fuer Registerinitialisierung mit erster gemessener Zeitspanne
#Reg_Time[0]		Real	Netzwerk 5 - Register fuer die aktuellsten Zeitspannen (Produktionsdauer eines Werkstuecks)
#Reg_Time[1]		Real	Netzwerk 5 - Register fuer die aktuellsten Zeitspannen (Produktionsdauer eines Werkstuecks)
#Reg_Time[2]		Real	Netzwerk 5 - Register fuer die aktuellsten Zeitspannen (Produktionsdauer eines Werkstuecks)
#Reg_Time[3]		Real	Netzwerk 5 - Register fuer die aktuellsten Zeitspannen (Produktionsdauer eines Werkstuecks)
#Reg_Time[4]		Real	Netzwerk 5 - Register fuer die aktuellsten Zeitspannen (Produktionsdauer eines Werkstuecks)
#Reg_Time[#Index]		Real	Netzwerk 5 - Register fuer die aktuellsten Zeitspannen (Produktionsdauer eines Werkstuecks)
#T_1_Real		Real	Netzwerk 5 - Zeitspanne1 zwischen 2 produzierten Stueck in REAL Datentyp
#T_2_Real		Real	Netzwerk 5 - Zeitspanne1 zwischen 2 produzierten Stueck in REAL Datentyp
#Time_max	T#900s	Time	maximale Zeitspanne die Timer zaehlt - wenn dieser Wert erreicht wird kann davon ausgegangen werden das es Probleme bei der Produktion gibt. REFERENZWERT
#Time_measure1		Time	Netzwerk 5 - Zeitspanne1 zwischen 2 produzierten Stueck (Produktionsdauer eines Werkstuecks)
#Time_measure2		Time	Netzwerk 5 - Zeitspanne2 zwischen 2 produzierten Stueck (Produktionsdauer eines Werkstuecks)

Netzwerk 6: Stückzähler Reset

Symbol	Adresse	Typ	Kommentar
#Init_Counter	0	DInt	Startwert des Counters; auch nachdem Reset
#MES_Qty		Int	vom OPC-UA Server abgegriffen
#MES_Start_Stop		Bool	vom OPC-UA Server abgegriffen

Netzwerk 7: Variablen Reset des Performance Stück pro Stunde-Netzwerks (Netzwerk 5)

Symbol	Adresse	Typ	Kommentar
#Index		Int	Netzwerk 5 - Index für Register Array
#MES_Start_Stop		Bool	vom OPC-UA Server abgegriffen
#modulo_QtyRest		Int	Netzwerk 5 - Ergebnis nach Modulo-2 Operation - um die 2 Timer abwechselnd zu starten
#modulo_Rest		Int	Netzwerk 5 - Ergebnis nach Modulo-5 (Registergroesse) Operation - um nach 5 Werten Register von vorne wieder ueberschreiben

8.5.4 Abschnitt 5.5.1 – Kommunikationsumgebung für MQTT

- Analyse des Netzwerkverkehrs zur Überprüfung des MQTT-Verbindungsaufbaus mit TLS

```
30 15.062548    10.0.0.139    10.0.0.100    TLSv1    274 Client Hello
31 15.086831    10.0.0.100    10.0.0.139    TLSv1    590 Server Hello
32 15.086832    10.0.0.100    10.0.0.139    TLSv1    590 Certificate [TCP segment of a reassembled PDU]
33 15.086832    10.0.0.100    10.0.0.139    TLSv1    344 Server Key Exchange, Server Hello Done
34 15.088112    10.0.0.139    10.0.0.100    TCP       60 41926 → 1883 [ACK] Seq=221 Ack=537 Win=30336 Len=0
35 15.088287    10.0.0.139    10.0.0.100    TCP       60 41926 → 1883 [ACK] Seq=221 Ack=1073 Win=31360 Len=0
36 15.088289    10.0.0.139    10.0.0.100    TCP       60 41926 → 1883 [ACK] Seq=221 Ack=1363 Win=32512 Len=0
38 15.107797    10.0.0.139    10.0.0.100    TLSv1    188 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
40 15.110000    10.0.0.100    10.0.0.139    TLSv1    304 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
41 15.112557    10.0.0.139    10.0.0.100    TLSv1    176 Application Data, Application Data
42 15.114195    10.0.0.100    10.0.0.139    TLSv1    128 Application Data, Application Data
```

