

Konsortialbericht zum Abschluss der Projektverlängerung

# Verwaltungsschale für den Leitungssatz

Berichtszeitraum 01.12.2024 – 30.06.2025

Förderkennzeichen: 13IK005



**Umsetzung der Verwaltungsschale als  
interoperabler Digitaler Zwilling für  
Entwicklung, Produktion und Montage  
des Leitungssatzes im Automobil  
entlang der Wertschöpfungskette**



## Inhaltsverzeichnis

Inhaltsverzeichnis .....	2
Projektsteckbrief der Projektverlängerung: VWS4LS im Überblick .....	5
1 Projektziele, -struktur und -ergebnisse im Überblick .....	7
1.1 Projektverlängerung – VWS4LS positioniert die Verwaltungsschale in einer zentralen Automotive-Wertkette .....	7
1.2 Projektstruktur der Projektverlängerung .....	9
1.3 Projekt- und Zeitplanung der Projektverlängerung .....	10
2 TP11 – Anbindung an Catena-X.....	11
2.1 Zielsetzung.....	11
2.2 Ergebnis.....	12
2.3 Begrifflichkeiten .....	12
2.4 AP 11.1 – Erarbeitung eines Konzepts zur Entscheidungsfindung.....	13
2.4.1 Vorgehensweise und Inhalte.....	14
2.4.2 Discovery Workflow .....	14
2.4.3 Dataspace Workflow .....	15
2.4.4 Zweite Fassung.....	18
2.5 AP 11.2 – Technische Umsetzung der Systemarchitektur .....	21
2.5.1 Vorgehensweise und Inhalte.....	21
2.5.2 Variationen.....	21
2.5.3 Deployment.....	26
2.5.4 Durchführung Szenario .....	27
2.5.5 Ergebnis.....	37
2.6 AP 11.3 – Identity & Access Management (IAM) .....	38
2.6.1 Zielsetzung .....	38
2.6.2 Vorgehensweise und Inhalte.....	38
2.6.3 Authentifizierung .....	38
2.6.4 Autorisierung.....	40
2.6.5 Regeln – IDTA Security Spezifikation.....	41
2.6.6 Identity Provider .....	41
2.6.7 Ergebnis.....	43
3 TP12 – VWS-Produktkatalog .....	44
3.1 AP 12.1 – Anforderungsanalyse .....	44
3.1.1 Ist-Situation .....	45
3.1.2 Anwendungsfälle.....	46
3.1.3 Typen von Digitalen Zwillingen .....	46
3.2 AP 12.2 - Datenmodellierung .....	48
3.2.1 Submodelle der Typ-VWS.....	48
3.2.2 Benennung der VWS .....	55
3.2.3 Benennung der Submodelle.....	57
3.2.4 Versionierung.....	57
3.2.5 Semantische Referenzierung in der VWS.....	58
3.2.6 Einheitendeklaration in der Concept Description .....	61
3.2.7 Anwendung des VEC .....	62
3.3 AP 12.3 - Extraktionslogik entwickeln .....	67
3.3.1 Komax Testing .....	68
3.3.2 COROFLEX.....	72
3.3.3 KOSTAL .....	74
3.4 AP 12.4 - Verwaltungsschalengenerierung.....	77
3.4.1 Komax Testing .....	77

3.4.2	KOSTAL .....	78
3.4.3	COROFLEX.....	79
<b>3.5</b>	<b>AP 12.5 - Integration und Bereitstellung .....</b>	<b>80</b>
3.5.1	Mnestix-Browser .....	81
3.5.2	Deployment ARENA2036.....	85
<b>3.6</b>	<b>AP 12.6 - Testen und Qualitätssicherung.....</b>	<b>88</b>
<b>3.7</b>	<b>AP 12.7 - Dokumentation und Schulung.....</b>	<b>89</b>
<b>4</b>	<b>TP13 – Standards und Middleware .....</b>	<b>90</b>
<b>4.1</b>	<b>AP 13.1 – Abgleich von Aspekt- und Teilmodellen .....</b>	<b>90</b>
<b>4.2</b>	<b>AP 13.2 – Freigabe der Teilmodelle „Process Parameters“ .....</b>	<b>90</b>
<b>4.3</b>	<b>AP 13.3 – OPC UA Begleitstandard "Wire Harness Manufacturing" .....</b>	<b>91</b>
4.3.1	Prüfprozesse identifizieren .....	91
4.3.2	Use Cases und Prozessparameter definieren.....	92
4.3.3	Abgleich mit Domänenexperten .....	95
4.3.4	Abgleich mit VEC .....	95
4.3.5	Übergabe an die VDMA-Arbeitsgruppe «Wire Harness Manufacturing» .....	96
<b>4.4</b>	<b>AP 13.4 – Semantische Durchgängigkeit .....</b>	<b>96</b>
4.4.1	Modellierung von Bereichswerten.....	96
4.4.2	UUID / GUID .....	97
4.4.3	IRDI (ISO 29005-5) .....	98
4.4.4	URI / IRI .....	100
4.4.5	Semantische Datenbanken.....	101
4.4.6	Klassifizierungssysteme für Einheiten .....	107
4.4.7	Klassifizierungssysteme für Dokumente .....	109
<b>4.5</b>	<b>AP 13.5 – Meta-Level AAS Designer .....</b>	<b>112</b>
4.5.1	Import von semantischen IDs aus Referenzkatalogen .....	112
4.5.2	Handover Documentation-Wizard .....	115
4.5.3	Auswahl des ContentType.....	115
4.5.4	ID-Validation.....	116
4.5.5	Auswahllisten für Produkthierarchien .....	116
4.5.6	ConceptDescription-Management .....	117
<b>4.6</b>	<b>AP 13.6 – XITASO Mnestix Viewer.....</b>	<b>118</b>
4.6.1	Funktionsweise.....	118
4.6.2	Bedienung .....	119
4.6.3	Einstellungen.....	121
4.6.4	Dashboard .....	123
4.6.5	Produktfilterung.....	123
4.6.6	Produktanzeige .....	124
<b>4.7</b>	<b>AP 13.7 – BaSyx .....</b>	<b>125</b>
4.7.1	Anpassungen in Eclipse BaSyx.....	125
4.7.2	BaSyx UI – Modul-Feature.....	125
4.7.3	BaSyx UI – AAS Editor .....	126
<b>5</b>	<b>TP14 – Demonstrator .....</b>	<b>140</b>
<b>5.1</b>	<b>Workflow-gesteuerte Auftragsabwicklung.....</b>	<b>140</b>
5.1.1	BaSyx UI – Leitungssatzherstellung (Demonstrator UI) .....	141
<b>5.2</b>	<b>Änderungsmanagement Einleitung .....</b>	<b>145</b>
5.2.1	Szenario 1: Abonnieren eines Teilmodellelements (update) .....	147
5.2.2	Szenario 2: Abonnieren eines neuen Teilmodells in einer VWS (create) .....	148
5.2.3	Szenario 3: Abonnieren einer neuen VWS (create).....	149
5.2.4	Rollen und ihre Interaktionen .....	150
<b>5.3</b>	<b>Anforderungsanalyse .....</b>	<b>151</b>
5.3.1	Die Arten der zu abonnierenden VWS-Elemente.....	151
5.3.2	Spezifizierung des Event-Elements.....	153

5.3.3	Integration des Events in einen Systementwurf .....	155
5.3.4	Übertragungstechnologie.....	159
5.3.5	Anforderungsliste.....	159
<b>5.4</b>	<b>Konzept .....</b>	<b>160</b>
5.4.1	Systemarchitektur .....	160
<b>5.5</b>	<b>Definition der Event-Metamodellelemente .....</b>	<b>161</b>
5.5.1	Festlegung der Eventtypen .....	163
5.5.2	Modellierung des Event-Elements .....	164
5.5.3	Spezifikation der Infrastrukturkomponenten-Schnittstellen .....	169
<b>5.6</b>	<b>BaSyx UI – Änderungsmanagement .....</b>	<b>174</b>
<b>5.7</b>	<b>Zusammenfassung .....</b>	<b>176</b>
<b>6</b>	<b>Ausblick.....</b>	<b>177</b>
<b>6.1</b>	<b>Offene Themenfelder und Ansätze zu weiterführenden Arbeiten.....</b>	<b>177</b>
6.1.1	DataSpecification – Templates.....	177
6.1.2	Events.....	177
6.1.3	Software als Produkt.....	177
6.1.4	Services als Produkt .....	178
6.1.5	VWS-Generierung mit KI .....	178
6.1.6	Produktkonfigurator.....	178
6.1.7	VWS-Qualitätssicherung und Validierung (regelbasiert und mit KI) .....	178
6.1.8	Produktsuche mit KI .....	178
6.1.9	VWS für e-Commerce.....	178
<b>6.2</b>	<b>Business-Sicht.....</b>	<b>179</b>
	Literaturverzeichnis .....	180
	Abbildungsverzeichnis .....	183
	Tabellenverzeichnis .....	186
	Abkürzungsverzeichnis.....	187
	Anhang .....	188
	Autorenverzeichnis.....	193

## Projektsteckbrief der Projektverlängerung: VWS4LS im Überblick

Das Projekt "**Verwaltungsschale für den Leitungssatz**" ([VWS4LS](#)) ist ein gefördertes Verbundvorhaben im Rahmen des Corona-Konjunkturpaketes, Ziffer 35c der Bundesregierung "**Zukunftsinvestitionen für Fahrzeughersteller und Zulieferindustrie**" von 2020 (<https://www.bmwi.de/Redaktion/DE/Textsammlungen/Industrie/zukunftsinvestitionen-fahrzeughersteller-zulieferindustrie.html>).

Dieses Vorhaben wurde als Teil des Modul a2 gefördert und startete in einer Gruppe weiterer Projekte im Rahmen der ersten Welle der Projektbewilligungen. Weitere Projekte dieser ersten Welle waren beispielsweise Catena-X größtes und SDM4FZI als zweitgrößtes Projekt der Ausschreibung. Die reguläre Projektlaufzeit begann planmäßig am 1. Dezember 2021 und endet am 30. November 2024. Mitte 2024 wurde eine Projektverlängerung beantragt, welcher durch den Projektträger stattgegeben wurde. Somit wurde das Projekt nochmal um 7 Monate bis zum 30.6.2025 verlängert. Die in diesem Dokument zu lesende Inhalte beziehen sich lediglich auf die Projektverlängerung.

Die folgenden 3 Unternehmen stellen zusammen mit der ARENA2036 das Projektkonsortium der Projektverlängerung dar:

	
Hauptansprechpartner: Michele Lagnese Global Director IT & Digital	Hauptansprechpartner: Pascal Neuperger Product Developer & Product Support Manager
	
Hauptansprechpartner: Jürgen Bücken Managing Director	Hauptansprechpartner: Georg Schnauffer Stv. Geschäftsführer

Als **assoziierte Partner** in der Projektverlängerung engagieren sich die in der regulären Projektlaufzeit beteiligten Teilnehmer Dräxlmaier, DI.iT, Festo, Kromberg & Schubert, und Wezag.

Die [ARENA2036 e.V.](#) fungierte als Verbundkoordinator. Die Projektträgerschaft lag bei der VDI Technologiezentrum GmbH [VDI-Technologiezentrum \(vdi-technologiezentrum.de\)](http://vdi-technologiezentrum.de).

Das initial beantragte Projektvolumen lag bei ca. 11,3 Mio. €, das Fördervolumen bei 6,1 Mio. €.

Folgende Veränderungen sind in der Projektlaufzeit im Konsortium aufgetreten:

- Bis zum September 2022 war die KUKA Deutschland GmbH Konsortialpartner von VWS4LS.
- TSK Prüfsysteme GmbH wurde im Oktober 2022 in Komax Testing Germany GmbH umfirmiert und führt das Logo der Komax Gruppe.

Die Konsortialpartner verfolgen mit ihrer Teilnahme an VWS4LS wesentliche Zielstellungen sowohl auf Ebene der Zusammenarbeit im gesamten Liefernetzwerk als auch auf der Ebene des jeweiligen Unternehmens selbst. Im Folgenden nehmen Vertreter des Managements Stellung zu ihren Perspektiven auf das Projektvorhaben und dessen strategischer Bedeutung.

# 1 Projektziele, -struktur und -ergebnisse im Überblick

## 1.1 Projektverlängerung – VWS4LS positioniert die Verwaltungsschale in einer zentralen Automotive-Wertkette

Der Leitungssatz ist das Nervensystem des Fahrzeugs: Energie und Daten werden über den Leitungssatz im gesamten Fahrzeug verteilt. Als eine der teuersten und komplexesten Einzelkomponenten steht die bisherige stark manuell geprägte Wertschöpfung der Leitungssatz-Konfektion vor großen Herausforderungen. Für viele dieser Herausforderungen ist die digitale Durchgängigkeit der Prozessketten eine wesentliche Voraussetzung – insbesondere unternehmensübergreifend. Das ist der Beitrag von „Verwaltungsschale für den Leitungssatz“ (VWS4LS) zur Transformation dieser Wertkette. Bei den in diesem Dokument dargestellten Ergebnissen handelt es sich um den Zeitraum der beantragten Projektverlängerung vom 01.12.2024 bis zum 30.06.2025. Die Ergebnisse aus der Regulären Projektlaufzeit vom 01.12.2021 – 30.11.2024 sind in einem bereits veröffentlichten Dokument auf der Projektwebseite veröffentlicht<sup>1</sup>.

VWS4LS hat in der regulären Projektlaufzeit Grundlagen gelegt, wie die Technologie Verwaltungsschale (VWS) in der Wertkette des Leitungssatzes implementiert werden kann. Einige Kernergebnisse sind nachfolgenden beschrieben:

- Pilotanbindung der Verwaltungsschale an Catena-X
  - Erste Prototypen zeigen, wie eine Integration zwischen VWS und Catena-X technisch realisiert werden kann.
  - Identifizierte Whitespots: fehlende Interoperabilität bei Submodelldefinitionen, unklare Verantwortlichkeiten in der Datenbereitstellung, mangelnde Standards für bestimmte Capability-Typen.
- Entwicklung und Anwendung von Submodellen im PPR-Kontext
  - Submodelle zur Beschreibung von Capabilities in den Bereichen Produkt, Prozess und Ressourcen (PPR-Modell) wurden spezifiziert und erfolgreich auf Use Cases angewendet.
  - Das PPR-Modell ermöglicht die strukturierte Abbildung der Fähigkeiten über den gesamten Fertigungsprozess hinweg.
- Nutzen der Capability-Beschreibungen auf dem Shopfloor
  - Fähigkeiten gestützte VWS ermöglichen eine dynamische Produktionsplanung durch gezielten Abgleich von Anforderungen und verfügbaren Ressourcen.
  - Nutzen auf dem Shopfloor: erhöhte Transparenz, flexible Umplanung, kürzere Rüstzeiten.
- Automatisierte Lösungsfindung basierend auf Capabilities
  - Implementierung eines Matching-Mechanismus, der anhand definierter Strategien (z. B. Effizienz, Nachhaltigkeit) automatisiert Produktionsressourcen auswählt, deren Capabilities die Fertigungsanforderungen erfüllen.
  - Demonstration anhand realitätsnaher Fertigungsszenarien.
- Vorgehen zur Integration von Domänenstandards und VWS
  - Entwicklung eines Mappings zwischen bestehenden Industrie-Standards (z. B. OPC UA, ISO 23247) und der Verwaltungsschale.
  - Ergebnis: Synergiepotenziale durch Wiederverwendung domänenspezifischer Informationen in der VWS-Struktur.

---

<sup>1</sup> [https://arena2036.de/files/FinaleBilder/02\\_Projekte/vws4ls/VWS4LS\\_Abschlussbericht\\_Gesamtprojekt.pdf](https://arena2036.de/files/FinaleBilder/02_Projekte/vws4ls/VWS4LS_Abschlussbericht_Gesamtprojekt.pdf)

- Architekturansätze für VWS-Management
  - Konzepte für Versionierung, Änderungsmanagement, Modularisierung, Synchronisation, Rückverfolgbarkeit und semantische Verlinkung innerhalb der VWS.
  - Erste Referenzarchitektur in Form eines modularen, servicebasierten VWS-Frameworks.
- Virtuelle Inbetriebnahme mit VWS
  - Anwendung der VWS zur Kopplung digitaler Zwillinge mit Simulationssystemen ermöglicht die virtuelle Inbetriebnahme von Produktionssystemen.
  - Reduktion von Fehlern in frühen Planungsphasen und schnellere Iteration in der Prozessentwicklung.
- Submodelle für Data-Retention-Policies und Bill-of-Process Parameters
  - Entwicklung und Pilotierung zweier spezialisierter Submodelle:
    - Data-Retention-Policies zur regelkonformen Datenhaltung.
    - Bill-of-Process Parameters zur standardisierten Beschreibung prozessrelevanter Parameter entlang der Fertigungskette.

In der Projektverlängerung von VWS4LS wird auf den bisher erzeugten Ergebnissen weiter aufgebaut und einige Aspekte weiterentwickelt.

In diesem Bericht werden die erarbeiteten Ergebnisse der Projektverlängerung aus den einzelnen Teilprojekten (TP) dokumentiert.

## 1.2 Projektstruktur der Projektverlängerung

Die Teilprojekte stellten in der Projektverlängerung die dominante Strukturierung des Gesamtprojektes dar. Thematisch wurden die folgenden Teilprojekte in der Projektverlängerung bearbeitet:

- TP11 – Anbindung an Catena-X
- TP12 – Produktkatalog
- TP13 – Standards und Middleware
- TP14 – Demonstration

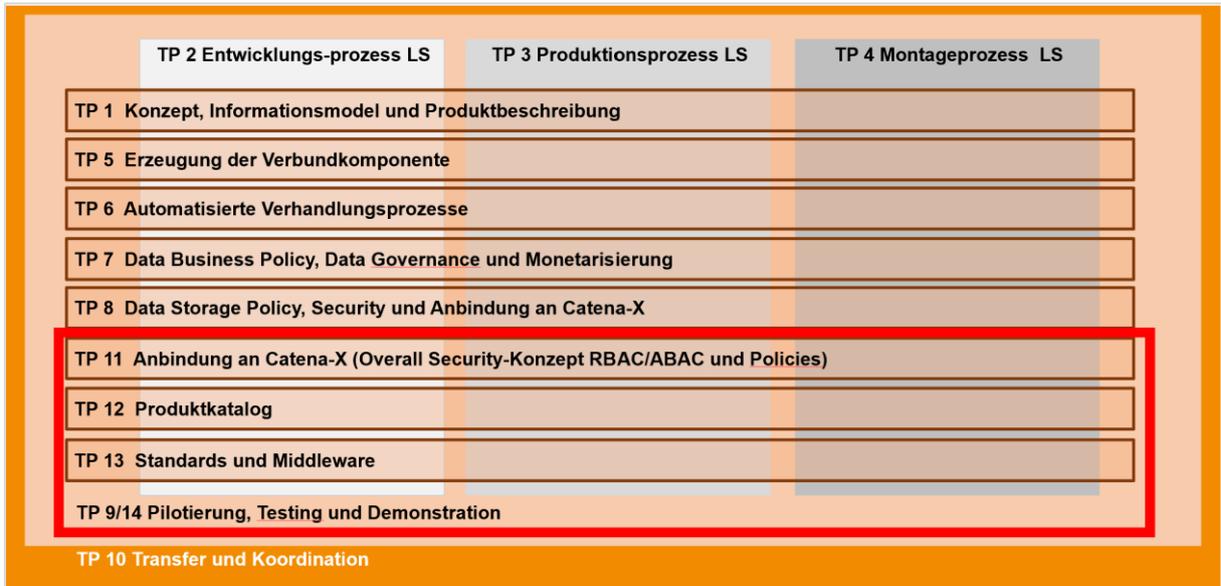


Abbildung 1-1: Matrixstruktur der Teilprojekte in VWS4LS

Für die Bearbeitung der einzelnen Inhalte in den TPs wurde auf Experten aus den verschiedenen Häusern der Partner zurückgegriffen (siehe Abbildung 1-2). Hinzu kamen weitere Experten wie bspw. [4Soft](#), [Fraunhofer IESE](#), [ifak](#), [Meta-Level](#), [XITASO](#) und [msg automotiv GmbH](#), die durch Unterbeauftragungen ihre Expertise in das Projekt im Bereich VWS, dem Vehicle-Electric-Container (VEC) bzw. der Kabelbaumliste (KBL), oder auch aus dem Catena-X Kontext einbringen konnten.

Teilprojekt					
		11	12	13	14
Partner	Haupt-Ansprechpartner	Anbindung an Catena-X	Produktkatalog	Standards/Middleware	Demonstrator
<b>ARENA2036 e.V.</b>	X	X	X	X	X
<b>Coroplast</b>	X	X	X	X	X
<b>Komax Testing</b>	X	X	X	X	X
<b>Kostal</b>	X	-	X	-	-
<b>MSG (Unterauftrag)</b>	-	X	-	-	-
<b>IESE (Unterauftrag)</b>	-	X	-	X	X
<b>ifak (Unterauftrag)</b>	-	-	-	-	X
<b>Meta-Level (Unterauftrag)</b>	-	-	X	-	-
<b>4Soft (Unterauftrag)</b>	-	-	X	-	-
<b>XITASO (Unterauftrag)</b>	-	-	X	-	-
<b>DliT (assoziiert)</b>	X	-	-	-	-
<b>Dräxlmaier (assoziiert)</b>	X	-	-	-	-
<b>Festo (assoziiert)</b>	X	-	-	-	-
<b>Kroschu (assoziiert)</b>	X	-	-	-	-

Abbildung 1-2: Projektteilnehmer VWS4LS (Fettgedruckt → TP-Leitung)

### 1.3 Projekt- und Zeitplanung der Projektverlängerung

Die Projektplanung für die Projektverlängerung sieht vor, alle 4 Teilprojekte (TP11-TP14) vom 1.12.2024 bis zum 30.6.2025 parallel von Beginn bis Ende laufen zu lassen. Der Entschluss alle TPs gemeinsam zu starten und zu enden, wurde gefasst, da es als nicht sinnvoll erachtet wurde aufgrund der Timeline innerhalb der Projektverlängerung die TPs sequenziell zu starten.

Jahresübergreifend wurden seit dem 1.12.2024 bis 30.06.2025 mehr als 50 TP-Termine wahrgenommen siehe Abbildung 1-3. Alle TP-Termine wurden durch das Project Office koordiniert und vom zentralen VWS4LS-Account dem jeweiligen Verteiler als persönliche Outlook-Einladung zur Verfügung gestellt. Zusätzlich zu diesen Terminen fanden innerhalb der TPs, AP-bezogene Termine statt. Diese Termine wurden von den jeweiligen Verantwortlichen selbst organisiert und sind nicht in der Abbildung 1-3 enthalten. Zusammenfassend lässt sich aus Projektsicht festhalten, dass intensiv und mit hohem Zeit- und Arbeitseinsatz an der Umsetzung der in der Projektziele gearbeitet wurde.

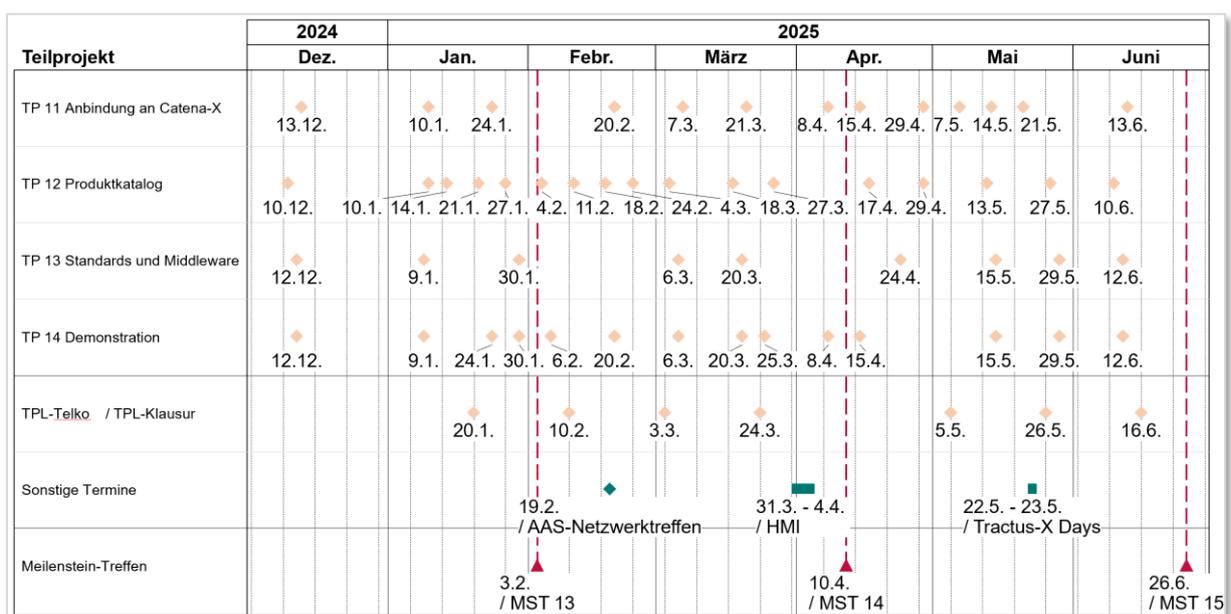


Abbildung 1-3: Projekttreffen Projektverlängerung

## 2 TP11 – Anbindung an Catena-X

Die Digitalisierung industrieller Wertschöpfungsketten erfordert standardisierte Datenmodelle sowie vertrauenswürdige Mechanismen für den Datenaustausch. Die lückenlose digitale Abbildung und Absicherung von Datenflüssen entlang der Wertschöpfungskette ist entscheidend um Effizienz, Rückverfolgbarkeit und Qualität sicherzustellen.

Zwei technologische Bausteine, Verwaltungsschale und Eclipse-Dataspace Connector (EDC) sollen dabei gemeinsam genutzt werden, um vor allem das Accessmanagement bis auf Attributebene zu ermöglichen.

- Die VWS ermöglicht, als standardisierte Grundlage die Erstellung Digitaler Zwillinge eines physischen oder immateriellen Assets, eine strukturierte und semantisch beschreibbare Bereitstellung von Informationen entlang des Lebenszyklus. Über integrierte Mechanismen wie Role-Based-Access-Control (RBAC) oder Attribute-Based-Access-Control (ABAC) lassen sich differenzierte Zugriffskonzepte auf Submodelle und Datenpunkte realisieren.
- Der EDC wiederum stellt eine interoperable Infrastruktur für den sicheren und souveränen Austausch von Daten auf unternehmensübergreifender-Ebene bereit. Über Policies und Contract Definitions lassen sich Nutzungsbedingungen und Zugriffskontrollen entlang von Datenflüssen definieren und automatisiert durchsetzen.

Gerade in der Leitungssatzwertkette – mit ihren hohen Anforderungen an Variantenvielfalt, Fertigungstiefe und Lieferanteneinbindung – eröffnet die Kombination beider Technologien erhebliches Potenzial. Die VWS sorgt für semantisch konsistente Datenstrukturen auf Asset-Ebene (Produktrepräsentanz), während der EDC die kontrollierte Verteilung dieser Informationen über Unternehmensgrenzen hinweg ermöglicht. Eine koordinierte Anwendung der jeweiligen Zugriffskontrollmechanismen – also VWS-internes RBAC/ABAC und EDC-Policies – schaffen dabei die Grundlage für Datensouveränität, Compliance und Effizienz in einem digitalisierten Wertschöpfungsnetzwerk.

Aktuell fehlt es jedoch an einem ersten technischen Durchstich, wie beide Technologien mit deren jeweiligen Konzepten kombiniert angewendet werden können. Eine solche Verbindung ist essenziell, um Zugriffsrechte und Nutzungsregeln konsistent zu übertragen und in durchsetzbare Vertragsbedingungen zu überführen. Der Aufbau eines interoperablen, sicheren Datenraums entlang der Leitungssatzwertkette stellt somit nicht nur einen wichtigen Use Case dar, sondern auch ein Schlüsselszenario für die Zukunft industrieller Datenökosysteme.

### 2.1 Zielsetzung

Die Aufgabenstellung wurde in mehrere Arbeitspakete unterteilt. Zuerst sollte ein detailliertes Konzept zur Entscheidungsfindung bezüglich der Systemarchitektur, insbesondere für den Datenaustausch auf Attribut-Ebene aus einer VWS erstellt werden.

Zu **Arbeitspaket 1** (AP) gehört zunächst die Analyse der Schnittstellen zwischen VWS, Submodellen und einzelnen Attributen, da der normale Transferprozess des EDC diesen spezifischen Schnitt nicht unterstützt. Der Fokus liegt darauf, klare Spezifikationen zu schaffen, die als Grundlage für die technischen Umsetzungen in den nachfolgenden Phasen dienen.

1.1 In *IDTA 01004-3-0-0: Specification of the Asset Administration Shell - Part 4 - Security* [1] wurde ein Security-Konzept entwickelt, welches innerhalb von BaSyx umgesetzt werden soll. Dieses soll die Auswahl der Daten auf Attributebene ermöglichen. Dafür wurden Meetings für die Synchronisation zwischen Fraunhofer IESE, ARENA2036 und msg durchgeführt, um diese Konzepte mit der aktuellen Situation in Catena-X zusammenzubringen.

1.2 Es soll geprüft werden, ob die für den Datenaustausch mit dem EDC gesetzten Policies dynamisch generiert werden können. Hierbei wird festgelegt, ob diese Policies durch einen externen Service, durch Integration in den EDC oder durch Einbau in eine VWS Registry generiert werden können.

In **AP2** soll das Szenario implementiert und folgend die Services auf einer aktuellen Tractus-X Version (Virtuelle Maschine in der IT-Infrastruktur) deployt und erprobt werden. Dazu gehört ebenfalls die Implementierung der attribut- oder rollenbasierten Sicherung von Submodellen innerhalb von BaSyx.

In **AP3** werden die bisherigen Arbeitspakete hinsichtlich der Security-Aspekte zusammengefasst. Im Rahmen dessen soll ein Rollen- und Rechtekonzept beschrieben werden, welches Erkenntnisse aus AP1 in Verbindung mit der Implementierung aus AP2 bringt. Zusätzlich werden Ausblicke zur weiteren Optimierung und zu möglichen Weiterentwicklungen gegeben.

## 2.2 Ergebnis

Das Ziel ist es zum einen, eine umfassende Architektur-Dokumentation, die neben präzisen Schnittstellenbeschreibungen und erforderlichen Anpassungen an bestehenden Komponenten auch konkrete Vorschläge für ein effektives Identity- und Access-Management enthält. Diese Dokumentation bildet die Grundlage für die nachfolgenden Implementierungsschritte und agile Entwicklungsprozesse.

Zum anderen soll als weiteres Ergebnis ein Deployment der EDCs sowie notwendiger föderierter Services aus dem Catena-X Datenökosystem stattfinden, welches die spezifizierten Anforderungen erfüllen kann. Außerdem soll das BaSyx-Deployment weiterentwickelt werden, sodass es in der Lage ist attribut- oder rollen basiert Verwaltungsschalen zu sichern.

Das Ziel ist die Erarbeitung einer Dokumentation zu den Arbeitspaketen, die die Herangehensweise und die finalen Ergebnisse beschreibt.

## 2.3 Begrifflichkeiten

Anbei eine Tabelle 2-1 mit einigen Begrifflichkeiten die für das Verständnisses des Gesamtkontextes wichtig sind.

Begriff	Englischer Begriff	Beschreibung
Authentifizierung	Authentication	Der Prozess, bei dem die Identität eines Benutzers, Geräts oder Systems überprüft wird. Es geht darum, sicherzustellen, dass jemand oder etwas tatsächlich der ist, für den er sich ausgibt. Frage: „Wer bist du?“
Autorisierung	Authorization	Der Prozess, der bestimmt, welche Rechte oder Zugriffsberechtigungen ein authentifizierter Benutzer, Gerät oder System hat. Es legt fest, was jemand tun darf, nachdem seine Identität bestätigt wurde. Frage: „Was darfst du tun?“
EDC	EDC	Der EDC ist eine zentrale Komponente zum sicheren, souveränen Datenaustausch und Zugriffskontrolle und Datenübertragung in föderierten Datenräumen.
EDC Management API	EDC Management API	REST-API zur Verwaltung von Assets, Policies, Verträgen und ihrer Verhandlung
EDC-Protokoll	EDC Protocol	Das EDC-Protokoll bezieht sich auf eine Implementierung des Dataspace Protokolls, welches durch die International Dataspaces Association (IDSA) definiert wurde
Tractus-X	Tractus-X	Referenzimplementierung der Catena-X-Dateninfrastruktur; beinhaltet zentrale Dienste wie Registries, Broker, Identity Provider, der Portal.

Angebot	Offer	Ein Angebot ist ein Eintrag innerhalb eines Katalogs. Er kann dazu verwendet werden, um eine Vereinbarung mit dem Aussteller des Katalogs über den Zugriff auf eine Ressource zu treffen.
Katalog	Catalog	Der Katalog wird bei Anfrage dem externen Benutzer zur Verfügung gestellt. Er enthält verschiedene Angebote für den Zugriff auf Ressourcen des Ausstellers des Katalogs.
Vertrag	Contract	Ein Vertrag ist die Vorstufe zu einem Angebot. Der Besitzer einer Ressource definiert im Normalfall mehrere Verträge und definiert dann Regeln, von wem und wann ein Vertrag abgerufen werden darf. Das Resultat ist dann ein Angebot.
Vereinbarung	Agreement	Eine Vereinbarung ist dann getroffen, wenn der Besitzer einer Ressource ein übermitteltes Angebot angenommen hat.
Regel/Policy	Policy	Der Zugriff auf Verträge wird über Regeln/Policies eingeschränkt.
Security-Submodell	Security-Submodell	Teilmodell einer VWS zur Definition von Zugriffs- und Filterregeln für Daten auf Attributebene.
ABAC	ABAC (Attribute-Based Access Control)	Attribut-basierte Zugriffskontrollmethode, bei der Entscheidungen auf der Ebene von Attributen (z.B. Nutzerrolle, Kontext) getroffen werden.
RBAC	RBAC (Role-Based Access Control)	Rollen-basierte Zugriffskontrolle basierend auf definierten Benutzerrollen.
Single-Sign-On	Single-Sign-On	Authentifizierungsmechanismus, der den Zugriff auf mehrere Systeme mit einmaligem Login ermöglicht.
BPN	BPN	Business Partner Number: Die BPN wird im Allgemeinen bei einer globalen Registrierungsstelle hinterlegt. Von dort kann sie dann abgerufen oder auch validiert werden.
Interner Identity Provider	Internal Identity Provider	Der interne Identity Provider wird vom Besitzer einer Ressource verwendet, um Access-Tokens zum Zugriff auf diese Ressource zu ermöglichen.
Externer Identity Provider	External Identity Provider	Der externe Identity Provider ist eine globale Stelle die Identitäten überprüft und Tokens ausstellen kann.
IDTA	IDTA	Industrial Digital Twin Association e. V.: Verein zur Standardisierung von Digitalen Zwillingen und der VWS
IDTA Security Spezifikation	IDTA Security Specification	Spezifiziert Sicherheitsanforderungen und -richtlinien für die VWS. Darunter auch Autorisierungsregeln zur Filterung von Daten beim Zugriff auf Ressourcen der AAS

Tabelle 2-1: Glossar Anbindung an Catena-X

## 2.4 AP 11.1 – Erarbeitung eines Konzepts zur Entscheidungsfindung

Das erste Arbeitspaket umfasst die Erarbeitung eines detaillierten Konzepts zur Entscheidungsfindung bezüglich der Systemarchitektur und Kommunikationsmethodik. Ziel ist es, klare Spezifikationen zu schaffen, auf deren Basis die technischen Umsetzungen in den nachfolgenden Phasen durchgeführt

werden können. Zusätzlich sollen mögliche Konflikte erkannt und diese in Zusammenarbeit von Fraunhofer IESE, ARENA2036, msg und Catena-X geklärt sowie mögliche Resolutionen erarbeitet werden.

Das Ergebnis dieses Arbeitspakets ist eine vollständige Architektur-Dokumentation, die neben einer präzisen Beschreibung der Schnittstellen, der erforderlichen Anpassungen an bestehenden Komponenten und der definierten Policies auch konkrete Vorschläge für ein effektives Identity- und Access-Management beinhaltet.

#### 2.4.1 Vorgehensweise und Inhalte

Zu Beginn erfolgt eine umfassende Analyse des Ist-Zustands durch einen semantischen und technischen Abgleich der vorhandenen VWS- und Aspektmodelle, um Überschneidungen, Konflikte und Lücken zu identifizieren. Anschließend wird eine detaillierte Systemarchitektur entwickelt, die insbesondere die Anbindung der VWS an den EDC über eine erweiterte Distributed Digital Twin Registry (dDTR) spezifiziert. Dabei sind Transformationsregeln und Application Programming Interfaces (API) zu definieren, um Metadaten und Zugriffsrechte konsistent zwischen den Systemen zu synchronisieren.

Ein weiterer zentraler Bestandteil ist die Konzeption eines Single-Sign-On Mechanismus, der sicherstellt, dass Nutzer nahtlos und sicher auf verschiedene Services zugreifen können. Abschließend erfolgt die Erstellung einer Entscheidungsvorlage in Form eines Summary-Whitepapers, das die Grundlage für die nachfolgenden Implementierungsschritte darstellt. Diese Unterlagen dienen außerdem als Grundlage für agile Entwicklungsprozesse, inklusive Aufwandsschätzungen und Planungen der nächsten Projektschritte.

