

# Implementierung eines digitalen Zwillings für die robocell und Evaluierung eines KI gestützten Konzepts zur Optimierung

## Bachelorarbeit

des Studiengangs Elektrotechnik an der Hochschule Heilbronn

von

Albert Heinke

6. Juli 2025

Bearbeitungszeitraum	4 Monate
Matrikelnummer, Kurs	212081, Elektrotechnik
Ausbildungsfirma	groninger & co. GmbH, Crailsheim
Betreuer der Ausbildungsfirma	Betreuer (Firma)
Gutachter der DHBW Mosbach	Betreuer (DHBW)

## Sperrvermerk

Die vorliegende Arbeit beinhaltet interne vertrauliche Informationen des Unternehmens groninger & Co. GmbH. Sie ist nur für die Beteiligten am Begutachtungs- und Evaluationsprozess bestimmt. Die Weitergabe des Inhalts der Arbeit im Gesamten oder in Teilen sowie das Anfertigen von Kopien oder Abschriften – auch in digitaler Form – sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung des Unternehmens groninger & Co. GmbH.

---

Name, Vorname des Betreuers/Gutachters/Prüfers  
(Bitte in Druckbuchstaben)

---

Crailsheim, 6. Juli 2025

Ort, Datum

---

Unterschrift des  
Betreuer/Gutachter/Prüfer

## **Ehrenwörtliche Erklärung**

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbständig angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Crailsheim, 6. Juli 2025

Ort, Datum

Studierende/r

# Inhaltsverzeichnis

Inhaltsverzeichnis	IV
Abbildungsverzeichnis	VII
Tabellenverzeichnis	VIII
Listings	IX
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung . . . . .	1
1.3 Vorgehensweise . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
2.1 Industrie 4.0 . . . . .	3
2.2 Künstliche Intelligenz . . . . .	5
2.3 Digitaler Zwilling . . . . .	7
2.4 Asset Administration Shell . . . . .	10
2.4.1 Aufbau und Struktur . . . . .	11
2.4.2 Informationsaustausch . . . . .	13
2.4.3 Sicherheit . . . . .	14
2.5 Digitaler Produktpass . . . . .	15
2.6 robocell . . . . .	17
2.7 Technologische Grundlagen . . . . .	18
2.7.1 AASX Package Explorer . . . . .	18
2.7.2 Eclipse BaSyx . . . . .	18
2.7.3 OPC Unified Architecture . . . . .	20
<b>3 Entwicklung</b>	<b>21</b>
3.1 Konzeptionierung des digitalen Zwilling . . . . .	21
3.1.1 Identifikation relevanter Datenquellen . . . . .	21
3.1.2 Auswahl geeigneter Teilmodelle . . . . .	22
3.2 Modellierung mit der AAS . . . . .	24
3.2.1 Umsetzung mit dem Package Explorer . . . . .	24
3.2.2 Validierung . . . . .	26
3.3 Technische Integration . . . . .	27
3.3.1 Bereitstellung der AAS . . . . .	27
3.3.2 Integration von Echtzeitdaten über OPC UA . . . . .	29
3.3.3 Verarbeitung von Zeitreihendaten . . . . .	31
3.4 Anwendungsfall Digitaler Produktpass . . . . .	33
3.4.1 Beschreibung . . . . .	33
3.4.2 Umsetzung mit dem Teilmodell Carbon Footprint . . . . .	33
3.5 Anwendungsfall automatisierte Generierung von AAS . . . . .	34
3.5.1 Erstellen von Submodel Templates . . . . .	34
3.5.2 Automatisiertes Befüllen mit strukturierten Daten . . . . .	35

3.5.3	Potenziale des KI-Einsatzes . . . . .	37
<b>4</b>	<b>Ergebnisse</b>	<b>37</b>
4.1	AAS-Demonstrator für die robocell . . . . .	37
4.1.1	Systemarchitektur . . . . .	37
4.1.2	Eingesetzte Teilmodelle . . . . .	37
4.1.3	Herausforderungen bei der Erstellung . . . . .	37
4.2	Anwendungsfall Digitaler Produktpass . . . . .	37
4.2.1	Implementierungskonzept . . . . .	37
4.2.2	Dynamische Berechnung des PCF . . . . .	37
4.3	Anwendungsfall automatisierte Generierung der AAS . . . . .	37
4.4	Einsatzmöglichkeiten von KI im Kontext der Verwaltungsschale . . .	37
4.4.1	Generierung von Verwaltungsschalen . . . . .	37
4.4.2	Anomaliererkennung . . . . .	37
4.4.3	Weiterführende Einsatzmöglichkeiten . . . . .	37
4.5	Evaluierung eingesetzter Tools und Software . . . . .	37
4.5.1	AASX Package Explorer . . . . .	37
4.5.2	Eclipse AASX Server . . . . .	37
4.5.3	BaSyx . . . . .	37
4.5.4	Mnestix Browser . . . . .	37
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>37</b>
5.1	Zusammenfassung der Arbeit . . . . .	37
5.2	Handlungsempfehlung für groninger . . . . .	37
5.3	Ausblick auf zukünftige Entwicklungen . . . . .	37
	<b>Glossar und Abkürzungsverzeichnis</b>	<b>38</b>
<b>A</b>	<b>Anhang 1</b>	<b>43</b>

# Abkürzungsverzeichnis

**AAS** Asset Administration Shell

**ABAC** Attribute Based Access Control

**Asset-ID** Asset Identifier

**BOM** Bill of Material

**CAD** Computer Aided Design

**CD** Concept Description

**DPP** Digitaler Produktpass

**ERP** Enterprise Resource Planing

**IDTA** Industrial Digital Twin Association

**KI** Künstliche Intelligenz

**MES** Manufacturing Execution System

**OPC UA** Open Platform Communications Unified Architecture

**PLM** Product Lifecycle Management

**RAMI 4.0** Referenzarchitekturmodell Industrie 4.0

## Abbildungsverzeichnis

1	Klassifizierung digitaler Zwilling . . . . .	8
2	Industrie 4.0 Komponente . . . . .	10
3	Vereinfachtes Metamodell der AAS . . . . .	12
4	Konzept des DPP 4.0 (eigene Darstellung in Anlehnung an [31]) . . .	16
5	Benutzeroberfläche AASX Package Explorer . . . . .	18
6	AAS-Konzept der robocell . . . . .	24
7	Bearbeitungsansicht eines Submodells im Package Explorer . . . . .	25
8	Konsolenausgabe nach erfolgreicher Validierung . . . . .	26
9	Metadaten-Konfiguration im Submodell Time Series Data . . . . .	31
10	Konfiguration des LinkedSegments im Submodell Time Series Data .	32

## Tabellenverzeichnis

1	REST-Endpunkte in Eclipse BaSyx . . . . .	19
2	Submodelle mit typischen Inhalten . . . . .	23
3	Bereitstellung einer AAS über die REST-API in Eclips BaSyx . . . .	28



## Listings

1	Beispielhafte JSON-Konfiguration einer Datenquelle . . . . .	30
2	Beispielhafte JSON-Konfiguration einer Datensenke . . . . .	30



# 1 Einleitung

## 1.1 Motivation

Im Rahmen von Industrie 4.0 gewinnen digitale Zwillinge zunehmend an Bedeutung. Sie ermöglichen die digitale Abbildung physischer Assets und sollen helfen, Prozesse transparenter und effizienter zu gestalten. Viele Unternehmen setzen bereits auf proprietäre Lösungen, die jedoch häufig nicht interoperabel sind und somit den Austausch von Daten über System -und Unternehmensgrenzen hinweg erschweren. Mit der steigenden Nachfrage nach digitalen Zwillingen wächst auch der Bedarf an standardisierten und interoperablen Lösungen. In den vergangenen Jahren hat sich die Industrial Digital Twin Association (IDTA) intensiv mit dieser Thematik auseinandergesetzt und als Lösung die sogenannte Asset Administration Shell (AAS) etabliert. Diese ermöglicht eine interoperable Umsetzung digitaler Zwillinge, wobei sämtliche relevanten Informationen zu einem Asset über dessen gesamten Lebenszyklus hinweg digital erfasst, strukturiert und dokumentiert werden können. Die zunehmende Beschäftigung zahlreicher Unternehmen mit dem Konzept der AAS, die Umsetzung in Pilotprojekten und ersten produktiven Anwendungen unterstreichen die hohe Relevanz und das Potenzial dieser standardisierten Lösung für die Digitalisierung und Vernetzung industrieller Systeme.

Aber auch regulatorische Anforderungen wie der digitale Produktpass, der von der EU entwickelt und künftig für viele Produktgruppen verpflichtend wird, zeigen die Notwendigkeit einheitlicher digitaler Zwillinge. Um am Puls der Zeit zu bleiben und den wachsenden Anforderungen an Transparenz, Rückverfolgbarkeit und Effizienz gerecht zu werden, ist es für Unternehmen unerlässlich, sich frühzeitig mit der AAS auseinanderzusetzen.

## 1.2 Zielsetzung

Ziel dieser Arbeit ist die prototypische Modellierung eines digitalen Zwillings für das Abfüll- und Verschleißmodul der robocell-Linie der Firma groninger. Dabei soll untersucht werden, ob der Einsatz der AAS für das Unternehmen einen Mehrwert bietet - insbesondere im Hinblick auf Interoperabilität, Produktivität und Nachverfolgbarkeit. Unter Interoperabilität wird dabei die herstellerunabhängige und standardisierte Kommunikation zwischen Maschinen und Systemen verstanden, die eine nahtlose Vernetzung über Unternehmensgrenzen hinweg ermöglicht. Hinsichtlich der Produktivität steht die effiziente Bereitstellung und der Austausch von Informationen im

Fokus, mit Potenzialen zur Optimierung von Prozessen, etwa durch vorausschauende Instandhaltungsmaßnahmen. Die Nachverfolgbarkeit schließlich umfasst die lückenlose Erfassung und Rückverfolgung aller relevanten Asset-Daten über den gesamten Lebenszyklus, wodurch Zustandsänderungen und Ereignisse transparent und nachvollziehbar werden. Für die Umsetzung sollen verschieden Open-Source-Tools zum Einsatz kommen, die unter anderem von der IDTA bereitgestellt oder im Rahmen des Eclipse-Projekts entwickelt wurden, wie etwa der AASX Package Explorer, die AASX Test Engine oder Eclipse BaSyx. Neben der Evaluierung dieser Werkzeuge soll zudem eine mögliche Architektur zur Verwaltung und Bereitstellung AAS-basierter digitaler Zwillinge aufgezeigt werden. Ebenfalls soll die Einsatzmöglichkeit von Künstlicher Intelligenz im Kontext der AAS diskutiert werden, um potenzielle Optimierungsansätze und weiterführende Anwendungsmöglichkeiten zu identifizieren. Auf Grundlage der gewonnen Erkenntnisse soll abschließend eine Handlungsempfehlung für groninger formuliert werden, die den möglichen Nutzen und die Relevanz des Einsatzes der AAS für das Unternehmen bewertet.

### 1.3 Vorgehensweise

Die Arbeit beginnt mit einer eingehenden Betrachtung grundlegender Konzepte, die für das Verständnis der Arbeit notwendig sind. Dabei wird insbesondere die Frage geklärt, was die Verwaltungsschale und der digitale Zwilling überhaupt sind und wie diese sich im industriellen Kontext - insbesondere im Rahmen von Industrie 4.0 - einordnen.

Anschließend wird auf die Entwicklung des digitalen Zwillings der robocell eingegangen. Dabei werden verschiedene Bereiche der Maschine betrachtet und in verschiedenen Submodellen der AAS modelliert. Unter anderem wird ein Anwendungsfall für den digitalen Produktpass sowie das automatisierte Erstellen der AAS umgesetzt. Aber auch das allgemeine Vorgehen bei der Erstellung eines digitalen Zwillings mithilfe der Verwaltungsschale wird aufgezeigt. Für die technische Umsetzung kommen verschiedene Tools und Technologien zum Einsatz, wie zum Beispiel der AASX Package Explorer zum Erstellen oder BaSyx als AAS-Laufzeitumgebung zum Verwalten der verschiedenen Verwaltungsschalen.

Im Ergebnisteil wird der AAS-Demonstrator für die robocell vorgestellt. Es werden sowohl die Systemarchitektur als auch die modellierten Teilmodelle präsentiert. Aber auch Herausforderungen bei der Implementierung werden beleuchtet. Zudem wird diskutiert, welche Potenziale Künstliche Intelligenz (KI) im Zusammenhang mit der AAS bietet. Außerdem werden die eingesetzten Open-Source-Lösungen zur

Modellierung und Verwaltung der AAS evaluiert.

Es folgt eine Zusammenfassung der Arbeit, bei der die wichtigsten Aspekte kurz zusammengefasst werden. Zudem wird eine Handlungsempfehlung für groninger formuliert und ein Ausblick auf zukünftige Entwicklungen geliefert.

## 2 Grundlagen

Das folgende Kapitel stellt die theoretischen und technologischen Grundlagen vor, die für das Verständnis und die Umsetzung dieser Arbeit erforderlich sind. Es beginnt mit einer Einführung in die Industrie 4.0, und vermittelt ein grundlegendes Verständnis für das Konzept des digitalen Zwillings. Anschließend werden die Struktur und Funktion der AAS sowie die Rolle des digitalen Produktpasses erläutert. Den Abschluss bilden die technologischen Voraussetzungen für die praktische Umsetzung, darunter beispielsweise die Open-Source-Plattform Eclipse BaSyx und der Kommunikationsstandard OPC UA.

### 2.1 Industrie 4.0

Der Begriff Industrie 4.0 wurde erstmals im Jahr 2011 im Rahmen eines von der deutschen Bundesregierung initiierten Zukunftsprojekts eingeführt, das auf die Förderung der Informatisierung in der industriellen Fertigung/Produktion abzielt. Angestrebt wird eine Stärkung der Wettbewerbsfähigkeit der deutschen Industrie sowie eine Verbesserung der Marktposition deutscher Unternehmen im globalen Wettbewerb. Industrie 4.0 steht dabei für die vierte industrielle Revolution und beschreibt die umfassende digitale Transformation industrieller Wertschöpfungsprozesse. Im Zentrum steht die intelligente Vernetzung von Menschen, Maschinen und Produkten über moderne digitale Kommunikationsnetzwerke, durch die eine weitreichende Integration der physischen mit der digitalen Welt ermöglicht wird.