Abbildung 88: TLS-Handshake während des MQTT-Verbindungsaufbaus

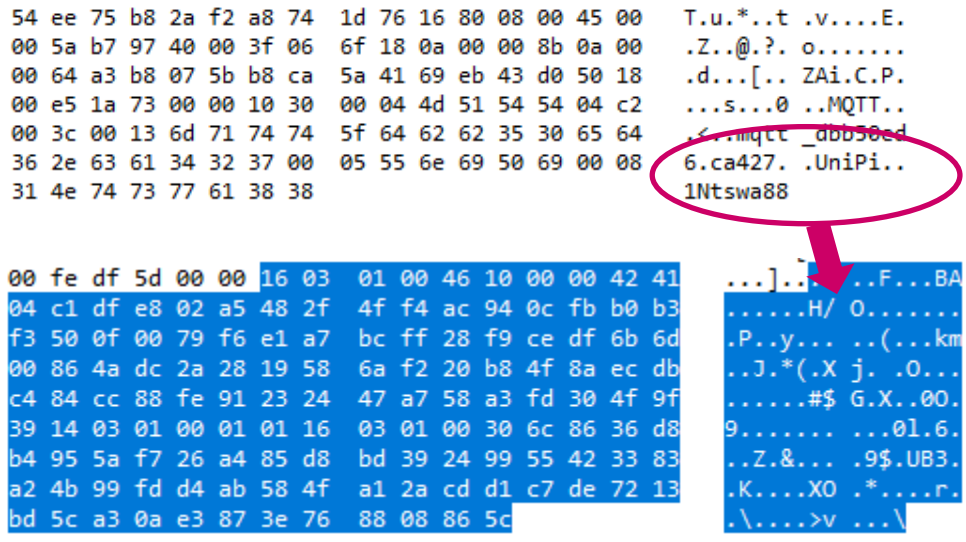


Abbildung 89: Gegenüberstellung der Authentifizierung bei MQTT ohne TLS und mit TLS