#### 2.4.2 Discovery Workflow

Der Discovery Workflow ist statisch definiert und wird sich im Rahmen des Projekts nicht ändern. Zum besseren Verständnis wurde er initial dokumentiert:

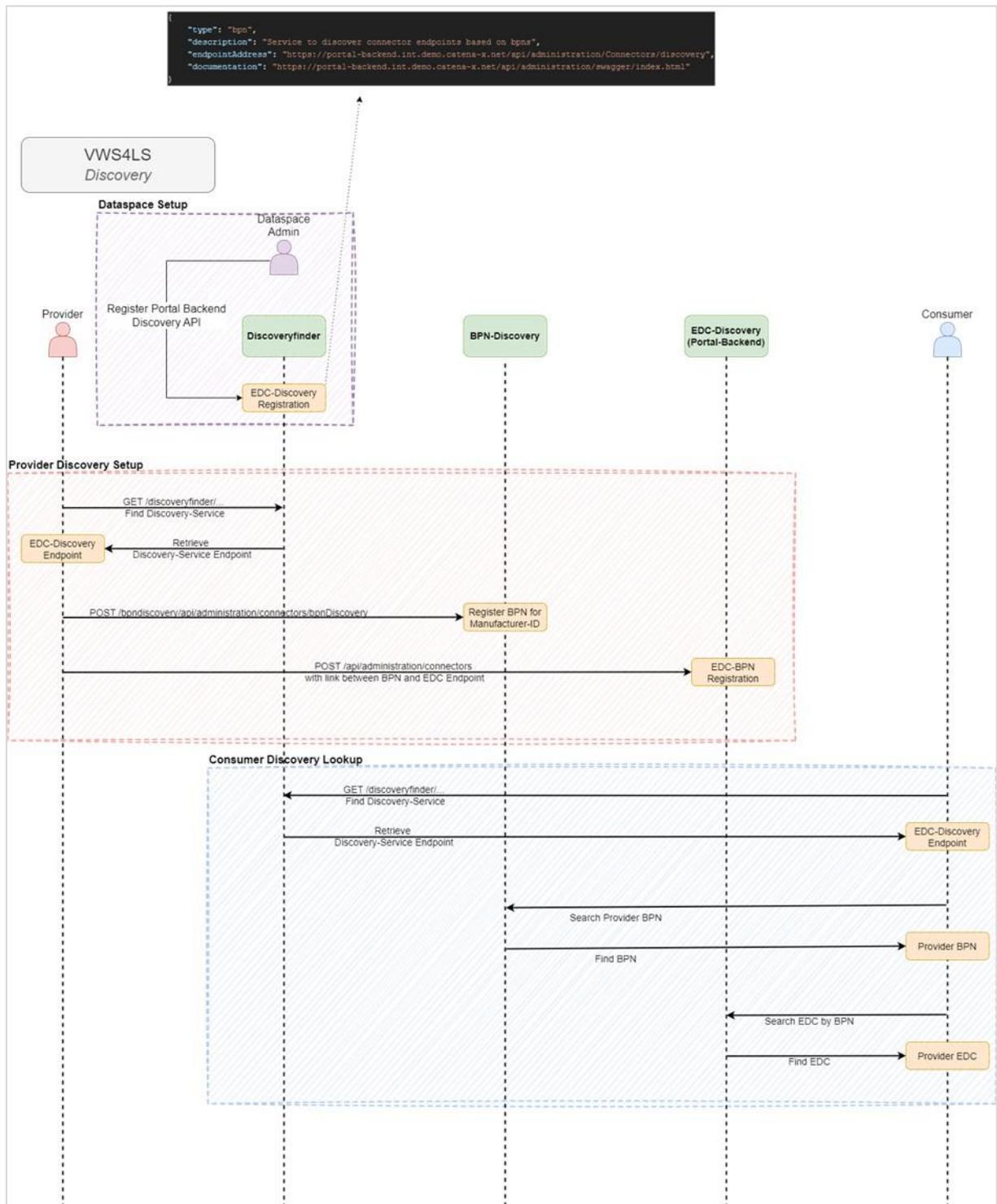


Abbildung 2-1: Sequenzdiagramm für den Discovery-Workflow

### 2.4.3 Dataspace Workflow

Um ein Verständnis für AP 2 zu gewinnen, wurde der vollständige Dataspace-Workflow in mehreren Fassungen zwischen den Projektpartnern diskutiert und weiterentwickelt.

#### 2.4.3.1 Dataspace Workflow - Initiale Fassung

Zunächst wird die initiale Fassung der Gespräche dargestellt. Bei dieser bleiben noch Fragen offen, welche im Folgenden behandelt werden, siehe Abbildung 2-2.

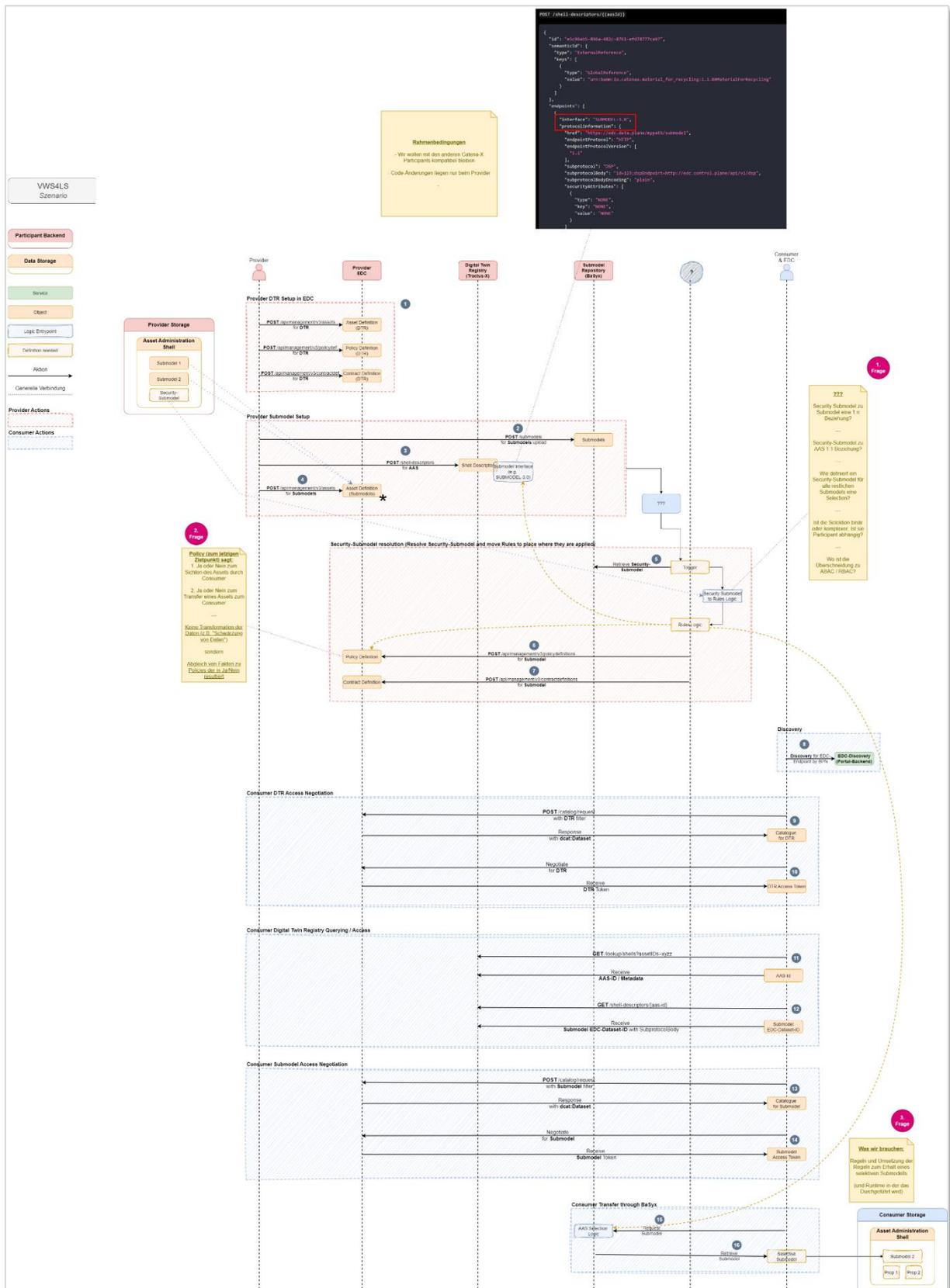


Abbildung 2-2: Initialer Catena-X EDC Transfer-Prozess

Die initiale Fassung beschreibt den vollständigen Catena-X-EDC Transfer-Prozess. Dieser beinhaltet:

- Anlegen von Asset, Policy und Contract für die Authentifizierung auf die Provider Digital Twin Registry für das Auffinden von Submodellen, die von Interesse sind

- In die Digital Twin Registry werden alle Metadaten der angebotenen Submodelle abgelegt, um eine Discovery zu ermöglichen
- Die Erstellung von Asset, Policy und Contract wird ebenfalls für die einzelnen Submodelle erstellt, um diese transferieren zu können
- Über einen noch zu definierenden Mechanismus sollen die logischen Regeln aus dem Security-Submodell extrahiert werden. Für die Anwendung dieser Regeln stehen zu diesem Zeitpunkt mehrere Möglichkeiten zur Verfügung:
  - Die Regeln werden über eine Erweiterung in der Digital Twin Registry umgesetzt. Dafür gibt es ungenutzte Security-Properties als Teil der Shell-Descriptor API, siehe [sldt-digital-twin-registry/docs/architecture/2-architecture-constraints.md](https://github.com/eclipse-tractusx/sldt-digital-twin-registry/blob/main/docs/architecture/2-architecture-constraints.md)<sup>2</sup>
  - Die Regeln werden als Teil der EDC-Policies umgesetzt. Dabei sind mehrere Fragestellungen hinsichtlich Umsetzbarkeit offen (siehe unten).
  - Die Regeln werden auf Seite des Submodel-Repositories bzw. auf BaSyx-Seite beim Provider umgesetzt.

Alle drei Optionen erfordern eine entsprechende Implementierung und sind von der Umsetzung sehr komplex, da sie mehrere Systeme beeinflussen.

Zudem müssen folgende Rahmenbedingungen beachtet werden:

- Potenzielle Code-Änderungen an EDCs, der Digital Twin Registry oder anderen Komponenten sollen nur beim Provider liegen.
- Die Implementierung soll mit anderen Catena-X Participants kompatibel bleiben. Dies ist ggf. umsetzbar, solange die Änderung ausschließlich beim Provider stattfindet und bspw. Consumer-EDCs nicht aktiv an dem Security-Mechanismus beteiligt sind.

### Offene Fragen: Security Submodell

Die folgenden Fragen haben sich zum Security-Submodell ergeben:

- Ist das Security-Submodell zum übertragenen Submodell eine 1:n oder eine 1:1 Beziehung?
- Wie definiert ein Security-Submodell eine Selektion für die zu übertragenden Submodelle?
- Ist die Selektion binär (Attribut anzeigen/nicht anzeigen) oder komplexer?
- Ist die Selektion Consumer-abhängig?
- Ist die Selektion allgemeingültig auf dem Submodell, unabhängig vom Consumer?
- Wo ist die Überschneidung zu ABAC / RBAC?

### Annahmen zu Policies

Eine Policy hat (zum jetzigen Zeitpunkt) folgende Limitationen:

- Die Evaluation der Policies ergibt eine binäre Antwort (Policy erfüllt / Policy nicht erfüllt) zum Sichten des Angebots durch den Consumer.
- Die Evaluation der Policies ergibt eine binäre Antwort (Policy erfüllt / Policy nicht erfüllt) zum Transfer des Angebots durch den Consumer.
- Es findet keine Transformation der Daten (z.B. "Schwärzung von Daten" / Filterung der Daten) statt, sondern nur ein Abgleich von Fakten zu Policies, der in der binären Antwort resultiert.

---

<sup>2</sup> <https://github.com/eclipse-tractusx/sldt-digital-twin-registry/blob/main/docs/architecture/2-architecture-constraints.md#asset-administration-shell-domain-model>

## Anforderungen an Security-Submodell

Was für die Umsetzung des Security-Submodell benötigt wird:

- Definition von Regeln und Umsetzung der Regeln zum Erhalt eines selektiven Submodells
- Eine Runtime, in der die Translation zu Regeln durchgeführt wird
- Eine Runtime, in der die Selektion des Submodells durchgeführt wird

### 2.4.4 Zweite Fassung

Für die zweite Fassung, siehe Abbildung 2-3 wurden folgende Änderungen am Konzept durchgeführt:

- Abwendung von Security-Submodels, da ein neues Security-Konzept durch die IDTA standardisiert wird.
- Die Security-Regeln sowie Format, Scope, Rahmenbedingungen und Funktionalität werden durch die IDTA Security Spezifikation definiert<sup>3</sup>.

Basierend darauf wurden folgende Entscheidungen getroffen:

- Policies werden zur Authentifizierung eines Participants für den Zugriff auf ein Token genutzt, welches auf das BaSyx-Backend autorisiert ist
  - Offene Punkt: Policies, die mit Catena-X kompatibel sind, könnten auch autogeneriert sein, aber eine Runtime dafür würde fehlen.
- Sowohl Tractus-X (Datenökosystem-Infrastruktur) als auch BaSyx (VWS-Backend) nutzen für Demonstrationszwecke dieselbe Keycloak-Instanz
  - Die Regeln werden dateibasiert und zentral an einer Stelle verwaltet.
  - Für eine hohe Anpassbarkeit, werden für alle BaSyx Repositories, Registries und die Discovery explizite Regeln definiert
  - Zunächst liegt der Fokus auf die Umsetzung RBAC Regeln, da diese den Anwendungsfall ausreichend abdecken und die Komplexität damit gering gehalten wird
  - Im Ausblick ist die Erweiterung auf weitere Quellen für Regeln möglich

---

<sup>3</sup> <https://industrialdigitaltwin.org/content-hub/aasspecifications/specification-of-the-asset-administration-shell-part-4-security-idta-number-01004>

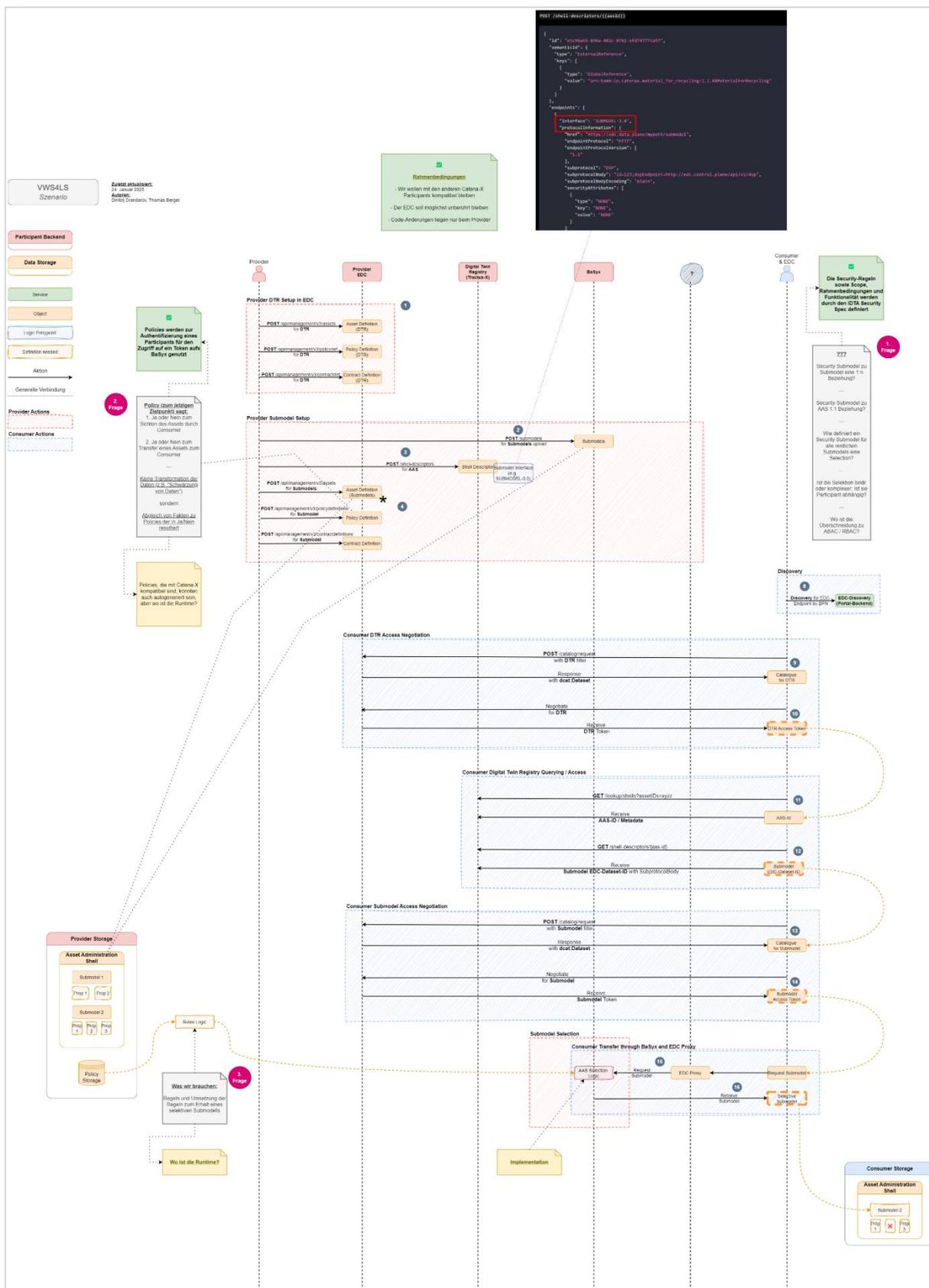


Abbildung 2-3: Zweite Fassung des Catena-X EDC Transfer-Prozess

### 2.4.4.1 Finale Fassung

In der finalen Fassung, siehe Abbildung 2-4 wurden folgenden Entscheidungen getroffen:

- Entscheidung mit Fraunhofer IESE zum Switch von der Tractus-X Registry auf die BaSyx Registry
  - Der Prozess wird dadurch signifikant simplifiziert, da nur noch eine Freigabe auf die BaSyx Infrastruktur stattfinden muss, statt einzelne Services freigeben zu müssen.
  - Dies Ermöglicht volle Kontrolle über den Autorisierungs- & Authentifizierungsprozess durch den Provider.

Folgende Entscheidungen wurden final bestätigt:

- Initial wird dieselbe Identity-Provider-Instanz für die Tractus-X und die BaSyx Infrastruktur genutzt.
- Die Security-Regeln sowie Scope, Rahmenbedingungen und Funktionalität werden durch die IDTA Security Spec. Definiert.

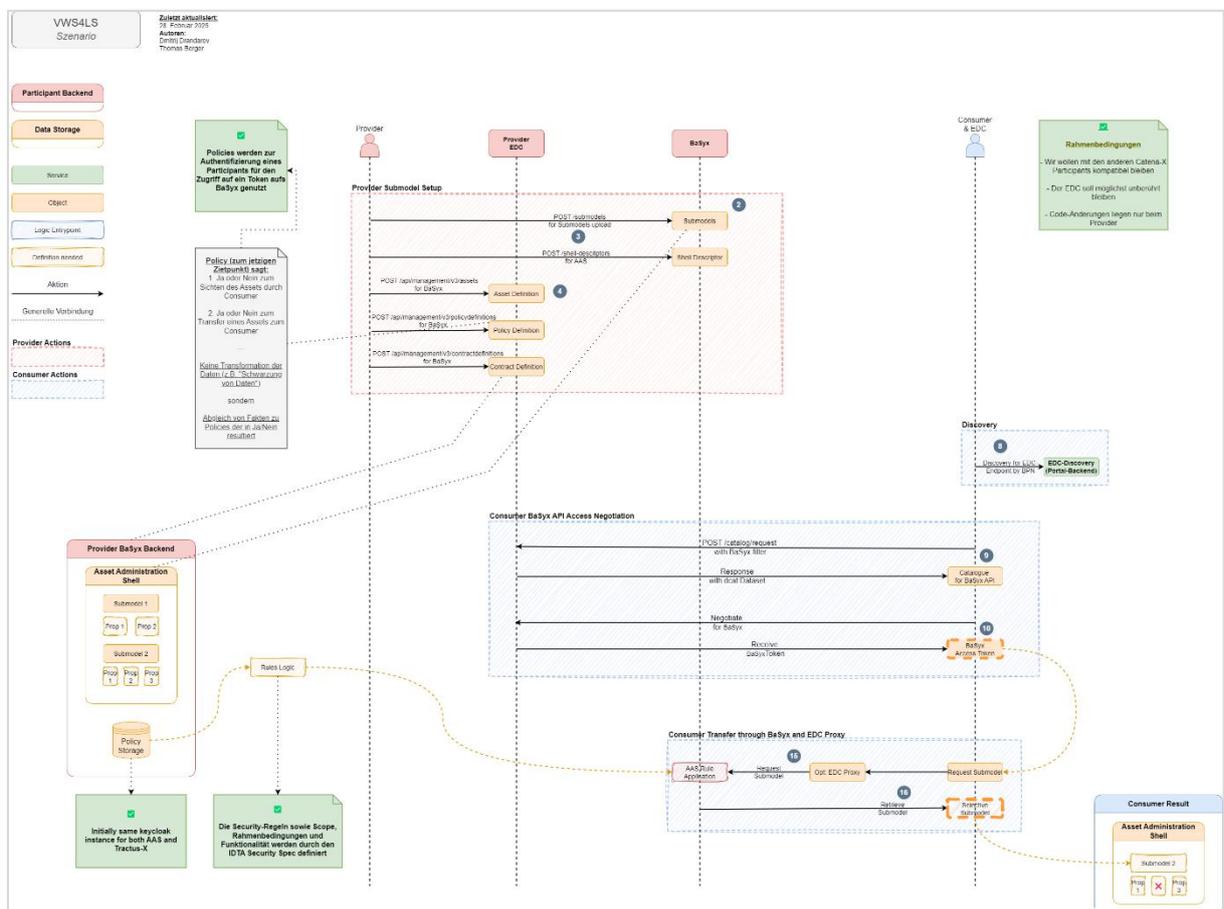


Abbildung 2-4: Finaler Catena-X EDC Transfer-Prozess

## 2.5 AP 11.2 – Technische Umsetzung der Systemarchitektur

Im zweiten Arbeitspaket liegt der Schwerpunkt auf der technischen Umsetzung des zuvor entwickelten Konzepts. Ziel ist die Implementierung eines funktionsfähigen Proof-of-Concepts (PoC), der den Transfer auf Attribut- bzw. Rollen-Ebene mit dem EDC, sowie BaSyx-Komponenten<sup>45</sup> ermöglicht.

### 2.5.1 Vorgehensweise und Inhalte

Die initiale Fassung der föderierten Services ([Industry Core](#)<sup>6</sup>) wird von dem Projekt der ARENA2036-X bereitgestellt und an den entsprechenden Stellen angepasst.

Zusätzlich werden dedizierte EDCs spezifisch für diesen Use-Case deployt mit denen das Szenario durchgespielt werden kann. Zudem wird eine aktualisierte Version von BaSyx auf der virtuellen Maschine Tractus-X 07 VM von Fraunhofer implementiert und deployt, auf welche der Datentransfer basieren soll.

### 2.5.2 Variationen

Im Rahmen dieses Projekts wurden verschiedene Variationen spezifiziert, um den Einsatz von Identity-Providern und Token-Konfigurationen zu optimieren. Diese Variationen zielen darauf ab, unterschiedliche Szenarien zu unterstützen und die Flexibilität sowie Sicherheit der Systemarchitektur zu erhöhen.

Die folgenden Abschnitte erläutern die einzelnen Varianten im Detail und bieten eine umfassende Übersicht über ihre Zielsetzung und Begründung sowie die Konfiguration von Keycloak.

Für die Umsetzung des Szenarios wird zunächst die Basisvariante umgesetzt und getestet, um einen PoC im Rahmen der noch verfügbaren Projektlaufzeit zu finalisieren. Diese Variante wird ausführlich in Kapitel 2.5.2.1 beschrieben. Weitere Varianten mit zwei Identity Providern und zwei Token werden in den Kapiteln 2.5.2.2 (Zwei Identity Providern) und 2.5.2.3 (Zwei Token Variante) beschrieben.

#### 2.5.2.1 Shared Identity-Provider / Basisvariante

Ein Keycloak-Server wird verwendet, um die Authentifizierung zu zentralisieren und den Zugriff auf geschützte Ressourcen zu steuern. Für die Kommunikation zwischen den Teilnehmern wird ein Tunnel zwischen dem Consumer und dem Provider über die EDCs der beiden Teilnehmer aufgebaut. Diese Kommunikation erfolgt durch eine Vertragsverhandlung über die EDC Control Plane.

Die Architektur des beschriebenen Systems umfasst mehrere wichtige Komponenten und Schritte:

- Ein einzelner Keycloak-Server: Dieser Server übernimmt die zentrale Verwaltung und Authentifizierung der Benutzer und Clients. Durch die Implementierung von RBAC/ABAC-Regeln auf dem BaSyx-Server gewährleistet der Keycloak-Server im Zusammenspiel mit dem BaSyx Authorization Feature eine präzise und flexible Steuerung des Zugriffs auf verschiedene Ressourcen.
  - Wegen der deutlichen höheren Komplexität bei der Implementierung von ABAC-Regeln in BaSyx, wird das Konzept vorerst mit RBAC-Regeln getestet
- EDC als Proxy und Kontrolleinheit: Der EDC dient als Proxy und ermöglicht die sichere Kommunikation zwischen Consumern und Providern. Er übernimmt die Rolle einer Kontroll- und Vermittlungseinheit, die den Zugriff auf den BaSyx-Server über die EDC Data Plane regelt.
- Aufbau des Tunnels: Die Verbindung zwischen den Teilnehmern erfolgt über die EDCs beider Parteien. Diese Konnektoren ermöglichen eine Vertragsverhandlung über die EDC Control Plane

---

<sup>4</sup> [https://wiki.basysx.org/en/latest/content/user\\_documentation/basysx\\_components/v2/VWS\\_repository/features/authorization.html](https://wiki.basysx.org/en/latest/content/user_documentation/basysx_components/v2/VWS_repository/features/authorization.html)

<sup>5</sup> <https://github.com/eclipse-basysx/basysx-java-server-sdk/tree/main/examples/BaSyxSecured>

<sup>6</sup> <https://catenax-ev.github.io/docs/next/standards/CX-0126-IndustryCorePartType>

Ein wichtiger Aspekt der Authentifizierung ist die Generierung von Tokens. Diese Tokens werden aus der Kombination von Client-ID, Token-URL und Client-Secret erzeugt, welche vom Provider im Asset definiert werden. Durch diese Methode wird sichergestellt, dass nur autorisierte Benutzer auf die Ressourcen zugreifen können.

Ein zentraler Bestandteil des Systems ist der Einsatz von Keycloak für die Authentifizierung und Autorisierungsregeln. Die Nutzung von Keycloak-Clients ermöglicht die Definition von RBAC und ABAC-Rollen und -Attributen. Der Provider hat die vollständige Kontrolle über die Client-Konfiguration, insbesondere hinsichtlich der Rechte auf den BaSyx Server. Dieser Client wird effektiv über den EDC angeboten.

Die Implementierung des beschriebenen Szenarios erfordert mehrere sorgfältig geplante Schritte:

- Bereitstellung und Konfiguration des Keycloak-Servers: Die Einrichtung und Konfiguration des Keycloak-Servers ist der erste Schritt. Hierbei werden die verschiedenen Benutzer, Clients und deren Rollen bzw. Attribute definiert.
- Deployment der EDC-Komponenten: Als nächster Schritt erfolgt die Bereitstellung der EDCs, welche die Verbindung zwischen den Konsumenten und Providern herstellen.
- Aufbau des Tunnels: Nachdem die EDCs bereitgestellt wurden, wird der Tunnel zwischen den Teilnehmern aufgebaut. Dieser Tunnel ermöglicht eine sichere und zuverlässige Kommunikation.
- Erstellung der Authorisierungsregeln in BaSyx: Für alle Ressourcen in der BaSyx Discovery, den Repositories und den Registries werden Zugriffsregeln in den JSON-Dateien definiert.
- Erstellung und Konfiguration der Keycloak-Clients: Die Keycloak-Clients werden erstellt und konfiguriert, um die notwendigen Rollen oder Attribute zu definieren. Diese Informationen werden mit den in BaSyx definierten RBAC/ABAC-Regeln abgeglichen, um den Zugriff auf den BaSyx Server zu steuern.
- Generierung und Verwaltung von Tokens: Schließlich erfolgt die Generierung und Verwaltung der Authentifizierungstokens, die für den Zugriff auf die Ressourcen benötigt werden.

Dieses Dokument hat die wesentlichen Aspekte der technischen Umsetzung eines Systems beschrieben, das auf einer einzigen Keycloak Instanz basiert. Durch die detaillierte Beschreibung der Komponenten und Implementierungsschritte wird ein umfassender Überblick über die erwarteten Ergebnisse und die Vorteile dieser Architektur geboten, siehe Abbildung 2-5.

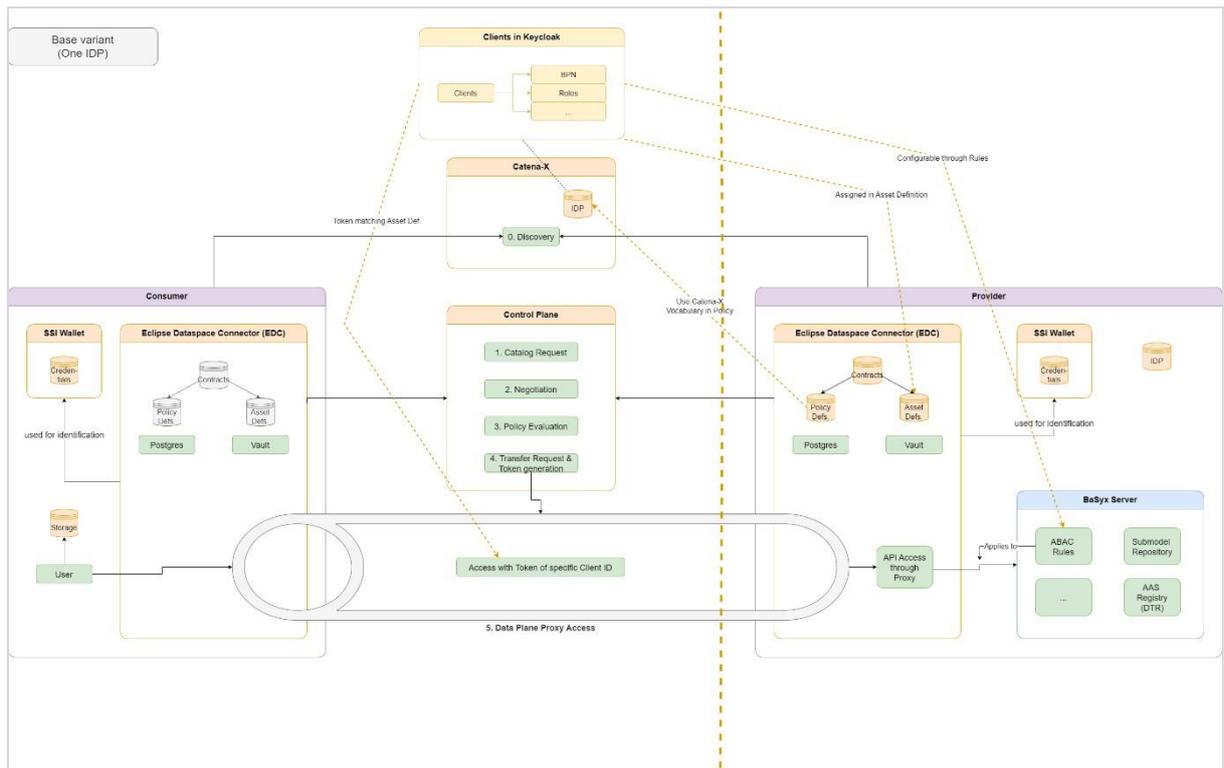


Abbildung 2-5: Architekturübersicht

### 2.5.2.2 Zwei Identity-Provider Variation

Für die Variante der Nutzung von zwei Identity-Providern wird ein Keycloak für die föderierten Tractus-X Services verwendet, während ein weiterer Keycloak vom Provider selbst eingesetzt wird. Diese Konfiguration bietet mehrere Implementierungsoptionen, welche den Grad der Kontrolle und Flexibilität des Providers erhöhen. Eine Übersicht des entwickelten Vorgehens ist in Abbildung 2-6 dargestellt.

Eine Möglichkeit besteht darin, die Provider-Clients und Keycloak-Clients über die Policy- und Asset-Definition zuzuordnen. In der Asset-Definition wird der passende Client beziehungsweise Business Partner zu der Business Partner Number (BPN) in der Policy-Definition angegeben. Dadurch können die Zuordnung und Verwaltung der Rechte und Rollen auf feingranulare Weise erfolgen, was den Bedürfnissen des Providers gerecht wird.

Eine alternative Implementierungsmethode ist die Föderierung der Keycloaks. Durch das Teilen der Clients in der föderierten Struktur können die Vorteile der Basisvariante genutzt werden. Dabei bleibt der Provider in der Lage, das Rollen- und Rechtemanagement effektiv zu steuern. Dies ermöglicht eine zentrale Verwaltung der Authentifizierungs- und Autorisierungsrichtlinien, während gleichzeitig eine flexible Handhabung der Clients und ihrer Ressourcen gewährleistet wird.

Die beschriebenen Ansätze bieten dem Provider die Möglichkeit, das Zugangskontrollsystem an die spezifischen Anforderungen und Geschäftsprozesse anzupassen. Diese Flexibilität ist besonders wichtig, um verschiedenen Sicherheitsanforderungen gerecht zu werden und die Benutzerfreundlichkeit für die Endbenutzer zu maximieren. Durch die detaillierte Planung und Implementierung der beschriebenen Szenarien kann ein robustes und anpassungsfähiges System zur Verwaltung von Identitäten und Zugriffsrechten geschaffen werden.

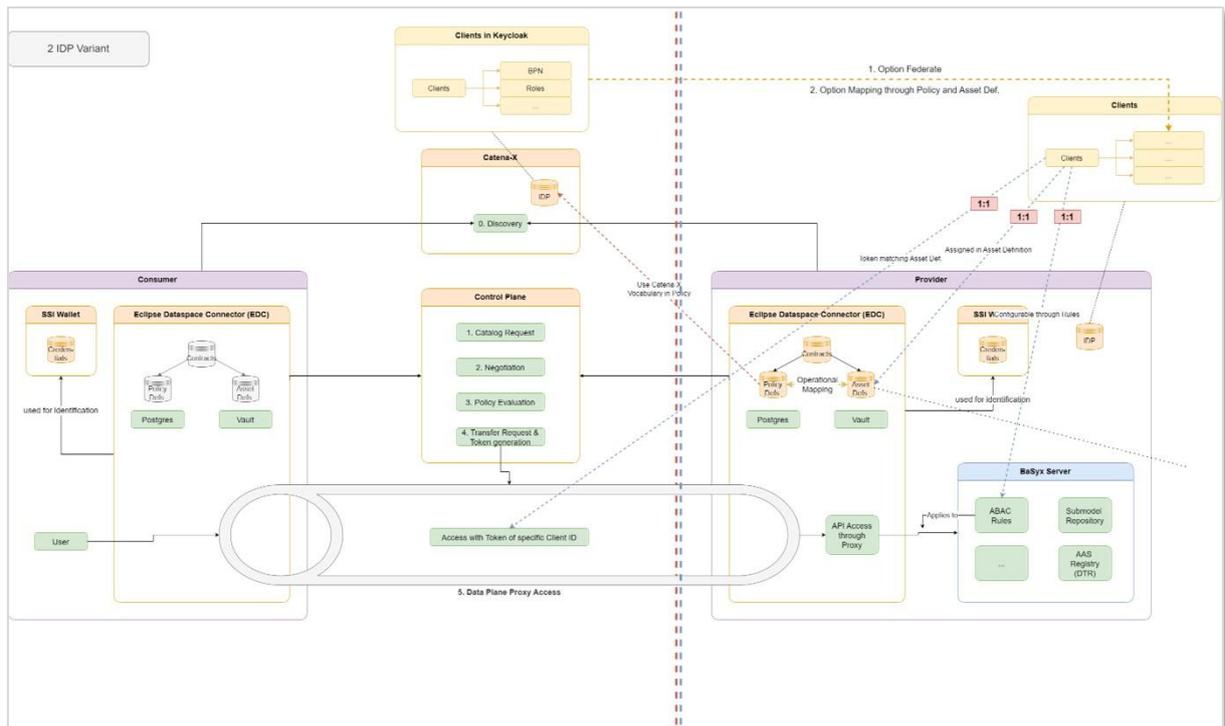


Abbildung 2-6: Architekturübersicht mit zwei Identity Providern

### 2.5.2.3 Zwei Token Variation

Die entscheidende Differenz bei dieser Variante besteht darin, dass nach der Weiterleitung des Requests über den EDC-Tunnel ein weiterer Token vom EDC generiert und intern für den Proxy genutzt wird. Eine Übersicht des entwickelten Vorgehens ist in Abbildung 2-7 dargestellt.

Dies bietet erweiterte Sicherheit, da der Zugang zu den Ressourcen noch stärker kontrolliert werden kann. Allerdings führt dies auch zu einem höheren Implementierungsaufwand beim EDC, weil zusätzliche Schritte zur Token-Generierung und Verwaltung erforderlich sind.

Diese zusätzliche Sicherheitsebene garantiert, dass nur autorisierte Anfragen den Proxy erreichen und bearbeitet werden, wodurch die Integrität und Vertraulichkeit der Daten gewahrt werden.