Zur besseren Einordnung von Industrie 4.0 ist ein Blick auf die vorangegangenen industriellen Revolutionen hilfreich. Die Industrialisierung begann bereits Mitte des 18. Jahrhunderts in Großbritannien und breitete sich von dort an weltweit aus. Mit der Entwicklung der ersten Dampfmaschine setzte die erste industrielle Revolution ein. Sie ermöglichte erstmals die Mechanisierung der Fertigung durch den Einsatz von Arbeits- und Kraftmaschinen. Dadurch konnten manuelle Tätigkeiten zunehmend durch Maschinenkraft ersetzt werden, insbesondere in der Textil-, Eisen- und Stahlindustrie, die zu den ersten Branchen gehörten, die von dieser Entwicklung profitierten. Die zweite industrielle Revolution setzte gegen Ende des 19. Jahrhunderts ein und

war maßgeblich durch den flächendeckenden Einsatz von Elektrizität geprägt. Mit der Erfindung elektrischer Antriebe und des Verbrennungsmotors konnten Maschinen nun auch dezentral betrieben werden. Sie waren nicht länger auf zentrale Kraftquellen wie Dampfmaschinen angewiesen. Dies ermöglichte eine flexiblere Gestaltung von Produktionsstätten und führte zur Entwicklung einer arbeitsteiligen Massenproduktion mithilfe von Fließ- und Förderbändern. Ausgehend von dem deutschen Wirtschaftswunder in den sechziger Jahren des 20. Jahrhunderts entstand in den folgenden Jahrzehnten die dritte industrielle Revolution. Diese zeichnet sich vor allem durch den Einsatz elektronischer Systeme sowie der Informations- und Kommunikationstechnologie zur Automatisierung aus und ist noch bis heute wirksam. Aufbauend auf den vorangegangenen industriellen Revolutionen strebt die vierte industrielle Revolution eine tiefgreifende Transformation industrieller Produktionsprozesse an. Im Fokus steht dabei die Vernetzung von Systemen, die über moderne Internettechnologien miteinander kommunizieren. Ziel dieser Entwicklung ist es, die industrielle Wertschöpfung deutlich flexibler und effizienter zu gestalten sowie eine stärkere Individualisierung von Produkten zu ermöglichen. [1][2]

Trotz der weiten Verbreitung des Begriffs Industrie 4.0 mangelt es in der Literatur und Forschung an einer einheitlichen Definition. Vor diesem Hintergrund nimmt insbesondere die Plattform Industrie 4.0 [3] eine zentrale Rolle in Deutschland ein. Dabei handelt es sich um eine gemeinsame Initiative des Bundesministeriums für Wirtschaft und Energie, des Bundesministerium für Forschung, Technologie und Raumfahrt sowie führender Industrieverbände (z.B. VDMA, Bitkom, ZVEI), Unternehmen, Forschungseinrichtungen und Gewerkschaften.

Die Plattform ist maßgeblich an der inhaltlichen und strategischen Ausarbeitung der Industrie 4.0 beteiligt und leistet einen entscheidenden Beitrag zur Begriffsdefinition. Sie definiert Industrie 4.0 als „die vierte industrielle Revolution, einer neuen Stufe der Organisation und Steuerung der gesamten Wertschöpfungskette über den Lebenszyklus von Produkten. Dieser Zyklus orientiert sich an den zunehmend individualisierten Kundenwünschen und erstreckt sich von der Idee, dem Auftrag über die Entwicklung und Fertigung, die Auslieferung eines Produkts an den Endkunden bis hin zum Recycling, einschließlich der damit verbundenen Dienstleistungen. Basis ist die Verfügbarkeit aller relevanten Informationen in Echtzeit durch Vernetzung aller an der Wertschöpfung beteiligten Instanzen sowie die Fähigkeit aus den Daten den zu jedem Zeitpunkt optimalen Wertschöpfungsfluss abzuleiten. Durch die Verbindung von Menschen, Objekten und Systemen entstehen dynamische, echtzeitoptimierte und selbst organisierende, unternehmensübergreifende Wertschöpfungsnetzwerke, die

sich nach unterschiedlichen Kriterien wie beispielsweise Kosten, Verfügbarkeit und Ressourcenverbrauch optimieren lassen “ [4, S. 8].

Die Definition der Plattform Industrie 4.0 verdeutlicht: Industrie 4.0 steht für einen grundlegenden Wandel in der industriellen Wertschöpfung. Im Zentrum stehen dabei nicht nur neue technologische Möglichkeiten, sondern vor allem das Potenzial, Produktions- und Geschäftsprozesse flexibler, effizienter und nachhaltiger zu gestalten. Durch die intelligente Vernetzung und die Nutzung von Echtzeitdaten entstehen selbstorganisierende Systeme, die auch über Unternehmensgrenzen hinweg kooperieren und tiefgreifende organisatorische Veränderungen erfordern. Auf diese Weise werden nicht nur Qualität und Transparenz gesteigert, sondern auch Ressourcen geschont und Kosten reduziert. Insgesamt trägt Industrie 4.0 somit nicht nur zur Verbesserung der Wirtschaftlichkeit bei, sondern leistet auch einen wichtigen Beitrag zur ökologischen Nachhaltigkeit.

## 2.2 Künstliche Intelligenz

Mit der zunehmenden Vernetzung von Maschinen und Anlagen entstehen immer größere Datenmengen entlang von Produktionsprozessen. Diese Daten bieten enorme Potenziale zur Optimierung industrieller Abläufe, beispielsweise zur Steigerung der Anlagenverfügbarkeit, zur Reduzierung von Ausschuss oder zur Verbesserung der Produktqualität. Zudem erlauben intelligente Produkte, dass Hersteller auch während der Nutzungsphase Informationen über den Einsatz und Zustand eines Produktes erhalten. Auf Basis dieser lassen sich neue Dienstleistungen und Anwendungen entwickeln, wie etwa die vorrauschaunde Instandhaltung oder die dynamische Optimierung von Betriebsparametern.

Damit solche datengetriebenen Ansätze erfolgreich umgesetzt werden können, benötigt es leistungsfähige Technologien zur Analyse und Interpretation. Genau hier setzt KI an. Insbesondere Methoden des maschinellen Lernens erlauben es, Zusammenhänge in großen Datenmengen zu erkennen, Zustände zu klassifizieren und sogar Vorhersagen über zukünftige Systementwicklungen zu treffen. [5] KI nimmt somit eine zentrale Rolle innerhalb der Industrie 4.0 ein, indem sie einen echten Mehrwert aus den gesammelten Daten generiert.

Allgemein bezeichnet der Begriff KI dabei eine Vielzahl von Methoden und Technologien, die es Computern ermöglichen, Aufgaben zu bewältigen, für die normalerweise menschliche Intelligenz erforderlich ist [6]. Dazu zählen insbesondere die Fähigkeit, aus Erfahrungen (Daten) zu lernen, Problemlösungen zu entwickeln sowie selbstständig

Entscheidungen zu treffen [7].

Ein zentrales Teilgebiet der KI ist das maschinelle Lernen (ML). Es umfasst Algorithmen, die selbstständig aus Daten lernen und darauf basierend Vorhersagen treffen können. Im Gegensatz zu klassischen Computerprogrammen folgt das Programm keinem Lösungsweg. Stattdessen erkennt es Muster in den vorliegenden Daten und kann auf Grundlage dieser bestimmte Aufgaben ausführen [8]. Aufgrund seiner zentralen Bedeutung für zahlreiche Anwendungen gilt maschinelles Lernen häufig auch als Schlüsseltechnologie der KI [9].

Grundsätzlich wird zwischen überwachtem Lernen (Supervised Learning) und unüberwachtem Lernen (Unsupervised Learning) unterschieden. Beim überwachten Lernen wird ein Modell anhand eines Datensatzes trainiert, der sowohl Eingabewerte als auch die dazugehörigen Ziel- bzw. Ausgabewerte enthält. Der Algorithmus lernt dabei, Zusammenhänge und Muster in den Daten zu erkennen, um daraus eine Funktion abzuleiten, die neue Eingaben korrekt vorhersagen kann. Typische Anwendungsbereiche sind die Klassifikation (z. B. die Einteilung von Bildern in Kategorien) und die Regression (z. B. die Vorhersage numerischer Werte wie Temperatur oder Energieverbrauch). Beim unüberwachtem Lernen hingegen wird ein Modell mit ungelabelten Daten trainiert, das heißt ohne vorgegeben Zielwerte. Der Algorithmus versucht selbstständig Strukturen oder Muster in den vorgegeben Daten zu finden. Ein häufig eingesetztes Verfahren ist die Clusteranalyse, bei der ähnliche Datenpunkte zu Gruppen, sogenannten Clustern, zusammengefasst werden. Eine typische Anwendung ist beispielsweise die Anomalieerkennung, bei der Abweichungen von einem erwarteten Normalverhalten erkannt werden können.

Ein besonders leistungsfähiger Ansatz innerhalb des maschinellen Lernens ist das sogenannte Deep Learning. Es basiert auf künstlichen neuronalen Netzen, die in ihrer Struktur an das Verhalten von Neuronen im menschlichen Gehirn angelehnt sind. Während klassische maschinelle Lernmodelle meist aus nur wenigen Schichten bestehen, kommen beim Deep Learning sogenannte tiefe neuronale Netze mit vielen hintereinandergeschachtelten Schichten zum Einsatz. Diese ermöglichen das Erkennen hochkomplexer und abstrakter Zusammenhänge in großen Datenmengen, etwa bei der Bilderkennung oder der Sprachverarbeitung. [10] Ein besonders aktuelles Beispiel für die Anwendung tiefer neuronaler Netzwerke ist die generative KI. Sie beschäftigt sich mit der Erstellung neuer Inhalte wie Text oder Sprache. Dabei kommen verschiedene Methoden des Deep Learnings zum Einsatz, um aus großen Datenmengen zu lernen, wie neue Inhalte erzeugt werden können [11].



## 2.3 Digitaler Zwilling

Digitale Zwillinge gelten als eine der Schlüsseltechnologien der Industrie 4.0. Als digitales Gegenstück eines physischen Objekts, sei es eine Maschine ein Produkt oder eine komplette Anlage, bilden sie dessen Zustand, Verhalten und Leistung virtuell ab. Dadurch ermöglichen sie eine konsistente Erfassung von Daten, das Simulieren von Prozessen und das frühzeitige Erkennen von Optimierungspotenzialen.

Das Konzept selbst wurde erstmals von Michael Grieves im Jahr 2003 in einer Präsentation zum Product Lifecycle Management (PLM) vorgestellt. Grieves definierte drei grundlegende Komponenten [12], die zusammen das Informationsmodell des digitalen Zwilling bilden:

- ein reales Objekt in der physischen Welt,
- ein digitales Abbild dieses Objekts in einem virtuellen Raum, sowie
- eine Schnittstelle, die den Informationsfluss zwischen diesen beiden ermöglicht.

Auf Basis des von Grieves entwickelten Informationsmodells hat sich der Begriff des digitalen Zwillings kontinuierlich weiterentwickelt. Aufgrund verschiedener Fachgebiete und inkonsistenter Definitionen haben sich in der Vergangenheit allerdings eine Vielzahl unterschiedlicher Ausprägungen des Begriffs gebildet. Diese unterscheiden sich insbesondere in der Tiefe der Datenintegration zwischen dem physischen Objekt und seinem virtuellen Abbild. Während ein einfacher digitaler Zwilling lediglich ein einfaches Modell mit statischen Daten ist, ermöglichen fortgeschrittene Zwillinge einen bidirektionalen Datenaustausch zwischen physischem und virtuellem Objekt.

Für ein besseres Verständnis und zur tieferen Klassifizierung ist es zunächst hilfreich, zwischen Typen und Instanzen des digitalen Zwillings zu unterscheiden. Typen sind allgemeine Abbilder, die grundlegende Eigenschaften und Verhaltensmodelle einer Produktgruppe beschreiben. Sie können mit einer Klasse in der Softwareentwicklung verglichen werden, die als Vorlage für konkrete Instanzen dienen. Typen können zum Beispiel einen bestimmten Maschinentyp hinsichtlich Aufbau, Struktur oder Schnittstellen beschreiben, ohne dabei Bezug zu einer einzelnen physischen Maschine zu nehmen. Instanzen wiederum sind einzigartig, und beschreiben ein konkretes Produkt, wie zum Beispiel eine Maschine, die einzigartig über eine Seriennummer identifizierbar ist. Häufig sind Instanzen Ausprägungen eines Types mit einer Verbindung zu einem realen Objekt, wodurch beispielsweise die Überwachung des Zustands einer Maschine ermöglicht wird. Analog zur Softwareentwicklung können diese als instanziiertes Objekt einer Klasse gesehen werden. [13]

Je nach Art des Informationsflusses sowie dem Grad der Ausprägung der Verbindung zur realen Welt werden Instanzen digitaler Zwillinge häufig in drei Kategorien eingeteilt: das digitale Modell, den digitalen Schatten und den digitalen Zwilling [14]. Obwohl diese Begriffe im allgemeinen Sprachgebrauch oft synonym verwendet werden, unterscheiden sie sich deutlich hinsichtlich ihrer Funktion und Kopplung zum realen Objekt. Digitale Modelle sind statische Abbilder physischer Objekte, haben jedoch keine Verbindung zu diesen. Oft werden sie zur Veranschaulichung oder Konstruktion genutzt, wie zum Beispiel ein 3D-Modell einer Maschine. Zwar können reale Daten, wie etwa Maße oder Materialeigenschaften einer Anlage oder Maschine in ein solches Modell integriert werden, allerdings erfolgt die Eingabe dabei immer manuell. Änderungen an dem realen Objekt werden nicht automatisch aktualisiert und bleiben somit ohne Einfluss auf das digitale Modell. Der digitale Schatten ergänzt das digitale Modell um eine unidirektionale Verbindung zum realen Objekt. Dabei fließen Daten des physischen Objekts meist in Echtzeit über zum Beispiel geeignete Sensoren zum digitalen Objekt. Der Schatten bildet den aktuellen Zustand des Objekts ab, hat aber keine Rückkopplung zu diesem. Ein typisches Beispiel für einen digitalen Schatten wäre das Condition Monitoring, wobei der Zustand einer Maschine mit geeigneten Sensoren abgebildet wird. Mit einer aktiven Rückkopplung zum realen Objekt wird der digitale Schatten zum digitalen Zwilling. Es entsteht eine Feedback-Schleife und erlaubt dem virtuellen Objekt Einfluss auf das reale System zu nehmen.

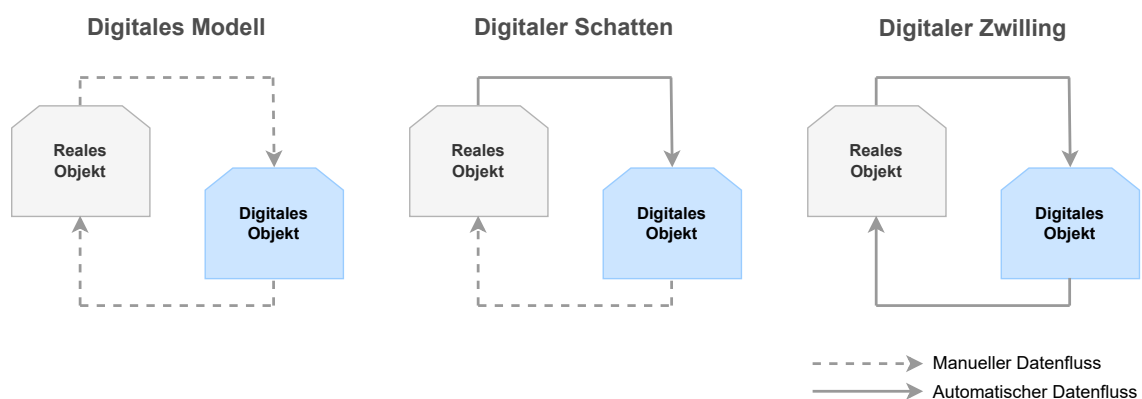


Abbildung 1: Klassifizierung digitaler Zwilling

Im industriellen Umfeld werden digitale Zwillinge in vielen unterschiedlichen Bereichen genutzt. Sie kommen entlang des gesamten Lebenszyklus eines Produkts oder Systems zum Einsatz - von der Entwicklung über die Produktion bis hin zum Betrieb und der Wartung. Dabei ist jedoch zu beachten, dass in der Praxis häufig auch digitale Modelle oder digitale Schatten als digitale Zwillinge bezeichnet werden, obwohl sie

technisch gesehen nicht alle Merkmale eines echten digitalen Zwillings aufweisen und somit ebenfalls nicht ihr volles Potenzial entfalten.