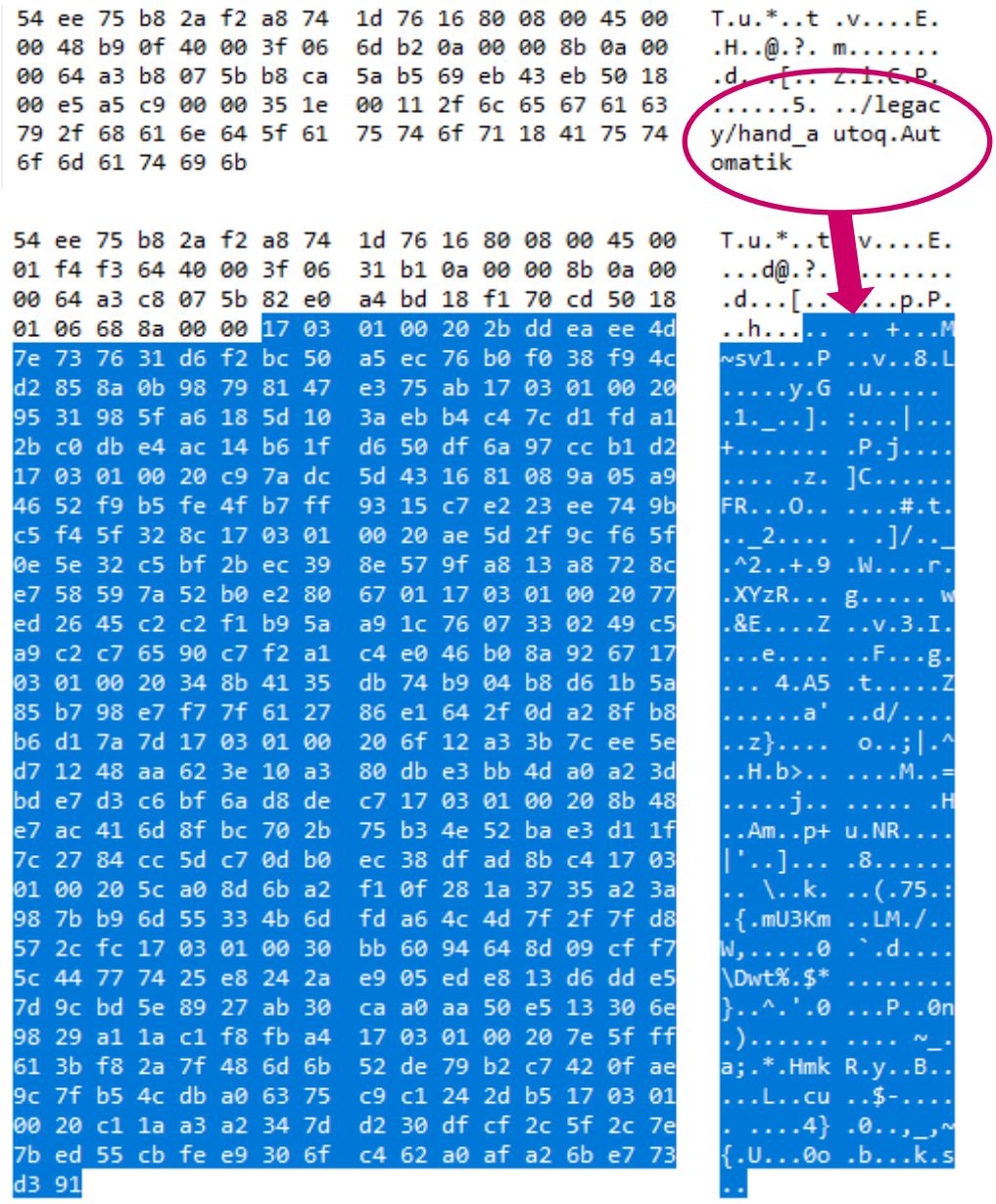


Abbildung 90: Gegenüberstellung der Datenübertragung mit MQTT ohne TLS und mit TLS

8.5.5 Abschnitt 5.5.2 – Kommunikationsumgebung für OPC UA

- Analyse des Netzwerkverkehrs zur Überprüfung der verschlüsselten Datenübertragung

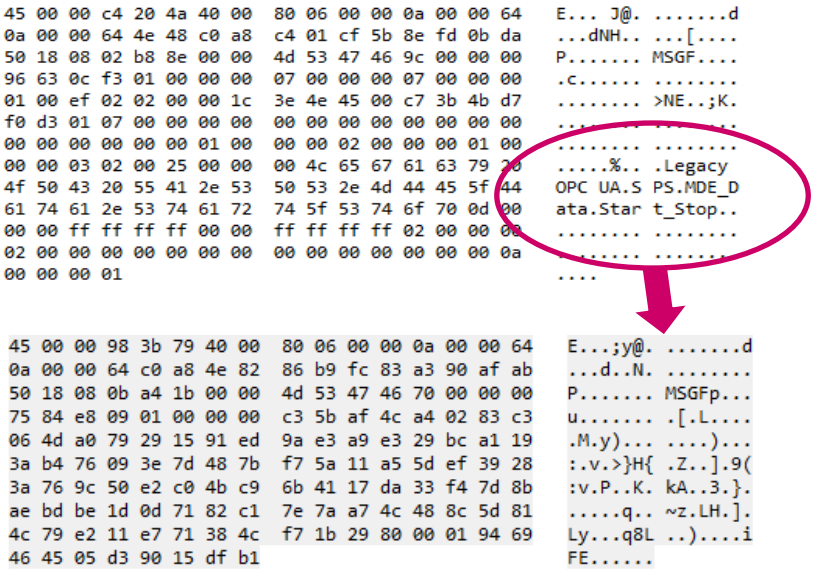


Abbildung 91: Gegenüberstellung der Datenübertragung mit OPC UA ohne und mit Verschlüsselung

9 Literaturverzeichnis

[1] J. Rinaldi, *OPC UA–Unified Architecture: The Everyman's Guide to the Most Important Information Technology in Industrial Automation*: CreateSpace Independent Publishing, 2016.

[2] M. Damm, S.-H. Leitner, and W. Mahnke, *OPC Unified Architecture* vol. 1, 2009.

[3] O. Foundation. *Unified Architecture*. [Online]. URL: https://opcfoundation.org/about/opc-technologies/opc-ua/. [Zugriff: 16.11.2017].