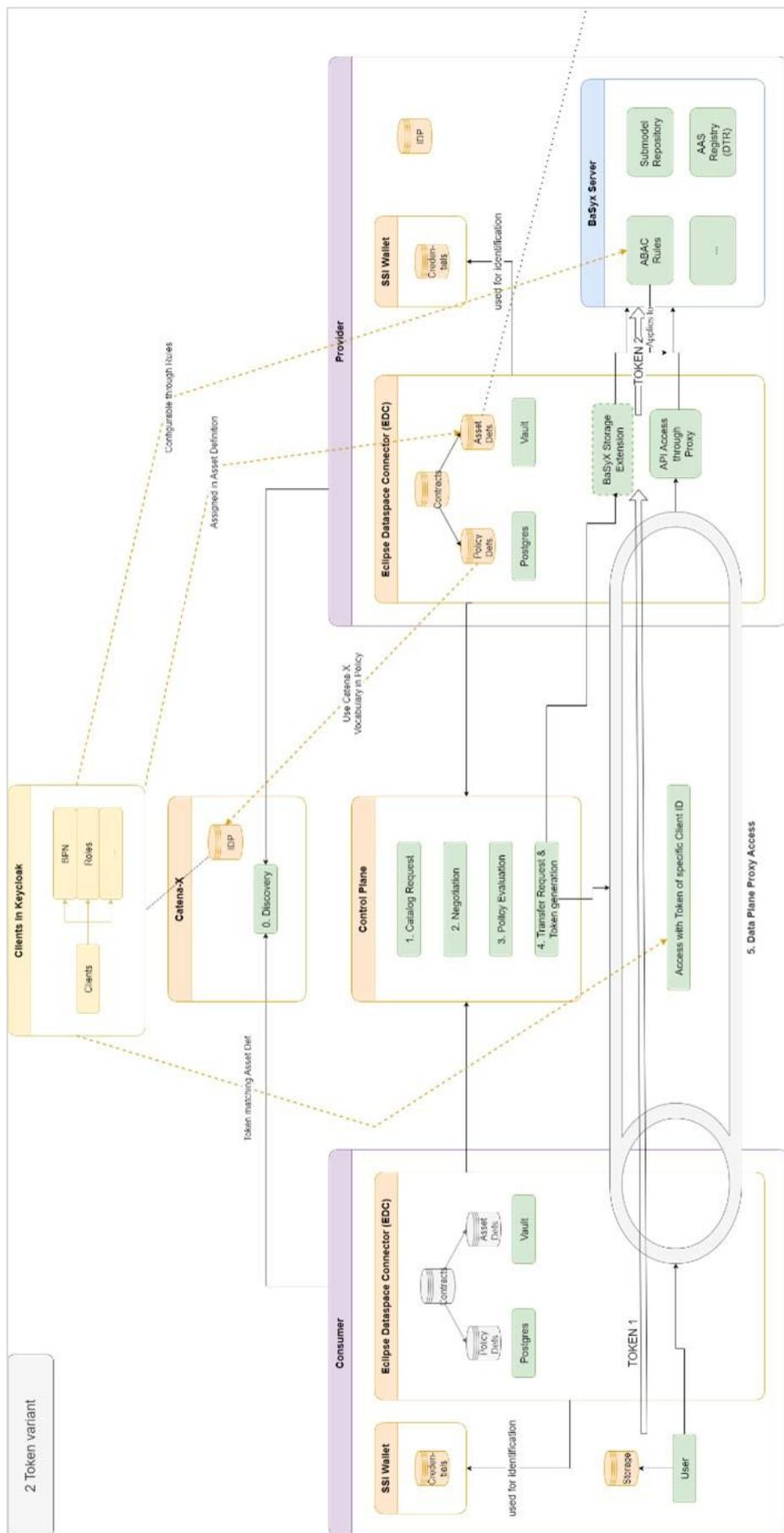


Abbildung 2-7: Architekturübersicht mit zwei Token Variation

### 2.5.3 Deployment

Zur Nutzung der bereitgestellten föderierten Services, werden erweiterte Umbrella-Charts von Tractus-X genutzt. In diesem Szenario wird ein Provider- und ein Consumer-EDC inkl. peripherer Service deployt, siehe Abbildung 2-8.

Die konkrete Definition der values.yaml kann in dem entsprechenden Repository überprüft werden. Die folgenden Schritte zeigen den genauen Ablauf dieses Prozesses:

1. Wechseln auf das Verzeichnis der Umbrella Charts:
  - `cd .\charts\umbrella\`
2. Erstellen der Abhängigkeiten der Helm Charts:
  - `helm repo add tractusx-dev https://eclipse-tractusx.github.io/charts/dev/`
  - `helm repo add runix https://helm.runix.net`
  - `helm dependency build`
3. Installieren der Charts mit dem folgenden Befehl auf der entsprechenden Kubernetes Umgebung:
  - `helm install --create-namespace -n vws4ls -f values-vws4ls.yaml vws4ls .`
4. Setze Secret für Keycloak Token Request in dem EDC-Vault-Pod per Shell:
  - `vault kv put secret/{{PROVIDER_CLIENT_SECRET_KEY}} content={{PROVIDER_CLIENT_SECRET}}`

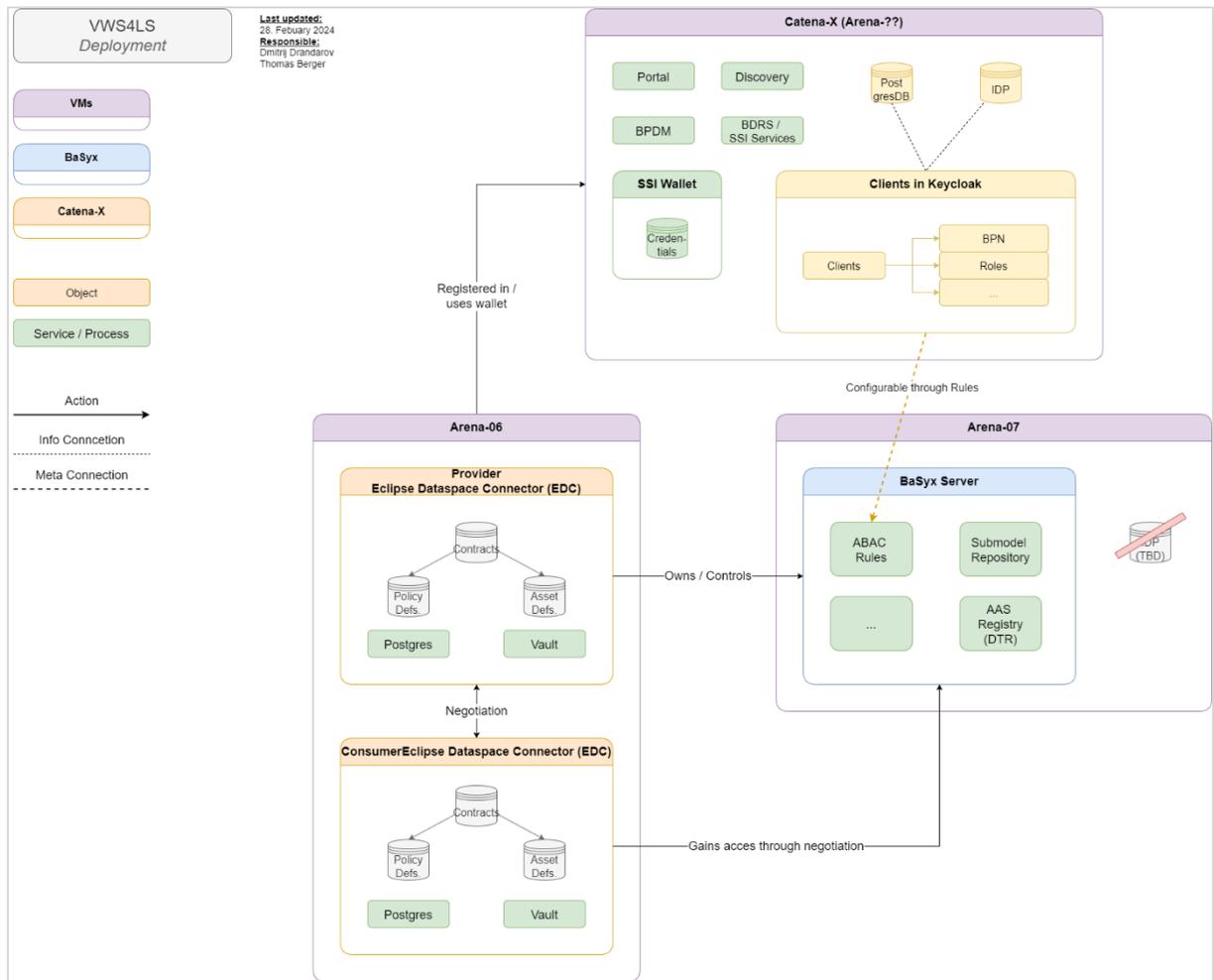


Abbildung 2-8: Deployment-Diagramm

### 2.5.4 Durchführung Szenario

Zu Beginn muss der EDC im Portal (siehe Abbildung 2-9) für den eigenen Nutzer registriert werden. Dies kann man unter dem Punkt „Connector Management“ durchführen:

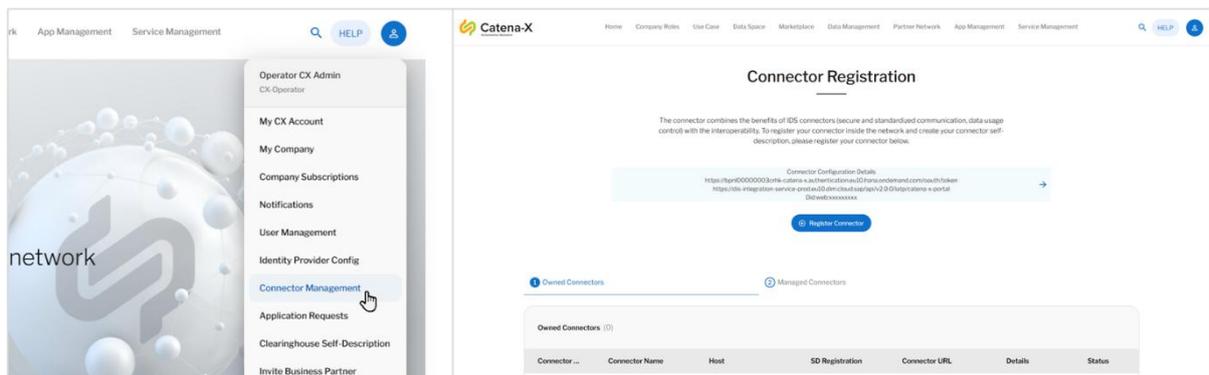


Abbildung 2-9: Connector Management (links) und -Registration (rechts)

Konkret sind die durchzuführenden Konfigurationen (siehe Abbildung 2-10) unter dem hell-blauen Panel ersichtlich:

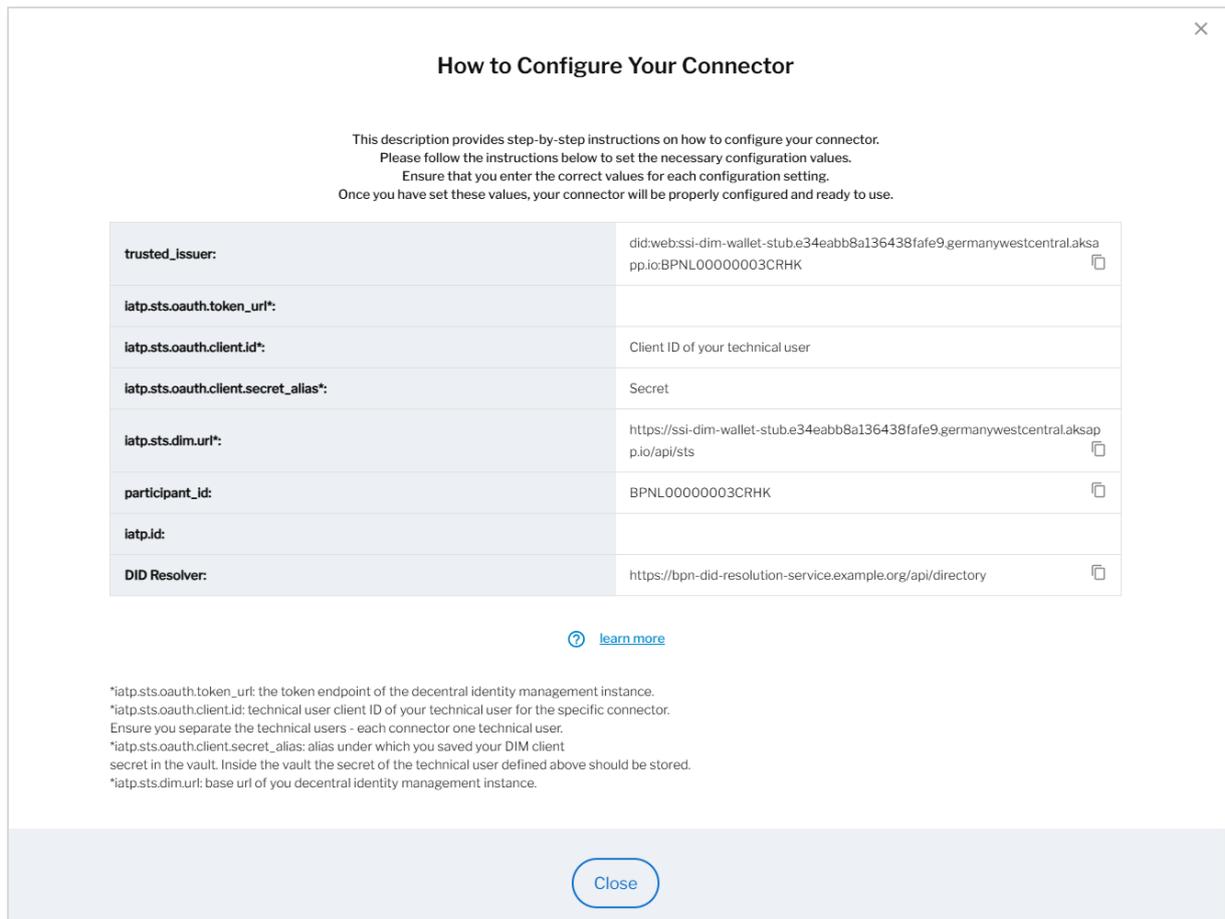


Abbildung 2-10: Connector Configuration

Diese User-spezifischen Werte müssen in den entsprechenden Umbrella-Charts hinterlegt werden. Für das vorliegende Szenario werden einfachheitshalber automatisch generierte EDCs der Umbrella-Charts genutzt, die mit den korrekten Daten vorbefüllt sind.

Zuletzt werden die deployten EDCs mit ihren konkreten Endpoints registriert, siehe Abbildung 2-11. Ein Beispiel für Endpoints ist folgende Adresse:

<https://dataprovider-vws4ls.e34eabb8a136438fafe9.germanywestcentral.aksapp.io/api/v1/dsp>

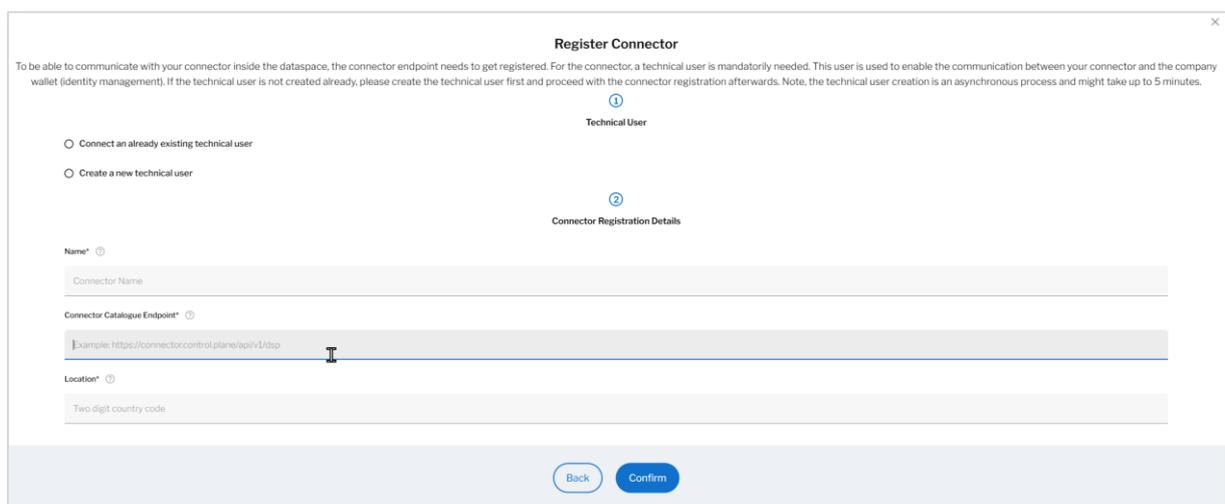


Abbildung 2-11: Abschluss Connector Registration

### 2.5.4.1 Data Provisioning

Folgend wird der Data-Provisioning-Teil beschrieben, bestehend aus der Population des BaSyx-Backends, sowie der Anlegung der Assets, Policies und Contracts im EDC.

Für die Durchführung des Szenarios wird eine Postman Collection beigelegt, welche alle Schritte von Provisioning bis Consumption definiert. Zusätzlich werden diese hier nochmal separat beschrieben.

Die folgenden Schritte müssen an den entsprechenden Endpoints der Provider EDC Management API und des BaSyx-Environments durchgeführt werden, um die Submodelle bereitzustellen, eine Contract Negotiation und den dazugehörigen Transfer des Keycloak-Tokens durchzuführen und schlussendlich den Zugriff auf die Submodelle zu ermöglichen.

#### 2.5.4.1.1 BaSyx-Backend

Im Rahmen des Berichtszeitraums wurde das Eclipse BaSyx Sicherheitsfeature eingesetzt und weiterentwickelt, um eine regelbasierte Zugriffssteuerung zu ermöglichen. BaSyx gliedert sich in mehrere Komponenten: das AAS-Repository, Submodel-Repository und ConceptDescription-Repository sowie die AAS-Registry, Submodel Registry und den AAS-Discovery Service. Die ersten drei Repository-Komponenten werden häufig als "AAS-Environment" zusammengefasst.

Das Sicherheitsfeature implementiert zunächst eine RBAC in Verbindung mit Keycloak, das als Identity Provider fungiert. Nutzer und deren Rollen werden in Keycloak gespeichert, während die Zugriffsregeln in BaSyx hinterlegt sind, um den Zugriff auf Ressourcen im Backend zu steuern. Diese Regeln werden in einer oder mehreren JSON-Dateien als Array abgelegt und folgen dem JSON-Schema in **Fehler! Verweisquelle konnte nicht gefunden werden.**, das die Definition von Rollen, erlaubten Aktionen und Zielinformationen, bzw. die betroffenen Ressourcen umfasst.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "RBAC Rules",
  "description": "A list of RBAC rules for roles and target components.",
  "type": "array",
  "items": {
    "type": "object",
    "required": ["role", "action", "targetInformation"],
    "properties": {
      "role": {
        "type": "string"
      },
      "action": {
        "oneOf": [
          {
            "type": "string",
            "enum": ["CREATE", "READ", "UPDATE", "DELETE", "EXECUTE"]
          },
          {
            "type": "array",
            "items": {
              "type": "string",
              "enum": ["CREATE", "READ", "UPDATE", "DELETE", "EXECUTE"]
            },
            "minItems": 1
          }
        ]
      },
      "targetInformation": {
        "type": "object",
        "required": ["@type"],
        "properties": {
          "@type": {
            "type": "string",
            "enum": [
              "aas",
              "submodel",
              "concept-description",
              "aas-environment",
              "aas-registry",
              "submodel-registry",
              "aas-discovery-service"
            ]
          }
        }
      }
    }
  }
}
```

```

    ]
  },
  "aasIds": {
    "oneOf": [
      { "type": "string" },
      {
        "type": "array",
        "items": { "type": "string" }
      },
      { "type": "null" }
    ]
  },
  "submodelIds": {
    "oneOf": [
      { "type": "string" },
      {
        "type": "array",
        "items": { "type": "string" }
      }
    ]
  },
  "submodelElementIdShortPaths": {
    "oneOf": [
      { "type": "string" },
      {
        "type": "array",
        "items": { "type": "string" }
      }
    ]
  },
  "conceptDescriptionIds": {
    "oneOf": [
      { "type": "string" },
      {
        "type": "array",
        "items": { "type": "string" }
      }
    ]
  },
  "assetIds": {
    "type": "array",
    "items": {
      "type": "object",
      "required": ["name", "value"],
      "properties": {
        "name": { "type": "string" },
        "value": { "type": "string" }
      }
    }
  }
},
"additionalProperties": false
}
},
"additionalProperties": false
}
}

```

*Listing 2-1: RBAC Rules JSON Schema*

Die als JSON-Objekte definierten Regeln bestehen aus mehreren Schlüssel-Properties. Die "role" beschreibt den Namen der Rolle, während "action" die erlaubten Aktionen wie "CREATE", "READ", "UPDATE", "DELETE" und "EXECUTE" umfasst. Die "targetInformation" gibt die Ressource an, auf welche die Regel angewendet wird. Hierbei ist "@type" entscheidend, da es den Typ der Ressource festlegt, wobei die Optionen "aas", "submodel", "concept-description", "aas-environment", "aas-registry", "submodel-registry" und "aas-discovery-service" zur Verfügung stehen. Zusätzlich können für die spezifischen Ressourcen, je nach Typ ("@type"), weitere Properties definiert werden, die die Identifizierung von AAS-, Submodel- oder Concept-Description-Repository, sowie den Registries und dem Discovery Service ermöglichen. Die Typen "aas", "submodel" und "concept-description" können auch hier wieder als "aas-environment" zusammengefasst werden

Für "@type": "aas" lautet die nachfolgende Property "aasIds", welche als Wert entweder einen String oder einen Array von Strings mit den AASIdentifiern, für die diese Regel gilt, hat.

### Beispiel

```
{
  "role": "basyx-reader",
  "action": "READ",
  "targetInformation": {
    "@type": "aas",
    "aasIds": "*"
  }
}
```

*Listing 2-2: Beispiel einer AAS Repository Rolle*

Für "@type": "submodel" lauten die nachfolgenden Properties "submodelIds" und optional "submodelElementIdShortPaths", welche als Wert entweder einen String oder einen Array von Strings mit dem SubmodelIdentifiern bzw. SubmodelElementIdShortPaths für die diese Regel gilt, haben. Werden die SubmodelElementIdShortPaths nicht explizit angegeben, sind implizit alle SubmodelElemente des betroffenen Submodels für die definierten Aktionen zugänglich.

### Beispiel

```
{
  "role": "basyx-reader",
  "action": "READ",
  "targetInformation": {
    "@type": "submodel",
    "submodelIds": "*",
    "submodelElementIdShortPaths": "*"
  }
}
```

*Listing 2-3: Beispiel einer Submodel Repository Rolle*

Für "@type": "concept-description" lautet die nachfolgende Property "conceptDescriptionIds", welche als Wert entweder einen String oder einen Array von Strings mit dem ConceptDescriptionIdentifiern, für welche diese Regel gilt, hat.

### Beispiel:

```
{
  "role": "basyx-reader",
  "action": "READ",
  "targetInformation": {
    "@type": "concept-description",
    "conceptDescriptionIds": "*"
  }
}
```

*Listing 2-4: Beispiel einer Concept Description Repository Rolle*

Für "@type": "aas-environment" können dieselben Properties der Typen "aas", "submodel" und "concept-description" folgen.

### Beispiel

```
{
  "role": "admin",
  "action": ["CREATE", "READ", "UPDATE", "DELETE"],
  "targetInformation": {
    "@type": "aas-environment",
    "aasIds": "*",
    "submodelIds": "*"
  }
}
```

*Listing 2-5: Beispiel einer AAS Environment Rolle*

Für "@type": "aas-registry" lautet die nachfolgende Property, wie bei "aas", ebenfalls "aasIds" welche als Wert entweder einen String oder einen Array von Strings mit dem AASIdentifiern, für welche diese Regel gilt, hat.

## Beispiel

```
{
  "role": "basyx-reader",
  "action": "READ",
  "targetInformation": {
    "@type": "aas-registry",
    "aasIds": "*"
  }
}
```

Listing 2-6: Beispiel einer AAS Registry Rolle

Für "@type": "submodel-registry" lautet die nachfolgende Property, wie bei "submodel", ebenfalls "submodelIds", welche als Wert entweder einen String oder einen Array von Strings mit dem Submodellidentifizieren, für welche diese Regel gilt, hat.

## Beispiel

```
{
  "role": "basyx-reader",
  "action": "READ",
  "targetInformation": {
    "@type": "submodel-registry",
    "submodelIds": "*"
  }
}
```

Listing 2-7: Beispiel einer Submodel Registry Rolle

Für "@type": "aas-discovery-service" lautet die nachfolgende Property, wie bei "aas" und "aas-registry", ebenfalls "aasIds", welche als Wert entweder einen String oder einen Array von Strings mit dem AAS-identifizieren, für welche diese Regel gilt, hat, gefolgt von der Property "assetIds", durch welche die erlaubten AssetIds bestimmt werden. Hier gilt die Besonderheit, dass eine AssetId selbst aus zwei Werten bestimmt wird, daher als Objekt, bestehend aus den Properties "name" und "value" angegeben wird. wenn keine AssetIds erlaubt sein sollen, muss diese Property als leeres Array angegeben werden. andernfalls als Array von Objekten mit den genannten Properties.

## Beispiel

```
{
  "role": "basyx-aas-discoverer",
  "action": "READ",
  "targetInformation": {
    "@type": "aas-discovery-service",
    "aasIds": null,
    "assetIds": [
      {
        "name": "*",
        "value": "*"
      }
    ]
  }
}
```

Listing 2-8: Beispiel AAS Discovery Service Rolle

Für Alle Properties nach @type kann auch das Wildcard Symbol "\*" angegeben werden, um alle Ressourcen der betroffenen Komponente zu erlauben (Wie in den Beispielen bereits exzessiv genutzt). Es existiert keine Ausschluss-Syntax, um auszudrücken, dass "alle außer ..." erlaubt sind. Daher müssen alle zulässigen Ressourcen stets explizit angegeben werden, sofern nicht alle Ressourcen erlaubt sind.

Für die Anbindung an Catena-X wurde eine klare Zuordnung zwischen Rolle und BPN geschaffen, indem die BPN selbst direkt als Rolle verwendet wurde. Eine Beispielregel könnte damit folgendermaßen aussehen:

```
{
  "role": "BPNL0000000EM1AT",
  "action": "READ",
  "targetInformation": {
    "@type": "aas-environment",
    "aasIds": "https://example.com/ids/sm/1203_7012_1142_1277_OEM1",
  }
}
```

```

"submodelIds": [
  "https://admin-shell.io/idta/SubmodelTemplate/DigitalNameplate/2/0_OEM1",
  "https://admin-shell.io/idta/SubmodelTemplate/HandoverDocumentation/1/0_OEM1",
  "https://admin-shell.io/idta-02023-1-0_OEM1",
  "https://admin-shell.io/ZVEI/TechnicalData/Submodel/1/2_OEM1",
  "https://admin-shell.io/IDTA-CATENAX/BatteryData/Submodel/1/0_OEM1",
  "https://example.com/ids/sm/8471_7012_1142_8921_OEM1",
  "https://example.com/ids/sm/1203_7012_1142_2277_OEM1"]
}

```

Listing 2-9: CATENA-X Regel mit BPN als Rolle

Diese Regeln werden in einer oder mehreren JSON-Dateien in einem bestimmten Verzeichnis gepflegt. Da die BaSys Komponenten jeweils in einem eigenen Docker Container liegen, wird dieses Verzeichnis entsprechend für jeden Container explizit gemountet, also in den jeweiligen Container gespiegelt.

Um den Authentifizierungsworkflow abzuschließen, ist es notwendig, die entsprechenden Nutzer und Rollen in Keycloak anzulegen.

### 2.5.4.1.2 Konnektoren

#### 2.5.4.1.2.1 Asset

**POST** {{PROVIDER\_MANAGEMENT\_URL}}/v3/assets

```

{
  "@context": {
    "edc": "https://w3id.org/edc/v0.0.1/ns/",
    "cx-common": "https://w3id.org/catenax/ontology/common#",
    "cx-taxo": "https://w3id.org/catenax/taxonomy#",
    "dct": "http://purl.org/dc/terms/"
  },
  "@id": "{{ASSET_ID}}",
  "properties": {
    "cx-common:version": "3.0"
  },
  "privateProperties": {
  },
  "dataAddress": {
    "@type": "DataAddress",
    "type": "HttpData",
    "baseUrl": "{{PROVIDER_VWS_ENVIRONMENT}}",
    "proxyQueryParams": "true",
    "proxyPath": "true",
    "proxyMethod": "false",
    "oauth2:tokenUrl": "{{CATENAX_CENTRALIDENTITY_PROVIDER_TOKEN_URL}}",
    "oauth2:clientId": "{{PROVIDER_CLIENT_ID}}",
    "oauth2:clientSecretKey": "{{PROVIDER_CLIENT_SECRET_KEY}}"
  }
}

```

Listing 2-10: Asset

#### 2.5.4.1.2.2 Policy

**POST** {{PROVIDER\_MANAGEMENT\_URL}}/v3/policydefinitions

```

{
  "@context": {
    "odr1": "http://www.w3.org/ns/odr1/2/"
  },
  "@type": "PolicyDefinitionRequestDto",
  "@id": "{{ASSET_ID}}-policy",
  "policy": {
    "@type": "odr1:Set",
    "odr1:permission": [
      {
        "odr1:action": "USE",
        "odr1:constraint": {
          "@type": "LogicalConstraint",
          "odr1:or": [
            {
              "@type": "Constraint",
              "odr1:leftOperand": {
                "@id": "BusinessPartnerNumber"
              }
            }
          ]
        }
      }
    ]
  }
}

```



```

    },
    "@type": "CatalogRequest",
    "counterPartyAddress": "{{PROVIDER_PROTOCOL_URL}}",
    "counterPartyId": "{{PROVIDER_BPN}}",
    "protocol": "dataspace-protocol-http",
    "querySpec": {
      "offset": 0,
      "limit": 50
    }
  }
}

```

Listing 2-13: Catalog Request

## Contract Negotiation

**POST** {{CONSUMER\_MANAGEMENT\_URL}}/v3/contractnegotiations

```

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@type": "NegotiationInitiateRequestDto",
  "counterPartyAddress": "{{PROVIDER_PROTOCOL_URL}}",
  "protocol": "dataspace-protocol-http",
  "policy": {
    "@context": "http://www.w3.org/ns/odrl.jsonld",
    "@type": "odrl:Offer",
    "@id": "{{OFFER_ID}}",
    "target": "{{ASSET_ID}}",
    "assigner": "{{PROVIDER_BPN}}",
    "odrl:permission": {
      "odrl:target": "{{ASSET_ID}}",
      "odrl:action": {
        "odrl:type": "USE"
      },
    },
    "odrl:constraint": {
      "odrl:or": {
        "odrl:leftOperand": "BusinessPartnerNumber",
        "odrl:operator": {
          "@id": "odrl:eq"
        },
        "odrl:rightOperand": "{{CONSUMER_BPN}}"
      }
    },
    "odrl:prohibition": [],
    "odrl:obligation": []
  }
}

```

Listing 2-14: Contract Negotiation

## Transfer Process

**GET** {{CONSUMER\_MANAGEMENT\_URL}}/v3/contractnegotiations/{{NEGOTIATION\_ID}}

**POST** {{CONSUMER\_MANAGEMENT\_URL}}/v3/transferprocesses

```

{
  "@context": {
    "@vocab": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@type": "TransferRequest",
  "protocol": "dataspace-protocol-http",
  "counterPartyAddress": "{{PROVIDER_PROTOCOL_URL}}",
  "contractId": "{{CONTRACT_AGREEMENT_ID}}",
  "transferType": "HttpData-PULL",
  "dataDestination": {
    "type": "HttpProxy"
  },
  "connectorId": "{{CONSUMER_BPN}}",
  "privateProperties": {},
  "callbackAddresses": []
}

```

Listing 2-15: Transfer Process

## Zugriff auf das Token

Der Datentransfer, der im vorherigen Kapitel durchgeführt wurde, ermöglicht die Generierung eines Access-Tokens für einen Client aus dem Keycloak Identity-Provider, welcher eine Berechtigung auf das BaSyx-Backend besitzt.

```
{
  "@context": "https://w3id.org/dspace/2024/1/context.json",
  "@type": "dspace:TransferStartMessage",
  "dspace:providerPid": "urn:uuid:a343fcbf-99fc-4ce8-8e9b-148c97605aab",
  "dspace:consumerPid": "urn:uuid:32541fe6-c580-409e-85a8-8a9a32fbe833",
  "dspace:dataAddress": {
    "dspace:endpointType": "https://w3id.org/idsa/v4.1/HTTP",
    "dspace:endpoint": "BASYX_BACKEND_API_ENDPOINT",
    "dspace:endpointProperties": [
      {
        "dspace:name": "https://w3id.org/edc/v0.0.1/ns/authorization",
        "dspace:value": "AUTH_TOKEN"
      },
      {
        "dspace:name": "https://w3id.org/edc/v0.0.1/ns/authType",
        "dspace:value": "bearer"
      },
      {
        "dspace:name": "https://w3id.org/tractusx/auth/refreshToken",
        "dspace:value": "REFRESH_TOKEN"
      },
      {
        "dspace:name": "https://w3id.org/tractusx/auth/expiresIn",
        "dspace:value": "300"
      },
      {
        "dspace:name": "https://w3id.org/tractusx/auth/refreshEndpoint",
        "dspace:value": "REFRESH_ENDPOINT"
      }
    ]
  }
}
```

Listing 2-16: Zugriff auf das Token

## Zugriff auf das BaSyx-Backend

Der Zugriff auf das BaSyx Backend ist durch OAuth 2.0 mit Json-Web-Tokens (JWT) gesichert und erfolgt in zwei Schritten (Siehe Kapitel 2.6.4.1.2). Dabei müssen sich die Benutzer gegenüber Keycloak mit Basic Auth authentifizieren. Dadurch erhalten sie ein JWT-Token, mit dem sie sich im Anschluss gegen das Backend authentifizieren können. Diese Authentifizierung gegen Keycloak erfolgt über die Access Token URL, die bei der verwendeten Version 25.0.6 stets dem Schema `https://<KEYCLOAK-HOST>/realms/<REALM-NAME>/protocol/openid-connect/token` folgt. Für die initiale Authentifizierung gibt es zwei Möglichkeiten:

- (1) Nur mit Client Credentials im HTTP-Header oder
- (2) durch die zusätzliche Übermittlung von Benutzeranmeldedaten im Body.

Im zweiten Fall (2) enthält das erhaltene Token Informationen über die Rolle des Nutzers, welche für die RBAC Authorisierung benötigt werden. In beiden Fällen (1 und 2) werden immer die Client Credentials als Basic Auth im Header übermittelt. Sie bestehen aus Client ID und Client Secret. Diese Informationen werden in dem Format `<Client ID>:<Client Secret>` in Base64 kodiert. Dann wird das Header-Feld "Authorization" erstellt und als Wert "Basic ", (gefolgt von einem Leerzeichen) und den kodierten Anmeldedaten übergeben. Die Client Credentials können in Keycloak in der *Clients* Sektion gefunden werden.

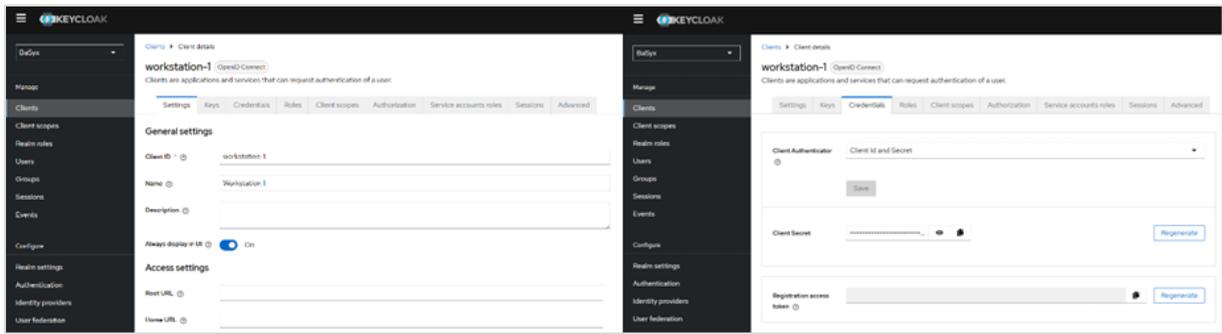


Abbildung 2-12: Keycloak Admin Console – Client ID + Client Secret

Dafür meldet man sich in der Keycloak Admin Console an, wechselt zu dem gewünschten Realm und wählt im Linken Menü *Clients* aus. Daraufhin wird eine Liste von Clients angezeigt, aus der man den betroffenen Client auswählt. Unter dem Reiter *Settings* findet man die Client ID und unter *Credentials* das Client Secret.

Um zusätzlich einen Nutzer zu Authentifizieren (2), werden die User Credentials im Body der Authentifizierungs Anfrage in dem Folgenden Format mitgegeben:

```
grant_type=password
username=<username>
password=<password>scope=openid
```

Listing 2-17: User Credentials im Body einer Authentifizierungsanfrage

Da die User Credentials im Body im Klartext übertragen werden und auch die Client Credentials im Header ohne weiteres in ein lesbares Format gebracht werden können, ist an dieser Stelle unbedingt die Verwendung von https zu empfehlen.

Nach erfolgreicher Authentifizierung stellt Keycloak ein JWT-Token aus. Dieses Token dient zur Authentifizierung des Benutzers gegenüber dem BaSyx Backend und enthält die Rollen und Berechtigungen, die für den Zugriff auf die Ressourcen erforderlich sind. Die definierten RBAC-Regeln steuern den Zugriff auf verschiedene Endpunkte der Repositories, Registries und des Discovery Services.

Der weitere Authentifizierungsfluss mit dem Token erfolgt so, dass das JWT-Token in den HTTP-Header jeder nachfolgenden Anfrage an das Backend eingefügt wird. Dies geschieht im Header-Feld "Authorization" mit dem Präfix "Bearer ". Das Token wird bei jeder Anfrage an das BaSyx Backend überprüft, um sicherzustellen, dass er gültig ist und die erforderlichen Berechtigungen für den angeforderten Zugriff vorhanden sind. Keycloak selbst wird nicht für jede Anfrage direkt angesprochen; stattdessen erfolgt die Validierung des Tokens durch die Backend-Anwendung, die sicherstellt, dass der Token nicht abgelaufen ist und die richtigen Rollen und Berechtigungen enthält.

## 2.5.5 Ergebnis

Als Ergebnis steht am Ende dieses Arbeitspakets ein voll funktionsfähiger Projektdemonstrator zur Verfügung, der nicht nur die technische Machbarkeit des PoCs beweist, sondern auch konkrete Vorschläge und technische Spezifikationen liefert. Diese können die Grundlage für zukünftige Standardisierungsprozesse innerhalb relevanter Organisationen wie Catena-X und der IDTA bilden.

## 2.6 AP 11.3 – Identity & Access Management (IAM)

### 2.6.1 Zielsetzung

In diesem Abschnitt werden die in den vorherigen Arbeitspaketen gewonnenen Erkenntnisse im Hinblick auf Security-Aspekte zusammengefasst. Zudem werden relevante Begriffe erläutert und ein Ausblick darauf gegeben, wie die erarbeiteten Lösungen in der Zukunft noch optimiert und weitergeführt werden können.

### 2.6.2 Vorgehensweise und Inhalte

Grundsätzlich lassen sich zwei Ebenen bei der Betrachtung von Security-Aspekten unterscheiden. Die erste Ebene bezieht sich auf die Authentifizierung und Autorisierung von Benutzern oder Systemen. Hier stellt sich die Frage, wer sich wie authentifizieren kann und welche zusätzlichen Informationen zur Autorisierung bereitgestellt werden müssen.

Die zweite Ebene bezieht sich auf die verwendete Infrastruktur. Dabei geht es darum, wie die Authentifizierung geregelt wird (zentral oder dezentral) und welche Kombinationen möglich sind.

Im ersten Abschnitt werden wir uns mit der Idee hinter der IDTA Security Spezifikation beschäftigen, die sich hauptsächlich auf RBAC und ABAC stützt. Dabei spielt auch die Verwendung von Identity Providern eine wichtige Rolle und wie diese in verschiedenen Szenarien eingesetzt werden können. Es werden verschiedene Szenarien durchgespielt, um zu verdeutlichen, wie Identity Provider in der Praxis eingesetzt werden können. Dabei wird auch auf die Vor- und Nachteile verschiedener Implementierungen eingegangen.

Nach Aufbau der Verbindung schlägt die IDTA-Sicherheitsspezifikation die Verwendung eines JSON-Web-Tokens (JWT-Tokens) zur Absicherung des Kommunikationskanals vor. JWT-Tokens sind in der aktuellen Kommunikation mit REST nicht mehr wegzudenken und hinlänglich bekannt. In einem der nachfolgenden Kapitel wird deshalb nur kurz auf JWT-Tokens und weitere Möglichkeiten eingegangen.

Um einen sicheren Kommunikationskanal einzurichten, ist es in der Regel erforderlich, sicherzustellen, dass der Kommunikationspartner ein berechtigtes Interesse an den zu übermittelnden Daten hat. In der Umgebung von Catena-X wird dazu eine Business Partner Number (BPN) verwendet, die entsprechend abzusichern ist. Die BPN kann durch eine Signatur des zentralen Anbieters geschützt oder alternativ eine Dezentrale ID (DID) verwendet werden.

### 2.6.3 Authentifizierung

#### 2.6.3.1 Der allgemeine Ablauf

Die Authentifizierung bei einem Data Provider erfolgt derzeit über die BPN, siehe Abbildung 2-13. Mit der BPN kann man bei einem Data Provider einen Katalog anfordern, welcher Informationen über den möglichen Zugriff enthält. Ohne die BPN stellt der Data Provider entweder keine oder nur allgemeine Daten zur Verfügung.

Eingeschränkt wird diese Liste vom Data Provider, durch die Angabe von Regeln (Policies). D.h. der Data Provider definiert Regeln, welche sich für eine Business Partner auf die Auswahl der Einträge im Katalog (catalog) auswirkt.

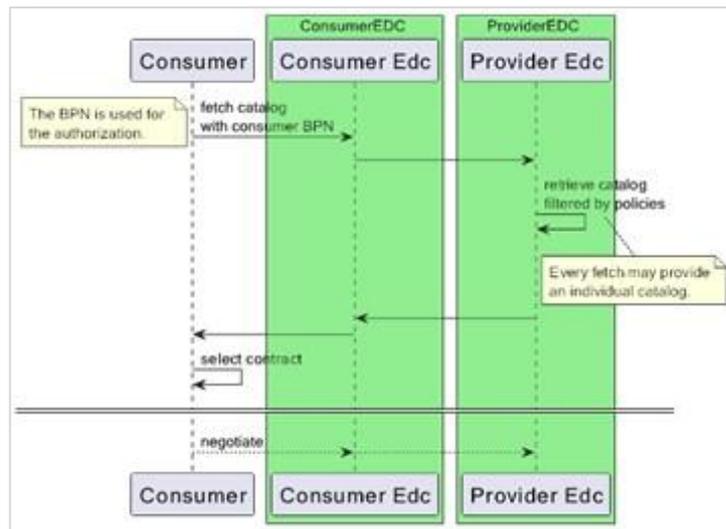


Abbildung 2-13: Authentifizierungs-Workflow

Aus dem Katalog (catalog) wird ein passendes Angebot (offer) ausgewählt. Der Data Provider bestätigt das Angebot anschließend in Form einer Vereinbarung (agreement). Damit kommt ein Vertrag (contract) zwischen dem Data Consumer und Data Provider zustande. Auf Basis der ausgetauschten Daten wird dann der Zugriff auf bestimmte Bereiche des Data Providers freigegeben.

#### 2.6.3.1.1 Policies

Der Katalog mit den Angeboten wird für einen Interessenten dynamisch erstellt. Um die Auswahl der relevanten Angebote für den Interessenten im Katalog einzuschränken, sind Auswahlregeln notwendig. Hierfür werden in diesem Fall Policies verwendet. Policies sind "Wenn-Dann"-Regeln – wenn die Bedingung erfüllt ist, wird das Angebot in den Katalog aufgenommen.

#### 2.6.3.1.2 IDTA Security Spezifikation

Diese Spezifikation zielt darauf ab Einschränkungen der zu übermittelten Daten (und nicht des Angebots) vorzunehmen. Grundsätzlich spricht nichts dagegen die Regeln der IDTA Security Spezifikation auch auf die Angebote des Katalogs anzuwenden. Im Moment findet die Auswahl des Angebots auf Basis der Policies statt. Um die IDTA Security Spezifikation auch auf die Filterung des Angebotskatalogs anzuwenden, müssen die bisherigen Implementierungen auf Basis der Policies angepasst oder ersetzt werden.

#### 2.6.3.2 Konsequenzen aus dem allgemeinen Ablauf

Wie aus den Ausführungen hervorgeht, gibt es auf Data Provider Seite keine speziellen Zugriffe für ausgewählte Personen, Personengruppen oder Subjekte (z.B. eine Maschine). Mit der BPN identifiziert man sich als interessierter Business Partner.

Die tatsächliche Auswahl geht vom Data Consumer aus, indem er aus dem Katalog ein Angebot auswählt. Beispielsweise gibt es auf Seiten des Consumers zwei verschiedene Gruppen. Beide benötigen verschiedene Informationen über die Produkte des Data Providers. Der Data Provider stellt also zwei unterschiedliche Angebote (im Katalog) für die beiden Gruppen zur Verfügung. Der Data Consumer wählt dann anhand der Zugehörigkeit der Person zu einer Gruppe das passende Angebot aus.

Exkurs: Durch die Einführung einer individuellen Anmeldung eines Users kann der Data Provider die Kontrolle über den Zugriff auf seine Daten verbessern. Erreichen könnte man dies, indem man einen speziellen Eintrag im Katalog anbietet, der eine individuelle Anmeldung eines Users erforderlich macht. Hierfür ist eine Erweiterung (Extension) des EDC auf Seiten des Data Providers notwendig. Zusätzlich muss der Data Provider eine Erfassung der angemeldeten User ermöglichen, indem er eine Identity Provider Funktionalität bereitstellt. Durch die individuelle Anmeldung eines Users kann der Data Provider sicherstellen, dass nur autorisierte Benutzer Zugriff auf seine Daten erhalten. Hierbei kann er auch die Rollen und Berechtigungen der Benutzer berücksichtigen und so ein höheres Maß an Sicherheit gewährleisten.

### 2.6.3.3 BPN Security

Aktuell sind zwei Verfahren spezifiziert, welche die Absicherung der BPN gewährleisten.

Bei dem vermutlich bekannteren Verfahren registriert man eine BPN bei einer zentralen Stelle. Nach der Registrierung kann eine signierte BPN angefordert werden, die dann an den Data Provider zur Authentifizierung des Business Partners weitergereicht werden kann. Dieses Verfahren ähnelt stark dem Austausch von Zertifikaten.

Eine weitere Möglichkeit ist die Verwendung einer DID. Dieses Verfahren ist ähnlich zum vorherigen, mit dem Unterschied, dass eine DID vom Besitzer angepasst werden kann. Der Aussteller bestätigt die DID und übermittelt sie an das Subject (z.B. die Person), welches sie beantragt hat. Die DID ist dann im Besitz des Antragsstellers. Anschließend kann der Inhalt an die jeweilige Aufgabe angepasst und in Form eines Verifiable Credential (VC) an den Data Provider übergeben werden. Für die Bestätigung der Gültigkeit der ursprünglich übermittelten Daten ist dann erneut die Bestätigung des Ausstellers (insbesondere der DID) erforderlich.

Exkurs: Da der Besitzer einer DID diese anpassen kann, könnte man sich beispielsweise vorstellen, dass der Consumer seine DID bzw. die VC so erweitert, dass der Data Provider zusätzliche Informationen erhält (z.B. eine Rolle), die es ermöglichen, den Katalog gezielter anzupassen. Im Gegensatz zur ursprünglichen Variante kann der Data Provider nun genauer entscheiden, welche Angebote er welchem Anfragenden anbietet.

### 2.6.4 Autorisierung

Die Autorisierung erfolgt in zwei verschiedenen Abschnitten. Zunächst findet beim Data Provider eine Filterung der möglichen Angebote anhand der BPN statt. Hat der Data Consumer dann einen Vertrag mit dem Data Provider geschlossen, bekommt der Data Consumer ein Access-Token, welches weitere Informationen (Claims) enthält. Diese Informationen machen bei Zugriffen auf die Daten des Data Provider anschließend eine weitere Filterung der Daten möglich.

Die Prüfungen, welche bei Zugriff erfolgen, sind in der IDTA Security Spezifikation beschrieben. Grundlage zur Autorisierung ist ein JWT-Token.

#### 2.6.4.1 Token

Tokens sind die etablierte Methode Security-Informationen über eine REST-Schnittstelle auszutauschen. Grundsätzlich sind zwei verschiedenen Arten von Tokens üblich. Das Opaque-Token wird in dieser Konstellation nicht verwendet und wird hier nur der Vollständigkeit halber ebenfalls beschrieben.

##### 2.6.4.1.1 Opaque-Token

Dieses Token ist ein einfacher String. Der Identity Provider des Data Provider muss immer bei dem ausstellenden Identity Provider die Gültigkeit des Tokens bestätigen lassen. Es enthält keine zusätzlichen Informationen über den Aufrufer. Der Vorteil ist, dass das Token jederzeit vom ausstellenden Identity Provider widerrufen werden kann.

##### 2.6.4.1.2 JWT (Json-Web-Token)

JWT-Tokens sollten aus der allgemeinen Literatur bereits bekannt sein. Ein kurzer Überblick über das Verfahren:

Ein JWT-Token besteht aus drei Teilen, die durch Punkte getrennt sind, und wird in der Regel über HTTPS übertragen. Der erste Teil enthält die sogenannten Header, also Metadaten über den Token. Der zweite Teil enthält die Nutzlast, also die eigentlichen Daten. Der dritte Teil ist die Signatur, die sich aus dem Hash des ersten und zweiten Teils sowie einem geheimen Schlüssel zusammensetzt. JWT-Tokens bieten eine sichere Möglichkeit, Ansprüche zwischen zwei Parteien auszutauschen, ohne dass eine zentrale Authentifizierungsinstanz erforderlich ist. Zur Validierung der Signatur wird der öffentliche Schlüssel des Ausstellers (Identity Provider) benötigt.

#### 2.6.4.1.3 Access-Token

Das Access-Token ist der generelle Schlüssel für die die Autorisierung innerhalb einer Anwendung. Die Nutzlast kann Daten (Claims) wie beispielsweise Rollen enthalten, die bei der Überprüfung von Zugriffsberechtigungen verwendet werden. Das Token kann nicht widerrufen werden, sondern enthält lediglich ein Ablaufdatum.

Da das Access-Token sehr viele Fragen bzgl. des Inhaltes offen lässt, wurde zusätzlich das ID-Token (Connect ID) spezifiziert welches die Inhalte genauer festlegt. Für die Sicherstellung, dass der Zugriff erlaubt ist, ist aber allein das Access-Token relevant.

#### 2.6.4.2 Claims, Scopes, Rollen

Ein JWT-Token kann neben der grundlegenden Authentifizierung auch zusätzliche Informationen in Form von Claims enthalten. Die folgenden Informationen dienen einer Erläuterung der Begriffe Claims, Scopes und Rollen.

##### 2.6.4.2.1 Claims

In einem JWT ist ein ‚Claim-Assertion‘ eine Information in Form eines Key-Value Paares. Der Wert kann ein einzelner Wert oder auch ein Array von Werten sein. Durch einen Claim-Assertion bestätigt der ausstellende Identity Provider die Gültigkeit eines bestimmten Attributes mit einem spezifischen Wert. Beispiele für Claims sind z.B. ‚sub‘ (Subject) für den Benutzernamen, ‚scope‘ für Berechtigungen oder ‚role‘ für Rollen.

##### 2.6.4.2.2 Scope

Ein Scope ist normalerweise ein festgelegter Begriff, der sowohl in Anfragen als auch in Antworten verwendet wird. Ein Scope definiert einen bestimmten Umfang an Claims, also die Informationen, die in einem Token enthalten sein sollen. Darüber hinaus kann ein Scope festlegen, welche Aktionen oder Ressourcen einem Benutzer erlaubt sind.

Wenn ein Scope in einer Anfrage angegeben ist, muss das zurückgegebene Token die entsprechenden Claims enthalten. Identity Provider können Scopes auch als optional definieren. Optionale Scopes werden vom Identity Provider standardmäßig bereitgestellt, falls sie nicht explizit in der Anfrage angegeben wurden.

Bei fehlenden, nicht als optional markierten Scopes, kann dies zu unterschiedlichen Folgen führen. Manche Identity Provider zeigen eine Fehlermeldung an, andere erzeugen ein Token ohne die notwendigen Berechtigungen.

##### 2.6.4.2.3 Rollen

Während standardisierte Claims wie ‚sub‘ oder ‚scope‘ feste Bedeutungen haben, sind Rollen nicht standardisiert. Sie können aber zusätzlich im JWT-Token angegeben werden, um dem Benutzer spezifische Rechte oder Verantwortlichkeiten zuzuordnen.

#### 2.6.5 Regeln – IDTA Security Spezifikation

Die Regeln zur Authentifizierung und Autorisierung sind in der IDTA Security Spezifikation beschrieben und haben folgendes Prinzip: Zum Vergleich werden die Attribute der Modelle und Submodelle herangezogen. Als Ziel der Regeln können einzelne Attribute, ganze Modelle oder Submodelle dienen.

Zum Vergleich stehen dann verschiedenen Optionen offen.

- Vergleiche mit anderen Attributen
- Festgelegte Werte (z.B. ein Datum ‚freigegeben ab‘)
- Claims aus den Access-Tokens (z.B. eine bestimmte Rolle)

#### 2.6.6 Identity Provider

Ein Identity Provider stellt das Access-Token zum Zugriff auf eine Ressource (Modelle und Submodelle) aus. Diese können in unterschiedlichen Varianten zentraler und dezentraler Identity Providers ausfallen.

Die grundlegende Herausforderung besteht darin, dass der Data Provider dem Identity Provider vertrauen muss. Vertrauen bedeutet in diesem Kontext, dass der Identity Provider die Access-Tokens signiert und der Data Provider anhand dieser Signatur die Gültigkeit des Tokens überprüfen kann. Wie bereits erwähnt, benötigt man einen zentralen Identity Provider, um eine überprüfbare BPN (Business Partner Number) zu gewährleisten. Zusätzlich werden oft weitere Claims benötigt, um die Sicht auf die BPN einzuschränken. Um solche Claims zu definieren, benötigt der Data Provider Zugriff auf den zentralen Identity Provider.

Ein Identity Provider stellt Access-Tokens zur Verfügung, um den Zugriff auf Ressourcen wie Modelle und Submodelle zu ermöglichen. Dabei gibt es verschiedene Varianten von Identity Providers – zentrale und dezentrale Identity Providers. Im Folgenden werden die unterschiedlichen Möglichkeiten betrachtet

### 2.6.6.1 Keine Authentifizierung/Autorisierung

Liegt keine Authentifizierung in Form einer BPN vor, können trotzdem die vom Data Provider als öffentlich gekennzeichneten Daten abgerufen werden, siehe Abbildung 2-14.

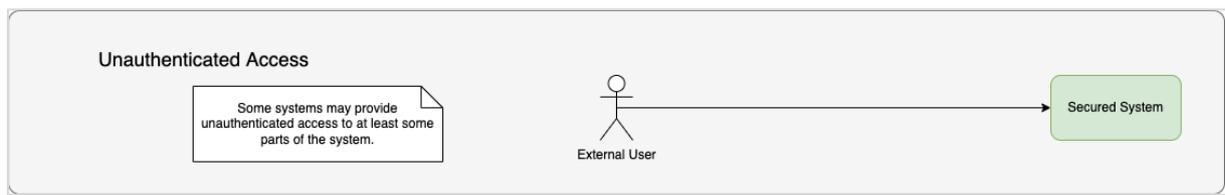


Abbildung 2-14: Unauthenticated Access

### 2.6.6.2 Interne Authentifizierung/Autorisierung

In diesem Fall ist der Identity Provider auf die Ressourcen des Data Providers abgestimmt. Allerdings muss der Data Provider in diesem Fall auch extern gepflegte Informationen, wie zum Beispiel die BPN, zusätzlich in seinem System hinterlegen. Dies kann zu Problemen führen, wenn diese externen Daten synchron gehalten werden müssen, siehe Abbildung 2-15.

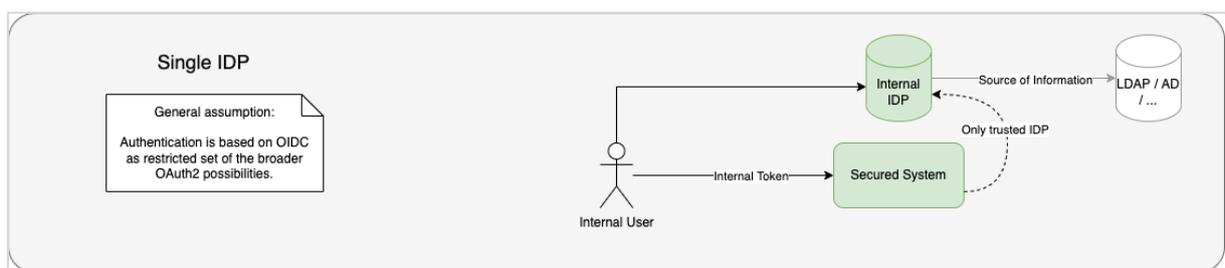


Abbildung 2-15: Internal Authentication

### 2.6.6.3 External und internal Identity Providers

In diesem Aufbau ist der externe Identity Provider für die Authentifizierung verantwortlich, siehe Abbildung 2-16. Dabei werden die folgenden Schritte durchlaufen:

1. Der externe Benutzer besorgt sich ein Token beim externen Identity Provider.
2. Mit dem Token vom externen Identity Provider weißt sich der externe Benutzer beim internen Identity Provider des Data Providers aus.
3. Der interne Identity Provider des Data Providers überprüft das Token vom externen Identity Provider und stellt das interne Access-Token aus.
4. Mit dem internen Access-Token weißt sich der externe Benutzer dann gegenüber dem Data Provider aus.

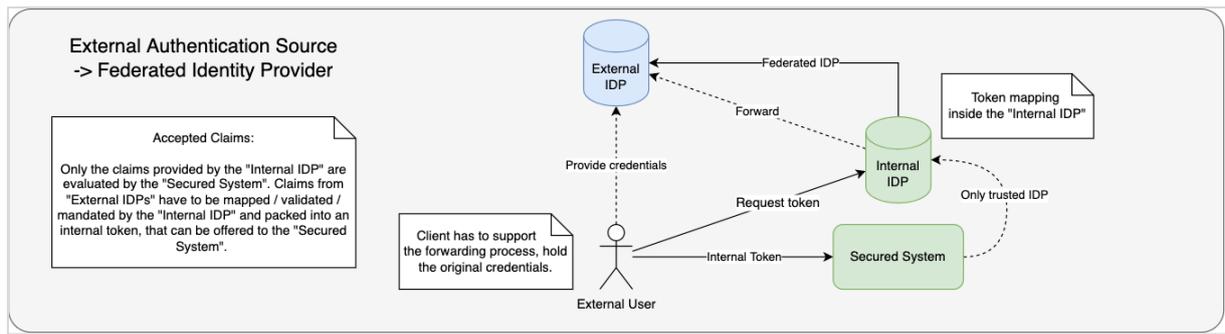


Abbildung 2-16: External Authentication

Ein Nachteil ist, dass externe Benutzer für jeden Data Provider einen separaten Token benötigen und sich bei jedem Data Provider erneut anmelden müssen. Durch die Möglichkeit das vom externen Identity Provider erhaltenen Token weiter zu verwenden, entfällt allerdings die erneute Authentifizierung.

Exkurs: Verschiedene Identity Providers (z. B. Keycloak) können so konfiguriert werden, dass die Authentifizierung über einen externen Identity Provider erfolgt. Das Access-Token wird weiterhin vom internen Identity Provider ausgestellt. Dadurch verändert sich das Szenario, da die zusätzliche Kommunikation mit dem externen- vom internen Identity Provider übernommen wird. Allerdings bedeutet dies, dass das Token vom externen Identity Provider nicht zur Verfügung steht. Die Authentifizierung muss dann bei jedem Data Provider neu vorgenommen werden.

#### 2.6.6.4 Gateway

Um der Flut von möglichen Access-Tokens entgegenzuwirken, kann die Verwendung eines Gateways von Nutzen sein. Der externe Anwender benötigt weiterhin das Token des externen Identity Providers. Damit meldet er sich beim Gateway des Data Provider an. Das Gateway kontaktiert den internen Identity Provider und übergibt ihm das externe Token. Konnte das externe Token validiert werden, generiert der interne Identity Provider ein internes Token, das für den Zugriff auf die Ressourcen des Data Provider verwendet wird, siehe Abbildung 2-17.

Bei jedem weiteren Zugriff wird geprüft, ob zum externen Token bereits ein interner Token existiert. In diesem Fall ist die erneute Generierung eines Tokens über den internen Identity Provider nicht mehr notwendig.

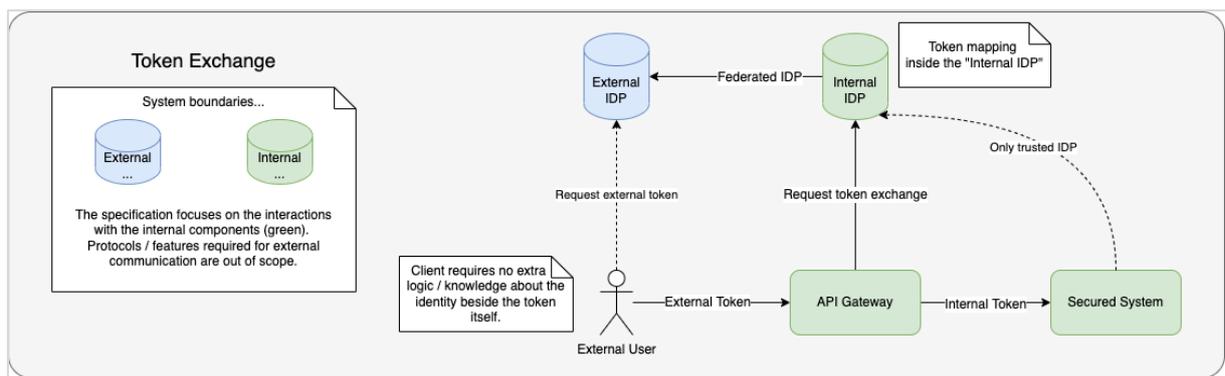


Abbildung 2-17: Token Exchange via Gateway

Der Vorteil dieses Ansatzes ist es, dass der externe Anwender nur noch den Token kennen muss, welcher er bei der Authentifizierung beim externen Identity Provider bekommen hat.

#### 2.6.7 Ergebnis

Die beschriebenen Methoden zum Identity- und Accessmanagement gewährleisten einen sicheren und kontrollierten Datenaustausch und schaffen Grundlagen für zukünftige Weiterentwicklungen und Standardisierungen. Es wird eine solide Basis gebildet für die Integration weiterer Sicherheitsmechanismen und die langfristige Vision einer einheitlichen und sicheren Datenarchitektur unterstützt.

### 3 TP12 – VWS-Produktkatalog

Im Teilprojekt 12 „Produktkatalog“ wurden auf Basis der in TP1 „Konzept, Informationsmodelle und Produktbeschreibung“, erarbeiteten Konzepte rund um das Informationsmodell und die Produktbeschreibung der Leitungssatzherstellung für die Automobilindustrie mittels Verwaltungsschalen (VWS), klare anwendungsorientierte Regeln für das Erzeugen und Verwalten von Typ-Verwaltungsschalen erarbeitet.

Zudem wurde ein Demonstrator erstellt, der einen beispielhaften herstellerübergreifenden Marktplatz basierend auf diesen Typ-Verwaltungsschalen abbildet. In diesem Kontext sollte auch untersucht werden, inwieweit die VWS zur Abwicklung der notwendigen e-Commerce-Prozesse anwendbar ist.

Die VWS nach IEC 63278-1 bietet als Schlüsselkonzept der Industrie 4.0 eine standardisierte Struktur für die Beschreibung von Assets (z.B. Maschinen, Komponenten oder Software) in einer digitalen Umgebung. Damit wird ein effizienter Datenaustausch zwischen verschiedenen Systemen und Anbietern ermöglicht.

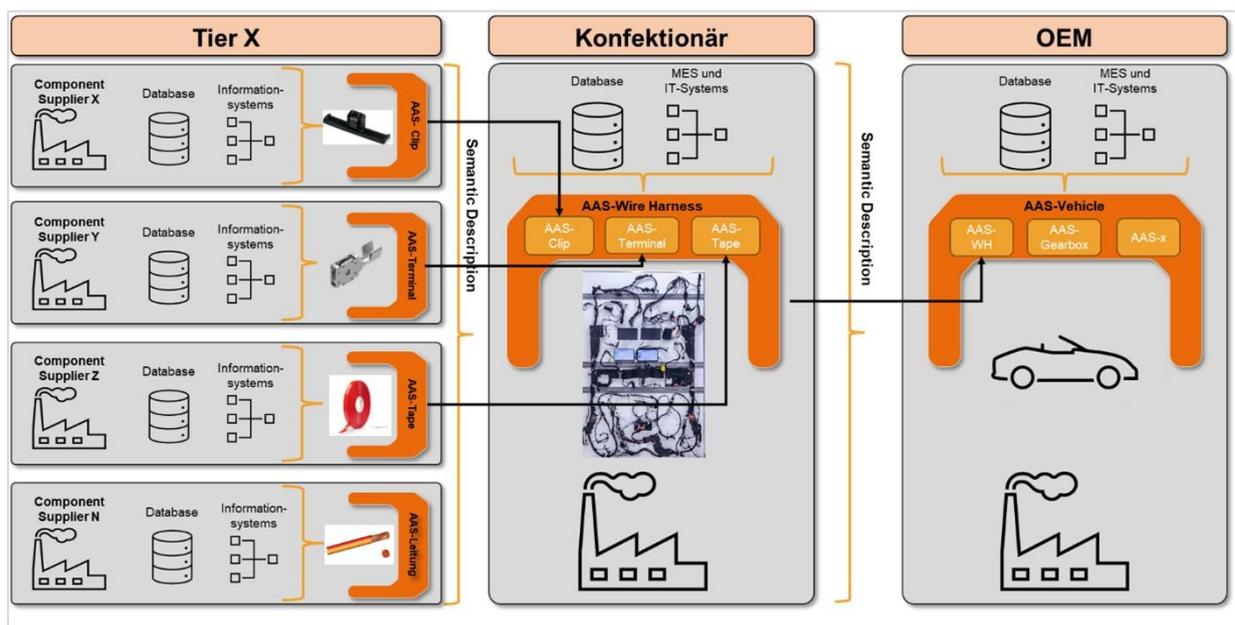


Abbildung 3-1: VWS in der Leitungssatz-Wertkette

In TP1 wurde daher basierend auf der VWS ein detailliertes und umfassendes Informationsmodell des Leitungssatzes spezifiziert, welches auch die Produktmodellierung des Leitungssatzes umfasst. Dies beinhaltet die detaillierten Anforderungen, die der Leitungssatz erfüllen muss.

Unverzichtbarer Startpunkt für das Arbeiten mit Verwaltungsschalen ist jedoch die Bereitstellung von Typ-Verwaltungsschalen mit digitalen Modellbeschreibungen für die im Leitungssatzkontext verwendeten Komponenten (siehe Spalte „Tier X“ in Abbildung 3-1). Im bisherigen Projektverlauf wurde dieses Thema nicht detailliert behandelt. Um die Einführungshürden für die betroffenen Unternehmen zu senken, widmete sich TP12 diesem Thema.

Das Teilprojekt wurde in Arbeitspakete aufgeteilt, die in den folgenden Kapiteln näher erläutert werden.

#### 3.1 AP 12.1 – Anforderungsanalyse

Im AP „Anforderungsanalyse“ wurden Grundlagen definiert sowie Zielsetzung und Use Cases für das Teilprojekt geklärt. Zusammenfassend geht es darum, eine minimal notwendige Produktdatenqualität für die Datenmodellierung und ein Konzept zur Generierung eines Produktkatalog aus Produktdatenbanken zu definieren und im Demonstrator der ARENA2036 sowie möglichst bei den Projektpartnern umzusetzen.

### 3.1.1 Ist-Situation

Produktinformationen werden oft ähnlich - aber uneinheitlich - in Online-Produktkatalogen dargestellt, die wesentlichen technischen Daten häufig als HTML-Tabellen bereitgestellt und zusätzliche Dokumente zum Download angeboten.

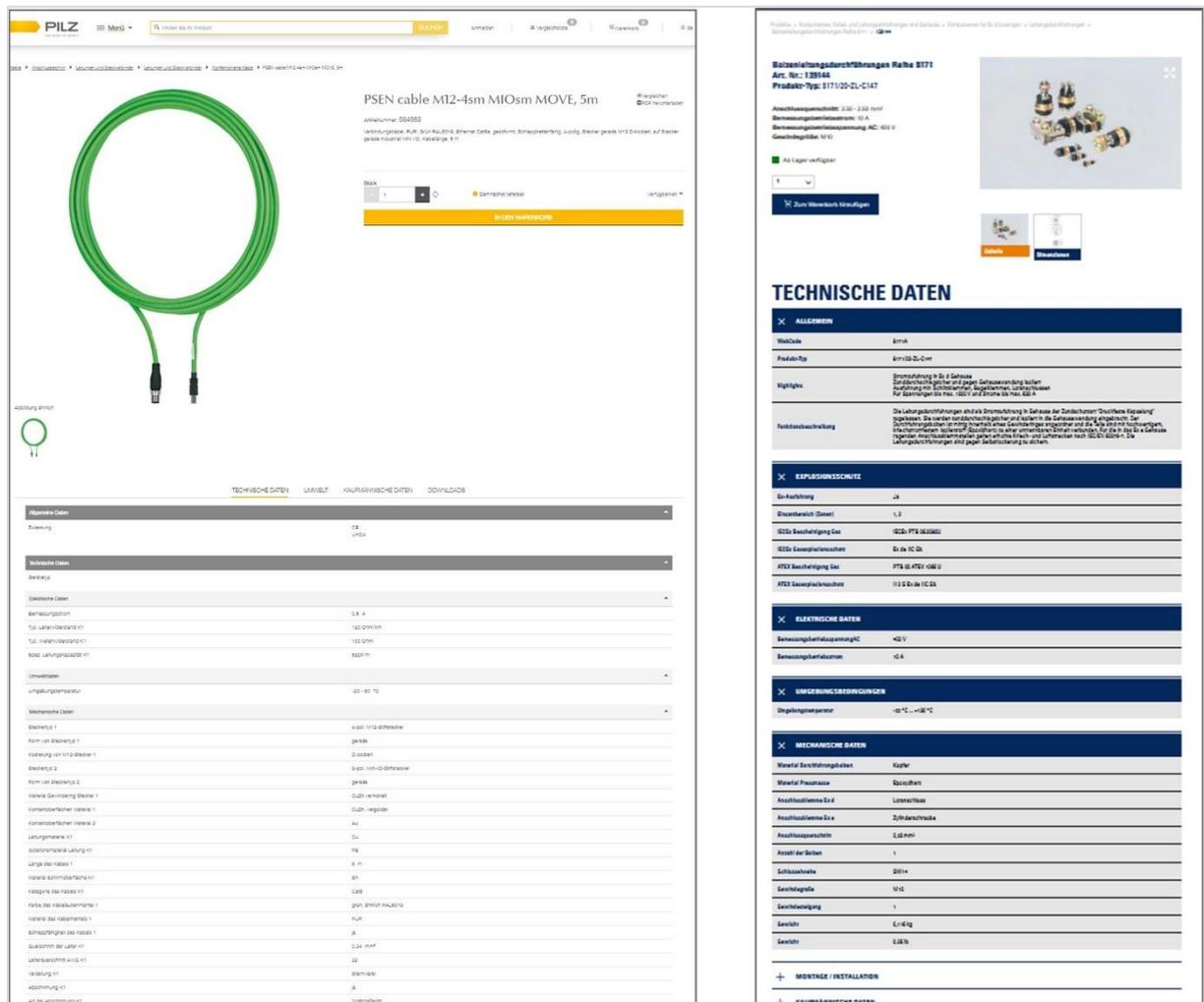


Abbildung 3-2: Typische Online-Produktkataloge

Derartige Bereitstellungen als HTML-Webseiten sowie PDFs sind jedoch nicht standardisiert maschinenlesbar oder -interpretierbar.

Eine semantische Identifizierbarkeit findet in der Regel nur auf Ebene der Produktkategorie statt, nicht auf Merkmalsebene, was unzureichend ist für die angestrebten typischen Anwendungsfälle.

### 3.1.2 Anwendungsfälle

Mittels eines VWS-basierten Produktkatalogs und semantischer Klassifizierung bis auf Merkmalsebene sollen die folgenden typischen Anwendungsfälle unterstützt werden (vgl. Abbildung 3-3):

- Herstellerübergreifende maschinelle Produktauswahl (Vergleichsportale, Marktplätze)
- Virtuelles, kollaboratives Engineering (z.B. ECAD, MCAD, VIBN) <sup>7</sup>
- Bereitstellung der Datengrundlage des „Product Carbon Footprint“ (PCF)

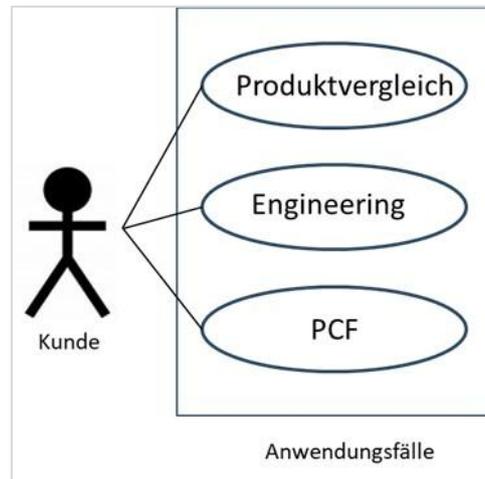


Abbildung 3-3: Anwendungsfälle des VWS-basierten Produktkatalogs

Das TP12 widmet sich im Besonderen dem Anwendungsfall „Herstellerübergreifende maschinelle Produktauswahl (Vergleichsportale, Marktplätze)“. Angerissen wird dabei auch das Thema e-Commerce unter Anwendung der VWS.

### 3.1.3 Typen von Digitalen Zwillingen

Digitale Zwillinge können in Untertypen unterteilt werden, zu denen der **Digitale Zwillingstyp** (DTP), die **Digitale Zwillinginstanz** (DTI) und das **Digitale Zwillingssaggregat** (DTA) gehören.

Der DTP besteht aus den Entwürfen, Analysen und Prozessen, d.h. Modellen, die ein physisches Produkt abbilden. Der DTP existiert, bevor es ein physisches Produkt gibt. Der DTI ist der digitale Zwilling jeder einzelnen Instanz des Produkts, sobald es hergestellt ist. Der DTI ist mit seinem physischen Gegenstück für die restliche Lebensdauer des physischen Gegenstücks verbunden. Der DTA ist die Zusammenfassung von DTIs, deren Daten und Informationen für Abfragen über das physische Produkt, Prognosen und Lernen verwendet werden können. Die spezifischen Informationen, die in den digitalen Zwillingen enthalten sind, werden durch Anwendungsfälle bestimmt. Der digitale Zwilling ist ein logisches Konstrukt, was bedeutet, dass die tatsächlichen Daten und Informationen in anderen Anwendungen enthalten sein können [2].

Das Konzept des DTP wird im Folgenden durch die Typ-VWS umgesetzt. Der DTI beschreibt das Konzept der Instanz-VWS. Der DTA ist nicht im Scope dieses Teilprojekts.

Folgende Klassen von Typ-VWS sind bei der Entstehung von Leitungssätzen relevant:

- Einzel-Komponenten-VWS: Diese modellieren die Einzelkomponenten als Typbeschreibungen, aus denen ein Leitungssatz besteht. Das können auch vorgefertigte zusammengesetzte Komponenten sein.

<sup>7</sup> <https://industrialdigitaltwin.org/use-cases/collaborative-engineering-die-technologische-grundlage-zur-digitalisierten-wertschoepfungskette>

- Halbfabrikat-VWS: Engineering- und Produktionsmodell für Halb-/Zwischenprodukten, bspw. eine geschnittene & abisolierte Leitung. Enthält das Teilmodell Bill of Material und Bill of Process Parameter mit Produktionsanweisungen.
- OEM-Leitungssatz-VWS: Modell des 150%-Leitungssatz. Enthält Produktspezifikation, Konfiguration, BOM mit Verweisen auf die Typ-VWSen (OEM-Artikelnummer) der beteiligten Komponenten.
- Konfektionär-Leitungssatz-VWS: Modell des 150%-Leitungssatz. Enthält Produktspezifikation, Konfiguration, BOM mit Verweisen auf die Typ-VWSen (Manufacturer-Artikelnummer) der beteiligten Komponenten und die MBOM.
- Produktions-VWS: Modell des Produktionsauftrags für den 100%-LS. Enthält Verweise auf eine definierte LS-Variante aus der LS-Typ-VWS (LS-BOM) sowie Verweise auf die Typ-VWSen der enthaltenen Komponenten und das Teilmodell BOPP mit Produktionsanweisungen.

Für diese Typ-VWS sind folgende Lebenszyklusphasen bzw. Anwendungsfälle relevant:

- Produktkatalog: Hier werden die einzelnen Komponenten und -VWSen, die zur Bildung des Endprodukts verwendet werden sollen, als digitale Produktmodelle bereitgestellt und können bspw. für Angebotsanfrage / Abgabe genutzt und auch im Weiteren als Vertragsgrundlage dafür dienen, was der Lieferant eigentlich liefern muss. Diese VWS müssen inhaltlich ausreichend befüllt und über Produktsuchen auffindbar sein. Eine Produktkatalog-VWS kann entweder Hardware, Software oder ein Serviceangebot abbilden. Als Filterregel für die Erkennung einer Produktkatalog-VWS soll die Attributbelegung `assetKind = Type` und das Vorhandensein des `SM TechnicalData` herangezogen werden. Andere VWSen sollten in einem Produktkatalog-Viewer nicht direkt angezeigt werden.
- Engineering: Hier werden mehrere Komponenten- und Halbfabrikat-VWSen virtuell zu einer Leitungssatz-VWS aggregiert. Ausserdem werden auch Halbfabrikat-VWSen erzeugt.
- Qualitätssicherung: Hier kann das Leitungssatz-Modell virtuell auf elektrische, mechanische und andere Eigenschaften geprüft werden, bevor er gefertigt wird.
- Produktion: In der Produktionsphase der Halbprodukte und des Leitungssatzes wird mit produktbezogenen Informationen aus den Typ-VWSen gearbeitet.

Im Folgenden soll die Zusammenstellung von Einzelkomponenten-VWSen zu Hersteller-Produktkatalogen betrachtet werden.

### 3.2 AP 12.2 - Datenmodellierung

Im AP „Entwicklung von Informationsmodellen und Zuweisung der Daten“ wurde basierend auf den folgenden vorhandenen Standards die Datenmodellierung von **Typ-Verwaltungsschalen** für den VWS4LS-Kontext untersucht

Das Ergebnis der Analyse zielte darauf ab, Best-Practice-Ansätze für die Nutzung der Standard-VWS-Metadatenmodelle zu identifizieren und in einem Leitfaden wie folgt zu beschreiben.

#### 3.2.1 Submodelle der Typ-VWS

Die im Folgenden detaillierter erläuterten Submodelle *IDTA 02003-1-2 Technical Data* [3], *IDTA 02004-1-2 Handover Documentation* [4] und *IDTA 02006-2-0 Digital Nameplate for Industrial Equipment* [5] werden als Minimalausstattung für die Typ-VWS von LS-Komponenten empfohlen.

Dies kann ergänzt werden durch weitere für den Produkt-Anwendungsfall sinnvolle Submodelle, bspw. *IDTA 02005-1-0 Provision of Simulation Models* [6], *IDTA 02026-1-0 Provision of 3D Models* [7], *IDTA 02013-1-0 Reliability* [8], *IDTA 02023-1-0 Carbon Footprint* [9], *IDTA 02002-1-0 Submodel for Contact Information* [10], etc.<sup>8</sup>

##### 3.2.1.1 SM Technical Data

*IDTA 02003-1-2 Technical Data* [3] dient dazu, technische Datenblätter in maschinenlesbarer Form und unter Angabe definierte Semantiken zur Verfügung zu stellen, siehe Abbildung 3-4. Der beabsichtigte Anwendungsfall ist, dass ein Hersteller sein Produkt, welches er auf den Markt bringt, mittels technischer Daten (Eigenschaften) beschreibt, die idealerweise mit Hilfe von Wörterbüchern wie ECLASS und IEC Common Data Dictionary (IEC CDD) interoperabel sind und von anderen Marktteilnehmern eindeutig verstanden werden.

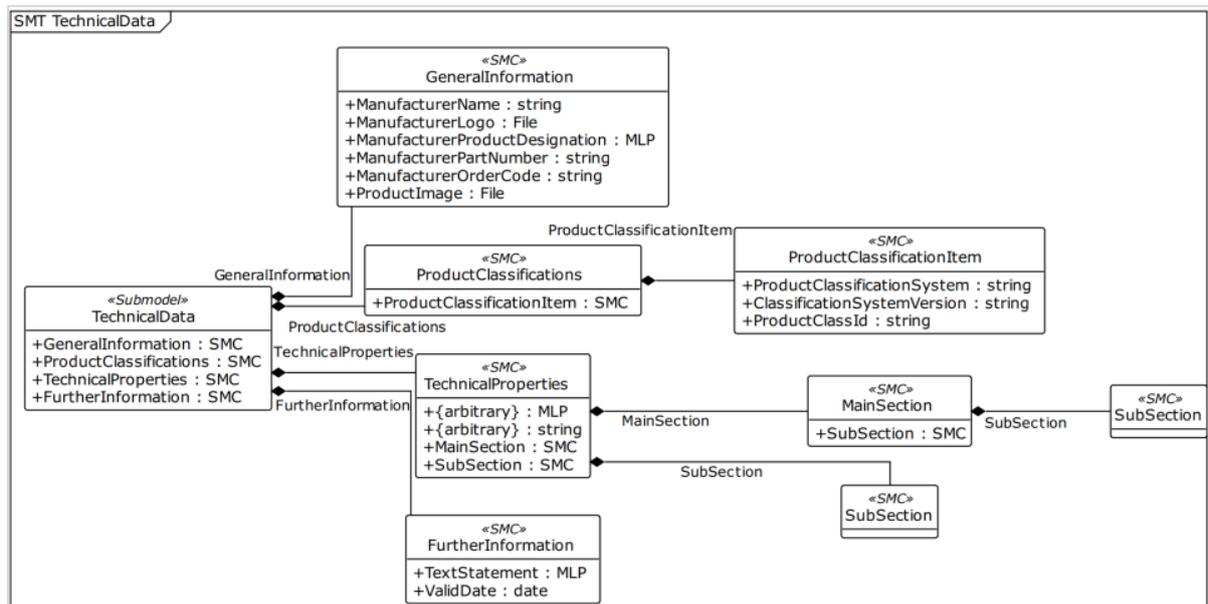


Abbildung 3-4: IDTA 02003-1-2 TechnicalData

Es wird für Komponenten-VWS im Leitungssatzbereich empfohlen, die Inhalte der *SubmodelElement-Collection (SMC) TechnicalProperties* möglichst weitgehend anhand der VEC-Ontologie zu definieren und mit den entsprechenden semantischen Referenzen zu versehen (siehe Kapitel 3.2.7).

<sup>8</sup> <https://industrialdigitaltwin.org/en/content-hub/submodels>

### 3.2.1.1.1 Produktklassifikation

Komponenten von Leitungssätzen können Steckverbinder, Leitungen, Verbindungen, Isolierungen und vieles mehr sein. Diese Produktklassifizierung muss eindeutig semantisch identifizierbar sein. ECLASS [11] als weit verbreitetes Klassifikationssystem für Produkte und Dienstleistungen bietet einige Klassifizierungs-Kategorien für den VWS4LS-Kontext:

- [44-04-01-01 Leitung \(KFZ, low voltage\)](#)
- [44-04-01-02 Leitungsstrang \(KFZ, low voltage\)](#)
- [44-04-01-03 Leitungsstrang, Ladekabel \(KFZ, low voltage\)](#)
- [44-04-01-05 Masseband, Batterieleitung \(KFZ, low voltage\)](#)
- [44-04-01-91 Bordnetz \(KFZ, low voltage, Teile\)](#)
- [44-04-02-01 Leitung \(KFZ, high voltage\)](#)
- [44-04-02-02 Leitungsstrang \(KFZ, high voltage\)](#)
- [44-04-02-03 Leitungsstrang, Ladekabel \(KFZ, high voltage\)](#)
- [44-04-02-05 Masseband, Batterieleitung \(KFZ, high voltage\)](#)
- [44-04-02-06 Leistungselektronik E-Motor \(KFZ, high voltage\)](#)
- [44-04-02-90 Bordnetz \(KFZ, high voltage, nicht spezifiziert\)](#)
- [19-21-20-90 Testing software \(client, server operating system, unspecified\)](#)
- [36-64-15 Prüfstand für Interieurkomponenten](#)
- [36-64-90 Prüfmaschine, Prüfstand \(Sonstige\)](#)
- [36-65-02-90 Verbindungs-/Fügestation \(Montage- und Handhabungstechnik, nicht spez.\)](#)

Das „Common Data Dictionary“ der IEC [12] bietet ebenfalls tendenziell für den Leitungssatz verwendbare Klassifizierungen, z.B.

- [0112/2///61360 4#AAA548 - car plug<sup>9</sup>](#)
- [0112/2///61360 4#AAA032 - conductor](#)
- [0112/2///61360 4#AAA034 - insulated conductor](#)
- [0112/2///61360 4#AAA035 - cable](#)
- [0112/2///61360 4#AAA036 - electric power cable<sup>10</sup>](#)
- [0112/2///61360 4#AAA148 - connector](#)
- [0112/2///61360 4#AAA612 - connector shell](#)
- [0112/2///61360 4#AAA524 - connector contact](#)
- [0112/2///61360 4#AAA611 - connector tool](#)
- [0112/2///61360 4#AAF249 - cable/wire](#)
- [0112/2///61360 4#AUA478 - cable \(multi-conductor\)](#)
- [0112/2///61360 4#AUA479 - insulated wire \(single conductor\)](#)

Für die Produktklassifizierung über den VEC eignet sich die Klasse [PrimaryPartType](#)<sup>11</sup>.

Um eine gezielte *Produktsuche* zu vereinfachen, wird empfohlen, Leitungssatzkomponenten in Produktkatalogen mit den oben aufgeführten Bezeichnern zu klassifizieren, siehe Tabelle 3-1.

ClassificationSystem	ClassificationSystemVersion	ProductClassId
VEC	2.1.0	PrimaryPartType_Wire
ECLASS	15.0 (BASIC)	44040101 oder 44040201
IEC	2.0018.0002	0112/2///61360_4#AAA036

Tabelle 3-1: Beispielinhalte für SMC ProductClassification

<sup>9</sup> [https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TU0/0112-2--61360\\_4%23AAA548](https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TU0/0112-2--61360_4%23AAA548)

<sup>10</sup> [https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/classes/0112-2--61360\\_4%23AAA036](https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/classes/0112-2--61360_4%23AAA036)

<sup>11</sup> <http://www.prostep.org/ontologies/ecad/2024/03/vec#PrimaryPartType>

### 3.2.1.2 SM Handover Documentation

*IDTA 02004-1-2 HandoverDocumentation* [4] (vgl. Abbildung 3-5) definiert ein standardisiertes Austauschformat für Dokumentationsartefakte zu einem Asset. Die übergebenen Dokumente können Informationen enthalten, die z.B. für die korrekte Konstruktion, Installation, Inbetriebnahme, Ersatzteilbevorzugung, Bedienung, Reinigung, Inspektion, Wartung oder Reparatur erforderlich sind. Darüber hinaus gibt es gesetzliche Regelungen, die das Vorhandensein bestimmter Herstellerdokumente vorschreiben können, wie z.B. Konformitätserklärungen der Communauté Européenne (CE), Atmosphères Explosives (ATEX) Zertifikate oder Materialzertifikate.

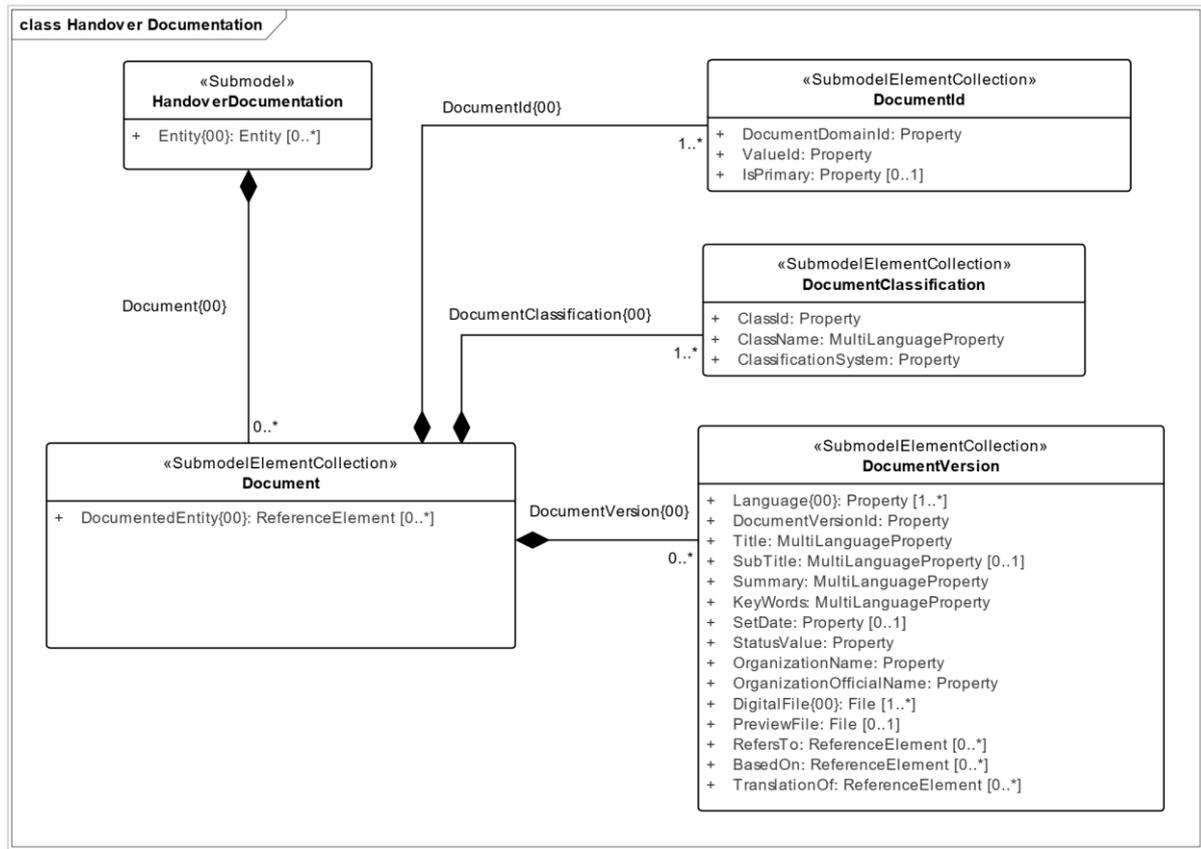


Abbildung 3-5: SM HandoverDocumentation

Im Teilmodell *HandoverDocumentation* müssen im SMC *DocumentVersion* insbesondere die Properties *Language*, *Title*, *OrganizationName*, *DigitalFile* und *PreviewFile* sowie die SMC *DocumentClassification* inhaltlich befüllt werden (siehe dazu auch die folgenden Unterkapitel und Kapitel 4.5.2).

#### 3.2.1.2.1 Dateitypen

Bei den *File*-Elementen muss auf den korrekten Medientyp (*MimeType*<sup>12</sup>) geachtet werden, damit das passende Anzeigetool ausgewählt und die Datei korrekt eingelesen werden kann. Siehe hierzu auch Kapitel 4.5.3 „Auswahl des ContentType“.

<sup>12</sup> <https://www.iana.org/assignments/media-types/media-types.xhtml>

### 3.2.1.2.2 Klassifizierungssysteme für Dokumente

Das SM *HandoverDocumentation* erlaubt die Klassifizierung von Dokumenten unter Verwendung von Referenzsystemen, siehe hierzu auch Kapitel 4.4.7 „Klassifizierungssysteme für Dokumente“.

Soweit vorhanden, wird empfohlen, die im folgenden aufgeführten Dokumenttypen in der VWS aufzuführen. Weiter wird empfohlen, die Dokumente nicht direkt in der VWS als Datei, sondern besser als Link auf einen Dokumentenserver bereitzustellen. Für die Preview-Files wird die Erstellung geeigneter Thumbnail-Bilddateien empfohlen, die für ganze Produktgruppen einheitlich genutzt werden können.

Als Mindestanforderung muss im SM *HandoverDocumentation* jedes Dokument nach VDI 2770 Blatt 1:2020-04 klassifiziert werden. Ergänzend kann nach IEC 61355-1:2008 klassifiziert werden.

Der Wert von *ClassificationSystem* ist dementsprechend auf „VDI2770 Blatt 1:2020“ oder „IEC 61355-1:2008“ zu setzen und der Wert von *ClassId* auf den entsprechenden Code aus den nachstehenden Tabellen:

#### 3.2.1.2.3 Datenblatt: Technische Beschreibung

ClassificationSystem	ClassID	ClassName (DE)	ClassName (EN)
VDI2770 Blatt 1:2020	02-01	Technische Spezifikation	Technical specification
IEC 61355-1:2008	DA	Datenblätter	Data sheets

Tabelle 3-2: Datenblatt: Inhalte für SMC DocumentClassification

#### 3.2.1.2.4 VEC: Technische Beschreibung

ClassificationSystem	ClassID	ClassName (DE)	ClassName (EN)
VDI2770 Blatt 1:2020	02-01	Technische Spezifikation	Technical specification
IEC 61355-1:2008	EC	Technische Spezifikations- / Anforderungsdokumente	Technical specification / requirement documents

Tabelle 3-3: Technische Spezifikation: Inhalte für SMC DocumentClassification

#### 3.2.1.2.5 Technische Zeichnung

ClassificationSystem	ClassID	ClassName (DE)	ClassName (EN)
VDI2770 Blatt 1:2020	02-02	Zeichnungen, Pläne	Drawings, plans
IEC 61355-1:2008	ED	Dimensionierungsdokumente	Dimensioning documents

Tabelle 3-4: Technische Zeichnung: Inhalte für SMC DocumentClassification

#### 3.2.1.2.6 Betriebsanleitung

ClassificationSystem	ClassID	ClassName (DE)	ClassName (EN)
VDI2770 Blatt 1:2020	03-02	Bedienung	Operation
IEC 61355-1:2008	DC	Anleitungen und Handbücher	Instructions and manuals

Tabelle 3-5: Betriebsanleitung: Inhalte für SMC DocumentClassification

#### 3.2.1.2.7 Konformitätserklärung

ClassificationSystem	ClassID	ClassName (DE)	ClassName (EN)
VDI2770 Blatt 1:2020	02-04	Zertifikate, Deklaration	Certificates, Declaration
IEC 61355-1:2008	EB	Normen und Richtlinien	Standards and regulations

Tabelle 3-6: Konformitätserklärung: Inhalte für SMC DocumentClassification

### 3.2.1.3 SM Digital Nameplate

IDTA 02006-2-0 *Digital Nameplate for Industrial Equipment* [5] (vgl. Abbildung 3-6) zielt darauf ab, den jeweiligen VWS interoperable Informationen über Anlagenkennzeichen zur Verfügung zu stellen.

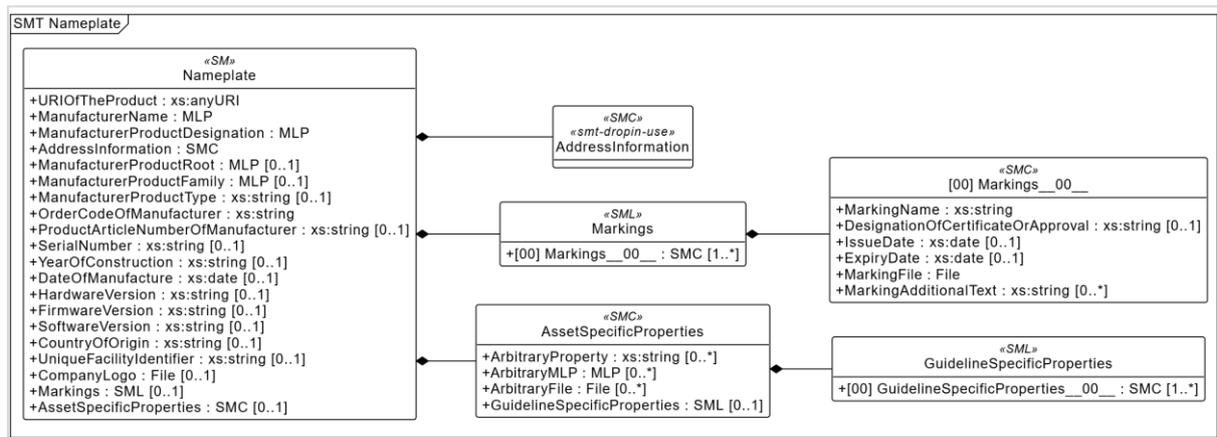


Abbildung 3-6: SM DigitalNameplate

Das SM *DigitalNameplate* enthält einige instanzbezogene Properties. Diese sollen für die Anwendung in der Typ-VWS nicht befüllt werden. Das betrifft vor allem *SerialNumber*, *DateOfManufacture* sowie *FirmwareVersion* und *SoftwareVersion*. Eine Ausnahme wird vorgeschlagen für *URIOfTheProduct*, wo der Link auf eine außerhalb des VWS-Ökosystems vorhandenen Online-Produktkatalogseite eingebracht werden kann, um hierüber eine sinnvolle Informationsverlinkung herzustellen.

Bei einigen Properties im SM *DigitalNameplate* gibt es Redundanzen zum SM *TechnicalData*, die beachtet und konsistent gehalten werden müssen, bei *ManufacturerName* besteht eine gewisse [Typinkonsistenz](#)<sup>13</sup>, siehe Tabelle 3-7:

SM „Digital Nameplate“	SM „Technical Data“	Inhalt
<b>ManufacturerName [MLP]</b>	ManufacturerName [Prop]	Textueller Bezeichner ohne Sonderzeichen
<b>ManufacturerProductDesignation [MLP]</b>	ManufacturerProductDesignation [MLP]	Beschreibender Kurztext, sprachabhängig
<b>ProductArticleNumberOfManufacturer [Prop]</b>	ManufacturerArticleNumber [Prop]	Textueller Bezeichner ohne Sonderzeichen
<b>OrderCodeOfManufacturer [Prop]</b>	ManufacturerOrderCode [Prop]	Textueller Bezeichner ohne Sonderzeichen
<b>CompanyLogo [File]</b>	ManufacturerLogo [File]	Langzeitstabile URL auf Bilddatei oder angehängte Datei

Tabelle 3-7: Redundante Properties in SM DigitalNameplate und SM TechnicalData

<sup>13</sup> <https://github.com/admin-shell-io/submodel-templates/issues/127>

### 3.2.1.3.1 Produkthierarchie

Das SM *DigitalNameplate* enthält als wesentliche Informationsbestandteile für den Produktkatalog die Produkthierarchie, welche über die folgenden Properties abgebildet wird, siehe Tabelle 3-8.

Attribute	Produkthierarchie
<i>ManufacturerProductRoot</i>	Top level of a 3-level manufacturer specific product hierarchy
<i>ManufacturerProductFamily</i>	Second level of a 3-level manufacturer specific product hierarchy
<i>ManufacturerProductDesignation</i>	Third or lowest level of a 3-level manufacturer specific product hierarchy
<i>ManufacturerProductType</i>	Characteristics to differentiate between different products of a product family or special variants

Tabelle 3-8: Properties für Produkthierarchie

Es wird empfohlen, dass jede Organisation für sich ein klar definiertes Konzept entwirft, wie diese Properties in der VWS konkret befüllt werden sollen. Siehe dazu auch Kapitel „

AP 12.3 - Extraktionslogik entwickeln“.

### 3.2.1.3.2 Kennzeichen

Die SMC *Markings* soll Verweise auf relevante Conformance-Standards und -Organisationen enthalten, idealerweise mit einem geeigneten Logo zur grafischen Darstellung (*MarkingFile*) auf dem Typenschild. In *DesignationOfCertificateOrApproval* sollte der jeweilige exakte Spezifikationsname verwendet werden, für *MarkingName* eine gängige Abkürzung.

Spezifikationen für Produktkonformität im Automotive-Umfeld sind exemplarisch in Tabelle 3-9 dargestellt.

<b>MarkingName / DesignationOfCertificateOrApproval</b>	
<b>LV216-2</b>	VDA-Standard für geschirmte Hochspannungsmantelleitungen für Kraftfahrzeuge und deren elektrische Antriebe
<b>BMW: GS 95007-6-2</b>	
<b>BMW: 9 327 162.9</b>	
<b>Mercedes: C51 / 12.14</b>	
<b>Daimler Truck: A0025460501</b>	
<b>VW: N 107 777</b>	
<b>VW: N 108 888</b>	

Tabelle 3-9: Automotive-Spezifikationen

Wenn für *MarkingFile* kein offizielles Logo in Form einer Bilddatei vorhanden ist, wird aus Darstellungsgründen empfohlen, ein Logo generativ mit dem Text des Dokuments zu erzeugen, siehe Abbildung 3-7: Logos für Kennzeichen.



Abbildung 3-7: Logos für Kennzeichen

Ergänzende Erklärungen können in *MarkingAdditionalText* untergebracht werden. Falls ergänzende Dokumente zu einzelnen Kennzeichen übermittelt werden sollen, müssen diese im Teilmodell *IDTA 02004-1-2 HandoverDocumentation* [4] untergebracht werden.

### 3.2.2 Benennung der VWS

Prägendes Element für die Benennung einer VWS ist die **idShort**. Es wird empfohlen, diese inhaltlich sprechend zu gestalten, so dass der **Herstellername**, die **Produktfamilie** und die **Artikelnummer** erhalten sind. Bei AASX-Dateiausleitungen sollte die **idShort** die Basis für den Dateinamen bilden.

Im Folgenden ist detaillierter erklärt, welche Aspekte bei der ID-Bildung beachtet werden sollten.

#### 3.2.2.1 ID-Bildung

Eine systematische ID-Definition ist relevant für die langfristige Wartbarkeit des VWS-Systems. Es wird eine methodische Vorgehensweise empfohlen, nach der die VWS-Veröffentlicher bei der ID-Bildung vorgehen können.

Es gibt zwei Arten von Identifizierern. Zusätzlich sind in Tabelle 3-10 die Metadaten beschrieben:

- Semantic-IDs, um Konzepte zu referenzieren
- IDs für Artefakte, d.h. Assets, VWS, Submodelle, Properties

VWS-Identifikator	Beschreibung
<b>idShort</b>	Textueller Bezeichner, darf nur Buchstaben, Ziffern und Unterstriche (``_``) enthalten; sie muss mit einem Buchstaben beginnen. Inhaltlich idealerweise identisch oder ähnlich der <i>ProductArticleNumberOfManufacturer</i> bzw. <i>ManufacturerArticleNumber</i>
<b>aasId</b>	URI/IRI
<b>globalAssetId</b>	URI/IRI nach IEC 61406 (Identification Link), wird als QR-Code aufgebracht und fungiert als Digitales Typenschild.
<b>semanticId</b>	GUID, IRDI, URI/IRI als Referenz auf einen semantischen Datensatz, z.B. im Concept Description Repository, der ECLASS-Datenbank oder dem IEC-CDD.

Tabelle 3-10: Identifikations-Metadaten der VWS

Die Anwendung von Uniform Resource Identifier (URI)/ Internationalized Resource Identifier (IRIs) wird als bevorzugte Methode (vor International Registration Data Identifier-IRDIs, Globally Unique Identifier-GUIDs) zur Bildung von global unverwechselbaren IDs empfohlen. Die VWS-Metamodell-Spezifikation [13] beschreibt dazu eine „Best Practice for Creating URI Identifiers“, basierend auf folgendem Grammatik-Schema:

```

Grammar:
<AAS URI> ::= <scheme> ":" <authority> [ <path> ]
<scheme> ::= a valid URI scheme
<authority> ::= Organization
<path> ::= <subunit> <domain> <release> <element>
<subunit> ::= { ("/" | ":") <Organizational Subunit/Document ID/Document subunit> }*
<domain> ::= [ ("/" | ":") <Submodel / Domain-ID>
<release> ::= [ ("/" | ":") <Version> [ ("/" | ":") <Revision> ]* ]
<element> ::= [ ("/" | ":" | "#") {( <Property/Element-ID> | <Instance number> )}*
]
    
```

Abbildung 3-8: Grammatik für URI Identifier

Es wird empfohlen, diesen generischen Strukturvorschlag in Abbildung 3-8 wie im Folgenden beschrieben in vereinfachter Form anzuwenden, ggf. im **<path>** auf die Elemente **<domain>** und **<release>** zu verzichten. Insbesondere **<release>** als Bestandteil der ID kann langfristig problematisch werden, sollte sich diese ändern. Im Falle der Notwendigkeit von geänderten Versionsständen (die aus Wartbarkeitsgründen generell vermieden werden sollten) könnten diese besser per Query-Parameter abgefragt werden.

Die **aasld** der VWS ist die global eindeutige Identifikations-Zeichenketten zur Kennung der VWS in Form eines URI/IRI nach [RFC 3987](#), wobei darauf zu achten ist, dass der verwendete Zeichensatz des Kennzeichnungssystems kompatibel zu RFC 3986 ist, d.h. aus sog. „unreserved characters“ besteht und prozentcodierte Bestandteile konsequent vermeidet.

Die **idShort** ist ein optionaler kurzer textueller Bezeichner, der nur in diesem Kontext eindeutig sein muss. Die **idShort** darf keine Leerzeichen oder Sonderzeichen enthalten und nur Ziffern, Buchstaben und das „\_“-Symbol enthalten ([a-zA-Z][a-zA-Z0-9\_]). Zudem muss die erste Ziffer ein Buchstabe sein.

Wenn eine VWS veröffentlicht - d.h. für die Produktion oder den Betrieb "freigegeben" - wird, ist es wichtig, eine **globalAssetId** als Zugriffslink zuzuweisen. Diese Asset-ID kann z.B. über einen QR-Code auf dem Asset, über ein RFID für das Asset, aus der Firmware des Assets oder aus einer Asset-Datenbank abgerufen werden. Die IEC 61406 (früher DIN SPEC 91406) definiert das Format solcher Asset-IDs und es wird empfohlen den *IdentificationLink* nach IEC 61406-1 [14] als Basis sowohl für die **id** als auch für die **globalAssetId** zu verwenden. Die URI/IRI kann systematisch so aufgebaut werden, dass Sie gleichzeitig zur produktbezogenen Referenzierung und Verlinkung in den Digitale Produkttyp (DTP) und die Digitale Produktinstanz (DPI) definiert werden kann:

Der *IdentificationLink* in der **globalAssetId** könnte sich demnach nach einem der folgenden Beispiele systematisch zusammensetzen:

- `https://{ManufacturerDomain}/{ManufacturerArticleNumber}`
- `https://{ManufacturerDomain}/{ManufacturerOrderCode}`
- `https://{ManufacturerDomain}/{ManufacturerArticleNumber}/{ManufacturerOrderCode}`

Für eine spätere Produkt-Instanz könnte man diese Typ-URI/IRI dann einfach um bspw. die *serialNumber* erweitern, um so eine Instanz-URI/IRI zu bilden:

- `https://{ManufacturerDomain}/{ManufacturerArticleNumber}/{ManufacturerOrderCode}/{SerialNumber}`

Alternativ kann die Linkbildung auch nach IEC61406-2 [15] erfolgen, wobei dann die Identifikationsparameter nicht im URL-Pfad codiert sind, sondern als Query-Parameter ausgeführt werden.

- `https://{manufacturerDomain}?1P={ManufacturerArticleNumber} S={ManufacturerOrderCode}`

Dieser *Identification Link* nach IEC 61406 [14] [15] verbindet ein Produkt nahtlos mit seinen Informationen im Internet – dem Digitalen Typenschild (Digital Nameplate, DNP) als Schlüsseltechnik für Industrie 4.0 -Anwendungen, wie bspw. den Digital Product Passport (DPP).

### 3.2.2.2 Best-Practice-Kochrezept

In Abbildung 3-9 ist eine Basis URL-Struktur dargestellt. Im Folgenden wird eine zusammenfassende Übersicht der Empfehlungen zur ID-Definition mittels URI/IRI präsentiert:



Abbildung 3-9: Basic URL Structure<sup>14</sup>

1. Protocol (scheme) „https://“ (nicht „http://“) verwenden wegen Linkfähigkeit als URL
2. Domain-Name (authority) des Komponentenherstellers verwenden
3. Möglichst kurze Root Domain
4. Subdomain oder Subdirectory „**aas**“ wird empfohlen für **aasld**
5. Subdomain oder Subdirectory „**asset**“ wird empfohlen für **globalAssetId**

<sup>14</sup> <https://www.hostwinds.com/blog/11-parts-of-url-complete-guide>

6. Im Pfad Sonderzeichen und Leerzeichen vermeiden (RFC 3986)
7. Keine Sprachabhängigkeiten im Pfad (z.B. „/en/“, „/de/“)
8. Zufallsstrings (z.B. UUIDs) als ID-Bestandteil vermeiden!
9. Pfadb Bestandteile ohne semantischen Informationsgehalt vermeiden
10. Auf bestehendem ID-System aufsetzen, bspw. Artikelnummern  
z.B. „https://aas.kostal.com/**10142491**“
11. Passendes gleichförmiges Schema für Submodell-IDs verwenden,  
z.B. „https://aas.kostal.com/**sm/nameplate/10142491**“
12. **IdShort**: menschenlesbarer kurzer String, ohne Sonderzeichen, ggf. nach dem Schema „{ManufacturerName}\_{ManufacturerProductFamily}\_{ManufacturerArticleNumber}“ aufbauen,  
z.B. **KOSTAL\_PLK14\_10142491**
13. Ggf. **IdShort** als ID-Bestandteil im URI/IRI-Pfad verwenden  
z.B. „https://aas.kostal.com/**KOSTAL\_PLK14\_10142491**“
14. Analog gleichförmiges Schema für Dateinamen anwenden,  
z.B. „**KOSTAL\_PLK14\_10142491\_v10.aasx**“

### 3.2.3 Benennung der Submodelle

Es wird empfohlen, die Benennungen der IDTA-Submodelle möglichst nicht zu verändern. In untergeordneten Strukturen werden allerdings für SMC-Listenelemente menschenlesbare Benennungen empfohlen, anstelle der in den SMT-Spezifikationen manchmal empfohlenen fortlaufenden Nummerierungen (siehe Beispiele in Abbildung 3-10).

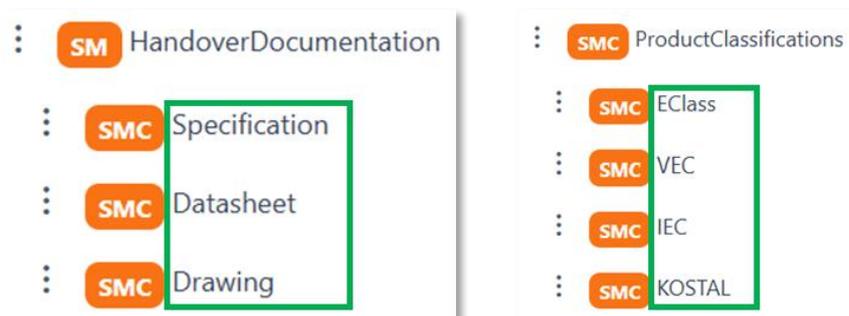


Abbildung 3-10: Benennung von Submodell-Elementen

### 3.2.4 Versionierung

Die Versionierung einer VWS erfolgt über die administrativen Metadaten (Version, Revision, Creator dargestellt in Abbildung 3-11) und sollte entsprechend der unternehmensinternen Prozesse gepflegt werden.

Administrative information	
Version	1
Revision	0
Creator	mrentsch
Template ID	https://aas.kostal.com/DigitalNameplate/1/0/

Abbildung 3-11: Versionierung

Es muss darauf hingewiesen werden, dass im VWS-Metamodell derzeit immer noch keine Metadaten für die Historieninformation verfügbar sind, siehe dazu auch „[AdministrativeInformation: New properties "createdAt" and "lastModified" Issue #520 · admin-shell-io/aas-specs-metamodel](https://github.com/admin-shell-io/aas-specs-metamodel/issues/520)“<sup>15</sup>.

<sup>15</sup> <https://github.com/admin-shell-io/aas-specs-metamodel/issues/520>

### 3.2.5 Semantische Referenzierung in der VWS

Semantische Referenzierung stellt sicher, dass Daten in der Verwaltungsschale nicht nur Werte enthalten, sondern auch deren Bedeutung klar definiert ist. Dies ermöglicht Interoperabilität zwischen Systemen, da die Daten kontextbezogen interpretiert werden können. Beispielsweise wird eine Eigenschaft wie "Temperatur" nicht nur als Zahl (z.B. 25) dargestellt, sondern mit einer eindeutigen Definition verknüpft (z.B. "Temperatur in Grad Celsius").

Dies geschieht hauptsächlich durch die *semanticId* und die *Concept Description*. Die *semanticId* verweist auf eine Definition (extern oder intern), während die *ConceptDescription* als eigenständiges Element in der Verwaltungsschale eine detaillierte semantische Beschreibung eines Konzepts (z.B. einer Eigenschaft, Einheit oder eines Objekts) bereitstellen kann.

Andere Systeme können die *semanticId* auswerten, die referenzierte Definition nachschlagen (entweder direkt oder über die *ConceptDescription*) und die Bedeutung der Daten verstehen. Grundsätzlich wird unterschieden:

- **Direkte Referenzierung:** Die *semanticId* einer Property kann direkt auf eine externe Definition (z. B. ECLASS, IEC 61360 oder VEC) verweisen (via *ExternalReference*). In diesem Fall ist **keine** *ConceptDescription* notwendig, siehe Listing 3-1.