Bereits bei der Entwicklung von Produkten können digitale Zwillinge einen erheblichen Vorteil bieten. Indem bereits frühzeitig digitale Modelle oder Simulationen eingesetzt werden, entfällt die Notwendigkeit physischer Prototypen, was Entwicklungszeiten und Kosten deutlich senken kann. Während der Produktion ermöglichen sie eine durchgängige Überwachung, Analyse und Optimierung von Fertigungsprozessen durch die Integration von Echtzeitdaten. Sie unterstützen die virtuelle Inbetriebnahme von Maschinen und dienen als Grundlage für die vorausschauende Wartung (Predictive Maintenance), wodurch Stillstandzeiten einer Maschine reduziert werden können. Nicht zuletzt dienen digitale Zwillinge als zentrale Datenplattform, in der alle relevanten Informationen aus verschiedenen Datenquellen gebündelt werden. Sie bilden somit eine konsistente Datenbasis und können beispielsweise bei dem Entwicklungsprozess eines Produktes helfen. [15]

Die Implementierung digitaler Zwillinge erweist sich in der Praxis jedoch oftmals als sehr anspruchsvoll. Eine zentrale Herausforderung ist die fehlende Interoperabilität zwischen verschiedenen IT-Systemen. Sowohl innerhalb eines Unternehmens als auch unternehmensübergreifend bilden sich dadurch häufig voneinander isolierte Datenbestände, die nicht systemübergreifend nutzbar sind. Solche sogenannten Informationssilos können die Umsetzung eines konsistenten digitalen Zwillings erheblich erschweren, da die relevanten Informationen und Daten zunächst aus unterschiedlichen Systemen wie Enterprise Resource Planning (ERP), Manufacturing Execution System (MES) oder Computer Aided Design (CAD) zusammengeführt werden müssen. Hinzu kommt, dass diese Daten oftmals in unterschiedlichen, nicht standardisierten Formaten vorliegen, was eine automatisierte Integration zusätzlich erschwert. Diese Problematik zeigt sich nicht nur innerhalb einzelner Unternehmen, sondern auch entlang der gesamten Wertschöpfungskette, etwa wenn verschiedene Akteure einer Lieferkette heterogene Datenformate und proprietäre Austauschprotokolle verwenden. Ein digitaler Zwilling, der in einem Unternehmen A erstellt wurde, kann dadurch von einer Anwendung oder einem weiteren digitalen Zwilling eines Unternehmens B nicht ohne Weiteres interpretiert oder verwendet werden. Es ist daher essenziell, digitale Zwillinge in einem interoperablen Format bereitzustellen, um eine einheitliche Interpretation und Nutzung auch über Unternehmensgrenzen hinweg zu ermöglichen. [16]

### 2.4 Asset Administration Shell

Die Asset Administration Shell (AAS) - deutsch Verwaltungsschale - ist eine Schlüsselkomponente innerhalb des Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0) [17] und bildet die Grundlage für die Umsetzung und Entwicklung interoperabler digitaler Zwillinge im industriellen Umfeld. Sie wurde maßgeblich von der Plattform Industrie 4.0 entwickelt und erstmalig im Jahr 2016 als Teil von RAMI 4.0 vorgestellt. Seit ihrer Einführung wurde die AAS kontinuierlich weiterentwickelt und ist mittlerweile in der internationalen Norm IEC 63278-1 [18] standardisiert.

Seit 2020 wird die Umsetzung und Weiterentwicklung der AAS von der Industrial Digital Twin Association (IDTA) [19] organisiert und gesteuert. Ziel der IDTA ist es, den digitalen Zwilling auf Basis der AAS zu standardisieren und in Form von Open-Source-Softwarelösungen in das industrielle Umfeld zu integrieren. Die AAS wird dabei in mehreren Spezifikationen der IDTA dokumentiert und beschrieben. Aktuell bildet die AAS Version 3 den neuesten Entwicklungsstand und ist ebenfalls die Basis für diese Arbeit.

Die AAS repräsentiert ein Asset digital, indem sie alle relevanten Daten, Eigenschaften und Funktionen über den gesamten Lebenszyklus hinweg in strukturierter und standardisierter Form bereitstellt. Sie fungiert somit als digitales Gegenstück eines realen Objekts - also als digitaler Zwilling. Die Informationen sind in sogenannten Submodellen organisiert, die jeweils spezifische Aspekte eines Assets abbilden. Dabei kann es sich sowohl um physische Assets (z.B. Maschinen, Anlagen) als auch um virtuelle Assets (z.B. Software, Konzepte) handeln. Eine AAS ist dabei stets einem Asset zugeordnet und global eindeutig identifizierbar. Durch die Kombination eines Assets mit seiner AAS entsteht eine sogenannte Industrie-4.0-Komponente.

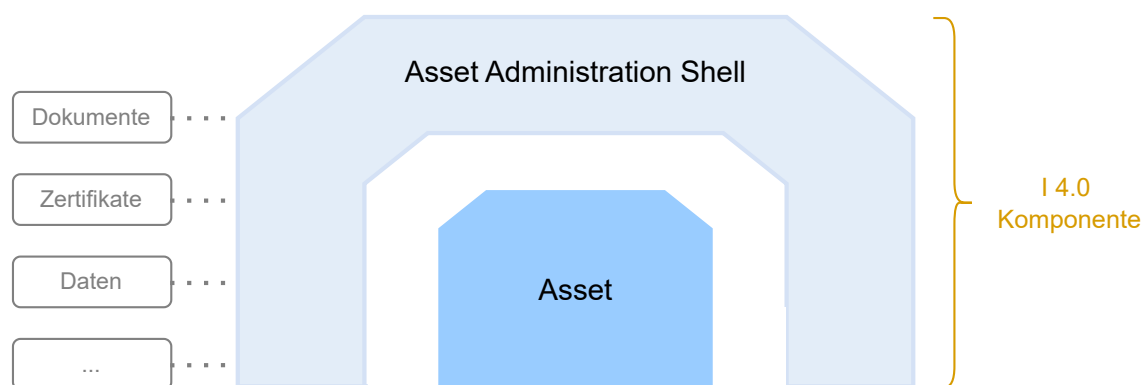


Abbildung 2: Industrie 4.0 Komponente

### 2.4.1 Aufbau und Struktur

Allgemein kann zwischen Typ -und Instanz-Verwaltungsschalen unterschieden werden. Typ-AAS beschreiben allgemeine Eigenschaften eines Produkttypen oder einer Produktklasse wie einen bestimmten Maschinen-Typ, während Instanz-Verwaltungsschalen immer einem spezifischen Objekt zugeordnet werden. Typen können zum Beispiel allgemeine Dokumente, Eigenschaften oder Merkmale enthalten, die für eine bestimmte Maschinenart gelten. Sie dienen als standardisierte, wiederverwendbare Vorlage für das Erstellen von Instanzen. Im Gegensatz dazu werden Instanzen immer einem konkreten physischen Objekt zugewiesen - etwa durch die Seriennummer, den aktuellen Standort oder den Betriebszustand einer Maschine.

Bestimmte Aspekte eines Assets werden gemäß der Spezifikation des Metamodells der AAS [20] in verschiedenen Submodellen verwaltet. Man kann sich dies wie ein Schubladensystem vorstellen, wobei jede Schublade einen bestimmten Bereich des Assets abdeckt - beispielsweise die technischen Stammdaten, das Typenschild, Wartungsinformationen oder Zustandswerte einer Maschine. Die Auswahl und Struktur der Submodelle ist domänenspezifisch und hängt stark vom konkreten Asset bzw. Anwendungsfall ab. Dabei kann eine AAS beliebig viele Submodelle enthalten, die bei Bedarf auch erweitert werden können.

Die Daten innerhalb eines Submodells werden in verschiedenen Submodellelementen strukturiert. Diese umfassen Dateneigenschaften, Operationen sowie weitere Substrukturelemente die für eine umfassende Beschreibung eines digitalen Modells eines Assets erforderlich sind. Das vermutlich am häufigsten verwendete Datenelement ist das Submodellelement Property. Es lässt sich mit einer Variablen aus der Softwareentwicklung vergleichen, da es einfache Merkmale wie etwa einen Namen oder eine Seriennummer repräsentiert und dabei über einen definierten Datentyp wie String, Integer oder Boolean verfügt. Neben Properties spielt zudem das Submodellelement File eine besonders wichtige Rolle. Es ermöglicht das Einbetten oder Referenzieren von Dateien in die AAS. Dabei werden gängige Dateiformate wie PDF, JPG oder STEP unterstützt, was besonders für technische Dokumentationen oder CAD-Modelle von Bedeutung ist. Neben diesen Datenelementen existieren noch weitere Submodellelemente, die spezifische Funktionen ermöglichen. So erlaubt beispielsweise das RelationshipElement die Modellierung von Beziehungen oder das ReferenceElement die Referenzierung von internen oder externen Inhalten. Zur besseren Veranschaulichung des zugrundeliegenden Metamodells wird dieses nachfolgend in einer vereinfachten Form dargestellt.

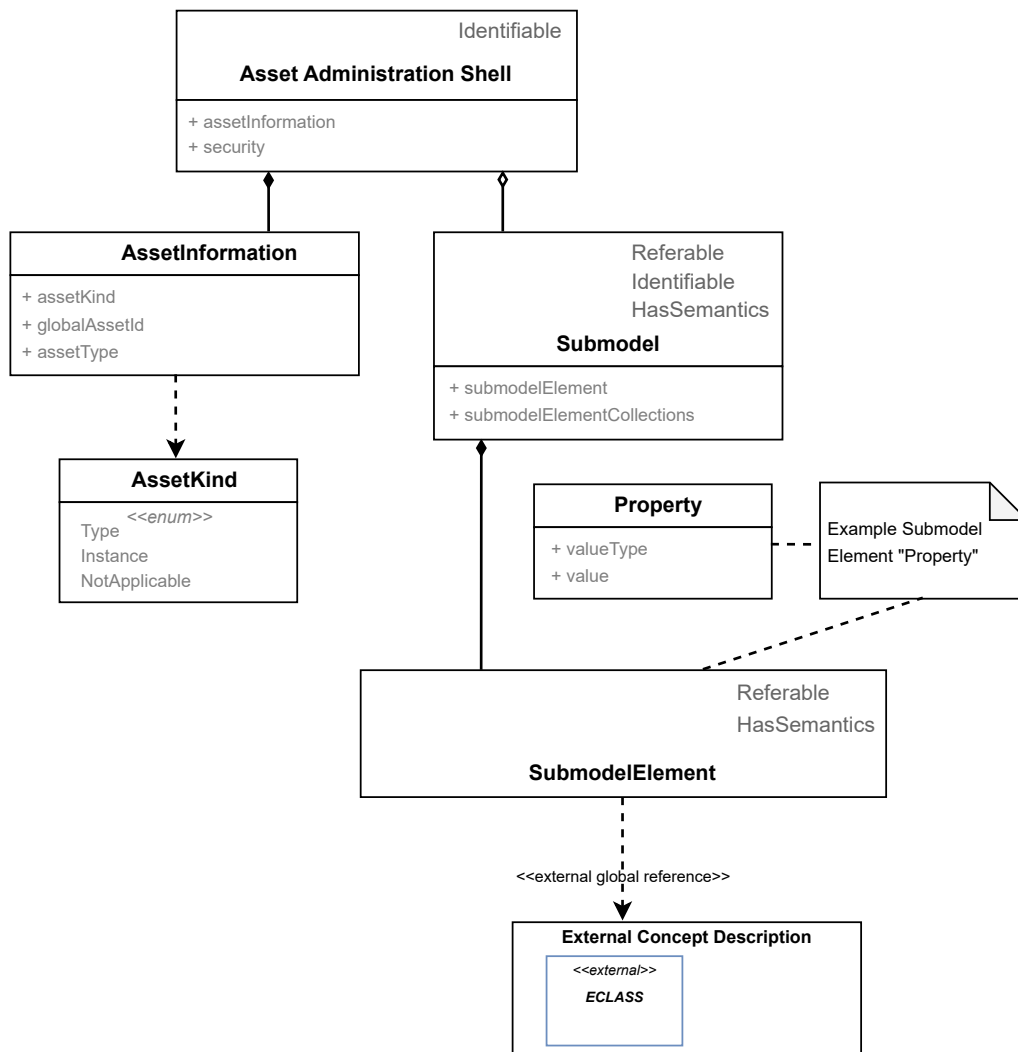


Abbildung 3: Vereinfachtes Metamodell der AAS

Wichtig ist, sowohl die AAS selbst als auch ihre Submodelle müssen global eindeutig identifizierbar sein. Dies wird durch die Verwendung von eindeutigen Identifikatoren der Klasse **Identifiable** wie einer URI (Uniform Resource Identifier) oder IRDI (International Registration Data Identifier) sichergestellt. Für die Elemente innerhalb eines Submodells ist eine lokale Kennung ausreichend. Dies erfolgt in der Regel anhand einer **idShort** der Klasse **Referable**, die einen kurzen, aussagekräftigen Namen enthält. Dabei ist wichtig alle Submodelle bzw. alle Elemente über eine eindeutige Semantik zu beschreiben. Hierfür gibt es eine sogenannte **semanticID** der Klasse **HasSemantics**, welche eine semantische Referenz enthält. Diese verweist entweder auf einen externen Standard oder auf eine lokale Concept Description (CD), die direkt in der AAS eingebettet ist. Ein häufig verwendeter externer Standard ist zum Beispiel ECLASS, welcher auf der Norm IEC 61360 [21] basiert. Darin wird eine standardisierte Struktur für Merkmale und ihre semantische Beschreibungen

definiert. Die Semantik enthält unter anderem Beschreibungen, Definitionen, Einheiten und externe Referenzen für bestimmte Submodelle oder Submodellelemente und hilft bei der Klassifizierung dieser. Dadurch wird ein gemeinsames Verständnis unterschiedlicher Systeme geschaffen. Fehlt die semantische Beschreibung, so kann es schnell zu Missverständnissen kommen. Liegt beispielsweise ein einfacher Wert 25 vor, ist ohne weitere Informationen unklar, was gemeint ist. Es könnte sich um 25 Euro, 25 Meter oder 25 Grad Celsius handeln. Erst durch die zugehörige semantische Beschreibung, in der die Einheit und Bedeutung definiert werden, wird der Wert eindeutig interpretierbar.

Um die Erstellung von Submodellen zu erleichtern und gleichzeitig Interoperabilität zu gewährleisten, stellt die IDTA standardisierte Submodellvorlagen - sogenannte Submodel Templates - zur Verfügung. Aktuell sind 34 dieser Templates veröffentlicht, viele weitere sind in der Entwicklung oder im Überprüfungsprozess und werden in Zukunft ergänzt. Die bereits verfügbaren Templates enthalten unter anderem Submodelle wie das digitale Typenschild oder den Carbon Footprint. Alle Eigenschaften innerhalb dieser Vorlagen werden dabei in Verbindung mit dem ECLASS-Standard einheitlich semantisch beschrieben. Diese Templates können über ein zentrales Repository [22] bezogen werden und bilden die Basis für eine interoperable semantische Datenstruktur.