[4] O. Foundation, "OPC Unified Architecture Specification," in *Part 6: Mappings*, ed, 2017.

[5] O. Foundation, "OPC Unified Architecture Specification," in *Part 1: Overview and Concepts*, ed, 2017.

[6] U. Enste and W. Mahnke, "OPC unified architecture," *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik,* vol. 59, pp. 397-404, 2011.

[7] O. Foundation, "OPC Unified Architecture Specification," in *Part 14: PubSub*, ed, 2017.

[8] O. Foundation, "OPC Unified Architecture Specification," in *Part 4: Services*, ed, 2017.

[9] R. Huang, F. Liu, and P. Dongbo, "Research on OPC UA security," in *Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on*, 2010, pp. 1439-1444.

[10] O. Foundation, "OPC Unified Architecture Specification," in *Part 5: Information Model*, ed, 2017.

[11] M. Schleipen, *Praxishandbuch OPC UA: Grundlagen - Implementierung - Nachrüstung - Praxisbeispiele* vol. 1, 2017.

[12] O. Foundation, "OPC Unified Architecture Specification," in *Part 3: Address Space Model*, ed, 2017.

[13] O. Foundation, "OPC Unified Architecture Specification," in *Part 12: Discovery*, ed, 2015.

[14] O. Foundation, "OPC Unified Architecture Specification," in *Part 2: Security Model*, ed, 2015.

[15] B. f. S. i. d. I. (DE), "Sicherheitsanalyse OPC UA," 25.04.2016. [Online]. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/OPCUA/OPCUA.pdf?__blob=publicationFile&v=2. [Zugriff: 02.08.2017].

[16] B. Farnham and R. Barillere, "MIGRATION FROM OPC-DA TO OPC-UA." [Online]. URL: http://cds.cern.ch/record/1404189/files/MOPMS025.pdf. [Zugriff: 25.10.2017].

[17] A. Banks and R. Gupta, "MQTT Version 3.1.1 Plus Errata 01," in *OASIS Standard Incorporating Approved Errata 01*, ed, 2015.

[18] ISO/IEC. *ISO/IEC 20922 Information technology - Message Queuing Telemetry Transport (MQTT) v3.1.1* 2016. [Online]. URL: https://www.iso.org/standard/69466.html. [Zugriff: 01.07.2017].

[19] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2014, pp. 1-6.

[20] M. Maritsch, C. Kittl, and T. Ebner, "Sichere Vernetzung von Geräten in Smart Factories mit MQTT," *Mensch und Computer 2015–Workshopband,* 2015.

[21] S. Katsikeas, K. Fysarakis, A. Miaoudakis, A. Van Bemten, I. Askoxylakis, I. Papaefstathiou*, et al.*, "Lightweight & Secure Industrial IoT Communications via the MQ Telemetry Transport Protocol," 2017.

[22] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," *Transaction on IoT and Cloud Computing,* vol. 3, pp. 11-17, 2015.

[23] E. P. R. *MQTT - MQ Telemetry Transport* 2017. [Online]. URL: http://peteregli.net/content/iot/MQTT/MQTT.html. [Zugriff: 01.08.2017].

[24] E. Linden, "A latency comparison of IoT protocols in MES," Master, Computer and Information Science, Linköping, 2017.

[25] o. V. *Securing MQTT Systems* 2017. [Online]. URL: http://www.hivemq.com/blog/mqtt-security-fundamentals-securing-mqtt-systems. [Zugriff: 14.07.2017].

[26] R. Drolshagen, "Evaluation von M2M-Kommunikationsansätzen für Industrie 4.0 und Herleitung von Entscheidungskriterien für deren Einsatz," Master, Fachbereich Design Informatik Medien - Studiengang Informatik, Hochschule RheinMain, 2015.

[27] K.-H. Niemann, "IT-Sicherheit in Produktionsanlagen," 2017. [Online]. URL: https://serwiss.bib.hs-hannover.de/files/1135/IT+Sicherheit+in+Produktionsanlagen_05-2017.pdf. [Zugriff: 12.12.2017].

[28] B. f. S. i. d. I. (DE), "Industrial Control System Security." [Online]. URL: https://www.allianz-fuer-cybersicherheit.de/ACS/DE/_/downloads/BSI-CS_005.pdf;jsessionid=51A8CB42BB9C305A079B576C76235F81.2_cid351?__blob=publicationFile&v=4. [Zugriff: 08.10.2017].