```
{
  "idShort": "Temperature",
  "modelType": "Property",
  "valueType": "xs:double",
  "value": "25.0",
  "semanticId": {
    "type": "ExternalReference",
    "keys": [
      {
        "type": "GlobalReference",
        "value": "0112/2///61360_4#AAE685"
      }
    ]
  }
}
```

Listing 3-3-1: Verweis auf externe semantische Beschreibung eines Property

- **Indirekte Referenzierung:** Die *semanticId* einer Property verweist auf eine *ConceptDescription* innerhalb der Verwaltungsschale (via *ModelReference*). Die *ConceptDescription* selbst kann über ihre *dataSpecifications* auf externe Standards verweisen.

**Beispiel:** Die folgende *ConceptDescription* beschreibt das Konzept von „Durchmesser“ mit der Einheit „Millimeter“ und kann von mehreren Elementen in der Verwaltungsschale referenziert werden, was die Wiederverwendbarkeit fördert. Intern verweist Sie über *dataSpecification* auf externe Definitionen (z.B. in VEC, ECLASS und IEC).

```
{
  "idShort": "Diameter",
  "modelType": "Property",
  "valueType": "xs:double",
  "value": "21.2",
  "semanticId": {
    "type": "ModelReference",
    "keys": [
      {
        "type": "ConceptDescription",
        "value": "http://example.com/concept/diameter"
      }
    ]
  }
}
```

Listing 3-2: Verweis auf externe semantische Beschreibung eines Property

```
}
  "modelType": "ConceptDescription",
  "id": "http://example.com/concept/diameter",
  "idShort": "diameter",
  "displayName": [
    {
      "language": "en",
      "text": "diameter"
    }
  ]
}
```

```

    ],
    "embeddedDataSpecifications": [
      {
        "dataSpecification": {
          "type": "ExternalReference",
          "keys": [
            {
              "type": "GlobalReference",
              "value": "http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360/3/0"
            }
          ]
        },
        "dataSpecificationContent": {
          "preferredName": [
            {
              "language": "en",
              "text": "diameter"
            },
            {
              "language": "de",
              "text": "Durchmesser"
            }
          ],
          "unit": "mm",
          "unitId": {
            "type": "ExternalReference",
            "keys": [
              {
                "type": "GlobalReference",
                "value": "0173-1#05-AAA480#004"
              }
            ]
          },
          "dataType": "REAL_MEASURE",
          "definition": [
            {
              "language": "de",
              "text": "Ausdehnung, gemessen als Abstand..."
            }
          ],
          "modelType": "DataSpecificationIec61360"
        }
      }
    ]
  }
}

```

Listing 3-4 Umsetzung der Concept Description auf der JSON-Ebene

Eine *ConceptDescription* fungiert als Metadaten-Container und enthält Informationen wie:

- Eindeutige Identifikation (*id*).
- Kurzbezeichnung (*idShort*).
- Beschreibungen in mehreren Sprachen (*preferredName*, *shortName*, *definition*).
- Verweise auf standardisierte Datenformate oder Ontologien (*dataSpecifications*).
- Optional: Zusätzliche Attribute wie Synonyme, Formeln oder physikalische Definitionen.

Die *semanticId* und die *ConceptDescription* zusammen ermöglichen eine vollständige semantische Referenzierung innerhalb des VWS-Ökosystems.



### 3.2.6 Einheitendeklaration in der Concept Description

Die [IDTA GUIDELINE: How to transport ECLASS in the Asset Administration Shell](#) [16] beschreibt die Anwendung von ECLASS und Einheitendefinitionen über *ConceptDescriptions*. Es wird allerdings empfohlen, auch die in dem Dokument nicht berücksichtigte Property *symbol* zu belegen, damit Viewer Tools diese sicher anzeigen können, siehe Abbildung 3-14.

SemanticID	
ID	<a href="http://www.prostep.org/ontologies/ecad/2024/03/vec#terminalTypeNominalSize">http://www.prostep.org/ontologies/ecad/2024/03/vec#terminalTypeNominalSize</a>
Definition	<b>Specifies the nominal size of terminals that fit into the cavity. (e.g. 2x4).</b>
Source of definition	<a href="http://qudt.org/vocab/unit/Millim">http://qudt.org/vocab/unit/Millim</a>
Unit	<b>Millimeter</b>
Symbol	<b>mm</b>
Datatype	<b>RealMeasure</b>

Abbildung 3-14: Beispiel zur Einheitendefinition mittels einer Concept Description

Alternativ zu ECLASS (welches ohnehin auf IEC 62720 referenziert) wird empfohlen, direkt auf Einheitendefinitionen in IEC 62720 zu verweisen, und zwar mittels URI-Referenzen über <http://qudt.org>, da diese Lösung gleichzeitig eine attraktive Onlinedarstellung bereitstellt (siehe Abbildung 3-15 bzw. <https://qudt.org/vocab/unit/Millim>):

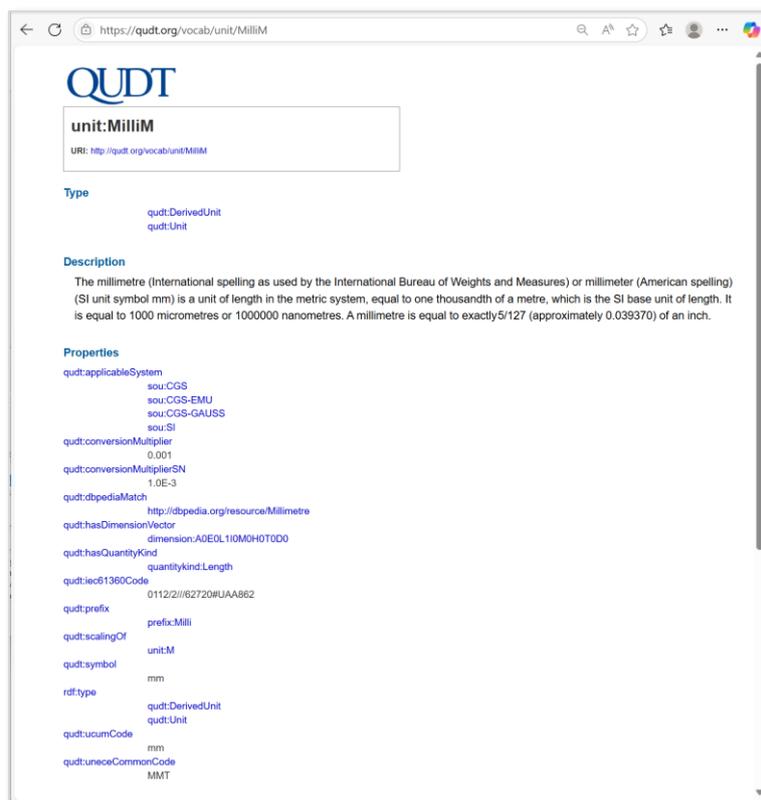


Abbildung 3-15: QUDT-Onlinedarstellung

Siehe zu diesem Thema auch „[Listing 4-4: Referenzbeispiele für Properties Klassifizierungssysteme für Einheiten](#)“.

### 3.2.7 Anwendung des VEC

VEC [17] ist das führende Modellierungssystem der Leitungssatzbranche und stellt präzise semantische Definitionen für Komponentenmerkmale und Datentypen bereit. Um Inkonsistenzen im Produktlebenszyklus zu vermeiden, wird empfohlen, in Produktkatalogen und Datenblättern nur Merkmale aufzuführen, die auch im VEC-Standard definiert sind. In der VWS sind diese mit einer entsprechenden semantischen Referenz auf die VEC-Definition zu versehen, um die maschinelle Interpretierbarkeit sicherzustellen. Im Idealfall erstellt bereits der Komponentenhersteller ergänzend eine komplette VEC-Spezifikation für seine jeweilige Komponente und fügt diese der VWS als Dateianhang hinzu, siehe Abbildung 3-16.

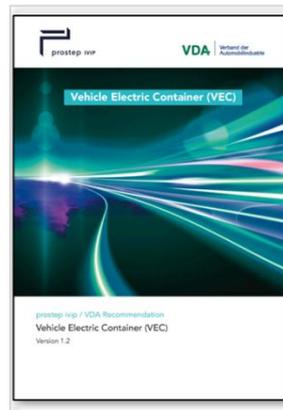


Abbildung 3-16: Spezifikation des Vehicle Electric Container

Für die Produktklassifizierung über den VEC eignet sich die Klasse [PrimaryPartType](#)<sup>18</sup>, siehe Abbildung 3-17.

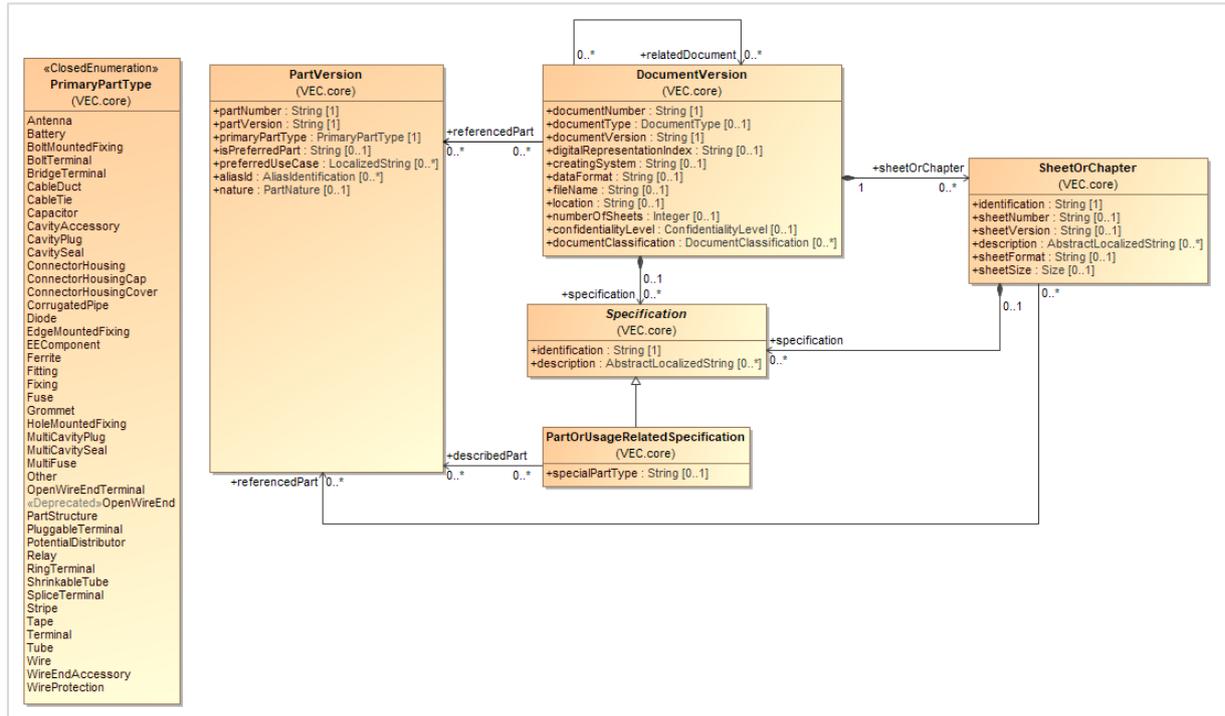


Abbildung 3-17: VEC-Datenmodell: Description of Parts

<sup>18</sup> <http://www.prostep.org/ontologies/ecad/2024/03/vec#PrimaryPartType>

Es liegt nahe, für leitungssatzspezifische Properties in Produktkatalogen auf die [Definitionen des VEC](#) zurückzugreifen und entsprechend im Submodell *TechnicalData* [3] in der SMC *TechnicalProperties* darauf zu verweisen, siehe für das Beispiel eines Terminals Abbildung 3-18. Unter <https://ecad.prostec.org/ontologies/2024/03/vec> wird eine Ontologie in „[RDF 1.1 Turtle](#)“-Syntax bereitgestellt. Deren Anwendung ist in Kapitel 0 erläutert.

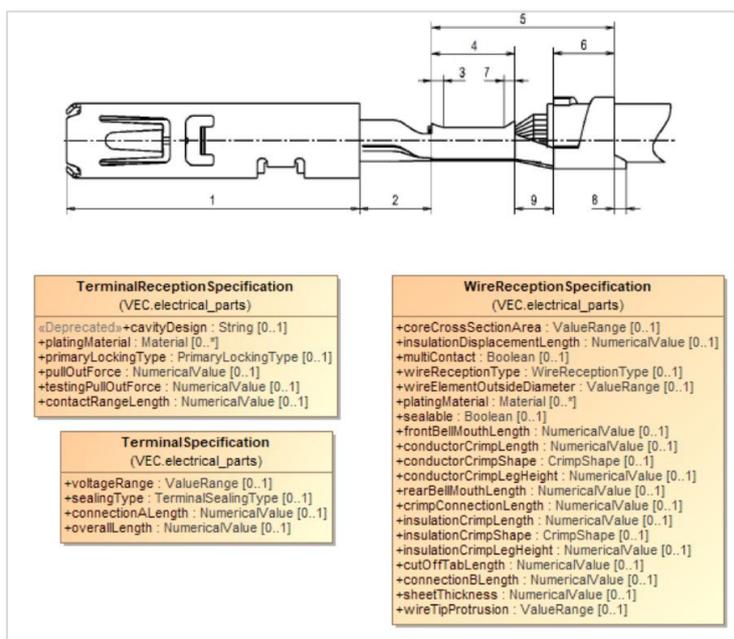


Abbildung 3-18: Beispiel VEC Terminal Datamodel (Quelle: VEC Recommendation V2.1)

Sowohl in der OPC UA Companion Spec. „OPC 40570: OPC UA for the Wire Harness Manufacturing Industry“ [18] als auch der [DIN 72036](#) [19] wird als semantische Referenzierung auf die [Ontologie des VEC](#) verwiesen.

Als bevorzugte Lösung wäre es daher wünschenswert, dass Komponentenlieferanten grundsätzlich ein vollständiges VEC-Modell als Handover-Dokumentation zu einer Komponente bereitstellen. Unabhängig davon wird der im Folgenden beschriebene Ansatz als Migrationspfad empfohlen, um eine Angleichung der Datenmodelle zwischen Produktkatalog und VEC zu erreichen.

Im Sinne konsistenter Datendefinitionen über den Produktlebenszyklus hinweg wurde daher versucht, auch die im Online-Katalog dargestellten Produktdaten (im Submodell *TechnicalData* enthalten) auf dem VEC-Datenmodell basierend zu definieren, um eine semantische Kompatibilität der im klassischen Produktdatenblatt eines Online-Katalog dargestellten Informationen zur VEC-Ontologie zu erreichen.

In Kapitel 0 ist das Prinzip der Referenzenbildung zur VEC-Ontologie erläutert.

### 3.2.7.1 NumericalValue (VEC) als Property (VWS) modellieren

Modellierung eines VEC-NumericalValue durch verschiedene VWS-Konzepte und deren Bewertung, siehe Abbildung 3-19:

#### Variante 1: Definition der Einheit in der *ConceptDescription*