### **2.4.2 Informationsaustausch**

Der Austausch von Informationen über die AAS kann auf unterschiedliche Weise erfolgen. Die einfachste Möglichkeit besteht im Dateiaustausch. Hierfür wurden speziell für die AAS sogenannte AASX-Dateien [23] entwickelt, die den einfachen Austausch statischer AAS (Typ 1) ermöglichen. Dabei werden sämtliche Daten, Beziehungen, Strukturen sowie zugehörige Dateien der AAS serialisiert und in ein AASX-ZIP-Dateiformat gespeichert. Diese Datei kann anschließend über ein digitales Medium, etwa per E-Mail oder eine Cloud-Plattform, weitergegeben werden. Eine Typ 2-AAS hingegen wird von einer Laufzeitumgebung gehostet, wodurch ein direkter und dynamischer Zugriff auf ihre Inhalte ermöglicht wird. Die Spezifikation Part 2: Application Programming Interfaces [24] beschreibt hierfür nicht nur standardisierte Schnittstellen, sondern auch ein ganzheitliches System für das Verwalten, Bereitstellen und Auffinden der AAS. Repositories dienen dabei als zentraler Speicherort für die Inhalte einer AAS, einschließlich ihrer Submodelle und Concept Descriptions. Die Aufgabe der Verwaltung und Registrierung wird von Registries übernommen. Sie ermöglichen das systemweite Auffinden von AAS und stellen sicher,

dass diese eindeutig referenzierbar sind. Ergänzend dazu bieten Discovery Services eine erweiterte Suchfunktionalität, indem sie Beziehungen verschiedener Entitäten mittels verschiedener Schlüsselwertpaare speichern. Eine AAS kann so zum Beispiel logisch mit einer Asset-ID (Asset Identifier) verknüpft werden und somit schnell innerhalb komplexer Systeme identifiziert werden. Der Zugriff auf diese Systeme bzw. ihrer Inhalte wird in Form von Schnittstellen standardisiert, wodurch eine hohe Interoperabilität gewährleistet wird. Ein besonderer Fokus liegt dabei auf HTTP/REST, welches durch definierte Zugriffsprinzipien strukturiert ist. Diese folgen dabei dem REST-Schema und unterstützen gängige HTTP-Methoden wie GET, POST, PUT oder DELETE. Die fortschrittlichste Form des Informationsaustausches stellt die Peer-to-peer Kommunikation dar, bei der I4.0-Komponenten (Typ 3-AAS) eigenständig über die I4.0-Sprache miteinander kommunizieren können.

### 2.4.3 Sicherheit

Gerade wenn Informationen aus der AAS über die Grenzen des eigenen Unternehmens hinweg bereitgestellt werden, ist es besonders wichtig, dass die enthaltenen Daten geschützt sind. Die neueste Spezifikation Part 4: Security [25] der IDTA liefert hierfür die technische und konzeptionelle Grundlage. Sie beschreibt, wie Zugriffe auf Daten in der AAS sicher gesteuert werden können, insbesondere in vernetzten Umgebungen wie Datenräumen. Zum Einsatz kommen dabei neue Dienste wie ein Identity Provider zur Authentifizierung oder ein Policy Service zur Durchsetzung von Richtlinien. Die Sicherheit wird dabei mithilfe eines attributbasierten Zugriffsmodells (Attribute Based Access Control (ABAC)) gewährleistet. Bei jeder Anfrage auf bestimmte Objekte innerhalb der AAS wird dabei anhand verschiedener Merkmale (Attribute) geprüft ob ein Zugriff erlaubt ist. Dazu zählen sogenannte Subjektattribute (also wer die Anfrage stellt), Objektattribute (z.B. welches Submodell, welche Property oder welches Submodellelement betroffen ist), die gewünschte Aktion (z.B. Lesen oder Schreiben) sowie kontextbezogene Bedingungen (z.B. Zeitpunkt der Anfrage oder Zustand des Systems). Die zur Prüfung notwendigen Informationen liefert in der Regel ein Token, das vom Identity Provider bereitgestellt wird. Die Spezifikation sieht hierfür die Nutzung sogenannter JSON Web Tokens (JWT) vor. Die Attribute werden schließlich von einem Policy Service mit den dort hinterlegten Zugriffsrichtlinien abgeglichen und basierend darauf eine Zugriffsentscheidung getroffen. Ein besonderer Vorteil des ABAC-Modells liegt dabei in seiner hohen Flexibilität. Rollen können ebenfalls als Attribute behandelt werden, wodurch sich auch problemlos rollenbasierte Zugriffskonzepte (RBAC) umsetzen lassen.



Die beschriebenen Kontrollmechanismen lassen sich nicht nur auf die Inhalte der AAS selbst, sondern insbesondere auch auf die Schnittstellen von Registries und Repositories anwenden. So kann beispielsweise sichergestellt werden, dass nur autorisierte Systeme Zugriff auf ein bestimmtes Submodell erhalten oder nur bestimmte Nutzergruppen neue AAS-Instanzen registrieren können. Diese Sicherheits-Konzepte sind jedoch noch vergleichsweise neu und müssen in der Praxis erst noch weiter erprobt werden. Erste Referenzimplementierungen liegen zwar bereits häufig schon in Form rollenbasierter Zugriffskontrollen vor, eine vollständige Integration des ABAC-Ansatzes steht jedoch noch aus.

## 2.5 Digitaler Produktpass

Der Digitale Produktpass (DPP) ist ein zentrales Instrument der Europäischen Union zur Umsetzung einer nachhaltigen, digitalen Transformation. Ziel ist es, die Transparenz über ökologische Merkmale von Produkten wie verwendete Materialien, Recyclebarkeit oder die CO<sub>2</sub>-Bilanz deutlich zu verbessern. Hierzu müssen produktspezifische Daten über den gesamten Lebenszyklus hinweg aufgezeichnet und in einem menschen- und maschinenlesbarem Format bereitgestellt werden. Langfristig soll dies zu einer Kreislaufwirtschaft und digitalen Wirtschaft innerhalb der EU führen.

Das Konzept des digitalen Produktpasses wurde erstmals im Rahmen des European Green Deal von der Europäischen Kommission im Jahr 2019 vorgestellt [26]. Im Zuge der Ökodesign-Verordnung (Eco Design for Sustainable Products Regulation (ESPR)) [27] wird der DPP aktuell als verpflichtendes Mittel für zahlreiche Produktgruppen eingeführt. Als erste konkrete Anwendung wird der digitale Produktpass erstmals im Jahr 2027 für Batterien verpflichtend, wie in der EU-Batterieverordnung festgelegt. Weitere Produktkategorien, darunter auch die Elektroindustrie und der Maschinen- und Anlagenbau werden in den nächsten Jahren folgen.

Die Bereitstellung der digitalen Produktpässe erfolgt gemäß den Anforderungen der ESPR in elektronischer Form. Dabei müssen diese untereinander interoperabel miteinander kommunizieren können. Daten innerhalb eines Passes müssen standardisiert und strukturiert in einem menschen- und maschinenlesbarem Format zur Verfügung gestellt werden. Je nach Art der Information werden verschiedene Zugriffsrechte für unterschiedliche Interessengruppen eingeführt. Damit soll der Schutz von geistigem Eigentum sichergestellt werden. Verwaltet werden sollen die Daten dabei über einen zentralen Server bzw. ein Registry, in dem die verschiedenen DPPs gespeichert bzw. zumindest registriert werden. [28]

Während die regulatorischen Rahmenbedingungen schon mehr oder weniger final ausgearbeitet sind, bleibt noch die Frage der konkreten technologischen Umsetzung. Eine dezentrale Lösung zur Umsetzung bildet der von der ZVEI vorgestellte digitale Produktpass für die Industrie 4.0 (DPP 4.0) [29]. Der DPP 4.0 basiert dabei auf zwei etablierten Standards. Zum Einen das digitale Typenschild, und zum Anderen die AAS (siehe auch Kapitel Asset Administration Shell). Das digitale Typenschild ermöglicht - gemäß der Norm IEC 61406 [30] - die eindeutige Identifikation von Produkten über eine einzigartige Asset-ID. Typischerweise wird diese in Form eines maschinenlesbarem Links oder QR-Codes an ein Produkt angebracht, wodurch ein direkter Zugriff auf den jeweiligen DPP ermöglicht wird.

Organisiert werden die Daten im DPP 4.0 in verschiedenen Submodellen der AAS. Standardisierte Teilmodelle wie das digitale Typenschild, Dokumentationen oder der Product Carbon Footprint (PCF) helfen bei der Umsetzung der im DPP geforderten Daten. Darüber hinaus können auch zusätzliche, nicht verpflichtende Informationen integriert werden, sofern sie für bestimmte Stakeholder einen Mehrwert bieten. Der Zugriff auf die Daten erfolgt über ein webbasiertes Portal. Verschiedene Interessengruppen erhalten dabei unterschiedliche Zugangsrechte. Hierfür werden bestimmte Submodelle gezielt für unterschiedliche Gruppen freigegeben oder eingeschränkt. Während beispielsweise das digitale Typenschild (DN) oder der PCF öffentlich zugänglich sind, werden sensible Informationen wie technische Dokumentationen oder sicherheitsrelevante Details nur bestimmten autorisierten Gruppen zugänglich gemacht. Eine schematische Darstellung des DPP 4.0 in Abbildung 4 zu sehen.

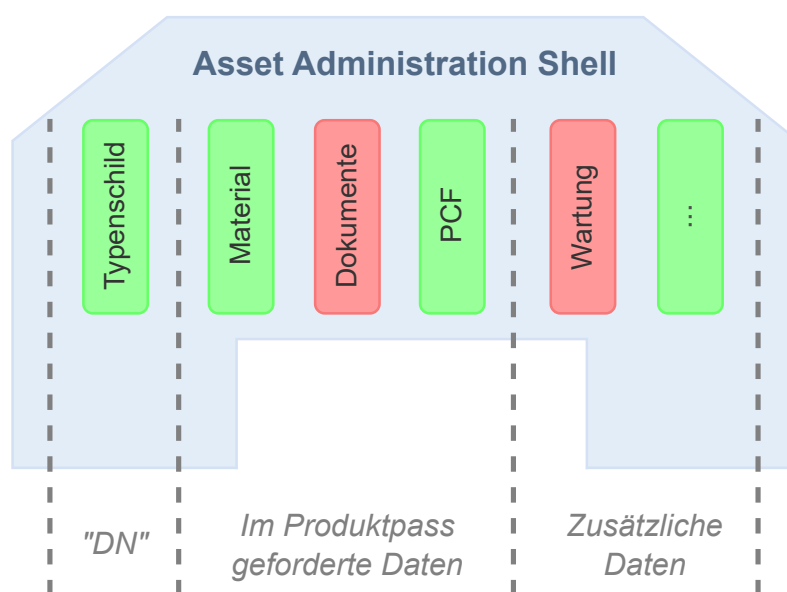


Abbildung 4: Konzept des DPP 4.0 (eigene Darstellung in Anlehnung an [31])

Das Konzept der ZVEI sieht darüber hinaus vor, dass Unternehmen ihre digitalen Produktpässe entgegen den Anforderungen der ESPR dezentral in einem eigenen Repository verwalten. Diese Repositories können entweder vom produzierenden Unternehmen selbst oder von Dritten - etwa Cloud-Dienstleistern - im Auftrag betrieben werden. Ziel ist es, Unternehmen die Möglichkeit zu geben, ihre Daten bei Bedarf eigenständig zu aktualisieren und gleichzeitig die Kontrolle über sensible Informationen zu behalten. Zur Koordination dieser dezentralen Systeme ist ein zentrales Registry vorgesehen, in der alle Repositories bzw. Server registriert werden. Über dieses zentrale Verzeichnis können interessierte Akteure relevante Server identifizieren und gezielt auf freigegebene Submodelle eines Produktpasses zugreifen. So wird sichergestellt, dass trotz der dezentralen Struktur eine durchgängige Interoperabilität gewährleistet ist, wie sie für die Umsetzung des digitalen Produktpasses auf europäischer Ebene erforderlich ist.

## **2.6 robocell**

Die robocell ist eine von groninger in Zusammenarbeit mit SKAN entwickelte Maschinenlinie zur aseptischen Abfüllung von genesteten Spritzen, Zylinderampullen und Vials. Sie zeichnet sich dadurch aus, dass alle Prozessschritte vollständig automatisiert ablaufen. Durch den gezielten Einsatz von Robotern kann so der menschliche Eingriff auf ein Minimum reduziert werden, wodurch maximale Sicherheit, Flexibilität und Sicherheit im Abfüllprozess gewährleistet wird [32].

Die Linie besteht dabei aus mehreren modular aufgebauten Einzelmaschinen, die jeweils spezifische Aufgaben entlang des Produktionsprozesses übernehmen. Im Rahmen dieser Arbeit liegt der Fokus auf dem hochautomatisierten Abfüll- und Verschleißmodul, das für das vollautomatisierte Abfüllen und Verschließen von Behältnissen verantwortlich ist.

### 2.7 Technologische Grundlagen

Im Folgenden werden die technologischen Grundlagen erläutert, die für das Verständnis und die Umsetzung dieser Arbeit besonders relevant sind.

#### 2.7.1 AASX Package Explorer

Der AASX Package Explorer, nachfolgend als Package Explorer bezeichnet, wurde als Referenzimplementierung für die AAS gemäß den Spezifikationen der IDTA entwickelt. Das Tool ist als Open-Source-Software [33] verfügbar und ermöglicht das Erstellen und Bearbeiten von AAS im standardisierten AASX-Dateiformat. Er verfügt über eine benutzerfreundliche grafische Oberfläche (siehe Abbildung 5), die insbesondere Einsteigern den Zugang zur Modellierung erleichtert. Dabei können Submodelle, Eigenschaften, semantische Referenzen sowie Metadaten strukturiert definiert und verwaltet werden. Gleichzeitig bietet der Package Explorer auch erweiterte Funktionen, wie das Erstellen von Submodell-Templates, wodurch er sich auch für den professionellen Einsatz eignet.

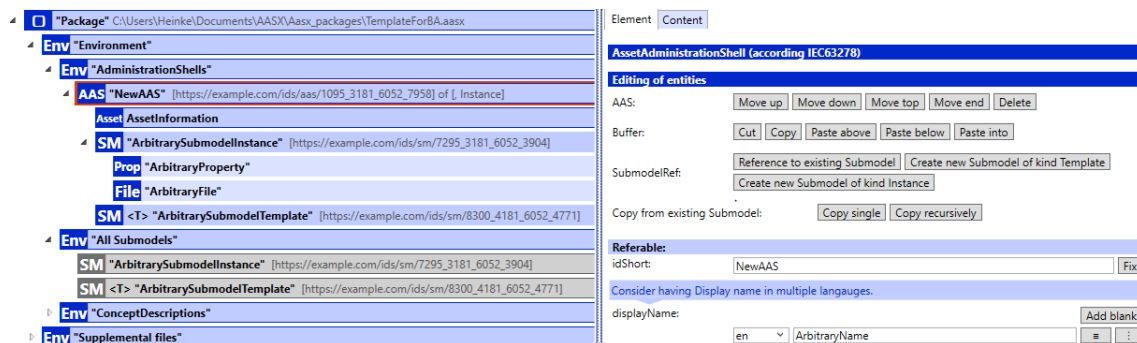


Abbildung 5: Benutzeroberfläche AASX Package Explorer

Neben der lokalen Modellierung erlaubt das Tool ebenfalls die Verbindung zu einem AAS-Server über standardisierte Schnittstellen (z.B. OPC UA oder HTTP/REST). Dies ermöglicht den Betrieb von AAS in verteilten Systemen. Besonders geeignet hierfür ist der Referenzserver des Eclipse-AAS-Projekts [34], der das Hosten und Bereitstellen von AASX-Paketen ermöglicht sowie eine nahtlose Integration mit dem Package Explorer erlaubt.

#### 2.7.2 Eclipse BaSyx

Eclipse BaSyx ist eine vom Fraunhofer-Institut für Experimentelles Software Engineering entwickelte Open-Source-Plattform für die Realisierung von Industrie 4.0-Anwendungen. Mittlerweile wird das Projekt unter dem Dach der Eclipse Foundation

weitergeführt. Der Fokus liegt auf einer einfachen Umsetzung einer Infrastruktur zur Erstellung und Verwaltung digitaler Zwillinge auf Basis der AAS. Die Software steht dabei allen Interessierten frei zur Verfügung und erlaubt individuelle Modifikationen an spezifische Anforderungen.