[29] P. Kobes, *Leitfaden Industrial Security : IEC 62443 einfach erklärt*. Berlin Offenbach: VDE Verlag GmbH, 2016.

[30] I. E. C. (IEC), "IEC 62443-1-1 " in *Security for Industrial Automation and Control Systems – Terminology, Concepts and Models program*, ed, 2007.

[31] E. Byres, "White Paper V1.2 Using ISA/IEC 62443 Standards to Improve Control System Security," Tofino Security. [Online]. URL: https://www.tofinosecurity.com/sites/default/files/common/white-papers/Using-ISA_IEC-62443-Standards-WP-v1.2%20(May%202014).pdf. [Zugriff: 15.11.2017].

[32] S. AG, "Industry White Paper V1.0 Industrial Security – Security Konzept zum Schutz industrieller Anlagen," ed, 2013.

[33] I. E. C. (IEC), "IEC 62443-2-1 Edition 1.0," in *Industrial communication networks – Network and system security – Part 2-1: Establishing an industrial automation and control system security program*, ed, 2010-2011.

[34] U. S. D. o. H. S. I. C. S. C. E. R. Team, "Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies." [Online]. URL: https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf. [Zugriff: 27.11.2017].

[35] A. I. S. b. V. u. H. AG, "Leitfaden Security für den Maschinen- und Anlagenbau - Der Weg durch die IEC 62443." [Online]. URL: https://industrialsecurity.vdma.org/documents/16227999/16499033/1492086887896_INS_NAM_2016_Industrial_Security_IEC62443.pdf/c2e80bdb-c820-42cb-b3cc-fedd68571e1e. [Zugriff: 29.12.2017].

10 Abbildungsverzeichnis

Abbildung 1: OPC UA auf allen Ebenen [2] 10
Abbildung 2: Übersicht OPC UA Spezifikationen [5] 11
Abbildung 3: Hierarchie der Spezifikationen [6] 12
Abbildung 4: OPC UA Client/Server-Architektur [5] 12
Abbildung 5: Ablauf des Aufbaus einer Kommunikationsumgebung 14
Abbildung 6: Parameter von MonitoredItems und Subscriptions [2], [8] 14
Abbildung 7: Detaillierte OPC UA Client/Server-Architektur [9] 15
Abbildung 8: Publish/Subscribe-Modell [7] 16
Abbildung 9: Mögliche schematische Kommunikationsstruktur mit Client/Server- und Publish/Subscribe-Modell [5] 17
Abbildung 10: Client/Server-Architektur [2] 17
Abbildung 11: Chained-Server-Architektur für vertikale OPC UA Integration [5] 18
Abbildung 12: Server/Server-Architektur [5] 18
Abbildung 13: Aggregating-Server-Architektur [2] 19
Abbildung 14: Hierarchie und Aufbau von Informationsmodellen [2] 21
Abbildung 15: OPC UA Objekt-Modell [12] 21
Abbildung 16: OPC UA Node-Modell [12] 22
Abbildung 17: OPC UA Server-Architektur [5] 24
Abbildung 18: OPC UA Client-Architektur [5] 28
Abbildung 19: Sicherheitsarchitektur von OPC UA [14] 29
Abbildung 20: Mapping-Möglichkeiten für Client/Server-Architekturen 31
Abbildung 21: Mapping zu einem Service Request vom Client an den Server [1] 31
Abbildung 22: Mögliches Client-Redundanzkonzept [2] 35
Abbildung 23: Prinzip der Transparent Server-Redundanz [2] 36
Abbildung 24: Konzepte der Non-Transparent Server-Redundanz [2] 37
Abbildung 25: Applikations-Architektur 38
Abbildung 26: Software-Layer bei der Client/Server-Interaktion 40
Abbildung 27: Publish/Subscribe Modell 41
Abbildung 28: Struktur MQTT Message [23] 43
Abbildung 29: Verbindungsaufbau zwischen Client und Broker 44
Abbildung 30: Nachrichtentransport von Client zu Broker bzw. Broker zu Client 45
Abbildung 31: Publish von Client zu Broker mit den drei QoS Level 46
Abbildung 32: Subscription eines oder mehrerer Topics 46
Abbildung 33: Priorisierung der Schutzziele im Office- bzw. Automatisierungstechnikbereich [27] 51
Abbildung 34: Elemente eines Cyber Security Management Systems [33] 55
Abbildung 35: Netzwerkstruktur der Ausgangssituation und nach Anbindung von segmentierten Netzwerken von Produktionsstandorten nach IEC 62443 57
Abbildung 36: Prinzipielle Vorgehensweise im Sinne der IEC 62443 59