```
{
  "idShort": "vec_terminalTypeNominalSize",
  "description": [
    {
      "language": "en",
      "text": "Specifies the nominal size of terminals that fit into the cavity. (e.g. 2x4).\"
    }
  ],
  "id": "http://www.prostep.org/ontologies/ecad/2024/03/vec#terminalTypeNominalSize",
  "embeddedDataSpecifications": [
    {
      "dataSpecificationContent": {
        "preferredName": [
          {
            "language": "en",
            "text": " Terminal Size"
          }
        ],
        "unit": "Millimeter",
        "sourceOfDefinition": "http://qudt.org/vocab/unit/MilliM",
        "symbol": "mm",
        "dataType": "REAL_MEASURE",
        "definition": [
          {
            "language": "en",
            "text": "Specifies the nominal size of terminals that fit into the cavity. (e.g. 2x4).\"
          },
          {
            "language": "de",
            "text": "Gibt die Nenngro\u00dfe der Klemmen an, die in den Hohlraum passen. (z. B. 2x4).\"
          }
        ],
        "modelType": "DataSpecificationIec61360"
      },
      "dataSpecification": {
        "type": "ExternalReference",
        "keys": [
          {
            "type": "GlobalReference",
            "value": "http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360/3/0"
          }
        ]
      }
    }
  ],
  "modelType": "ConceptDescription"
}
```

Listing 3-3 Definition der Einheit in der *ConceptDescription*

SemanticID	
ID	http://www.prostep.org/ontologies/ecad/2024/03/vec#terminalTypeNominalSize
Definition	<b>Specifies the nominal size of terminals that fit into the cavity. (e.g. 2x4).</b>
Source of definition	<b>http://qudt.org/vocab/unit/MilliM</b>
Unit	<b>Millimeter</b>
Symbol	<b>mm</b>
Datatype	<b>RealMeasure</b>

Abbildung 3-19: Einheitendefinition in der *Concept Description* f\u00fcr ein VEC NumericalValue

#### Problem:

- Die Einheit (z.B. m, mm, Zoll) ist damit global f\u00fcr die referenzierte VEC-Eigenschaft f\u00fcr alle VWS auf dem VWS-Server definiert.

- Bei Einheitenwechsel aus einer nativen Quelle kann eine Konvertierung mit Rundungsfehlern auftreten, da es im Kontext Einheiten gibt, die nicht direkt umrechenbar sind (z.B. Leitungsquerschnitte in mm<sup>2</sup> vs. American Wire Gauge (AWG) analog wie zöllige und metrische Gewinde). Die meisten CAD-Formate (bspw. STEP oder JT) sind Einheiten agnostisch.

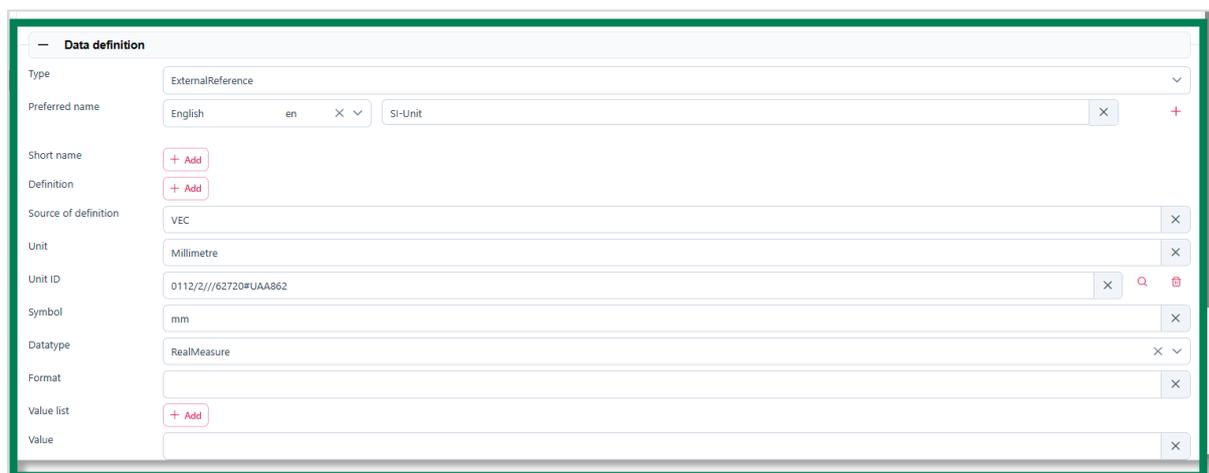
Betrachtung des folgenden Szenarios: Hersteller X gibt seine Daten frei (in Inches) und will seine Daten in einem zentralen VWS Marketplace einstellen. Dort werden jetzt die Inches in Meter umgerechnet. Falls nun diese digitalen Daten auch für digitale Verhandlungsprozesse genutzt werden und dann auch Vertragsgrundlage sind, wer verantwortet es, wenn die Daten aufgrund von Rundungsfehlern bei der Umrechnung nicht stimmen? Der Hoster des Marketplace? Der Einsteller der Daten, weil er das, was umgerechnet wurde, nicht nochmal geprüft hat, der Datenempfänger, weil er sich auf die Daten verlassen hat, oder der Hersteller der VWS-Repository Service?

**Variante 2:** Erstellen einer *ConceptDescription* für jede Einheit-Eigenschafts-Kombination, z. B.

- vec#thickness\_m
- vec#thickness\_mm
- vec#thickness\_inches

Problem: Der semantische Bezug zwischen den *ConceptDescription* und der VEC-Spezifikation würde verloren gehen.

**Variante 3:** Definition der Einheit in der eingebetteten Datenspezifikation jedes VEC Properties, siehe Abbildung 3-20.



Data definition	
Type	ExternalReference
Preferred name	English en × SI-Unit × +
Short name	+ Add
Definition	+ Add
Source of definition	VEC ×
Unit	Millimetre ×
Unit ID	0112/2//62720#UAA862 × 🔍 🗑️
Symbol	mm ×
Datatype	RealMeasure ×
Format	×
Value list	+ Add
Value	×

Abbildung 3-20: Definition der Einheit in der eingebetteten Datenspezifikation der Property

Problem: Erzeugt potenzielle Redundanz.

Basierend auf der dargestellten Analyse wird für metrische Einheiten die Anwendung der Variante 1 als Standard empfohlen, für andere Einheitensysteme die Anwendung der Variante 3, um das Risiko von Rundungsfehlern bei der Einheitenkonvertierung in der Verantwortung des Datenempfängers zu belassen.

### 3.2.7.2 VEC als VWS-Submodell

Für die [OSS Harness Lib](#) wurde seitens 4Soft ein „experimental“ Modul „[vec-VWS-conversion](#)“<sup>19</sup> erstellt, um einen VEC verlustfrei in ein VWS-Submodell konvertieren zu können. Dabei wurden Modellierungsschwächen im Bereich der VWS-Properties offenbart, insbesondere zur Definition von Bereichswerten und Toleranzen.

Die Behebung dieser Modellierungsschwächen wurde daraufhin bei der IDTA in die Wege geleitet.

Der Ansatz der verlustlosen Transformation des VEC in ein VWS-Submodell wurde nicht weiterverfolgt, da der VEC als formal definierte Spezifikation vorliegt und als solcher im Originalformat an die VWS angehängt werden kann.

### 3.2.7.3 VEC als HandoverDocumentation

Als praktikabelste Lösung wurde daher die Einbindung der VEC-Datei als *HandoverDocument* angesehen und auch zur Anwendung empfohlen. Das *File-* (oder *Blob-*)Element sollte dabei den Medientyp (*mimeType*) „text/xml“ besitzen, damit entsprechende Softwarewerkzeuge die Datei korrekt einlesen können. Sinnvoll wäre auch, bei der IANA einen dedizierten [MimeType](#)<sup>20</sup> für VEC zu reservieren (bspw. „application/vec+xml“), wie bereits hier angestrebt wurde: [\[KBLFRM-1254\] Registrierung von IANA Media Types für den VEC · Issue #1118 · prostep-ivip-e-V/vec](#)<sup>21</sup>.

### 3.2.7.4 Referenzieren innerhalb des VEC

Wie bereits im [Architekturkapitel](#) [20] beschrieben, wäre die Referenzierung spezifischer Informationsfragmente innerhalb der VEC-Datei ein interessanter Ansatz. Hierzu haben ab Version 2.1 VEC-Elemente die Fähigkeit erhalten, global eindeutige IDs zu tragen; (URNs). Diese können als Referenzierungsanker verwendet werden und ggf. mit einer einfachen Ergänzung der VWS-REST-API direkt zugänglich gemacht werden.

---

<sup>19</sup> <https://github.com/4Soft-de/harness-model/pull/313>

<sup>20</sup> <https://www.iana.org/assignments/media-types/media-types.xhtml>

<sup>21</sup> <https://github.com/prostep-ivip-e-V/vec/issues/1118>

### 3.3 AP 12.3 - Extraktionslogik entwickeln

Im Arbeitspaket „Extraktionslogik entwickeln“ wurden folgende Aspekte für die Erzeugung von Komponenten-VWS betrachtet:

- Unternehmensinterne Datenquellen auf VWS-Struktur mappen.
- Datenquellen analysieren und auf Standards abstimmen.
- Schnittstellen zu Datenquellen implementieren.
- Extract-Transform-Load (ETL)-Prozesse erstellen.

Für die Industrialisierung der VWS müssen von den Komponentenherstellern VWS für Ihre Katalog-Produkte bereitgestellt werden, um deren weitere Verwendung in branchentypischen Engineering-Systemen zu ermöglichen. Heutige Produktdatenhaltungssysteme bieten üblicherweise noch keine VWS-Ausleitungsschnittstelle an.

Deshalb wurde bei interessierten Projektpartnern die interne Datenbereitstellung analysiert und Lösungen zum ETL-Prozess für die maschinell unterstützte Erzeugung der Produkt-VWS vorgeschlagen. Ebenso wurden den Projektpartnern Vorschläge zur Ertüchtigung Ihrer IT-Landschaft erarbeitet, um einen VWS-basierten Produktkatalog bereitstellen zu können. Ergänzend wurden VWS für repräsentative Produkte bzw. Produktfamilien der Projektpartner erzeugt und in einer geeigneten Demonstrator-Umgebung der ARENA2036 bereitgestellt <sup>22</sup>.

Zur Ermittlung der notwendigen Informationen wurden mit den Projektpartnern (Komax Testing, Coroflex, Kostal) Workshops durchgeführt. Dazu wurden bei allen Partnern folgende Themen betrachtet und erarbeitet:

1. Definition des aktuellen Stands bei der Veröffentlichung von Produktinformationen (30min)
2. Erarbeitung der Enterprise Architektur mit den relevanten (externen) Akteuren, deren Touchpoints, betroffene Abteilungen und Rollen im Unternehmen sowie die internen Prozesse, Systeme und Daten (1,5 – 2h)
3. Vorstellung des ETL-Prozesses mit Eclipse Mnestix (15min)
4. Zusammenfassung (15min)
5. Feedback (5min)

Die Ergebnisse zu diesen Agendapunkten wurden auf einem digitalen Whiteboard uniform mit folgendem Säulenschema festgehalten, siehe



Abbildung 3-21: Workshop-Schema

<sup>22</sup> <https://marketplace.arena2036.app>

### 3.3.1 Komax Testing

Der Workshop fand am 12. Mai in digitaler Form statt und wurde in einem digitalen Whiteboard dokumentiert. Zuerst wurde der aktuelle Stand bzgl. der Darstellung der Komax Testing-Produkte erarbeitet. Neben den Inhalten auf der Webseite<sup>23</sup> ist hierbei das Komax Testing-eigene Kundenportal *TSK Direct* hervorzuheben, ein Kundenportal für gelieferte Produkte. Abbildung 3-22 zeigt exemplarisch die Übergabedokumentation für einen Prüfadapter im TSK Direct.

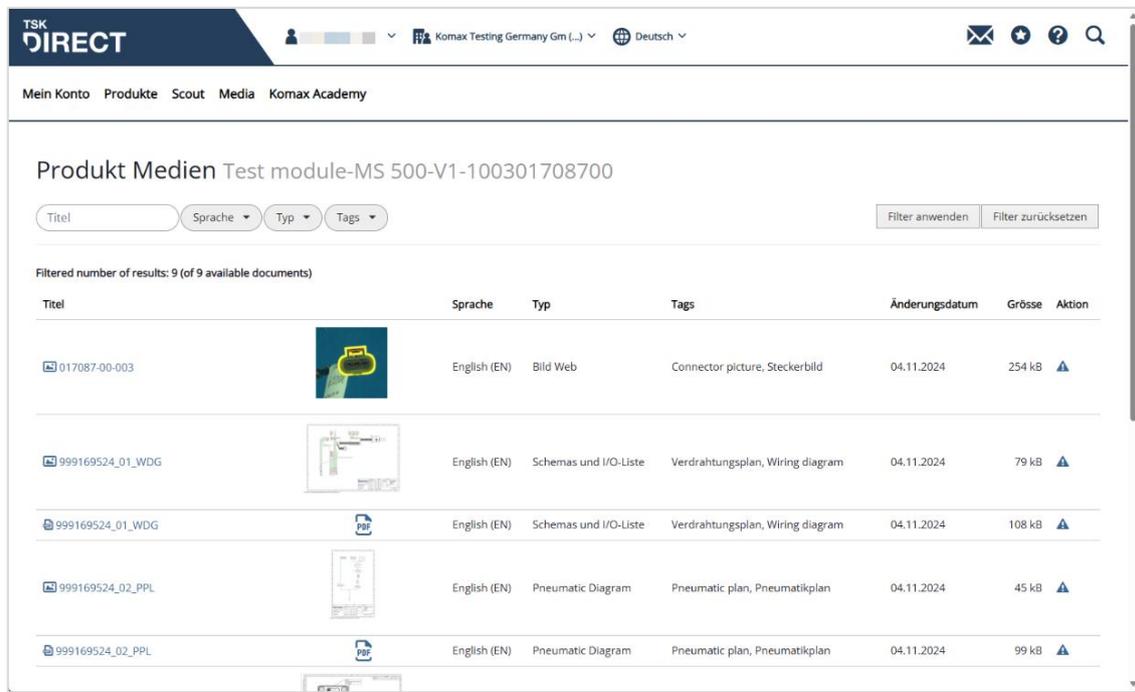


Abbildung 3-22: Übergabedokumentation im TSK Direct von Komax Testing

Bei Komax Testing sind die klassischen IT-Systeme wie Enterprise Resource Planning (ERP), Product Lifecycle Management (PLM) und Product Information Management (PIM) vorhanden, aus denen die Webseite und das Kundenportal mit Inhalten gespeist werden. Weitere Informationen werden von den Dateiservern hinzugezogen. Aktuell werden noch keine Verwaltungsschalen bereitgestellt. Die vorhandene Infrastruktur kann aber als Quellsystemen für einen ETL-Prozesses zur Erzeugung von VWS genutzt werden.

Hervorzuheben ist die mögliche Struktur der VWS, die durch die Art der Produkte von Komax Testing bedingt ist und sich deutlich von den anderen Projektpartnern unterscheidet:

- Systemtypen (als Typ-VWS) analog zur Webseite und für den Produktkatalog
  - *DigitalNameplate* mit Herstellerinformationen und Klassifizierung
  - *TechnicalData* mit allgemeinen technischen Daten (teilweise mit Wertebereichen)
  - *HandoverDocumentation* mit Produktflyer und Handbücher
- Systemkonfiguration (als Typ-VWS) mit produktspezifischen Konfigurationen
  - *DigitalNameplate* mit Herstellerinformationen und der spezifischeren Klassifizierung
  - *TechnicalData* mit der spezifischen Konfiguration
  - *BillOfMaterial* (one down)
  - *HandoverDocumentation* mit spezifischen Zeichnungen und weiteren Dokumenten
- Systeminstanzen (als Instanz-VWS)
  - *DigitalNameplate* mit Seriennummer, Herstelldatum und CE-Kennzeichnung
  - *HandoverDocumentation* mit Prüfprotokollen und Zertifikaten

<sup>23</sup> <https://komaxgroup.com/testing>

Im Kontext des Vorhabens und mit dem Fokus auf den Produktkatalog wurden Systemtypen exemplarisch als VWS erzeugt und in einem automatischen Prozess ausgeleitet. Das genaue Vorgehen ist in Kapitel 3.4.1 erläutert.

### 3.3.1.1 Bildung der Produkthierarchien

Die Organisation der Produktkategorien auf der Website und im TSK Direct - sowohl in den URLs als auch den Navigationsleisten - liefern konkrete Ansatzpunkte für die Abbildung von Produkthierarchien in einem semantischen Produktkatalog.

Abbildung 3-23 zeigt die Navigationsebenen auf der Website von Komax Testing. Tabelle 3-11 stellt die semantische Zuordnung der Produkthierarchie mit Referenz zu Abbildung 3-23 dar.

Referenz	Property in der VWS	Beispiel
1	<i>ManufacturerProductRoot</i>	Testing
2	<i>ManufacturerProductFamily</i>	Low Voltage Testing
3	<i>ManufacturerProductType</i>	TS1500
4	<i>ManufacturerProductDesignation</i>	TS1500 800x600 mm
5	<i>ManufacturerPartNumber / ManufacturerOrderCode</i>	Nicht vorhanden

Tabelle 3-11: Zuordnung der Produkthierarchie bei Komax Testing

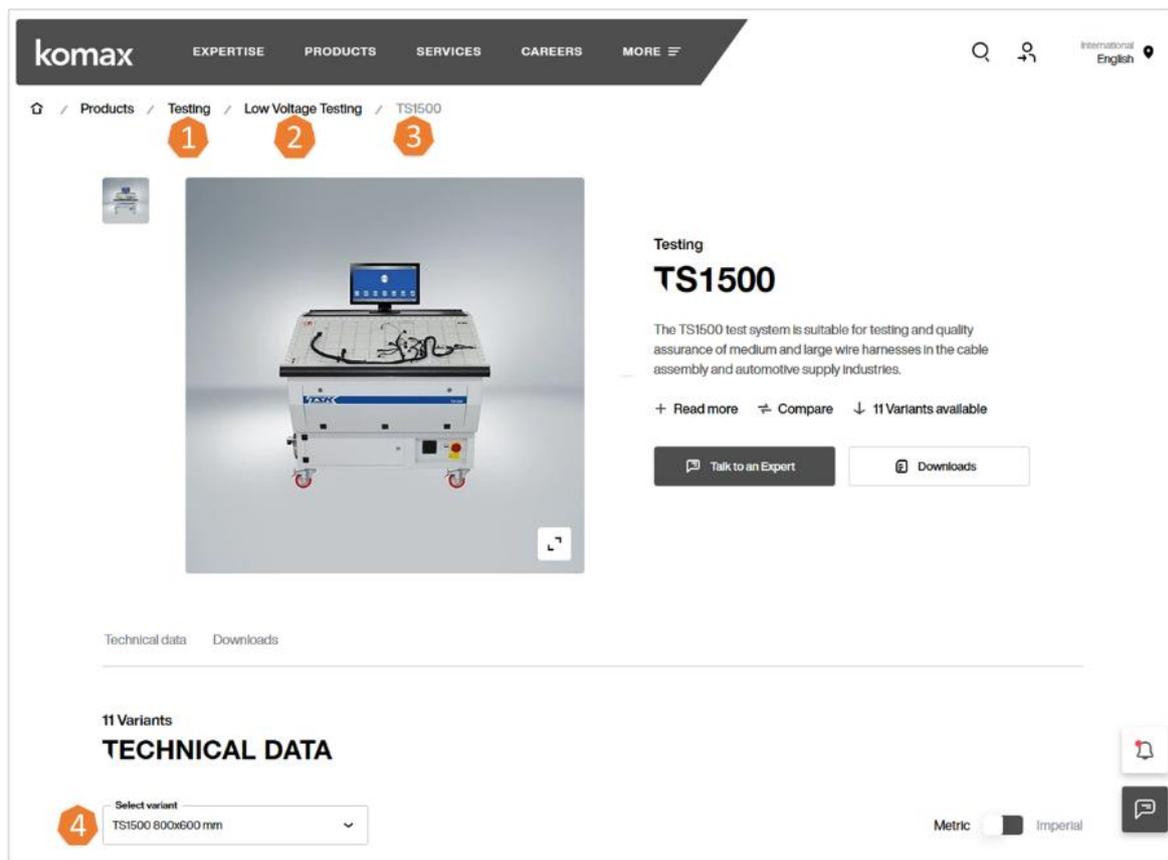


Abbildung 3-23: Produktstruktur Komax Testing

### 3.3.1.2 URL im Produktkatalog

Die folgenden beiden URLs illustrieren den Aufbau der Produkthierarchie auf der Website in den beiden Sprachversionen deutsch und englisch:

- <https://komaxgroup.com/en/products/testing/low-voltage-testing/ts1500>
- <https://komaxgroup.com/en/products/testing/low-voltage-testing/ts1500>

Für die Verwendung in einem semantischen Produktkatalog mit «sprechenden» URLs bietet es sich an die *ManufacturerPartNumber* bzw. *ManufacturerOrderCode* zu ergänzen und die Sprachauswahl mittels Query-Parameter abzubilden.

Durch die *ManufacturerPartNumber* bzw. *ManufacturerOrderCode* wird sichergestellt, dass eine spezifische Produktauswahl möglich ist, ohne zusätzliche Filterung durch manuelle Auswahl über die Variantenauswahl:

- <https://komaxgroup.com/en/products/testing/low-voltage-testing/ts1500/401165>

Die Nutzung von Query-Parametern für die Sprachauswahl würde zudem die Nutzung der URL als spezifische Asset-ID (*specificAssetId*) ermöglichen, da der Query-Parameter in diesem Fall weggelassen werden könnte, die URL somit auf das generische Produkt verweist:

- <https://komaxgroup.com/products/testing/low-voltage-testing/ts1500/401165?lang=en>

Des Weiteren könnte der Pfadbestandteil «products» in den Domänennamen überführt werden, sodass der Pfad genau die Inhalte enthält, die für die eindeutige Produktidentifikation mittels «sprechenden» URLs notwendig sind:

- <https://products.komaxgroup.com/testing/low-voltage-testing/ts1500/401165?lang=en>

### 3.3.1.3 Identifikation in der Verwaltungsschale

Für die Identifikation von Produkten in der VWS sind drei Bestandteile von Bedeutung:

- *IdShort* – als Kurzbezeichner für das Asset
- *aasId* – Als global eindeutiges Identifikationsmerkmal für die spezifische VWS
- *assetId* – Als global eindeutiges Identifikationsmerkmal für das Asset

Die *IdShort* kann im einfachsten Fall direkt aus den Bestandteilen der Produkthierarchie abgeleitet werden. Hierbei ist wichtig Leerzeichen und weitere Sonderzeichen gemäß dem Metamodell der Verwaltungsschale zu vermeiden:

- $\{ManufacturerName\}_{ManufacturerProductFamily}_{ManufacturerArticleNumber}$

*Beispiel:*

- KomaxTesting\_TS1500\_401165

Die *aasId* als global eindeutiges Identifikationsmerkmal für die VWS sollte in Form einer URI oder URL abgebildet und die Empfehlungen aus dem Metamodell der VWS beachtet werden. Es bietet sich an, das Schema nach etablierten Modellen aufzubauen.

*Beispiel:*

- <http://komaxgroup.com/testing/aas/1/0/401165>

Die *globalAssetId* - als global eindeutiges Identifikationsmerkmal für das Asset - kann wiederum einer ähnlichen Struktur folgen, wie die URL auf der Website. Es ist jedoch nicht ratsam die URL der Website direkt zu verwenden, da zukünftige Änderungen diese ungültig machen könnte. Stattdessen bietet es sich an das Schema nach etablierten Modellen aufzubauen. Bei einer Instanz ergänzt bspw. um die Seriennummer.

Für eine VWS vom Typ Instanz kann zudem das heute übliche Typenschild als Hinweisgeber verwendet werden. Die Abbildung 3-24 illustriert den Aufbau eines Typenschilds bei Komax Testing:

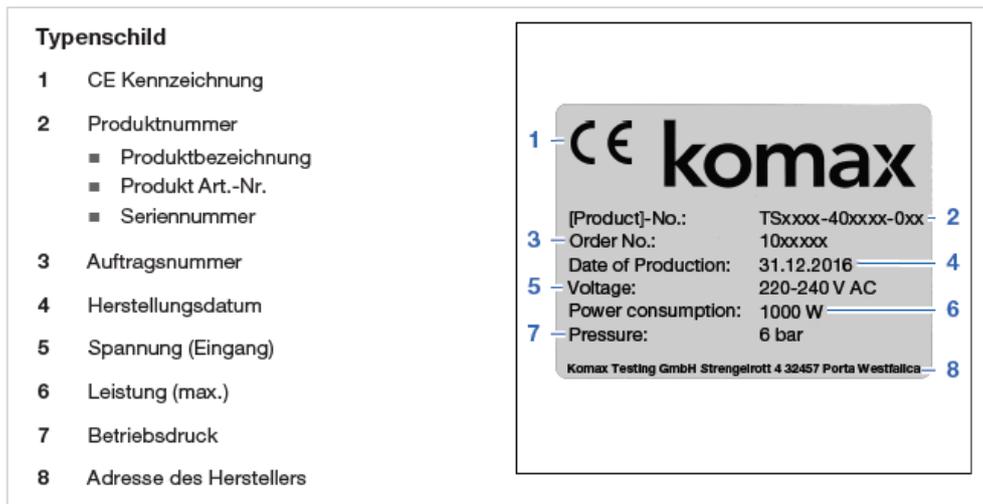


Abbildung 3-24: Definition eines Typenschildes von Komax Testing

Für die *globalAssetId* ist der Punkt (2 - Produktnummer) entscheidend. Diese kann in abgewandelter Form übernommen und als URL dargestellt werden:

- <http://asset.komaxgroup.com/testing/{Produktbezeichnung}/{Produkt Art.-Nr.}/{Seriennummer}>

*Beispiel:*

- Typ: <http://asset.komaxgroup.com/testing/ts1400/401165>
- Instanz: <http://asset.komaxgroup.com/testing/ts1400/401165/001>

Anmerkung: Es wird generell empfohlen keine zählenden Seriennummern als Identifikationsmerkmale zu verwenden, da diese erratbar und somit unter Umständen gegen ungewollten Zugriff geschützt sind. Stattdessen können nach IEC 61406 (*Identification Link*) [14] [15] auch Identifier mit zufälligen Buchstaben- und Zahlenkombinationen zum Einsatz kommen.

### 3.3.2 COROFLEX

Der Workshop wurde am 13. Mai durchgeführt und in einem digitalen Whiteboard dokumentiert. Zuerst wurde der aktuelle Stand bzgl. der Darstellung der Coroflex-Produkte erarbeitet. Aktuell sind die Produkte mit einigen technischen Merkmalen auf der Webseite gelistet. Detaillierte technische Informationen erhält man auf Anfrage durch den zuständigen Vertriebsmitarbeiter.

Bei Coroflex ist das führende System ein ERP-System mit Materialstammdaten, Produktklassifikationen und -Merkmalen. Die technischen Daten für die Kunden werden analog von der Entwicklungsabteilung vorbereitet und verwaltet und an die Kunden weitergegeben. Diese IT-Landschaft macht einen automatisierten Prozess schwierig.

Im Vorhaben wurden die Informationen für einige Produktfamilien (inkl. deren Varianten) in eine Excel-Tabelle überführt. Daraus konnten die Verwaltungsschalen erfolgreich automatisiert erstellt werden.

#### 3.3.2.1 Bildung der Produkthierarchien

Der im Online-Produktkatalog nach aussen sichtbarem Zustand liefert einige Informationen zur Organisation der Produktkategorien, sowohl in den URLs als auch den Navigationshierarchien. Weitere wesentliche Informationsquellen waren ein zur Verfügung gestelltes Excelsheet mit PLM-Properties, Entwürfen der neuen Produktstruktur sowie einer Anzahl Produktdatenblätter.

Abbildung 3-25 zeigt die Navigationsebenen auf der Website von Coroflex. Tabelle 3-12 stellt die semantische Zuordnung der Produkthierarchie mit Referenz zu Abbildung 3-25 dar.

Referenz	Property in der VWS	PLM-Property
1 / 2	<i>ManufacturerProductRoot</i>	External product name / category_name
3	<i>ManufacturerProductFamily</i>	Cable Design / description_short
3 / 5	<i>ManufacturerProductDesignation</i>	Cable description + Conductor size + Temperature Class + Voltage Class
5	<i>ManufacturerProductType</i>	n/a
5	<i>ManufacturerPartNumber / ManufacturerOrderCode</i>	Coroflex Type no.

Tabelle 3-12: Zuordnung der Produkthierarchie bei Coroflex

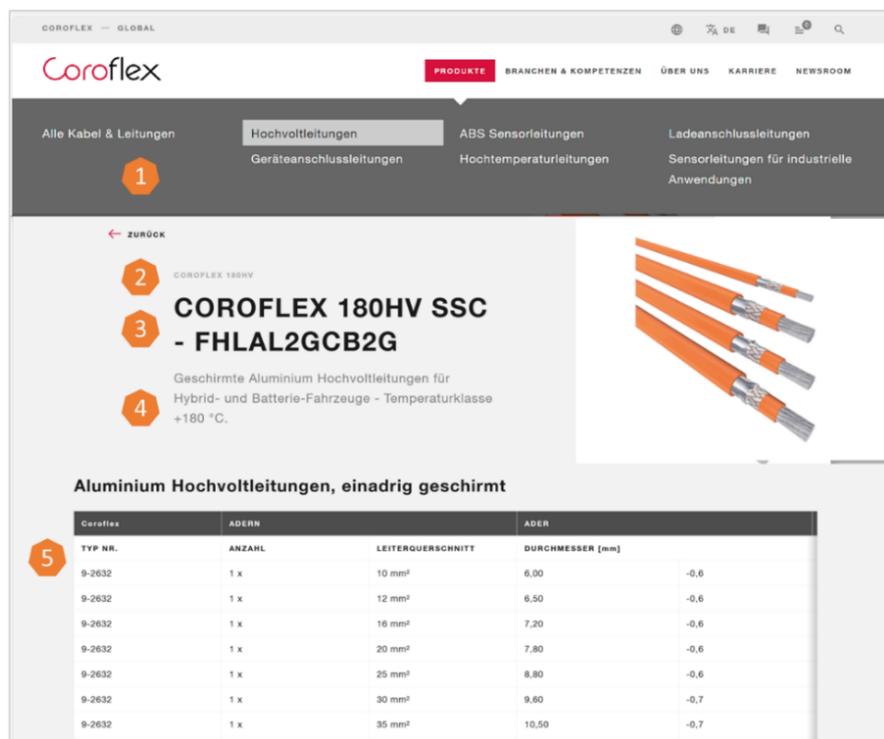


Abbildung 3-25: Produktstruktur COROFLEX

### 3.3.2.2 URL im Produktkatalog

Die folgenden beiden URLs illustrieren den Aufbau der Produkthierarchie auf der Website in den beiden Sprachversionen deutsch und englisch:

- <https://www.coroflex-cable.com/en/high-voltage-cables/coroflex-180hv-ssc-fhlal2gcb2g/>
- <https://www.coroflex-cable.com/de/hochvoltleitungen/coroflex-180hv-ssc-fhlal2gcb2g/>

Bei einer Neuentwicklung des Online-Produktkatalogs wird empfohlen, die Sprachabhängigkeiten in der URL zu beseitigen und eine Linkbildung idealerweise nur mit dem englischsprachigen Produktbezeichner als Identifizierungsbestandteil in der URL und einem Query-Parameter zur Übergabe der gewünschten Anzeigesprache anzustreben, bspw.:

- <https://www.coroflex-cable.com/high-voltage-cables/coroflex-180hv-ssc-fhlal2gcb2g/?lang=en>

Ggf. könnte auch die Verwendung einer Subdomain "products" vorteilhaft sein, um den Online-Produktkatalog über eine eigene Domäne abzugrenzen:

- <https://products.coroflex-cable.com/high-voltage-cables/coroflex-180hv-ssc-fhlal2gcb2g>

### 3.3.2.3 Identifikation in der Verwaltungsschale

Die *idShort* kann im einfachsten Fall direkt aus dem bestehenden Benennungsschema abgeleitet werden, hier noch hinreichend granular ergänzt für die 35mm<sup>2</sup>-Variante des Produkts:

- Coroflex\_180hv\_ssc\_fhlal2gcb2g\_35

Die *aasId* als global eindeutiges Identifikationsmerkmal für die Verwaltungsschale sollte in Form einer URI oder URL abgebildet werden. Es bietet sich an, das Schema unter Verwendung der *idShort* aufzubauen, bspw.:

- [https://aas.coroplast.de/coroflex\\_180hv\\_ssc\\_fhlal2gcb2g\\_35](https://aas.coroplast.de/coroflex_180hv_ssc_fhlal2gcb2g_35)

Die *assetId* als global eindeutiges Identifikationsmerkmal für das Asset kann wiederum einer ähnlichen Struktur folgen, wie die URL auf der Website. Zur Abgrenzung bietet es sich an das Schema mit der Subdomain „asset“ aufzubauen, bspw.

- [https://asset.coroplast.de/coroflex\\_180hv\\_ssc\\_fhlal2gcb2g\\_35](https://asset.coroplast.de/coroflex_180hv_ssc_fhlal2gcb2g_35)

### 3.3.2.4 Properties

Die folgenden COROFLEX-spezifischen Properties wurden im Rahmen der VWS-Erstellung identifiziert und im SM *TechnicalData* - SMC *TechnicalProperties* implementiert, siehe Tabelle 3-13. Die Einheitenreferenz wurde im Regelfall als *ConceptDescription* angelegt, eine etwaige WerteReferenz bei *valueId* hinterlegt.

COROFLEX Property	VEC-Referenz, ergänzen mit <a href="https://ecad.prostep.org/ontologies/2024/03/">https://ecad.prostep.org/ontologies/2024/03/</a>	Einheiten- oder Werte-Referenz
<b>MassInformation</b>	vec#conductorSpecificationMassInformation	<a href="https://qudt.org/vocab/unit/GM-PER-M">https://qudt.org/vocab/unit/GM-PER-M</a>
<b>ConductorSize</b>	vec#conductorSpecificationCrossSectionArea	<a href="https://qudt.org/vocab/unit/Millim2">https://qudt.org/vocab/unit/Millim2</a>
<b>ConductorDiameter</b>	vec#wireElementSpecificationOutsideDiameter	<a href="http://qudt.org/vocab/unit/Millim">http://qudt.org/vocab/unit/Millim</a>
<b>ConductorMaterial</b>	vec#conductorSpecificationMaterial	Tabelle 4-11: IEC Conductor properties
<b>CoreInsulationMaterial</b>	vec#insulationSpecificationMaterial	Tabelle 4-11: IEC Conductor properties
<b>JacketMaterial</b>	vec#insulationSpecificationMaterial	Tabelle 4-11: IEC Conductor properties
<b>TemperatureType</b>	vec#RobustnessClass_AmbientTemperature	
<b>NominalVoltage</b>	vec#NominalVoltage	

Tabelle 3-13: COROFLEX Properties

Es ist anzumerken, dass einige Bestandsdaten nicht sauber auf VEC-Definitionen gemappt werden konnten, da sie nicht immer formal passend definiert oder eindeutig befüllt sind. Es wird in diesen Fällen empfohlen, die Datenmodellierung im PLM- bzw. PIM-System entsprechend nachzujustieren.

### 3.3.3 KOSTAL

Der Workshop wurde am 27. Mai durchgeführt und in einem digitalen Whiteboard<sup>24</sup> dokumentiert. Zuerst wurde der aktuelle Stand bzgl. der Darstellung der KOSTAL-Produkte erarbeitet. Aktuell sind die Produkte mit einigen technischen Merkmalen auf der Webseite gelistet und können gefiltert werden. Detaillierte technische Informationen erhält man nur nach Anmeldung. Basierend darauf wurden ca. 15 VWS für unterschiedliche Produktgruppen erstellt.

Bei KOSTAL ist das führende System ein ERP-System mit Materialstammdaten und Produktklassifikationen. Zudem existiert ein PDM-System. Vom PDM werden die Information in ein PIM exportiert und um Marketing-relevante Informationen und Dokumente angereichert. Diese Daten bilden die Basis für den Produktkatalog auf der Webseite. Die Produktdatenblätter werden von der Produktentwicklung gepflegt und sowohl im PDM als auch im PIM hinterlegt.

Diese IT-Landschaft erlaubt einen automatisierten Prozess für die Erstellung von VWS. Da hochwertige VWS bereits manuell erstellt wurden, wurde auf einen automatisierten Prozess verzichtet.

#### 3.3.3.1 Bildung der Produkthierarchien

Derzeit wird bei KOSTAL der Online-Produktkatalog neu definiert und der aktuelle Zustand kann zwar als veraltet angesehen werden, liefert jedoch einige Informationen zur Organisation der Produktkategorien, sowohl in den URLs als auch den Breadcrumbs, siehe hierfür Tabelle 3-14 und Abbildung 3-26.

Referenz	Property in der VWS	PLM-Property
1/2	ManufacturerProductRoot	n/a
3	ManufacturerProductFamily	n/a
4	ManufacturerProductDesignation	n/a
n/a	ManufacturerProductType	n/a
5	ManufacturerPartNumber/ManufacturerOrderCode	n/a

Tabelle 3-14: KOSTAL: Bildung der Produkthierarchie

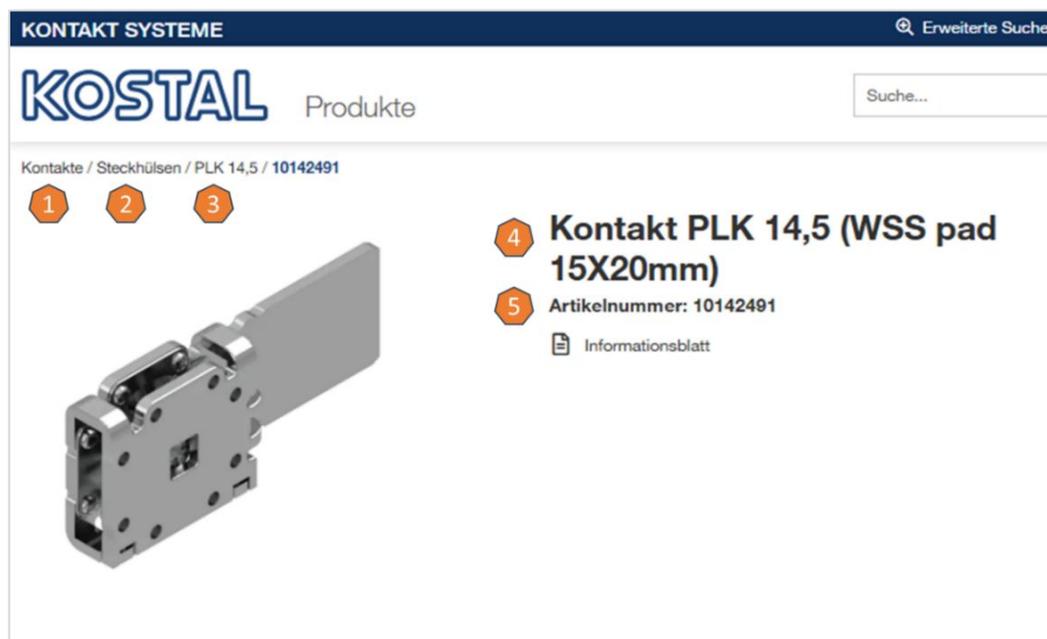


Abbildung 3-26: Produktstruktur KOSTAL

#### 3.3.3.2 URL im Produktkatalog

Die folgenden beiden URLs illustrieren den Aufbau der Produkthierarchie auf der Website in den beiden Sprachversionen deutsch und englisch:

<sup>24</sup> <https://metroretro.io/BOB8XF0YLGSU>

- [https://produktkatalog.kostal-kontakt-systeme.com/web/kostal/en/KOSTAL/1544095543984/Terminal%20PLK%2014%2C5%20%28WSS%20pad%2015X20mm%29/\\$catalogue/KOSTALData/PR/10142491/](https://produktkatalog.kostal-kontakt-systeme.com/web/kostal/en/KOSTAL/1544095543984/Terminal%20PLK%2014%2C5%20%28WSS%20pad%2015X20mm%29/$catalogue/KOSTALData/PR/10142491/)
- [https://produktkatalog.kostal-kontakt-systeme.com/web/kostal/de/KOSTAL/1544095543984/Kontakt%20PLK%2014%2C5%20%28WSS%20pad%2015X20mm%29/\\$catalogue/KOSTALData/PR/10142491/](https://produktkatalog.kostal-kontakt-systeme.com/web/kostal/de/KOSTAL/1544095543984/Kontakt%20PLK%2014%2C5%20%28WSS%20pad%2015X20mm%29/$catalogue/KOSTALData/PR/10142491/)

Es bestehen starke Sprachabhängigkeiten in der URL (de/en, Produktbezeichnung), die Produktbezeichnung enthält Sonderzeichen (%20, %28, %29, %2C) als Pfadbestandteil, der Firmenname kommt als redundanter Pfadbestandteil vor (/KOSTAL/), zudem gibt es Pfadbestandteile ohne semantischen Gehalt (/web/, /1544095543984/, /\$catalogue/KOSTALData/PR/).

Bei einer Neuentwicklung des Online-Produktkatalogs wird empfohlen, diese Punkte zu beseitigen und eine Linkbildung idealerweise nur mit der Artikelnummer als Produkt-Identifikator in der URL anzustreben sowie einem Query-Parameter zur Übergabe der gewünschten Anzeigesprache, bspw.:

<https://produktkatalog.kostal-kontakt-systeme.com/10142491?lang=en>

### 3.3.3.3 Identifikation in der Verwaltungsschale

Die *idShort* sollte am besten direkt aus den bestehenden Benennungsschemas abgeleitet werden, bspw.:

- KOSTAL\_PLK14\_10142491

Die *aas/d* als global eindeutiges Identifikationsmerkmal für die Verwaltungsschale sollte in Form einer URI oder URL abgebildet werden. Es bietet sich an, das Schema unter Verwendung der *idShort* aufzubauen, bspw.:

- [https://aas.kostal.com/KOSTAL\\_PLK14\\_10142491](https://aas.kostal.com/KOSTAL_PLK14_10142491)

Die *asset/d* als global eindeutiges Identifikationsmerkmal für das Asset kann wiederum einer ähnlichen Struktur folgen, wie die URL auf der Website. Zur Abgrenzung bietet es sich an das Schema mit der Subdomain „asset“ aufzubauen, bspw.

- [https://asset.kostal.com/KOSTAL\\_PLK14\\_10142491](https://asset.kostal.com/KOSTAL_PLK14_10142491)

### 3.3.3.4 Properties

Die folgenden KOSTAL-spezifischen Properties wurden im Rahmen der VWS-Erstellung identifiziert und im SM *TechnicalData* - SMC *TechnicalProperties* implementiert, siehe Tabelle 3-15. Die Einheitenreferenz wurde im Regelfall als *ConceptDescription* angelegt, eine etwaige WerteReferenz bei *valued* hinterlegt.

KOSTAL Property	VEC-Referenz, (ergänzen mit <a href="https://ecad.prostep.org/ontologies/2024/03/">https://ecad.prostep.org/ontologies/2024/03/</a> )	Einheiten- oder Werte-Referenz
CONNECT_WIRE_SIZE	vec#terminalCurrentInformationCore-CrossSectionArea	<a href="https://qudt.org/vocab/unit/Millim2">https://qudt.org/vocab/unit/Millim2</a>
CONNECT_TERMINAL_SIZE	vec#terminalTypeNominalSize	<a href="https://qudt.org/vocab/unit/Millim">https://qudt.org/vocab/unit/Millim</a>
CONNECT_SURFACE_LAMINA	vec#materialDescription	-
CONNECT_OPERATING_TEMPERATURE	vec#TemperatureType_OperatingTemperature	<a href="https://qudt.org/vocab/unit/DEG_C">https://qudt.org/vocab/unit/DEG_C</a>
CONNECT_MATING_FORCE_TERMINAL	vec#terminalPairingMatingForce	<a href="https://qudt.org/vocab/unit/N">https://qudt.org/vocab/unit/N</a>
CONNECT_CURRENT_CARRYING_CAPACITY_AT_80DEGREE	vec#terminalCurrentInformationCurrentRange	<a href="https://qudt.org/vocab/unit/A">https://qudt.org/vocab/unit/A</a>
CONNECT_UNMATING_FORCE_TERMINAL	vec#terminalPairingUnmatingForce	<a href="https://qudt.org/vocab/unit/N">https://qudt.org/vocab/unit/N</a>
CONNECT_TECH	vec#WireReceptionSpecification	vec#WireReceptionType_Crimp
Contact_family	vec#TerminalType	
CONNECT_NOMINAL_CAVITY_DIAMETER	vec#cavitySealSpecificationWireElementOutsideDiameter	<a href="https://qudt.org/vocab/unit/Millim">https://qudt.org/vocab/unit/Millim</a>
Colour	vec#generalTechnicalPartSpecificationColorInformation	Tabelle 4-10: IEC Colour properties
cont_house_type	vec#housingComponentHousingSpecification	
CONNECT_WIRE_OUTER_DIAMETER	vec#wireElementSpecificationOutsideDiameter	<a href="https://qudt.org/vocab/unit/Millim">https://qudt.org/vocab/unit/Millim</a>
Qty_pins	vec#slotSpecificationNumberOfCavities	0112/2///62720#UAD146
CONNECT_CODE	vec#variantCodeDescription	-
CONNECT_SEALING_PRINCIPLE	vec#SlotSealingType	vec#SlotSealingType_SingleSealing
CONNECT_CPA	vec#connectorHousingSpecification-ConnectorPositionAssuranceType	-
CONNECT_VOLTAGE_LEVEL_MAX	vec#NominalVoltage_48V	-
CONNECT_PREASSEMBLED	vec#cavityHasIntegratedTerminal	-

Tabelle 3-15: KOSTAL *TechnicalProperties*

### 3.4 AP 12.4 - Verwaltungsschalengenerierung

Das Arbeitspaket „Verwaltungsschalengenerierung“ beschäftigte sich mit der Implementierung der Erzeugung und Verwaltung von Typ-Verwaltungsschalen. Hierzu wurden eine Reihe von Referenzprodukten in Verwaltungsschalen modelliert, die im Produktkatalog-Demonstrator der ARENA2036 dargestellt werden.

#### 3.4.1 Komax Testing

Die VWS von Komax Testing wurden generativ auf Basis von ERP-Daten erzeugt und mittels REST API auf dem Demo-Verwaltungsschalen-Server abgelegt.<sup>25</sup>

Um eine semantische Durchgängigkeit zu gewährleisten, wurden die erforderlichen Daten aus den Teilmodellen der IDTA abgeleitet und entsprechende Parameter im ERP-System identifiziert. Die identifizierten Parameter wurden anschließend in einem Export-Prozess verfügbar gemacht und mittels eines entsprechenden Transformators in Teilmodelle und Verwaltungsschalen übersetzt. Der Fokus lag hierbei auf Typ-Verwaltungsschalen von Prüfmaschinen.

Eine wesentliche Erkenntnis war die Notwendigkeit für einen Transformator, der die Daten in der Verwaltungsschale stufenweise anreichern kann, um zu ermöglichen, dass Verwaltungsschalen zunächst mit unvollständigen Daten aufgebaut und anschließend mit weiteren Daten ergänzt oder aktualisiert werden können.

Auf dieser Grundlage können auch nachträgliche Korrekturen an der semantischen Struktur von Verwaltungsschalen mit geringem Aufwand durchgeführt werden. Beispielsweise für die Korrektur von Merkmalen, die in einem ersten Durchlauf fehlerhaft transformiert wurden.

Um die Bereitstellung von unvollständigen oder fehlerhaften Verwaltungsschalen auszuschließen, wurde zudem eine zweistufige Veröffentlichungsstrategie gewählt. In der ersten Stufe werden die Verwaltungsschale auf einer geschlossenen Testumgebung erzeugt. Diese kann mit geringem Aufwand verworfen und neu erzeugt werden. Erst in der zweiten Stufe werden die Verwaltungsschalen auf die Demo-Umgebung ausgeleitet.

In Abbildung 3-27 sind exemplarisch einige der erzeugten Verwaltungsschalen auf der Demo-Umgebung <https://marketplace.arena2036.app/marketplace/catalog?manufacturer=KOMAX> dargestellt.

Image	Manufacturer name	Manufacturer product designation	Asset ID	AAS ID
	Komax Testing Germany GmbH	TS1600 2400x850MM CT 90-260V		<a href="https://www.komaxgroup.com/testing/aas/1/0/401107">https://www.komaxgroup.com/testing/aas/1/0/401107</a>
	Komax Testing Germany GmbH	TS1600 4800x850MM CT 90-260V		<a href="https://www.komaxgroup.com/testing/aas/1/0/401109">https://www.komaxgroup.com/testing/aas/1/0/401109</a>
	Komax Testing Germany GmbH	TS1700 1200x1200MM CT 90-260V		<a href="https://www.komaxgroup.com/testing/aas/1/0/401110">https://www.komaxgroup.com/testing/aas/1/0/401110</a>
	Komax Testing Germany GmbH	TS1500 1200x600 WEE-Tester		<a href="https://www.komaxgroup.com/testing/aas/1/0/401132">https://www.komaxgroup.com/testing/aas/1/0/401132</a>
	Komax Testing Germany GmbH	TS1500 800x600 ET 90-260V		<a href="https://www.komaxgroup.com/testing/aas/1/0/401165">https://www.komaxgroup.com/testing/aas/1/0/401165</a>

Abbildung 3-27: Verwaltungsschalen von Komax Testing auf der Demo-Umgebung

<sup>25</sup> <https://vws4ls.dev.mnestix.xitaso.net/de/marketplace/catalog?manufacturer=KOMAX>

### 3.4.2 KOSTAL

Die [KOSTAL-Verwaltungsschalen](#)<sup>26</sup> in Abbildung 3-28 wurden manuell auf Basis von PLM-Daten erzeugt.

Image	Manufacturer name	Manufacturer product designation
	KOSTAL, Kostal Systeme GmbH & Co. KG	
	KOSTAL, Kostal Systeme GmbH & Co. KG	Receptacle DUK 1.2 SW6
	KOSTAL, Kostal Systeme GmbH & Co. KG	Terminal Header 180° K1 0.4
	KOSTAL, Kostal Systeme GmbH & Co. KG	Receptacle housing 130
	KOSTAL, Kostal Systeme GmbH & Co. KG	Receptacle MKX 1.2 ELA 5m
	KOSTAL, Kostal Systeme GmbH & Co. KG	
	KOSTAL, Kostal Systeme GmbH & Co. KG	Tab header 3 way
	KOSTAL, Kostal Systeme GmbH & Co. KG	Receptor cap 19P 25 mm²
	KOSTAL, Kostal Systeme GmbH & Co. KG	Receptor cap 20P 25 mm²
	KOSTAL, Kostal Systeme GmbH & Co. KG	Tab housing 18X 2.8 2 way SW5
	KOSTAL, Kostal Systeme GmbH & Co. KG	Tab header
	KOSTAL, Kostal Systeme GmbH & Co. KG	Single Wire Sealing PKX 14.5

Abbildung 3-28 : Kostal Produktverwaltungsschalen

<sup>26</sup> <https://mnestix.arena2036.app/catalog?manufacturer=KOSTAL>

### 3.4.3 COROFLEX

Die [Coroflex-Verwaltungsschalen](#)<sup>27</sup> wurden exemplarisch auf Grundlage existierender Daten erstellt. Dafür wurde ein manueller Export nach Excel aus dem bestehenden Produktdaten System von Coroflex erstellt, was den "Extract" Schritt repräsentiert. Diese Daten wurden dann zu einer JSON-Datei umgewandelt (Transform) um am Ende in das System geladen zu werden (Load). Die Grundlage für die zu erstellenden Verwaltungsschalen wurde vorab für einen Produkttyp händisch erstellt, um dann die Daten aus der JSON zu verknüpfen.

Ein typischer Prozess um mittels Eclipse Mnestix automatisiert Verwaltungsschalen zu erstellen sieht wie folgt aus. Zuerst müssen die Daten aus heterogenen Datenquellen in ein homogenes Datenformat übertragen werden. Hierfür können gängige ETL-Tools verwendet werden. Exemplarisch wurde hierfür der Export nach Excel und die Transformation in eine JSON-Datei verwendet. Die Transformation nach JSON wird verwendet, da die meisten Objekte nach JSON serialisierbar sind und somit eine gute Grundlage für das Laden und Erstellen der Verwaltungsschalen bildet. Diese JSON kann dann in den VWS-Generator geladen werden über eine HTTP/REST Schnittstelle. Der VWS-Generator [kann auf Docker gefunden werden](#). Zusammen mit einem vorher erstellten Blueprint werden dann die Verwaltungsschalen erstellt. Eine Übersicht ist in der folgenden Abbildung 3-29 zu sehen:

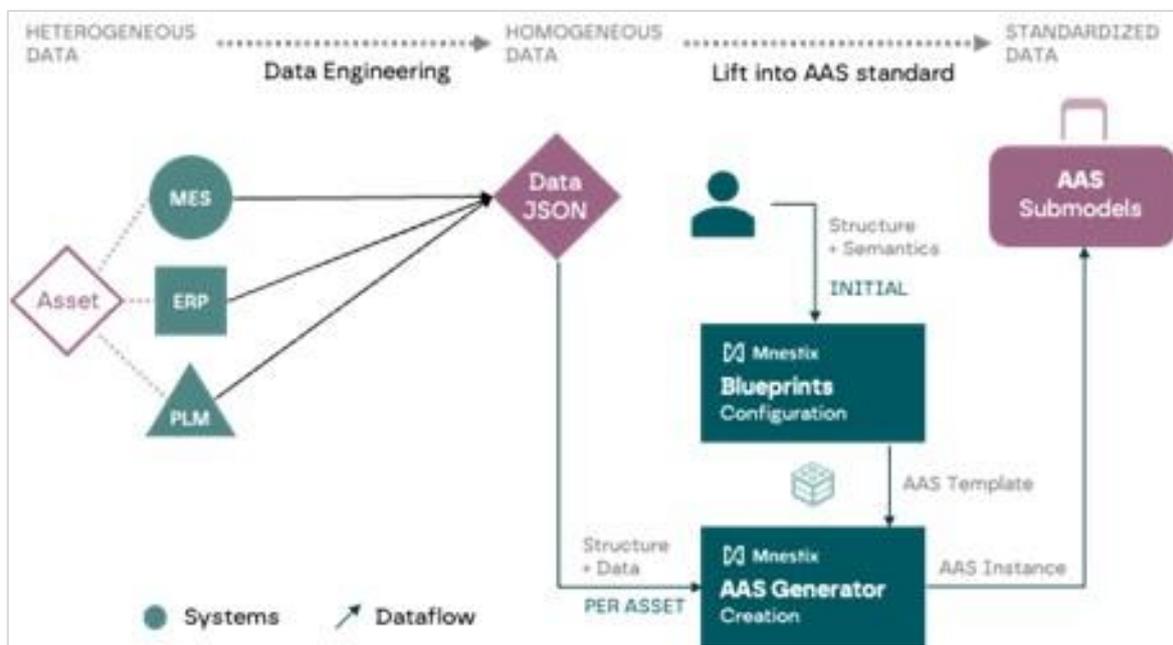


Abbildung 3-29: Mnestix VWS-Generator

Folgend sind Ausschnitte der verwendeten Excel Liste, in Abbildung 3-30 dargestellt, sowie der daraus erstellten JSON-Dateien, in Abbildung 3-31 dargestellt zu sehen, die für die Generierung der VWS verwendet worden sind:

ID-Webseite	Internal remarks	External product name	Coroflex Type no.	Cable description	Conductor size	Application	Cable Design	Customer Ident	Conductor material	Core Insulation	Jacket Material	Cable diameter
1001	HV, Kupfer, ungeschirmt, T180	Coroflex High Voltage Cable	9-2652	FHL2G	10 mm <sup>2</sup>	LV216	Single Core	LV216-1	Copper	Silicone	Silicone	6,2 mm (± 0,3)
1001	HV, Kupfer, ungeschirmt, T180	Coroflex High Voltage Cable	9-2652	FHL2G	16 mm <sup>2</sup>	LV216	Single Core	LV216-1	Copper	Silicone	Silicone	8,0 mm (± 0,3)
1001	HV, Kupfer, ungeschirmt, T180	Coroflex High Voltage Cable	9-2652	FHL2G	25 mm <sup>2</sup>	LV216	Single Core	LV216-1, VW: N 108 957	Copper	Silicone	Silicone	9,7 mm (± 0,3)
1001	HV, Kupfer, ungeschirmt, T180	Coroflex High Voltage Cable	9-2652	FHL2G	35 mm <sup>2</sup>	LV216	Single Core	LV216-1, VW: N 108 174	Copper	Silicone	Silicone	10,7 mm (± 0,3)
1001	HV, Kupfer, ungeschirmt, T180	Coroflex High Voltage Cable	9-2652	FHL2G	50 mm <sup>2</sup>	LV216	Single Core	LV216-1, VW: N 108 958	Copper	Silicone	Silicone	12,8 mm (± 0,4)
1001	HV, Kupfer, ungeschirmt, T180	Coroflex High Voltage Cable	9-2652	FHL2G	70 mm <sup>2</sup>	LV216	Single Core	LV216-1, VW: N 108 959, BMW: 5A320C9	Copper	Silicone	Silicone	14,7 mm (± 0,4)
1001	HV, Kupfer, ungeschirmt, T180	Coroflex High Voltage Cable	9-2652	FHL2G	95 mm <sup>2</sup>	LV216	Single Core	LV216-1, VW: N 108 960, BMW: 5A320D0	Copper	Silicone	Silicone	16,9 mm (± 0,5)
1001	HV, Kupfer, ungeschirmt, T180	Coroflex High Voltage Cable	9-2652	FHL2G	120 mm <sup>2</sup>	LV216	Single Core	LV216-1	Copper	Silicone	Silicone	19,0 mm (± 0,5)
1002	HV, Kupfer, geschirmt, T180	Coroflex High Voltage Cable	9-2611	FHLR2GCB2G	4,00 mm <sup>2</sup>	LV216	Shielded Single Core Cab	LV216-2, Mercedes: C 51 / 6.14	Copper	Silicone	Silicone	5,6 mm (± 0,2)
1002	HV, Kupfer, geschirmt, T180	Coroflex High Voltage Cable	9-2611	FHLR2GCB2G	6,0 mm <sup>2</sup>	LV216	Shielded Single Core Cab	LV216-2, Mercedes: C 51 / 8.14, BMW: 5A	Copper	Silicone	Silicone	6,3 mm (± 0,2)
1002	HV, Kupfer, geschirmt, T180	Coroflex High Voltage Cable	9-2611	FHLR2GCB2G	10 mm <sup>2</sup>	LV216	Shielded Single Core Cab	LV216-2, Mercedes: C 51 / 9.14	Copper	Silicone	Silicone	8,5 mm (± 0,3)
1002	HV, Kupfer, geschirmt, T180	Coroflex High Voltage Cable	9-2611	FHLR2GCB2G	16 mm <sup>2</sup>	LV216	Shielded Single Core Cab	LV216-2, Mercedes: C 51 / 10.14, Daimler	Copper	Silicone	Silicone	9,9 mm (± 0,3)
1002	HV, Kupfer, geschirmt, T180	Coroflex High Voltage Cable	9-2611	FHLR2GCB2G	25 mm <sup>2</sup>	LV216	Shielded Single Core Cab	LV216-2, Mercedes: C 51 / 11.14, VW: N 10	Copper	Silicone	Silicone	11,9 mm (± 0,3)

Abbildung 3-30: Ausschnitt aus Excel Datenquelle

<sup>27</sup> <https://mnestix.arena2036.app/marketplace/catalog?manufacturer=COROFLEX>