Die Softwarearchitektur basiert auf einer Vielzahl von Standardkomponenten (Off-the-Shelf), die alle als Docker-Container frei zugänglich sind und somit eine nahtlose Integration in bestehende Docker-Umgebungen erlauben. Eine der wichtigsten Komponenten ist die Registry. Sie ist, genau wie alle anderen Komponenten, auf den Spezifikationen der Verwaltungsschale aufgebaut, insbesondere auf der Spezifikation Part 2: Application Programming Interfaces [24]. In ihr können neue AAS registriert und bereits vorhandene Verwaltungsschalen anhand ihrer eindeutigen Kennung gesucht werden. Sie bildet damit die zentrale Anlaufstelle für Geräte und Anwendungen innerhalb eines Industrie-4.0-Systems. Analog dazu gibt es eine separate Registry für die Verwaltung von Submodellen.

Die eigentlichen Daten der AAS werden in der sogenannten AAS Environment gespeichert und organisiert. Sie umfasst Repositories für AAS, Submodelle und Concept Descriptions. In der Regel ist eine Datenbank, standardmäßig eine MongoDB als persistenter Speicher hinterlegt. Wie auch alle anderen Komponente stellen diese Repositories standardisierte Schnittstellen basierend auf der API-Spezifikation zur Verfügung. Dies erlaubt z.B. das Abfragen, Erstellen oder Aktualisieren von AAS und deren Submodellen. Alle verfügbaren Endpunkte dieser Schnittstellen können unter anderem in der automatisch generierten Swagger-Dokumentation eingesehen und ausgeführt werden. Typische REST-Endpunkte sind beispielsweise:

Tabelle 1: REST-Endpunkte in Eclipse BaSyx

Method	Endpoint	Beschreibung
<i>GET</i>	/shells	Liste aller AAS abrufen
<i>GET</i>	/shells/{aasIdentifier}	Bestimmte AAS anzeigen
<i>GET</i>	/submodels	Liste aller Submodelle aufrufen
<i>POST</i>	/shells	Neue AAS erstellen
<i>POST</i>	/submodels	Neues Submodell erstellen
<i>DELETE</i>	/shells/{aasIdentifier}	AAS löschen

Im BaSyx-System ermöglicht ein sogenannter Discovery Service die Verknüpfung physischer Assets mit ihren zugehörigen AAS. Dabei wird eine spezifische Asset-ID

mit der entsprechenden AAS-ID verlinkt. Dies ist insbesondere für die Abbildung von hierarchischen Strukturen wie Stücklisten (Bill of Material (BOM)) von großer Bedeutung. Ein übergeordnetes Asset (z.B. Maschine) kann so beispielsweise mit untergeordneten Komponenten (z.B. Antrieb, Sensoren) logisch über deren AAS verbunden werden. Einträge in den Discovery Service müssen derzeit allerdings noch manuell über die API vorgenommen werden.

Zur benutzerfreundlichen Visualisierung und Interaktion kann die sogenannte AAS Web Ui genutzt werden. Die webbasierte Benutzeroberfläche wurde mit dem JavaScript-Framework Vue.js entwickelt und kommuniziert über die standardisierte REST-API mit den zentralen Komponenten der BaSyx-Plattform, darunter die Repositories, Registries und der Discovery Service. Sie zeigt alle registrierten AAS in einer Liste an und bietet die Möglichkeit, einzelne AAS in einer Baumstruktur sowohl zu visualisieren als auch zu bearbeiten. Darüber hinaus unterstützt die Anwendung das Hochladen von Typ-1-AAS in Form von AASX-Dateien. Diese werden automatisch registriert und in eine Typ-2-AAS überführt, wodurch sie direkt in das System eingebunden werden können. Ein weiteres zentrales Merkmal der AAS Web UI ist der sogenannte AAS-Viewer. Dieser erlaubt die Visualisierung von Submodellen und deren Elementen anhand ihrer semantischen ID. Hierfür stehen verschiedene vordefinierte Plugins zur Verfügung, die bestimmte Submodelle - wie beispielsweise das Typenschild oder hierarchische Strukturen - grafisch darstellen. Da die Lösung Open Source ist besteht zudem die Möglichkeit, eigene benutzerdefinierte Plugins für weitere Submodelle zu erstellen. [35] [36]

### 2.7.3 OPC Unified Architecture

OPC UA (Open Platform Communications Unified Architecture) ist ein plattformübergreifender Kommunikationsstandard, der speziell für die Anforderungen der industriellen Automatisierung entwickelt wurde. Ziel ist ein herstellerübergreifender, sicherer und standardisierter Datenaustausch. In RAMI 4.0 [17] wird OPC UA als empfohlener Standard für die Kommunikationsschicht definiert und bildet damit die Grundlage für die Interoperabilität zwischen Maschinen, Anlagen und IT-Systemen verschiedener Hersteller.

Die grundlegende Idee von OPC UA besteht darin, dass ein Maschinenhersteller einen OPC UA Server bereitstellt, der einen standardisierten und herstellerunabhängigen Zugriff auf eine Maschine ermöglicht. Der Server dient hierbei als zentrale Schnittstelle zur Außenwelt. Er implementiert den OPC Standard und stellt strukturierte Informationen sowie Zugriffsmöglichkeiten auf Maschinenzustände und -daten bereit.

Im Inneren kommuniziert der Server dabei über ein herstellerspezifisches, proprietäres Protokoll mit der Steuerung. Zum Auslesen oder Austauschen dieser Daten wird ein OPC UA Client benötigt. Dieser agiert als Kommunikationspartner des Servers, stellt die Verbindung her und ermöglicht den bidirektionalen Datentransfer. [37]

## 3 Entwicklung

In diesem Kapitel wird die schrittweise Entwicklung des digitalen Zwillings der robocell beschrieben. Zunächst werden alle relevanten Datenquellen identifiziert und geeignete Submodelle der AAS ausgewählt. Darauf aufbauend folgt die Modellierung dieser mithilfe des Package Explorers sowie die Validierung mit einer Test Engine. Im Anschluss wird gezeigt, wie die AAS mithilfe von Eclipse BaSyx bereitgestellt und verwaltet werden kann. Dabei werden sowohl Echtzeidaten über OPC UA als auch Zeitreihendaten über eine InfluxDB eingebunden. Darüber hinaus wird auf die Entwicklung von drei exemplarischen Anwendungsfällen eingegangen. Dazu zählt unter anderem der digitale Produktpass, die automatisierte Generierung der AAS sowie der Einsatz von KI.

### 3.1 Konzeptionierung des digitalen Zwillings

Ziel dieses Abschnittes ist es, eine grundlegende Basis für die Erstellung des digitalen Zwillings zu schaffen. Dabei wird untersucht, welche Daten für die Modellierung erforderlich sind, wo diese herkommen und wie sie in (standardisierten) Teilmodellen der AAS strukturiert werden können.

#### 3.1.1 Identifikation relevanter Datenquellen

Ein digitaler Zwilling basiert immer auf einer Vielzahl unterschiedlicher Daten, die gemeinsam ein umfassendes digitales Abbild eines Assets ermöglichen. Dabei werden sowohl statische Informationen (z.B. Datenblätter oder Konstruktionsdaten) als auch dynamische Daten, die während des Betriebs einer Maschine anfallen, benötigt. Im ersten Schritt gilt es daher, alle relevanten Datenquellen zu identifizieren, die für die Modellierung des digitalen Zwillings erforderlich sind.

In industriellen Umgebungen kommen typischerweise verschiedene Systeme zur Erfassung, Verwaltung und Speicherung von Maschinendaten zum Einsatz. Bei groninger übernimmt diese Funktion das PLM-System Agile, das eng mit dem ERP-System PSI Penta verknüpft ist. Darin sind unter anderem Stücklisten, technische Spezifikationen,

CAD-Dateien sowie allgemeine Dokumente hinterlegt, die die statische Grundlage für den digitalen Zwilling bilden.

Neben den Informationen aus den Unternehmenssystemen spielen aber auch Laufzeitdaten, wie sie durch Sensoren oder Steuerungssysteme erzeugt werden, eine zentrale Rolle. Da im Rahmen dieser Arbeit keine reale Maschine angebunden ist, werden diese Daten simuliert. Hierfür kommt eine in Node.js entwickelte Anwendung zum Einsatz, die im Folgenden als Datengenerator bezeichnet wird und sowohl Prozess- als auch Betriebsdaten generiert. Ergänzend dazu wird ein Maschinensimulator verwendet, der einen PackML-Zustandsautomaten abbildet und typische Maschinenzustände sowie deren Übergänge simuliert. Beide Komponenten stehen als Docker Container zur Verfügung und stellen die Daten über einen OPC UA Server bereit, wodurch eine realitätsnahe Datenbasis geschaffen wird.

#### 3.1.2 Auswahl geeigneter Teilmodelle

Aufbauend auf den zuvor betrachteten Informationsquellen gilt es nun zu entscheiden, welche Aspekte der Maschine im digitalen Zwilling, oder besser gesagt in der AAS, abgebildet werden sollen. Im nächsten Schritt ist daher die Auswahl bzw. der Entwurf geeigneter Submodelle erforderlich, die die relevanten Informationen strukturiert bereitstellen.

Als Orientierung dienen die von der IDTA bereitgestellten Submodel Templates [22], die bereits viele typische Anwendungsfälle standardisiert abdecken. Diese sind jeweils in einer Submodellspezifikation der IDTA dokumentiert. Darüber hinaus besteht jedoch auch die Möglichkeit, eigene Submodelle zu entwerfen, die gezielt auf projektspezifische Anforderungen zugeschnitten sind. Diese können entweder vollständig neu konzipiert oder aus bestehenden Vorlagen abgeleitet werden.

Die konkrete Auswahl der Submodelle in dieser Arbeit orientiert sich hauptsächlich an typischen Industrie 4.0-Anwendungsfällen, die unter anderem auf der Website der IDTA dokumentiert sind [38]. Diese Anwendungsfälle zeigen auf, welche Submodelle in der Praxis besonders relevant sind. Eines der wichtigsten ist vermutlich das digitale Typenschild, da dieses häufig die erste Anlaufstelle für die Identifikation und grundlegende Informationen eines Assets darstellt. Daneben wurden aber auch projektspezifische Anforderungen berücksichtigt, die sich aus den verfügbaren Daten sowie dem fachlichen Austausch mit Industriepartnern wie Wittenstein ergaben.

Tabelle 2 liefert einen Überblick über die initiale Auswahl dieser Submodelle sowie ihrer typischen Inhalte. In der Spalte Standardisierung ist jeweils das zugehörige



Submodel Template durch die zugehörige Dokumentennummer der IDTA angegeben, in der die entsprechende Version dokumentiert ist. Diese Submodelle werden in späteren Anwendungsfällen gezielt erweitert. Ein wesentlicher Vorteil der AAS besteht nämlich in ihrer Flexibilität. Submodelle können sukzessive ergänzt, angepasst oder auch wieder entfernt werden, ohne dabei die bestehende Struktur des digitalen Zwillings zu verändern.

Tabelle 2: Submodelle mit typischen Inhalten

Submodell	Typische Inhalte	Standardisierung
Typenschild	Hersteller Seriennummer Adressinformationen	IDTA 02006-3-0 [39]
Dokumentation	Allgemeine Dokumente Betriebsanleitungen Projektzeichnungen	IDTA 02004-1-2 [40]
3D-Modelle	Konstruktionsmodelle	IDTA 02026-1-0 [41]
Technische Daten	Generelle Informationen Technische Eigenschaften	IDTA 02003 [42]
BOM	Strukturierte Stücklisten Komponentenbeziehungen	IDTA 02011-1-1 [43]
Wartung	Wartungsinformationen Wartungsintervalle	-
Prozessdaten	Messwerte	-
Zeitreihendaten	Zeitreihen	IDTA 02008-1-1 [44]
Kontrollkomponente	Betriebsmodi Schnittstelle zur Automatisierung	-

Ein besonders wichtiges Submodell stellt die Stückliste (BOM) dar. Mit diesem kann die hierarchische Struktur eines Assets abgebildet werden, das aus mehreren Komponenten besteht. Bei komplexeren Anlagen oder Maschinen, wie auch der robocell, bietet es sich an, einzelne Komponenten jeweils in einer eigenen AAS-Instanz zu modellieren und diese später in der Haupt-AAS zu referenzieren. Auf diese Weise kann eine höhere Modularität und Wiederverwendbarkeit erreicht werden. Da die robocell jedoch aus einer Vielzahl von Komponenten besteht, wird sich diese Arbeit im weiteren Verlauf auf eine kleine Auswahl beschränken.

Zur Verdeutlichung der zugrunde liegenden Architektur sowie der Beziehungen

zwischen den zuvor identifizierten Datenquellen, AAS und Submodellen dient Abbildung 6. Die AAS der robocell bildet dabei das zentrale digitale Abbild, in das die relevanten Informationen über die in diesem Kapitel ausgewählten Submodelle integriert werden sollen. Einzelne Maschinenkomponenten werden durch eigene Komponenten-AAS repräsentiert. Dabei ist zu beachten, dass in der Abbildung lediglich die erste Ebene dieser Komponentenstruktur visualisiert ist. Weitere, hierarchisch untergeordnete Bauteile können analog ebenfalls durch eigene AAS-Instanzen modelliert und über entsprechende Referenzen in die Haupt-AAS der robocell bzw. in übergeordnete Strukturen eingebunden werden.

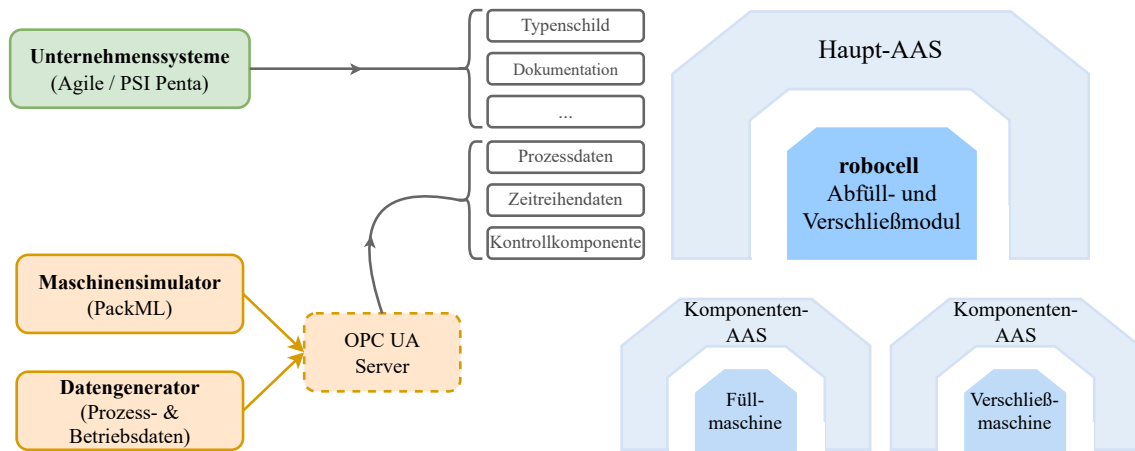


Abbildung 6: AAS-Konzept der robocell

## 3.2 Modellierung mit der AAS

In diesem Abschnitt wird erläutert, wie die zuvor ausgewählten Submodelle mit konkreten Daten befüllt werden können und welche Aspekte bei der Modellierung beachtet werden müssen. Zudem wird gezeigt, wie die erstellte AAS mithilfe einer Test Engine auf eine korrekte und vollständige Struktur überprüft werden kann.