Abbildung 37: Schwachstellen- und Bedrohungsanalyse von Assets 62
Abbildung 38: Foundational Requirements 63
Abbildung 39: Festlegung der Schutzziele 63
Abbildung 40: Risikomatrix für ein Asset und der dafür relevanten Bedrohungen.....65
Abbildung 41: Definition der Security-Level (SL) [29].....65
Abbildung 42: Zuordnung der zu erreichenden Security Levels (SL-T) 66
Abbildung 43 : Beispiel für ein segmentiertes Netzwerk und Conduits [30].....67
Abbildung 44: Beispiel für ein simples Datenflussdiagramm [31] 69
Abbildung 45: Zuordnung der Assets zu Zonen 69
Abbildung 46: gewählte Schutzmaßnahmen 70
Abbildung 47: Bewertung funktionaler und organisatorischer Maßnahmen mit Protection Level (PL) [29].....72
Abbildung 48: Überprüfung im laufenden Betrieb mit Protection Levels.....72
Abbildung 49: Segmentiertes Netzwerk gemäß der Anforderungen von IEC 62443.74
Abbildung 50: Hauptkomponenten und das Gesamtsystem.....76
Abbildung 51: Schnittstellen des UniPi v1.1 78
Abbildung 52: Siemens S7-1200 – CPU 1212C DC/DC/DC 80
Abbildung 53: Evaluierungsumgebung.....82
Abbildung 54: Das Steuerboard mit den verwendeten Komponenten 83
Abbildung 55: Anschluss der Signalleitungen an die digitalen Eingänge des UniPi v1.1 85
Abbildung 56: Anschluss der Signalleitungen an die digitalen Eingänge der SPS 86
Abbildung 57: Schematische Evaluierungsumgebung – separate Integration von MQTT- und OPC UA-Daten in der Visualisierung 90
Abbildung 58: Konfiguration Firewall-Interface – Extern (WAN).....91
Abbildung 59: Konfiguration Firewall-Interface – Intern (LAN) 91
Abbildung 60: IP-Masquerading und 1:1 NAT.....91
Abbildung 61: Regelsätze der Firewall.....91
Abbildung 62: Programmierung des Raspberry Pi – Node-RED Flow.....93
Abbildung 63: SPS Funktionsbaustein 94
Abbildung 64: TLS-Konfiguration der MQTT-Nodes in Node-RED.....96
Abbildung 65: Erfassung der Maschinendaten vom MQTT-Broker und Weitergabe an die Visualisierung in Node-RED am Laptop 97
Abbildung 66: Abfrage und Definition der Betriebszustände der Maschine.....98
Abbildung 67: Anpassung der Visualisierung von den Maschinenbetriebszuständen im Template-Node.....98
Abbildung 68: Struktur am OPC UA Server.....99
Abbildung 69: OPC UA Client zum Datenabgriff vom OPC UA Server und zur Weitergabe an die Visualisierung in Node-RED.....100
Abbildung 70: Definition der OpcUa-Item-Nodes in Node-RED für Start_Stop-Tag und Qty-Tag 100