### 3.2.1 Umsetzung mit dem Package Explorer

Für die manuelle Erstellung der AAS kann der Package Explorer eingesetzt werden. Dieser erlaubt eine intuitive Modellierung aller relevanten Elemente und erleichtert so den strukturierten Aufbau der AAS.

Bevor die gewünschten Submodelle hinzugefügt werden können, muss zuerst eine neue AAS-Paket erstellt werden. Dazu kann im Package Explorer eine neue Umgebung geöffnet werden, die als Container für die Inhalte eines Assets dient. Im Anschluss kann eine neue AAS hinzugefügt werden, die allgemeine Informationen sowie assetspezifische Daten enthält.

Neben der Auswahl des Asset-Typs (Instanz oder Typ) spielt insbesondere die eindeutige Identifikation eine wichtige Rolle. Das Asset wird über eine `globalAssetId` identifiziert, während die AAS selbst eine eigene ID sowie eine `idShort` erhält. Es bietet sich an, hierfür zunächst Beispiel-IDs zu verwenden. Diese können direkt im Package Explorer generiert werden. Dies ist besonders wichtig für den späteren Austausch der AAS sowie das systemweite Auffinden innerhalb eines Industrie 4.0-Ökosystems. Darüber hinaus besteht ebenfalls die Möglichkeit, ein Thumbmail in Form einer Datei (z.B. png oder jpeg) zu hinterlegen.

Im nächsten Schritt müssen die benötigten Submodelle hinzugefügt werden. Auch diese lassen sich entweder als Instanz oder Typ anlegen. Dabei besteht die Möglichkeit, entweder ein leeres Submodell manuell mit verschiedenen Submodellelementen zu erstellen oder auf ein vorhandenes Submodell Template zurückzugreifen. Letztere können als AASX-Datei, beispielsweise über das Repository der IDTA [22], heruntergeladen und anschließend über ein sogenanntes Auxiliary AAS in die Umgebung geladen werden. Sie stehen dann zur Verfügung und können in die AAS kopiert werden.

Nachdem alle Submodelle erstellt bzw. eingebunden sind, müssen diese mit den entsprechenden Inhalten gefüllt werden. Hierzu können konkrete Werte, Dateien oder Referenzen direkt in den jeweiligen Submodellelementen eingetragen bzw. eingebettet werden. Besonders bei der Verwendung von Templates muss darauf geachtet werden, dass die Struktur dieser nicht verändert wird, da dies sonst zu Abweichungen von der Submodellspezifikation führen kann. Eine Übersicht über die Modellierungsumgebung im Package Explorer mit einem geöffneten Submodell ist in Abbildung 7 dargestellt.

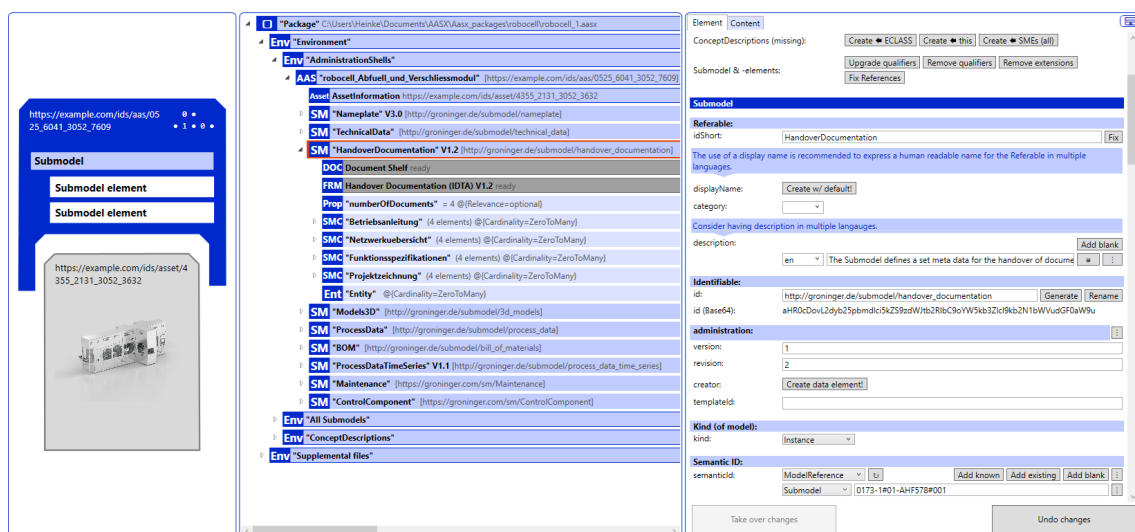


Abbildung 7: Bearbeitungsansicht eines Submodells im Package Explorer

Wie auch in der Abbildung zu erkennen, kann jedem Submodell bzw. jedem Submodellelement eine `semanticId` zugewiesen werden. Diese dient der einheitlichen semantischen Beschreibung und verweist entweder auf externe Standards wie ECLASS oder auf lokale Concept Descriptions innerhalb der AAS-Umgebung. Der Package Explorer bietet hierfür eine erweiterte Funktion, bei der sich vorgefertigte ECLASS-Kataloge importieren lassen. Die darin enthalten Begriffe können direkt im Package Explorer durchsucht, ausgewählt und den entsprechenden Submodellen bzw. Submodellelementen zugewiesen werden.

Sobald alle gewünschten Submodelle mit Inhalten gefüllt und semantisch beschrieben sind, muss die AAS gespeichert und exportiert werden. In diesem Projekt erfolgt dies bevorzugt im AASX-Format, das sich als standardisierte Austauschform für die AAS etabliert hat und eine einfache Weitergabe ermöglicht.

#### 3.2.2 Validierung

Im Anschluss an die Erstellung der AAS sollte eine Überprüfung der Konformität erfolgen. Hierzu kann eine von der IDTA bereitgestellte Test Engine [45] eingesetzt werden. Diese lässt sich direkt mit `pip`, dem Paketmanager von Python, installieren und anschließend über die Kommandozeile nutzen.

Mit dem Befehl `aas_test_engines check_file robocell.aasx` kann die zuvor erstellte AASX-Datei der `robocell` validiert werden. Dabei wird zunächst geprüft, ob die AASX-Datei formal korrekt aufgebaut ist, insbesondere hinsichtlich der internen Struktur und ihrer Beziehungen. Anschließend erfolgt die Kontrolle der enthaltenen AAS gegen die Metamodell-Spezifikationen der IDTA (Teil 1 [20] und 3a [46]). Zuletzt erfolgt ein Abgleich der Submodelle mit ihren entsprechenden Submodel Templates. Wenn im ganzen Prozess keine Fehler oder Abweichungen gefunden werden, folgt eine entsprechende Bestätigung der Test Engine (siehe Abbildung 8).

```
Checking AASX package
  Checking [Content_Types].xml
  Checking relationships
    Checking root relationship
    Relationship aasx/aasx-origin is of type http://admin-shell.io/aasx/relationships/aasx-origin
    Checking relationships of aasx/aasx-origin
      Relationship aasx/robocell_Abfuell_und_Verschliessmodul/robocell_Abfuell_und_Verschliessmodul.aas.xml is of type http://admin-shell.io/aasx/relationships/aas-suppl
    Checking relationships of aasx/robocell_Abfuell_und_Verschliessmodul/robocell_Abfuell_und_Verschliessmodul.aas.xml
      Relationship robocell_FS2_rgb_Logo.png is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Technische_Daten/Projektzeichnung.pdf is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Handover_Documentation/01_1354471_Betriebsanleitung_de.pdf is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Handover_Documentation/Netzwerk_bersicht.pdf is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Nameplate/groninger_Logo_Transparent_RGB.png is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Handover_Documentation/Funktionsspezifikationen.pdf is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Handover_Documentation/betriebsanleitung.PNG is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Handover_Documentation/Funktionsspezifikationen.PNG is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Handover_Documentation/Projektzeichnung.pdf is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Handover_Documentation/projektzeichnung.PNG is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Models3D/7001115840_0.dwf is of type http://admin-shell.io/aasx/relationships/aas-suppl
      Relationship aasx/Models3D/_aasx_files_AC_Servo_Drive_NG_Bgl.stp is of type http://admin-shell.io/aasx/relationships/aas-suppl
  Checking files
    Checking aasx/robocell_Abfuell_und_Verschliessmodul/robocell_Abfuell_und_Verschliessmodul.aas.xml
    Check
    Check meta model
    Check constraints
```

Abbildung 8: Konsolenausgabe nach erfolgreicher Validierung

### 3.3 Technische Integration

Im Anschluss an die Konzeption und Modellierung des digitalen Zwillings liegt der Fokus dieses Kapitels auf der technischen Integration der zuvor erstellten AAS in eine Industrie 4.0-kompatible Umgebung. Mithilfe der Eclipse BaSyx-Plattform wird gezeigt, wie die statisch modellierte AAS in eine Typ-2-AAS überführt und systemseitig bereitgestellt werden kann. Darüber hinaus wird die Erweiterung um dynamische Informationen anhand der Anbindung von Echtzeit- und Zeitreihendaten über standardisierte Schnittstellen erläutert.

#### 3.3.1 Bereitstellung der AAS

Für die Bereitstellung der AAS stehen mehrere Open-Source-Lösungen zur Verfügung. Eine Lösung, die sich besonders gut für den Einstieg eignet, ist der AasxServerBlazor [34], der das serverseitige Gegenstück zum Package Explorer bildet. Diese Kombination ermöglicht eine einfache Visualisierung sowie Verwaltung von AAS-Paketen und ist vorallem für erste Tests hilfreich.

Für komplexerer Anwendungsfälle erweist sich der AasxServerBlazor jedoch als nicht optimal, insbesondere im Hinblick auf die Integration von Echtzeitdaten sowie die dynamische Erweibarkeit von Submodellen. Aus diesem Grund wird im weiteren Verlauf dieses Projekts die Eclipse BaSyx-Plattform eingesetzt. Durch ihre modulare Architektur und die klare Trennung verschiedener Komponenten wie den Registries oder den Repositories bietet BaSyx eine deutlich flexiblere Grundlage für die Umsetzung praxisnaher Industrie 4.0-Szenarien.

Die einfachste Möglichkeit, um BaSyx zu installieren, ist über Docker. Alle benötigten Komponenten stehen hierfür als vorgefertigtes Image öffentlich über den Docker Hub zur Verfügung. Alternativ kann auch der Source Code von Github bezogen werden, um einzelne Komponenten individuell anzupassen oder zu erweitern. Die verschiedenen Services, darunter die AAS Environment, Registries für AAS und Submodelle, der Discovery Service, sowie die MongoDB als persistenter Speicher, können alle in einer zentralen docker-compose.yml Datei verwaltet werden. Die Konfiguration erfolgt in der Regel über Umgebungsvariablen oder separate Konfigurationsdateien, die in der docker-compose.yml referenziert werden. Schließlich können alle Komponenten gemeinsam mit dem Befehl `docker compose up` gestartet werden.

Nach erfolgreichem Start der BaSyx Umgebung, gibt es mehrere Möglichkeiten, eine AAS bereitzustellen und zu registrieren. Eine besonders einfache Variante besteht darin, dass die AASX-Datei in ein gemountetes Volume der AAS Environment

abgelegt wird. Bei einem erneuten Start der Anwendung wird diese Datei automatisch erkannt, registriert und in das BaSyx System eingebunden. Die dabei erzeugten Daten, darunter Informationen zu AAS, Submodellen, Concept Descriptions sowie Registrierungsdaten, werden in verschiedenen Tabellen der angebunden MongoDB gespeichert. Dies gewährleistet, dass alle relevanten Daten auch bei einem Neustart der Docker Container erhalten bleiben.

Alternativ kann die Bereitstellung auch direkt über die AAS Web Ui erfolgen. Über die Benutzeroberfläche kann eine AASX-Datei manuell importiert werden, wodurch sie direkt im laufenden System registriert und eingebunden wird. Diese Methode eignet sich besonders gut für Tests oder kleinere Anpassungen, da eine AAS auf diese Weise schnell und ohne direkten Zugriff auf das zugrunde liegende Dateisystem eingebunden werden kann.

Die flexibelste aber technisch anspruchsvollste Möglichkeit ist die manuelle Registrierung über die REST-API. Dabei kann nicht einfach eine AASX-Datei hochgeladen werden, sondern es müssen die AAS, ihre Submodelle sowie ihre Beziehungen eigenständig über die bereitgestellte Schnittstellen angelegt werden. Dies erfolgt über JSON-Datenstrukturen, die im Body der jeweiligen Anfrage übermittelt werden. Ein typischer Ablauf mit den zugehörigen REST-Endpunkten ist in Tabelle 3 dargestellt. Die Endpunkte gehören dabei zu verschiedenen Services innerhalb der AAS Environment.

Tabelle 3: Bereitstellung einer AAS über die REST-API in Eclipse BaSyx

Schritt	Service	Endpunkte
1. AAS erstellen	AAS Repository	/shells
2. Submodell(e) erstellen	Submodel Repository	/submodels
3. Submodell(e) mit AAS verknüpfen	AAS Repository	/shells/{aasIdentifier} /submodel-refs
4. CD anlegen	CD Repository	/concept-descriptions

Darüber hinaus besteht ebenfalls die Möglichkeit, eine AAS im Discovery Service zu registrieren. Über einen sogenannten assetLink kann diese dabei logisch mit einem Asset verknüpft werden. Die Registrierung erfolgt analog zur AAS Environment über einen REST-Endpunkt. Besonders bei der Darstellung von hierarchischen Strukturen, etwa einer BOM mit mehreren verschachtelten Assets, hilft der Discovery Service bei der eindeutigen Zuordnung der jeweiligen AAS zu ihrem physischen Gegenstück.

### 3.3.2 Integration von Echtzeitdaten über OPC UA

Nach der Erstellung der statischen AAS der robocell und der Integration in das BaSyx-System gilt es nun, diese um dynamische Informationen zu erweitern. Diese sind essenziell, um den aktuellen Zustand einer Maschine abbilden zu können. Die Datenbasis bilden die beiden zuvor vorgestellten Anwendungen, die Maschinen- bzw. Sensordaten über einen OPC UA Server bereitstellen.

Die Integration von Echtzeitdaten in die AAS wird im Folgenden am Beispiel des Submodells Prozessdaten erläutert. Innerhalb dieses Submodells existieren verschiedene Properties, die jeweils bestimmte Werte, wie beispielsweise den Druck oder die Anzahl abgefüllter Einheiten, repräsentieren. Diese Properties sollen im weiteren Verlauf dynamisch mit den simulierten Werten, die über OPC UA bereitgestellt werden, aktualisiert werden.

Das Eclipse BaSyx-Projekt stellt hierfür eine weitere Komponente bereit, die sogenannte Databridge [47]. Diese steht, wie alle anderen Komponenten auch, als Docker Container zur Verfügung und ermöglicht die Anbindung verschiedenster Datenquellen an eine AAS. Sie unterstützt eine Vielzahl von Protokollen, darunter insbesondere auch OPC UA oder MQTT. Dabei dient sie als Vermittler zwischen einem Datenendpunkt und einem Submodell innerhalb der AAS.

Die Konfiguration der Databridge erfolgt über mehrere JSON-Dateien. In einer zentralen Konfigurationsdatei werden sowohl die Datenquelle als auch die Datensinke definiert. Die Aktualisierung der Daten erfolgt dabei ereignisbasiert anhand eines Event-Triggers.

Als Datenquelle muss der OPC UA Server des Datengenerators angegeben werden, dessen Konfiguration typischerweise in einer separaten JSON-Datei erfolgt. Darin sind sowohl die Verbindungsparameter des Servers (z.B. URL und Port) als auch die zu überwachenden Knoten anzugeben. Der OPC UA Server stellt die Werte hierfür in einer hierarchischen Struktur bereit, wobei jeder Knoten über einen Namespaceindex (ns) und eine NodeId (i) eindeutig adressierbar ist.

Darüber hinaus muss der zu verwendende OPC UA Client festgelegt werden. In diesem Projekt wird Eclipse Milo eingesetzt, der auf einem Subscription-Modell basiert. Im Gegensatz zu einem Polling-Ansatz werden hier gezielt bestimmte Knoten (Nodes) abonniert, sodass neue Werte automatisch bei Änderung übermittelt werden.

Ein Beispiel für die JSON-Konfiguration einer solchen Datenquelle ist in Listing 1 für die Druck-Node dargestellt. Weitere optionale Parameter, wie etwa Sicherheits-

einstellungen oder das Übertragungsintervall können zusätzlich angegeben werden, wurden hier jedoch zur besseren Übersicht weggelassen.

Listing 1: Beispielhafte JSON-Konfiguration einer Datenquelle

```
1 {  
2   "uniqueId"      : "pressure",  
3   "nodeInformation": "ns=4;i=113",  
4   "serverUrl"     : "opcua-server",  
5   "serverPort"    : 4840,  
6   "pathToService" : "milo"  
7 }
```

Zur Anpassung der eingehenden Werte an die Struktur der gewünschten Property können verschiedene Transformatoren eingesetzt werden. Die Databridge unterstützt hierfür unter anderem das Verwenden von JSONata-Ausdrücken oder den Einsatz von `JsonJacksonTransformers`. Mit diesen lassen sich die vom OPC UA Server empfangenen Rohdaten in ein JSON-Objekt überführen und anschließend die konkreten Datenwerte der jeweiligen Nodes gezielt extrahieren

Anschließend muss die Datensenke konfiguriert werden, welche die transformierten Werte in die entsprechenden Properties des Submodells Prozessdaten überträgt. Dies erfolgt analog zur Konfiguration der Datenquelle über eine JSON-Datei. In dieser müssen der Endpoint des Submodells, der `idShortPath` der gewünschten Property sowie die verwendete API-Version angegeben werden. Eine beispielhafte Konfiguration ist in Listing 2 dargestellt. Der Platzhalter `{smId}` steht dabei für die Base64-kodierte ID des Submodells Prozessdaten, wie sie in der REST-API der AAS verwendet wird.

Listing 2: Beispielhafte JSON-Konfiguration einer Datensenke

```
1 {  
2   "uniqueId"      : "Submodel/ProcessData/Pressure",  
3   "submodelEndpoint": "http://aas-env:8081/submodels/{smId}",  
4   "idShortPath"   : "Druck",  
5   "api"           : "DotAAS-V3"  
6 }
```

Nach erfolgreicher Konfiguration der Databridge werden die OPC UA-Werte automatisiert in die jeweilige Property des Submodells geschrieben. Dieser Mechanismus lässt sich nicht nur für die Übertragung von Prozesswerten, sondern auch zur Darstellung des aktuellen Maschinenzustands nutzen und bietet somit eine flexible Grundlage für die Integration von Prozess- und Betriebsdaten in die AAS.



### 3.3.3 Verarbeitung von Zeitreihendaten

Grundsätzlich lassen sich Zeitreihendaten auf unterschiedlichste Weise in eine AAS einbinden. Das Submodel Template Time Series Data [44] bietet hierfür mehrere standardisierte Lösungsansätze. Eine Möglichkeit besteht darin, diese direkt über ein InternalSegment in der AAS zu speichern. Diese Variante eignet sich jedoch nur für kleinere Datenmengen. Alternativ können die Daten in Form einer Datei abgespeichert werden. Diese können dann entweder direkt in die AAS eingebunden oder extern über ein ExternalSegment referenziert werden. Für größere Datenmengen bietet sich die externe Speicherung an einem separaten Ort, wie etwa einer Datenbank, an. Diese kann über ein LinkedSegment mit der AAS verknüpft werden.

Im Folgenden wird die zuletzt genannte Option näher betrachtet. Hierzu werden die simulierten Werte für Druck und Temperatur des Datengenerators extern in einer InfluxDB gespeichert. Die über OPC UA bereitgestellten Daten werden dabei mithilfe von Telegraf, einem leichtgewichtigen Agenten zur Datenerfassung und -weiterleitung [48], kontinuierlich in eine speziell für diesen Anwendungsfall angelegte Tabelle geschrieben. Beide Komponenten stehen als Docker-Container zur Verfügung und lassen sich nahtlos in das bestehende BaSyx-System integrieren.

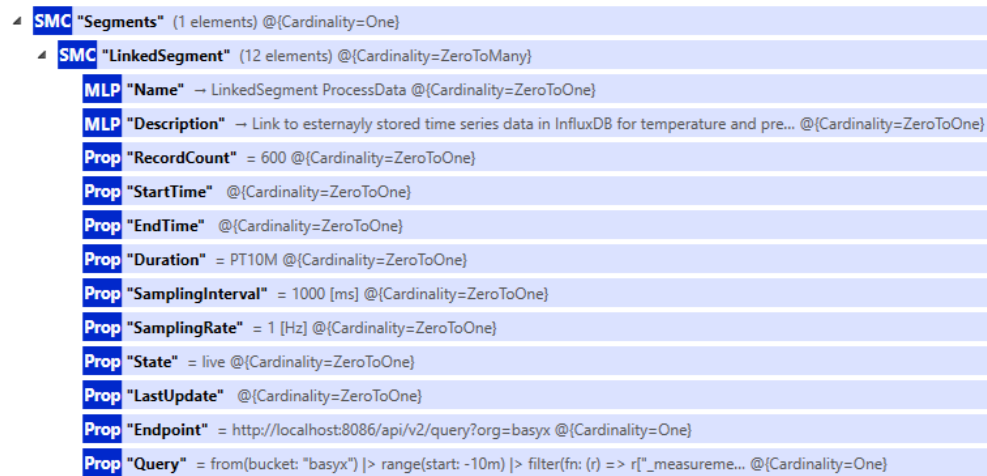
Um die in der Datenbank gespeicherten Daten in die AAS einzubinden, kann das bereits genannte Submodel Template Time Series Data verwendet werden. Zunächst müssen die Metadaten eingetragen werden (siehe Abbildung 9). Dazu gehören ein eindeutiger Name, eine Beschreibung sowie die Definition der Datenpunkte (Records), die hiermit aufgezeichnet werden sollen.

SMC	"Metadata"	(3 elements) @Cardinality=One
MLP	"name"	→ ProcessDataTimeSeries @Cardinality=One
MLP	"Description"	→ Metadata definition for time series data including pressure and temperature values @Cardinality=ZeroToOne
SMC	"Record"	(3 elements) @Cardinality=One
Prop	"time"	= 2025-06-30T12:00:00Z @Cardinality=OneToMany @AllowedIdShort=Time[2,3]
Prop	"pressure"	[bar] @Cardinality=ZeroToOne
Prop	"temperature"	[°C] @Cardinality=ZeroToOne

Abbildung 9: Metadaten-Konfiguration im Submodell Time Series Data

Anschließend folgt die Konfiguration des LinkedSegments (siehe Abbildung 10). Dieses stellt die eigentliche Verbindung zu den extern gespeicherten Zeitreihendaten her. Es enthält unter anderem den Endpunkt der InfluxDB, sowie die zugehörige Abfrage (Query), mit der die gewünschten Werte abgefragt werden können. Zusätzlich können weitere Informationen angegeben werden, wie beispielsweise die Abtastrate (samplingRate), der Zeitraum, den das Segment abdeckt, oder ein recordCount, der

angibt, wie viele Einträge innerhalb dieses Zeitraums erwartet werden.



SMC	"Segments"	(1 elements) @({Cardinality=One})
SMC	"LinkedSegment"	(12 elements) @({Cardinality=ZeroToMany})
MLP	"Name"	→ LinkedSegment ProcessData @({Cardinality=ZeroToOne})
MLP	"Description"	→ Link to esternayly stored time series data in InfluxDB for temperature and pre... @({Cardinality=ZeroToOne})
Prop	"RecordCount"	= 600 @({Cardinality=ZeroToOne})
Prop	"StartTime"	@({Cardinality=ZeroToOne})
Prop	"EndTime"	@({Cardinality=ZeroToOne})
Prop	"Duration"	= PT10M @({Cardinality=ZeroToOne})
Prop	"SamplingInterval"	= 1000 [ms] @({Cardinality=ZeroToOne})
Prop	"SamplingRate"	= 1 [Hz] @({Cardinality=ZeroToOne})
Prop	"State"	= live @({Cardinality=ZeroToOne})
Prop	"LastUpdate"	@({Cardinality=ZeroToOne})
Prop	"Endpoint"	= http://localhost:8086/api/v2/query?org=basyx @({Cardinality=One})
Prop	"Query"	= from(bucket: "basyx")  > range(start: -10m)  > filter(fn: (r) => r["_measureme... @({Cardinality=One})

Abbildung 10: Konfiguration des LinkedSegments im Submodell Time Series Data

Zur Visualisierung der Zeitreihendaten bietet das BaSyx-System eine praktische Lösung. Über ein entsprechendes Plugin können die im Submodell Time Series Data enthaltenen Informationen direkt in der AAS Web Ui dargestellt werden. Die über das LinkedSegment referenzierten Zeitreihendaten lassen sich dort in verschiedenen Diagrammen visualisieren, beispielsweise als Linien- oder Balkendiagramm. Dadurch wird eine benutzerfreundliche Darstellung der Daten ermöglicht, ohne dass diese physisch in der AAS gespeichert werden müssen.

## **3.4 Anwendungsfall Digitaler Produktpass**

Hier steht etwas

### **3.4.1 Beschreibung**

### **3.4.2 Umsetzung mit dem Teilmodell Carbon Footprint**

## 3.5 Anwendungsfall automatisierte Generierung von AAS

Ziel dieses Anwendungsfalles ist es, den Prozess der automatisierten Generierung und Bereitstellung einer AAS darzustellen. Dazu wird zunächst ein Submodel Template im Package Explorer erstellt und anschließend in ein Typ-Submodell überführt. Dieses wird anschließend automatisiert mit Daten befüllt, in eine AAS-Instanz eingebettet und schließlich in ein Industrie 4.0-System eingebunden.

### 3.5.1 Erstellen von Submodel Templates

Das Erstellen von Submodel Templates spielt eine zentrale Rolle für den effizienten Einsatz der AAS in Industrie 4.0-Anwendungen. Ein einmal definiertes Template kann für beliebig viele Instanzen eines digitalen Zwillings wiederverwendet werden. Dadurch wird nicht nur die Konsistenz der Datenstruktur sichergestellt, sondern auch der Entwicklungsaufwand erheblich reduziert.

Grundsätzlich gibt es mehrere Möglichkeiten ein solches Template zu erstellen. Es kann entweder vollständig neu modelliert oder von bestehenden Submodel Templates abgeleitet werden, wie es im Folgenden exemplarisch anhand des Submodells Technische Daten [42] gezeigt wird. Dieses Template stellt eine standardisierte Struktur zur Beschreibung technischer Merkmale eines Produkts bereit.

Für die technische Umsetzung kann der Package Explorer eingesetzt werden. Hierfür muss zunächst das Submodel Template Technische Daten importiert werden. Dabei handelt es sich um ein generisches Template, das lediglich grundlegende Strukturkategorien vorgibt. Es enthält beispielsweise SubmodelElementCollections wie Generelle Informationen, Technische Informationen oder Produktklassifikation, die allerdings noch nicht mit konkreten Eigenschaften gefüllt sind.

Auf Unternehmensebene kann dieses Template für eine bestimmte Produktgruppe mit zusätzlichen Merkmalen individualisiert werden. So lassen sich beispielsweise spezifische Anforderungen bzw. produktspezifische Eigenschaften wie Umgebungsbedingungen oder elektrische Daten einer Maschine ergänzen.

Aus dem angepassten Template kann anschließend eine Typ-AAS erstellt werden, in die bereits konkrete allgemeingültige Merkmale einer Produktgruppe eingetragen werden. Diese dient als Vorlage für die spätere Instanziierung konkreter digitaler Zwillinge einzelner Produkte, bei denen dann auch die produktspezifischen Werte ergänzt werden. Zur Weiterverwendung kann das Submodell direkt im Package Explorer als JSON-Datei exportiert werden.

### 3.5.2 Automatisiertes Befüllen mit strukturierten Daten

Nach der Erstellung eines Typ-Submodells stellt sich die Frage, wie dieses effizient mit konkreten Informationen befüllt und anschließend in eine vollständige AAS-Instanz eingebettet werden kann. In der Praxis liegen die hierfür benötigten Daten häufig bereits strukturiert in bestehenden Unternehmenssystemen vor, beispielsweise in PLM- oder ERP-Systemen.

Da eine direkte Anbindung solcher Systeme im Rahmen dieser Arbeit nicht möglich ist, wird das automatisierte Befüllen exemplarisch anhand vorgegebener Beispieldaten umgesetzt. Diese enthalten technische Informationen, wie sie typischerweise in den Unternehmenssystemen vorliegen, und bilden die Grundlage für die automatisierte Erstellung des Instanz-Submodells Technische Daten.

Für die technische Umsetzung bietet sich ein Skript an, das diese Daten automatisiert in die vorgegebene Submodellstruktur überträgt und anschließend in Form einer Typ-2-AAS bereitstellt. In diesem Projekt wird dazu die serverseitige JavaScript-Laufzeitumgebung Node.js [49] verwendet, da sie eine einfache Verarbeitung von JSON-Daten sowie eine unkomplizierte Kommunikation mit REST-Schnittstellen ermöglicht. Als Basis werden drei zentrale Komponenten benötigt:

1. **Datenquelle:** Beispieldaten technische Produktinformationen
2. **Submodell-Vorlage:** Struktur und Semantik des Submodells
3. **AAS-Vorlage:** Aufbau und Struktur der AAS-Instanz

Die Datenquelle liegt im JSON-Format vor und ist hierarchisch aufgebaut. Sie besteht aus mehreren verschachtelten Schlüssel-Wert-Paaren. Jeder Schlüssel dient als eindeutiger Bezeichner und entspricht einem idShort-Wert eines Submodellelements innerhalb der Submodell-Vorlage. Diese Vorlage ist so konzipiert, dass an den relevanten Stellen Platzhalter vorhanden sind, die während der Ausführung des Skripts mit den zugehörigen Werten aus der Datenquelle befüllt werden können.

Nach dem Befüllen des Submodells mit konkreten Werten muss dieses in eine zugehörige AAS-Instanz eingebunden werden. Die AAS-Vorlage definiert hierfür die grundlegende Struktur der AAS in einer separaten JSON-Datei. Diese enthält jedoch noch keine spezifischen Identifikatoren, wie beispielsweise die eindeutige ID der AAS oder des zugehörigen Assets. Um diese hinzuzufügen, können während der Skriptausführung zufällige UUIDv4-Werte generiert und an den entsprechenden Stellen in die Vorlage eingefügt werden.

Im Anschluss daran gilt es, die AAS sowie das Submodell in einer Industrie-4.0-

Anwendung bereitzustellen. Hierfür kann erneut Eclipse BaSyx genutzt werden, wobei die AAS und das Submodell, analog zur in Kapitel 3.3.1 beschriebenen Methode der manuellen Registrierung, über die REST-API bereitgestellt werden können.