Abbildung 71: Konfiguration des OpcUa-Client-Node in Node-RED 101
Abbildung 72: Abfrage und Definition der Betriebszustände der Maschine 101
Abbildung 73: Client-Zertifikat am Server als vertrauenswürdig einstufen 102
Abbildung 74: Struktur des Dashboards in Node-RED 103
Abbildung 75: Ergebnisse der Live-Anzeige von den Maschinendaten 104
Abbildung 76: Lesezugriff auf die Stückzahl für den authentifizierten User am OPC UA Server gesperrt 105
Abbildung 77: Struktur und Inhalte der IEC 62443 [29] 113
Abbildung 78: Zuordnung der Teile der Norm zu den jeweiligen Rollen [35] 114
Abbildung 79: Beispiel eines Datenerhebungsformulars für IACS [33] 116
Abbildung 80: Beispiel eines Netzwerkdiagramms [33] 117
Abbildung 81: Referenz-Modell [30] 124
Abbildung 82: Beispiel eines Asset-Modells [30] 125
Abbildung 83: Beispiel einer Referenzarchitektur [30] 126
Abbildung 84: Beispiel eines Zonen-Modells mit Sub-Zonen [30] 127
Abbildung 85: Beispiel eines separierten Zonen-Modells [30] 127
Abbildung 86: Beispiel für die Dokumentation der Attribute einer Zone [31] 128
Abbildung 87: Schaltplan und Klemmenbezeichnungen des Steuerboards 131
Abbildung 88: TLS-Handshake während des MQTT-Verbindungsaufbaus 137
Abbildung 89: Gegenüberstellung der Authentifizierung bei MQTT ohne TLS und mit TLS 137
Abbildung 90: Gegenüberstellung der Datenübertragung mit MQTT ohne TLS und mit TLS 138
Abbildung 91: Gegenüberstellung der Datenübertragung mit OPC UA ohne und mit Verschlüsselung 139

11 Tabellenverzeichnis

Tabelle 1: Charakteristika von OPC UA [2] 11
Tabelle 2: Service Sets in OPC UA [8] 26
Tabelle 3: MQTT Control Packets 43
Tabelle 4: Parameter des PUBLISH Control Packets 45
Tabelle 5: Top 10 Bedrohungen für IACS aus dem Jahr 2016 [28] 51
Tabelle 6: Systeminformationen des Raspberry Pi und verwendete Software 85
Tabelle 7: Signalzuordnung für das UniPi v1.1 85
Tabelle 8: Signalzuordnung für die SPS 86
Tabelle 9: Informationen der S7-1200 SPS und der verwendeten Software zur Programmierung 87
Tabelle 10: Systeminformationen des Laptops und die eingesetzte Software zur Datenerfassung und Visualisierung 89
Tabelle 11: Services in OPC UA [8] 112
Tabelle 12: Vorschläge für die Strukturanalyse 115
Tabelle 13: Bedrohungen [30] 119
Tabelle 14: FR, SR und RE [29], [35] 123
Tabelle 15: Sammlung von Gegenmaßnahmen 130

12 Abkürzungsverzeichnis

AMQP Advanced Message Queuing Protocol
APIs Application Programming Interfaces
CoAP Constrained Application Protocol
COM Component Object Model
CSMS Cyber Security Management System
DCOM Distributed Component Object Model
DDoS Distributed Denial of Service
DDS Data Distribution Service for Real-Time Systems
DMZ Demilitarisierte Zone, Demilitarized Zone
DoS Denial of Service
DPWS Devices Profile for Web Services
ERP Enterprise Resource Planning
FR Foundational Requirements
FUP Funktionsplan
GDS Global Discovery Server
GPIO General Purpose Input/Output
IACS Industrial Automation and Control Systems
ICMP Internet control message protocol
IIoT Industrial Internet of Things
IoT Internet of Things
KOP Kontaktplan
LDS Local Discovery Server
LDS-ME Local Discovery Server with Multicast Extension
M2M Maschine-zu-Maschine
MES Manufacturing Execution System
ML Maturity Level
MQTT Message Queue Telemetry Protocol
NAT Network Address Translation
OASIS Organization for the Advancement of Structured Information Standards
PL Protection Level
QoS Quality of Service
RE Requirement Enhancement
REST Representational State Transfer
SCADA Supervisory Control and Data Aquisition
SL Security Level
SL-A Achieved Security Level
SL-C Capability Security Level
SL-T Taget Security Level
SOA Service Oriented Architecture
SoC System-on-a-Chip
SR System Requirement
SSH Secure Shell
SSL Secure Socket Layer
ST Strukturierter Text
TLS Transport Layer Security
TSN Time Sensitive Network
UDP User Datagram Protocol
VPN Virtual Private Network
XMPP Extensible Messaging and Presence Protocol

Eidesstaatliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct, insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In– noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Lilienfeld, im September 2018

_____

Thomas, Siedler