### 3.5.3 Potenziale des KI-Einsatzes

## 4 Ergebnisse

### 4.1 AAS-Demonstrator für die robocell

#### 4.1.1 Systemarchitektur

#### 4.1.2 Eingesetzte Teilmodelle

#### 4.1.3 Herausforderungen bei der Erstellung

### 4.2 Anwendungsfall Digitaler Produktpass

#### 4.2.1 Implementierungskonzept

#### 4.2.2 Dynamische Berechnung des PCF

### 4.3 Anwendungsfall automatisierte Generierung der AAS

### 4.4 Einsatzmöglichkeiten von KI im Kontext der Verwaltungsschale

#### 4.4.1 Generierung von Verwaltungsschalen

#### 4.4.2 Anomaliererkennung

#### 4.4.3 Weiterführende Einsatzmöglichkeiten

### 4.5 Evaluierung eingesetzter Tools und Software

#### 4.5.1 AASX Package Explorer

#### 4.5.2 Eclipse AASX Server

#### 4.5.3 BaSyx

#### 4.5.4 Mnestix Browser

## 5 Zusammenfassung und Ausblick

### 5.1 Zusammenfassung der Arbeit

### 5.2 Handlungsempfehlung für groninger

### 5.3 Ausblick auf zukünftige Entwicklungen

## **Glossar und Abkürzungsverzeichnis**



## Glossar

- [1] Thomas Bauernhansl, Michael ten Hompel und Birgit Vogel-Heuser. *Industrie 4.0 in Produktion, Automatisierung und Logistik*. 2014. DOI: 10.1007/978-3-658-04682-8.
- [2] Armin Roth. *Einführung und Umsetzung von Industrie 4.0*. 2016. DOI: 10.1007/978-3-662-48505-7.
- [3] Plattform Industrie 4.0. 2025. URL: <https://www.plattform-i40.de/> (besucht am 15.06.2025).
- [4] BITKOM, VDMA und ZVEI. *Umsetzungsstrategie Industrie 4.0*. Apr. 2015. URL: <https://www.zvei.org/presse-medien/publikationen/umsetzungsstrategie-industrie-40> (besucht am 15.06.2025).
- [5] Joachim Metternich u. a. *Künstliche Intelligenz zur Umsetzung von Industrie 4.0 im Mittelstand*. 2021. URL: [https://www.ptw.tu-darmstadt.de/media/fachgebietptw/dokumente\\_3/wissenssammlung\\_ptw/leitfaeden\\_2/Kuenstliche\\_Intelligenz\\_zur\\_Umsetzung\\_\\_von\\_Industrie\\_4.0\\_im\\_Mittelstand.pdf](https://www.ptw.tu-darmstadt.de/media/fachgebietptw/dokumente_3/wissenssammlung_ptw/leitfaeden_2/Kuenstliche_Intelligenz_zur_Umsetzung__von_Industrie_4.0_im_Mittelstand.pdf) (besucht am 23.06.2025).
- [6] Gabler Wirtschaftslexikon. *Künstliche Intelligenz (KI)*. 2018. URL: <https://wirtschaftslexikon.gabler.de/definition/kuenstliche-intelligenz-ki-40285/version-263673> (besucht am 23.06.2025).
- [7] Bundesministerium für Wirtschaft und Energie (BMWi). *Technologieszenario „Künstliche Intelligenz in der Industrie 4.0“*. 2019. URL: <https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/KI-industrie-40.pdf> (besucht am 23.06.2025).
- [8] Fraunhofer IKS. *Künstliche Intelligenz (KI) und maschinelles Lernen*. o. J. URL: <https://www.iks.fraunhofer.de/de/themen/kuenstliche-intelligenz.html> (besucht am 24.06.2025).
- [9] Inga Döbel u. a. *Maschinelles Lernen - Eine Analyse zu Kompetenzen, Forschung und Anwendung*. 2018. URL: <https://www.bigdata-ai.fraunhofer.de/de/publikationen/ml-studie.html> (besucht am 24.06.2025).
- [10] Benny Botsch. *Maschinelles Lernen - Grundlagen und Anwendungen*. 2023. DOI: 10.1007/978-3-662-67277-8.
- [11] Christiane Köllner. *Warum KI nicht gleich KI ist*. Dez. 2023. URL: <https://www.springerprofessional.de/en/kuenstliche-intelligenz/neuronale-netze/warum-ki-nicht-gleich-ki-ist/26478010> (besucht am 24.06.2025).

- [12] Michael Grieves. *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*. White Paper. März 2015. URL: <https://www.researchgate.net/publication/275211047> (besucht am 20.05.2025).
- [13] Marco Grafe. *Der digitale Zwilling als eine Säule von Industrie 4.0*. Blogbeitrag, ZEISS Digital Innovation. Juni 2023. URL: <https://blogs.zeiss.com/digital-innovation/de/digitaler-zwilling-saeule/> (besucht am 20.05.2025).
- [14] Krizinger Werner u. a. “Digital Twin in manufacturing: A categorical literature review and classification”. In: *IFAC-PapersOnLine* 51.11 (2018), S. 1016–1022. DOI: 10.1016/j.ifacol.2018.08.474.
- [15] Mohsen Soori, Behrooz Arezoo und Roza Dastres. “Digital Twin for Smart Manufacturing, A Review”. In: *Sustainable Manufacturing and Service Economics* 2 (2023), S. 100017. DOI: 10.1016/j.smse.2023.100017.
- [16] Birgit Boss u. a. *Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0*. Joint Whitepaper. 2021. URL: <https://www.iiconsortium.org/pdf/Digital-Twin-and-Asset-Administration-Shell-Concepts-and-Application-Joint-Whitepaper.pdf> (besucht am 06.06.2025).
- [17] Beuth Verlag. *DIN SPEC 91345:2016-04 - Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)*. Apr. 2016. DOI: 10.31030/2436156.
- [18] International Electrotechnical Commission (IEC). *IEC 63278-1:2023 - Asset Administration Shell for industrial applications - Part 1: Asset Administration Shell structure*. Internationale Norm, IEC. Dez. 2023. URL: <https://webstore.iec.ch/en/publication/65628>.
- [19] Industrial Digital Twin Association (IDTA). *IDTA - Der Standard für den Digitalen Zwilling*. 2025. URL: <https://industrialdigitaltwin.org/> (besucht am 24.05.2025).
- [20] Industrial Digital Twin Association. *Specification of the Asset Administration Shell - Part 1: Metamodel*. Version 3.0.1. Juni 2024. URL: <https://industrialdigitaltwin.org/en/content-hub/aasspecifications> (besucht am 26.05.2025).
- [21] International Electrotechnical Commission (IEC). *IEC 61360-1:2017 - Standard data element types with associated classification scheme - Part 1: Definitions - Principles and methods*. Internationale Norm, IEC. Juli 2017. URL: <https://webstore.iec.ch/en/publication/28560>.

- [22] Industrial Digital Twin Association. *submodel-templates*. 2025. URL: <https://github.com/admin-shell-io/submodel-templates/tree/main> (besucht am 26.05.2025).
- [23] Industrial Digital Twin Association. *Specification of the Asset Administration Shell - Part 5: Package File Format (AASX)*. Version 3.0.1. Juni 2024. URL: <https://industrialdigitaltwin.org/en/content-hub/aasspecifications> (besucht am 26.05.2025).
- [24] Industrial Digital Twin Association. *Specification of the Asset Administration Shell - Part 2: Application Programming Interfaces*. Version 3.0.3. Okt. 2024. URL: <https://industrialdigitaltwin.org/en/content-hub/aasspecifications> (besucht am 26.05.2025).
- [25] Industrial Digital Twin Association. *Specification of the Asset Administration Shell - Part 4: Security*. Version 3.0. Mai 2025. URL: <https://industrialdigitaltwin.org/en/content-hub/aasspecifications> (besucht am 16.06.2025).
- [26] Europäische Kommission. *Der europäische Grüne Deal*. Dez. 2019. URL: <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:52019DC0640> (besucht am 28.05.2025).
- [27] Europäische Union. *Ökodesign-Anforderungen für nachhaltige Produkte*. Juni 2024. URL: <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:52019DC0640> (besucht am 28.05.2025).
- [28] CIRPASS Project. *The Digital Product Passport as defined in the Proposal for Eco-design for Sustainable Product Regulation (ESPR)*. März 2023. URL: <https://cirpassproject.eu/wp-content/uploads/2023/03/ESPR-short-summary-Final.pdf> (besucht am 28.05.2025).
- [29] Kai Garrels u. a. *DPP 4.0: An Architecture Proposal for a DPP-System to implement the EU Digital Product Passport for Industrial Products*. Version 1.1. ZVEI Discussion Paper. Dez. 2023. URL: [https://www.zvei.org/fileadmin/user\\_upload/Themen/Industrie/Fachverband\\_Automation/2023-12\\_Discussion\\_Paper\\_DPP4.0\\_Architecture\\_v1.1.pdf](https://www.zvei.org/fileadmin/user_upload/Themen/Industrie/Fachverband_Automation/2023-12_Discussion_Paper_DPP4.0_Architecture_v1.1.pdf) (besucht am 30.05.2025).
- [30] International Electrotechnical Commission (IEC). *IEC 61406-1:2022 - Identification Link - Part 1: General requirements*. Internationale Norm, IEC. Sep. 2022. URL: <https://webstore.iec.ch/en/publication/67673>.
- [31] Dieter Wegener. *DPP4.0 as a Digital Enabler for “Green Transformation”*. Nov. 2024. URL: <https://www.bundesnetzagentur.de/DE/Fachthemen/Telekommunikation/Technik/DMUEF/Vortraege2023/Wegener.pdf> (besucht am 30.05.2025).

- [32] groninger & co. gmbh. *robocell - Handschuhlose Abfüllmaschine für Spritzen, Vials und Karpulen*. o. J. URL: <https://www.groninger-group.com/de/pharma/aseptische-abfuellung/abfuellanlagen-fuer-rtu-vials-karpulen-und-spritzen/robocell/> (besucht am 20.06.2025).
- [33] Eclipse Foundation. *Eclipse AASX Package Explorer*. 2025. URL: <https://github.com/eclipse-aaspe/package-explorer> (besucht am 18.06.2025).
- [34] Eclipse Foundation. *Eclipse AASX Server*. 2025. URL: <https://github.com/eclipse-aaspe/server> (besucht am 18.06.2025).
- [35] Eclipse BaSyx Community. *BaSyx Wiki*. 2024. URL: <https://wiki.basysx.org/> (besucht am 03.06.2025).
- [36] Eclipse Foundation. *Eclipse BaSyx: Industry 4.0 Operating System*. 2023. URL: <https://eclipse.dev/basysx/> (besucht am 03.06.2025).
- [37] inray Industriesoftware GmbH. *Was ist OPC UA? Die wichtigsten Begriffe im Überblick*. o. J. URL: <https://www.opc-router.de/was-ist-opc-ua/> (besucht am 21.06.2025).
- [38] Industrial Digital Twin Association. *Use Cases - der Digitale Zwilling in der Praxis*. 2025. URL: <https://industrialdigitaltwin.org/use-cases> (besucht am 27.06.2025).
- [39] Industrial Digital Twin Association. *IDTA 02006-3-0 - Digital Nameplate for Industrial Equipment*. Version 3.0. Nov. 2024. URL: [https://github.com/admin-shell-io/submodel-templates/blob/main/published/Digital%20nameplate/3/0/IDTA%2002006-3-0\\_Submodel\\_Digital%20Nameplate.pdf](https://github.com/admin-shell-io/submodel-templates/blob/main/published/Digital%20nameplate/3/0/IDTA%2002006-3-0_Submodel_Digital%20Nameplate.pdf) (besucht am 27.06.2025).
- [40] Industrial Digital Twin Association. *IDTA 02004-1-2 - Handover Documentation*. Version 1.2. März 2023. URL: [https://github.com/admin-shell-io/submodel-templates/blob/main/published/Handover%20Documentation/1/2/IDTA%2002004-1-2\\_Submodel\\_Handover%20Documentation.pdf](https://github.com/admin-shell-io/submodel-templates/blob/main/published/Handover%20Documentation/1/2/IDTA%2002004-1-2_Submodel_Handover%20Documentation.pdf) (besucht am 27.06.2025).
- [41] Industrial Digital Twin Association. *IDTA 02026-1-0 - Provision of 3D Models*. Version 1.0. Juni 2024. URL: [https://github.com/admin-shell-io/submodel-templates/blob/main/published/Provision%20of%203D%20Models/1/0/IDTA\\_02026-1-0\\_Submodel\\_ProvisionOf3DModels.pdf](https://github.com/admin-shell-io/submodel-templates/blob/main/published/Provision%20of%203D%20Models/1/0/IDTA_02026-1-0_Submodel_ProvisionOf3DModels.pdf) (besucht am 27.06.2025).
- [42] Industrial Digital Twin Association. *IDTA 02003 - Generic Frame for Technical Data for Industrial Equipment in Manufacturing*. Version 2.0. März 2025. URL: <https://github.com/admin-shell-io/submodel-templates/blob/main/>

- published/Technical\_Data/2/IDTA%2002003\_Template\_TechnicalData.pdf (besucht am 27.06.2025).
- [43] Industrial Digital Twin Association. *IDTA 02011-1-1 - Hierarchical Structures enabling Bills of Material*. Version 1.1. Juni 2024. URL: [https://github.com/admin-shell-io/submodel-templates/blob/main/published/Hierarchical%20Structures%20enabling%20Bills%20of%20Material/1/1/IDTA%2002011-1-1\\_Submodel\\_HierarchicalStructuresEnablingBoM.pdf](https://github.com/admin-shell-io/submodel-templates/blob/main/published/Hierarchical%20Structures%20enabling%20Bills%20of%20Material/1/1/IDTA%2002011-1-1_Submodel_HierarchicalStructuresEnablingBoM.pdf) (besucht am 27.06.2025).
- [44] Industrial Digital Twin Association. *IDTA 02008-1-1 - Time Series Data*. Version 1.1. März 2023. URL: [https://github.com/admin-shell-io/submodel-templates/blob/main/published/Time%20Series%20Data/1/1/IDTA%2002008-1-1\\_Submodel\\_TimeSeriesData.pdf](https://github.com/admin-shell-io/submodel-templates/blob/main/published/Time%20Series%20Data/1/1/IDTA%2002008-1-1_Submodel_TimeSeriesData.pdf) (besucht am 01.07.2025).
- [45] Industrial Digital Twin Association. *aas-test-engines*. 2025. URL: <https://github.com/admin-shell-io/aas-test-engines> (besucht am 29.06.2025).
- [46] Industrial Digital Twin Association. *Specification of the Asset Administration Shell - Part 3a: Data Specification - IEC 61360*. Version 3.0.2. Juli 2024. URL: <https://industrialdigitaltwin.org/en/content-hub/aasspecifications> (besucht am 26.05.2025).
- [47] Eclipse BaSyx Community. *DataBridge Component*. 2024. URL: [https://wiki.basyx.org/en/latest/content/user\\_documentation/basyx\\_components/databridge/](https://wiki.basyx.org/en/latest/content/user_documentation/basyx_components/databridge/) (besucht am 30.06.2025).
- [48] InfluxData. *Telegraf Documentation*. 2025. URL: <https://docs.influxdata.com/telegraf/v1/> (besucht am 01.07.2025).
- [49] OpenJS Foundation. *Introduction to Node.js*. o. J. URL: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs> (besucht am 04.07.2025).

## A Anhang 1

Hier Anhang einfügen