```

"aas_id": "https://aas.coroplast.de/COROFLEX_9_2611_FHLR2GCB2G_35",
"asset_id": "https://asset.coroplast.de/9-2611-35",
"id_short": "COROFLEX_9_2611_FHLR2GCB2G_35",
"id_website": "1002",
"submodel_ids": {
  "HandoverDocumentation": "https://aas.coroplast.de/sm/9_2611_35/HandoverDocumentation",
  "DigitalNameplate": "https://aas.coroplast.de/sm/9_2611_35/DigitalNameplate",
  "TechnicalData": "https://aas.coroplast.de/sm/9_2611_35/TechnicalData"
},
"eclass": "44040201",
"markings": [
  "Daimler Truck: A0025460501",
  "LV216-2",
  "BMW: 9 327 162",
  "Mercedes: C51 / 12.14",
  "VW: N 107 755"
],
"internal_remarks": "HV, Kupfer, geschirmt, T180",
"auswahanwendungen": "",
"external_product_name": "Coroflex High-Voltage Cable",
"coroflex_type_no": "9-2611",
"cable_description": "FHLR2GCB2G",
"cable_uri": "https://www.coroflex-cable.com/de/hochvoltleitungen/coroflex-180hv-ssc-fhldr2gcb2g/",
"thumbnail": "https://www.coroflex-cable.com/fileadmin/_processed_/1/7/csm_COROFLEX-High-Voltage-Cable-Shielded-Single",
"name": "Coroflex High-Voltage Cable",
"type": "9-2611",
"description": "FHLR2GCB2G",
"product_designation": "FHLR2GCB2G 35 mm²/ 0.21 T180 0.6/1.0 kV",
"manufacturer_statement": "Coroflex Single Core High-Voltage Cable, Copper, shielded",
"conductor_size": "35 mm²",
"conductor_size_value": "35",
"application": {

```

Abbildung 3-31: Ausschnitt aus verwendeter JSON-Datei

### 3.5 AP 12.5 - Integration und Bereitstellung

Das Arbeitspaket „Integration und Bereitstellung“ beschäftigte sich mit der Integration des Deployment und Monitoring des Typ-Verwaltungsschalen-Generators und Produktkatalogs in die jeweilige IT-Infrastruktur.

Hierzu wurde bei den Projektpartnern im Rahmen jeweils eines Workshops eine Bestandsaufnahme der internen IT-Architektur analysiert und Lösungsvorschläge zur VWS-Implementierung und webbasierten Bereitstellung erarbeitet.

In Bezug auf TP 14 „Demonstrator“ wurde das Open-Source-Tool Mnestix weiterentwickelt, um eine Produktkatalog-VWS-Infrastruktur entsprechend navigieren zu können.

### 3.5.1 Mnestix-Browser

Im Zuge des Teilprojekts 12 erfolgte die begleitende Weiterentwicklung des OpenSource-VWS-Viewers „Mnestix“<sup>28</sup> um mit für Produktkataloge üblichen Such- und Filterfunktionen, insbesondere die Weiterentwicklung einzelner Viewer-Features zur optimalen Darstellung der Submodelle „*TechnicalData*“, „*HandoverDocumentation*“ und „*DigitalNameplate*“ in einem ansprechenden Produktkatalog-Erscheinungsbild.

Im Entwurfsprozess, siehe Abbildung 3-32 bis Abbildung 3-34 wurden in diversen Anforderungsworkshops UI-Entwürfe und ein klickbarer Prototyp erstellt, um die Designideen und Workflows zu sammeln und zu verifizieren:

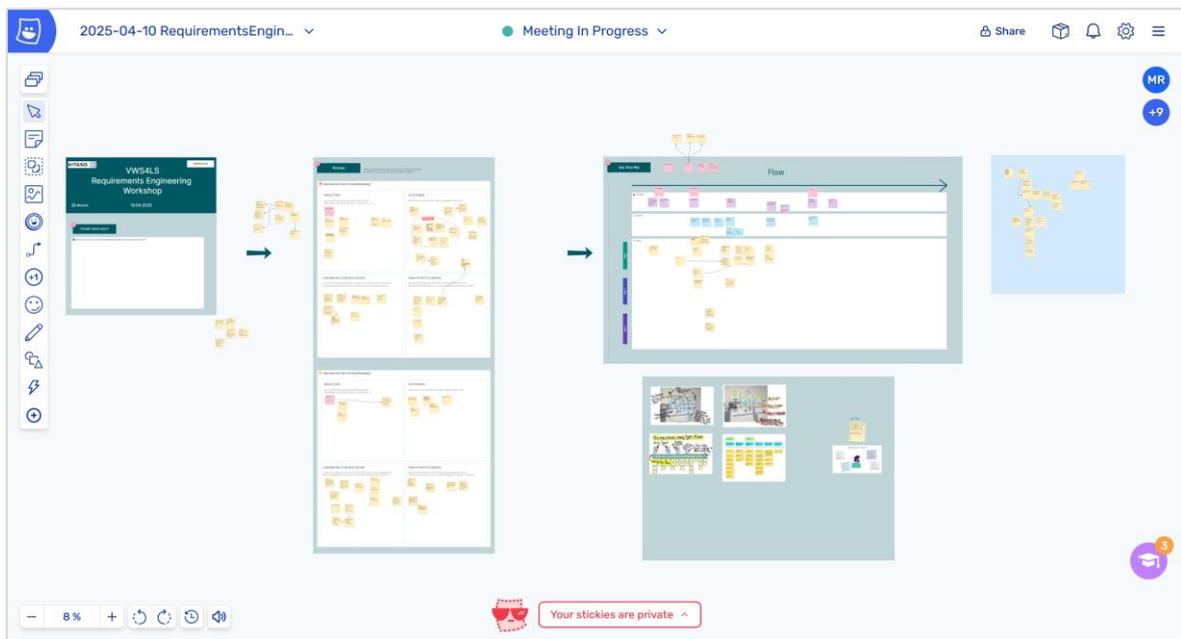


Abbildung 3-32: Anforderungsworkshop

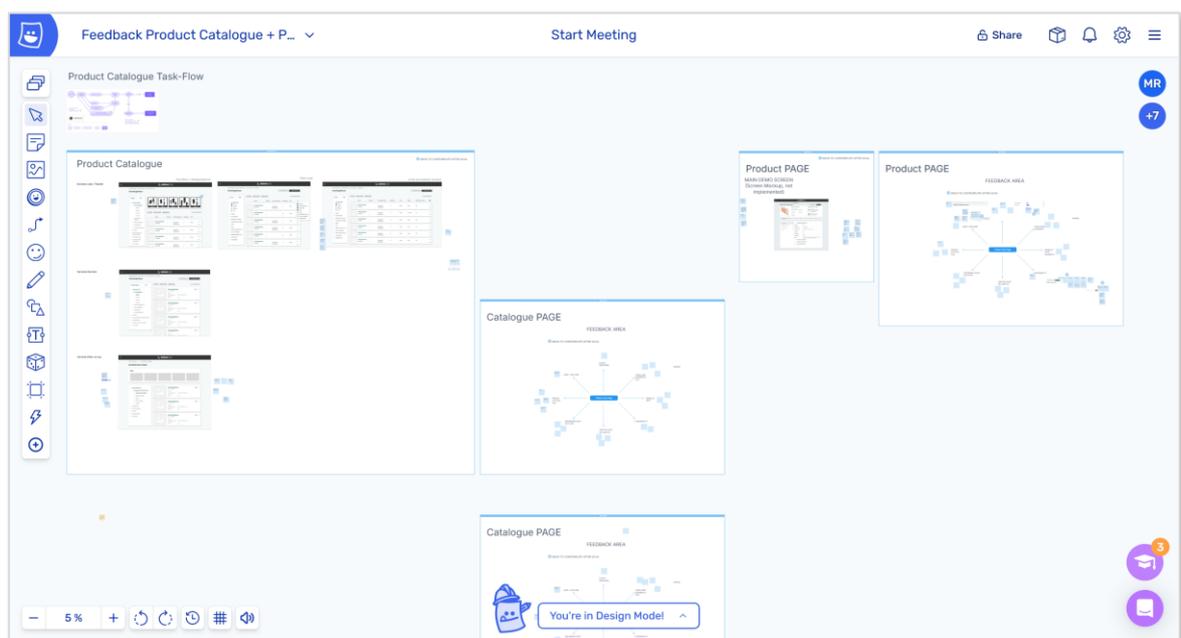


Abbildung 3-33: Feedbackworkshop

<sup>28</sup> <https://github.com/eclipse-mnestix/mnestix-browser>

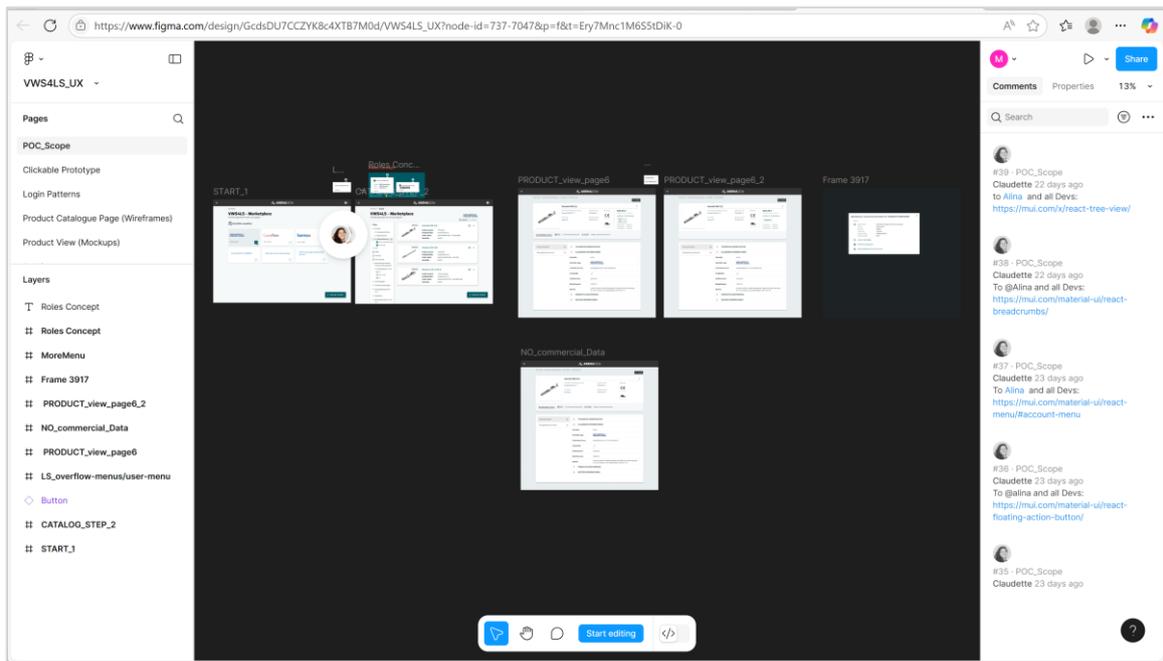


Abbildung 3-34: Figma Demonstrator

Dabei wurde die folgenden grundlegenden Dialoge und Workflows als Anforderungssammlung für die Implementierung festgelegt.

### 3.5.1.1 Produktkatalog-Auswahl

In diesem Dialog als sog. „Landing Page“ sollte entweder direkt ein Herstellerkatalog ausgewählt oder per Chatbot-Funktion eine katalogübergreifende Produktsuche durchgeführt werden, siehe Abbildung 3-35.

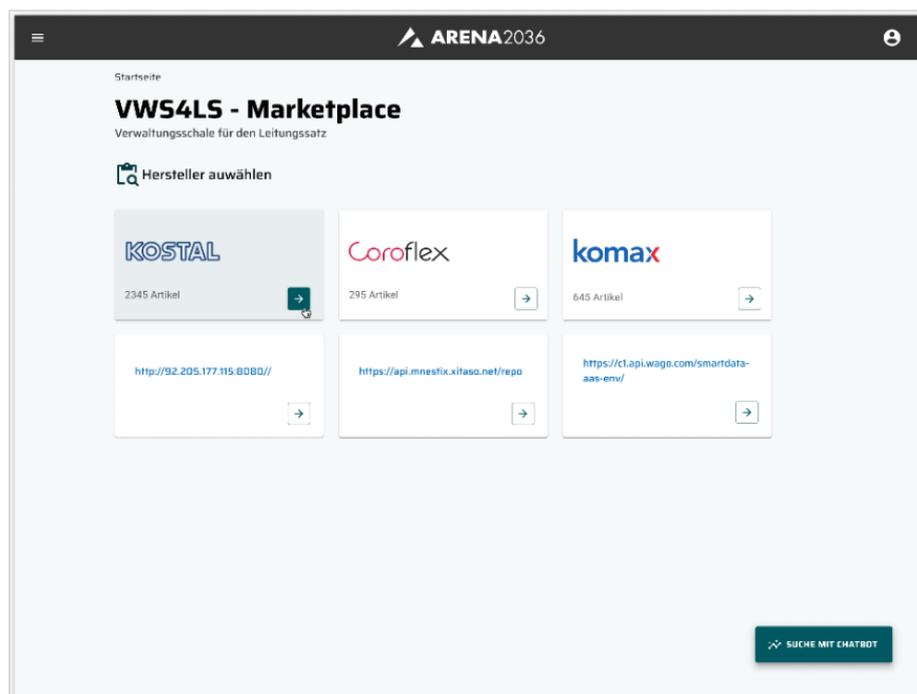


Abbildung 3-35: Mnestix Auswahl Produktkatalog

### 3.5.1.2 Produktfilterung

In diesem Dialog sollte innerhalb des ausgewählten Herstellerkatalogs mit detaillierten Filtern sowie Tabellensortierung die Produktsuche gezielt verfeinert werden, siehe Abbildung 3-36.

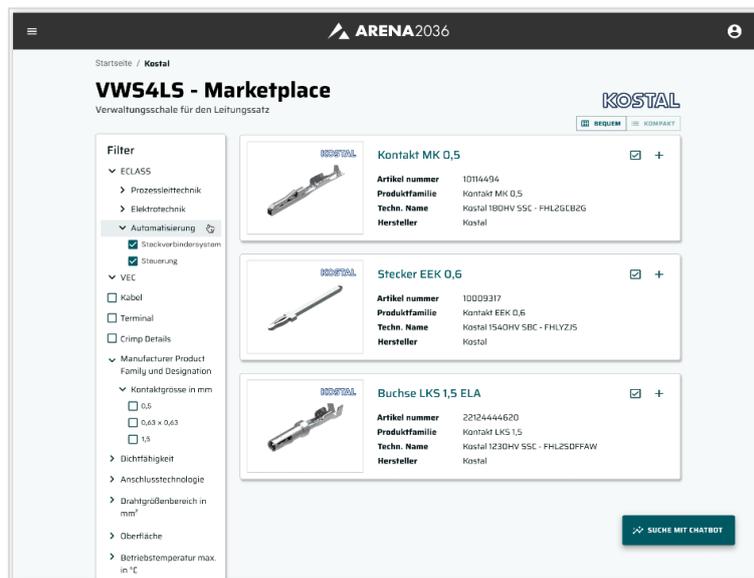


Abbildung 3-36: Mnestix Produktfilterung

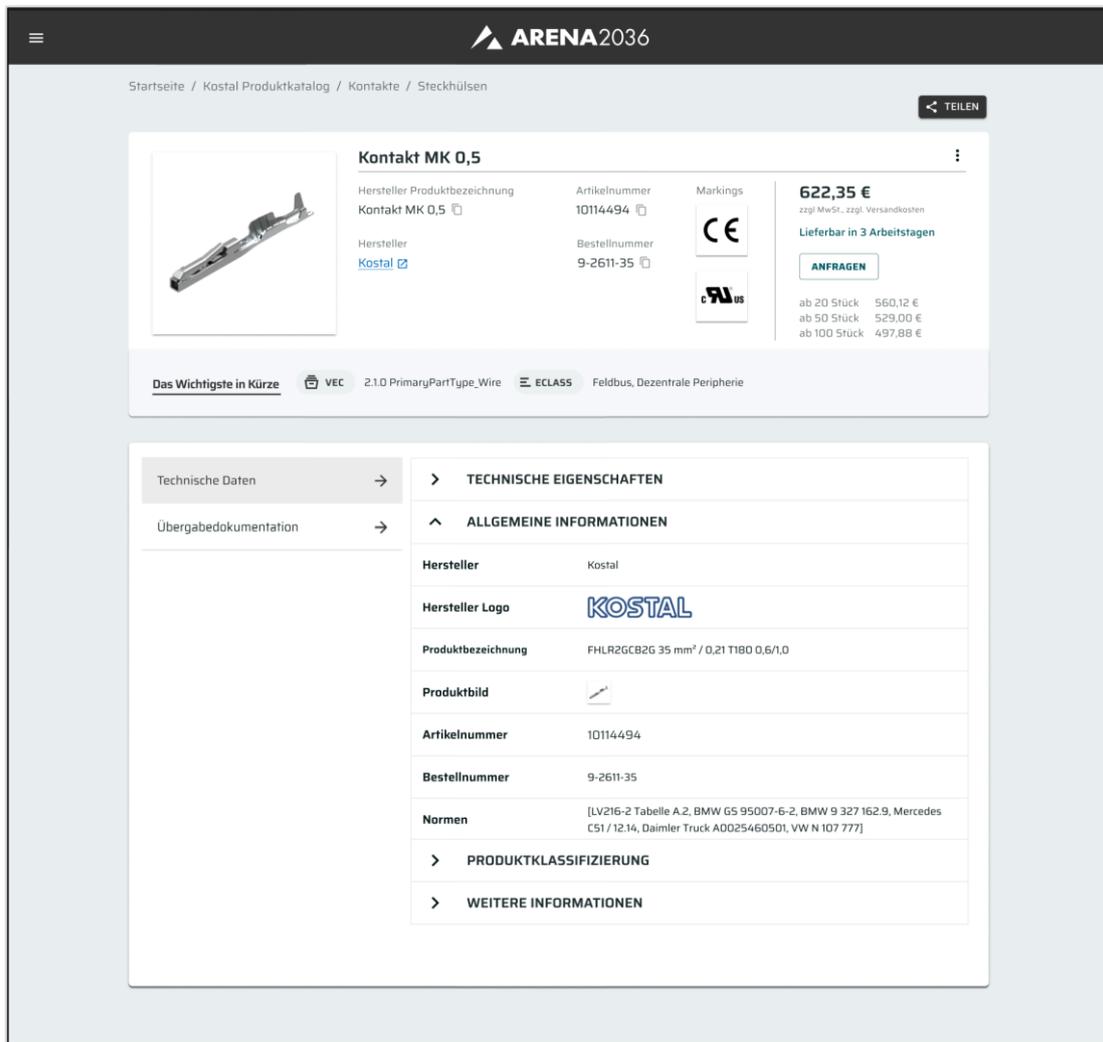
Der Sourcecode zur Filterfunktionalität liegt unter <https://github.com/mnestix/mnestix-searcher/tree/main/MnestixSearcher>. Dabei wird auf Anforderung, bspw. einmal pro Tag, ein beim Deployment spezifiziertes VWS Repository indexiert. Die Indexierung kann über [eine HTTP/REST Schnittstelle](#) angesprochen werden. Nach der Indexierung können die indexierten Verwaltungsschalen [mittels GraphQL durchsucht werden](#). Diese GraphQL Schnittstelle benutzt Eclipse Mnestix Browser um die Filter zu realisieren.

Dieser Ansatz wurde gewählt um die technische Umsetzbarkeit von Filtern basierend auf der Verwaltungsschale zu zeigen. Im nächsten Schritt kann die GraphQL Query erweitert werden um die [Asset Administration Shell Query Language](#)<sup>29</sup> [21] zu realisieren.

<sup>29</sup> <https://industrialdigitaltwin.io/aas-specifications/IDTA-01002/v3.1/query-language.html>

### 3.5.1.3 Produktanzeige

In diesem Dialog sollte das final ausgewählte Produkt, seine kommerziellen Informationen angezeigt und die Interaktionselemente zu einem e-Shop bereitgestellt werden, siehe Abbildung 3-37.



The screenshot shows the product page for 'Kontakt MK 0,5' in the ARENA2036 system. The page includes a product image, a price of 622,35 € (including shipping), and a delivery time of 3 working days. Below the product details, there is a table with technical and general information.

TECHNISCHE EIGENSCHAFTEN	
<b>ALLGEMEINE INFORMATIONEN</b>	
Hersteller	Kostal
Hersteller Logo	
Produktbezeichnung	FHLR2GCB2G 35 mm <sup>2</sup> / 0,21 T180 0,6/1,0
Produktbild	
Artikelnummer	10114494
Bestellnummer	9-2611-35
Normen	[LV216-2 Tabelle A.2, BMW G5 95007-6-2, BMW 9 327 162.9, Mercedes C51 / 12.14, Daimler Truck A0025460501, VW N 107 777]
<b>PRODUKTKLASSIFIZIERUNG</b>	
<b>WEITERE INFORMATIONEN</b>	

Abbildung 3-37: Mnestix Produktanzeige

Davon abgeleitet - jedoch nicht vollständig identisch aufgrund der Ressourcensituation - wurde die Implementierung erstellt, welche auf dem ARENA236-Demonstrator ausgebracht wurde und deren Bedienung im Kapitel 4.6 beschrieben ist.

Für die Abbildung der kommerziellen Informationen (siehe Abbildung 3-37: Mnestix Produktanzeige, oben rechts) ist die Definition eines neuen Teilmodells notwendig. Hierzu wurde vor Abschluss des Projekts ein Beitrag „[Utilization of Asset Administration Shells for e-Commerce Applications](#)“<sup>30</sup> bei der „30th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)“ eingereicht. Die Annahme oder Ablehnung war zum Zeitpunkt der Beendigung des Projekts noch nicht bekannt.

<sup>30</sup> <https://github.com/VWS4LS/vws4ls-subproject-results/upload/main/TP12>

### 3.5.2 Deployment ARENA2036

Die AAS- und Mnestix-Serverinfrastruktur soll im ARENA2036-X-Netzwerk deployt werden (siehe Abbildung 3-38), mit jeweils separaten Repositories für Coroflex, Kostal und Komax Testing. Ein weiteres Repository „vws4ls“ soll als interne Experimentierplattform dienen. Das aktuelle Docker Compose für Mnestix liegt unter:

<https://github.com/eclipse-mnestix/mnestix-browser/blob/main/compose.yml>

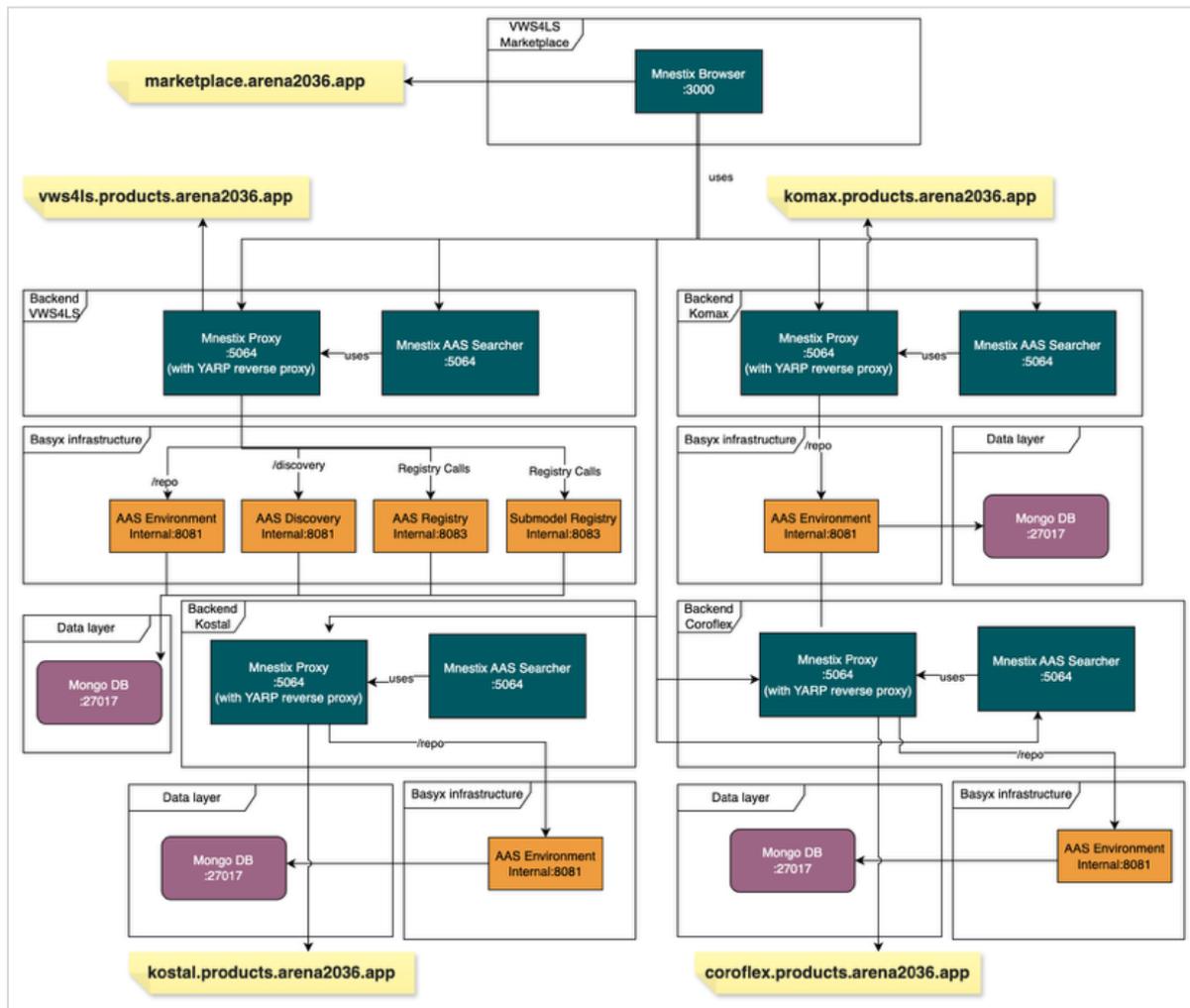


Abbildung 3-38: ARENA-X Deployment Architektur

#### 3.5.2.1 Domänenbildung

Top-Level-Domains (TLDs) unterliegen unterschiedlichen DNS- und Sicherheitsanforderungen: Bspw. ist „**app**“ in der sog. HTTP Strict Transport Security<sup>31</sup>-Preload-Liste enthalten, was bedeutet, dass moderne Browser automatisch eine sichere Verbindung erzwingen und SSL-Zertifikate benötigt werden. Dies ist etwa bei „**tech**“ nicht der Fall.

Als Domainname wurde „**arena2036.app**“ reserviert, was flexibel mit Subdomänen erweitert werden kann. Bis ein internes Hosting im ARENA-X-Kontext erfolgen kann, soll wie folgt umgeleitet werden, siehe Tabelle 3-16:

<sup>31</sup>[https://de.wikipedia.org/wiki/HTTP\\_Strict\\_Transport\\_Security](https://de.wikipedia.org/wiki/HTTP_Strict_Transport_Security)

Subdomäne	Anwendung	Wird umgeleitet nach
<a href="https://arena2036.app">https://arena2036.app</a>	Default Landing Page (HTML-Infoseite mit weiterführenden Links)	arena2036.de?
<a href="https://vws4ls.arena2036.app">vws4ls.arena2036.app</a>	Landing Page für vws4ls	<a href="https://vws4ls.dev.mnestix.xitaso.net/de/list">https://vws4ls.dev.mnestix.xitaso.net/de/list</a>
<a href="https://marketplace.arena2036.app">marketplace.arena2036.app</a>	Landing Page für Mnestix Marketplace	<a href="https://vws4ls.dev.mnestix.xitaso.net/de/marketplace">https://vws4ls.dev.mnestix.xitaso.net/de/marketplace</a>
<a href="https://mnestix.arena2036.app">mnestix.arena2036.app</a>	AAS-Server	<a href="https://vws4ls-api.dev.mnestix.xitaso.net/repo/">https://vws4ls-api.dev.mnestix.xitaso.net/repo/</a>

Tabelle 3-16: Subdomänen für "arena2036.app"

Die Domäne „marketplace.arena2036.app“ wurde als CNAME Eintrag konfiguriert, welcher direkt auf [vws4ls.dev.mnestix.xitaso.net](https://vws4ls.dev.mnestix.xitaso.net) zeigt. Somit kann ein Nutzer direkt auf der Domäne „marketplace.arena2036.app“ das System sehen ohne im Browser weitergeleitet zu werden.

Etwaige SSL-Zertifikate werden über gängige Tools, in diesem Fall [Traefik](#)<sup>32</sup>, automatisch generiert. Somit ist ein Zugriff über „HTTPS“ möglich.

### 3.5.2.2 Authentifizierung externer AAS Repositories

Von Drittparteien über das Internet zur Verfügung gestellte AAS-Server, wie bspw. von [Festo](#)<sup>33</sup> sind teilweise nur nach Authentifizierung zugreifbar. Um die dortigen Authentifizierungsmechanismen an Mnestix anbinden zu können, sind folgende Lösungsansätze denkbar:

#### 3.5.2.2.1 Token-Exchange

Analog zu dem im Kapitel 2.6.6.3 beschriebenen Ansatz wird der Authentifizierungsserver eines Drittanbieters als föderierte Autorität innerhalb der bestehenden Keycloak-Instanz für Mnestix konfiguriert. Die Benutzer können dann den föderierten Server für die Authentifizierung auswählen. Der primäre Authentifizierungsserver speichert dann das föderierte „Access Token“ und kann, falls unterstützt, einen Token-Exchange Flow initiieren, um ein „Access Token“ von der dritten Partei zu erhalten, siehe Abbildung 3-39.

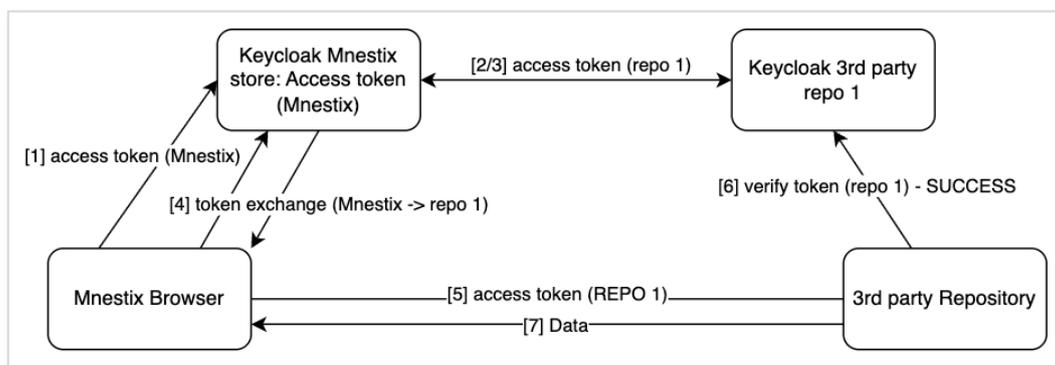


Abbildung 3-39: Ablauf Token-Exchange

Nur das vom Authentifizierungsserver des Drittanbieters ausgestellte „Exchange Token“ sollte von dem Repository des Drittanbieters akzeptiert werden. Dieser Austausch ist notwendig, da das Repository des Drittanbieters kein Token akzeptiert, welcher vom ursprünglichen Mnestix-Repository ausgestellt wurde.

#### 3.5.2.2.2 Authorization Code Flow

Bei fortgeschrittenen Konfigurationen mit mehreren Repositories kann es sinnvoll sein, separate Authentifizierungsdaten für jedes Repository zu definieren, siehe Abbildung 3-40. Bei diesem Ansatz würde jeder Authentifizierungsserver eines Drittanbieters einen eigenen Client für Mnestix bereitstellen,

<sup>32</sup> <https://doc.traefik.io/traefik/>

<sup>33</sup> <https://basyx.festo.com/aas-ui/>

der zusammen mit allen erforderlichen OpenID-Endpunkten, wie den Autorisierungs- und Token-Endpunkten, konfiguriert werden kann. Nach der Anmeldung über den Server des Drittanbieters würde der Benutzer nur Zugriff auf das spezifische Repository erhalten, das für diesen Server konfiguriert ist.

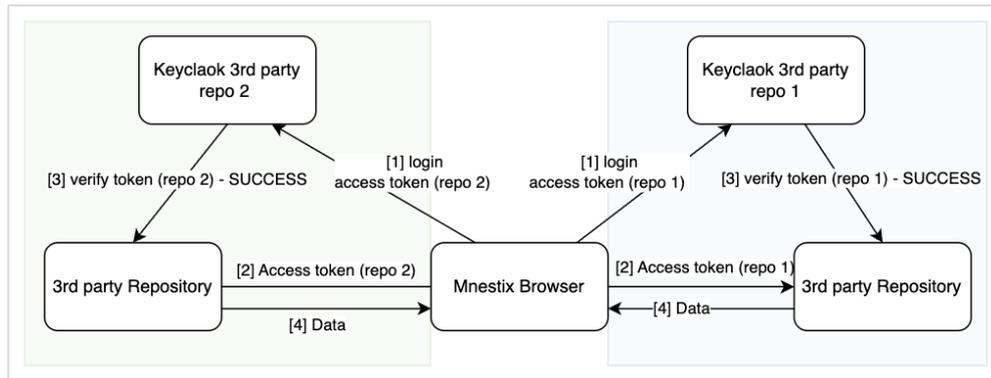


Abbildung 3-40: Ablauf Authorization Code Flow

Sowohl diese Methode als auch der Ansatz des Token-Exchange würden bei jedem Zugriff auf ein anderes Repository eine erneute Authentifizierung erfordern. Dieser Prozess erfordert eine komplexe Konfiguration pro Repository und muss durch gut durchdachte Sicherheitsmechanismen unterstützt werden. Wegen der erheblichen Komplexität wird dieser Ansatz jedoch derzeit nicht unterstützt.

### 3.5.2.2.3 Client-Credential

Bei diesem Ansatz nach Kapitel 2.6.6.4 stellt der Authentifizierungsserver eines Drittanbieters einem Client ein Secret zur Verfügung, das die Proxy-Komponente verwendet, um sich gegenüber dem Zielsystem zu authentifizieren und Zugang zu AAS und Teilmodellen zu erhalten, siehe Abbildung 3-41.

Der AAS-Browser kennt keinen Authentifizierungsmechanismus, er fordert lediglich Daten des Repositories an. Der Proxy wickelt also den Authentifizierungsvorgang ab, fügt das „Access Token“ an die nachgelagerte Serveranfrage an und gibt die Daten als authentifizierter Benutzer an den Browser zurück.

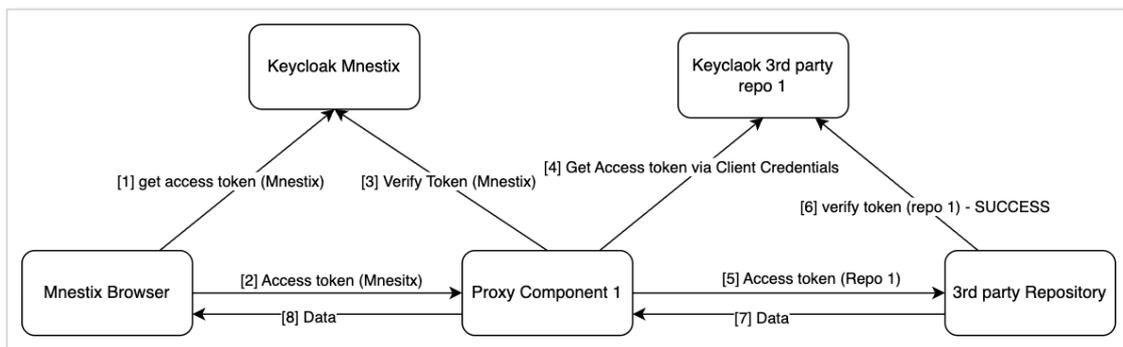


Abbildung 3-41: Ablauf Client Credential

Dieser Ansatz ist flexibel, da der Proxy mit einem Mnestix-„Access Token“ gesichert werden kann, wodurch sichergestellt wird, dass eine Authentifizierung innerhalb von Mnestix weiterhin erforderlich ist. Erst nach erfolgreicher Authentifizierung würde der Proxy die Anfrage weiterleiten.

Diese Flexibilität ermöglicht die Unterstützung mehrerer AAS-Repositories und Proxys von Drittanbietern, während sich der Benutzer nur einmal in Mnestix authentifizieren muss. Für eine feingranulare Kontrolle könnten individuelle Rollen angewendet werden, um den Zugriff auf bestimmte Repositories zu beschränken, die alle über einen einzigen Authentifizierungsserver verwaltet werden.

### 3.6 AP 12.6 - Testen und Qualitätssicherung

Zum Testen des Mnestix-basierten VWS4LS-Marketplace und der erstellten Referenz-Verwaltungsschalen wurden vergleichende Tests zwischen Mnestix-Viewer, dem Viewer der AAS-Suite und wo verfügbar den jeweiligen nativen BaSyx-basierten Viewern durchgeführt. Zudem wurden andere VWS-Server eingebunden (HARTING, WAGO, FESTO).

Im Zuge der Qualitätssicherungsaktivitäten wurden u.A. die folgenden Punkte identifiziert und an die entsprechenden Organisationen weitergetragen:

- (1) [AdministrativeInformation: New properties "createdAt" and "lastModified" · Issue #520 · admin-shell-io/aas-specs-metamodel](#)
- (2) [Revisit <<experimental>> SubmodelElement Range · Issue #450 · admin-shell-io/aas-specs-metamodel](#)
- (3) [Datatype "Identifier" should always be an URI · Issue #595 · admin-shell-io/aas-specs-metamodel](#)
- (4) [AAS Server Self Identification · Issue #461 · admin-shell-io/aas-specs-api](#)
- (5) [Add total amount to pagination metadata · Issue #468 · admin-shell-io/aas-specs-api](#)
- (6) [Define systematic URLs into the online repo · Issue #463 · admin-shell-io/aas-specs-api](#)
- (7) [How to cite IDTA publications ? · Issue #23 · admin-shell-io/aas-specifications](#)
- (8) [\[BUG\] Inconsistent definitions for image/picture file formats · Issue #121 · admin-shell-io/submodel-templates](#)
- (9) [ManufacturerName should be Property, not MLP · Issue #127 · admin-shell-io/submodel-templates](#)
- (10) [SMT DigitalNameplate: Define possible relation between Markings and Documentation · Issue #132 · admin-shell-io/submodel-templates](#)
- (11) [\[IDTA-01004-3-0 working\] Doc issue in file documentation/IDTA-01004/modules/ROOT/pages/access-rule-model.adoc · Issue #5 · admin-shell-io/aas-specs-security](#)
- (12) [Invalid IRDI used: 0112/2///62720#UAD718 · Issue #1118 · qudt/qudt-public-repo](#)
- (13) [how to represent fractional units \(Price and CurrencyExchange\) · Issue #1207 · qudt/qudt-public-repo](#)
- (14) [\[BUG\] Serialization Endpoint should give better Error Messages · Issue #753 · eclipse-basyx/basyx-java-server-sdk](#)
- (15) [\[FEATURE\] Detection for duplicate globalAssetIDs · Issue #733 · eclipse-basyx/basyx-java-server-sdk](#)
- (16) [\[FEATURE\] Concept Description Manager · Issue #525 · eclipse-basyx/basyx-aas-web-ui](#)
- (17) [AAS Editor Overview · Issue #8 · eclipse-basyx/basyx-aas-web-ui](#)
- (18) [Create/Edit/Delete AAS · Issue #10 · eclipse-basyx/basyx-aas-web-ui](#)
- (19) [F.03 Integration des Nameplate-Generators · Issue #36 · DHBW-TINF23F/Team2-BaSyx-WebClient](#)
- (20) [Implementierung F.03 Integration Nameplate-Generator · Issue #43 · DHBW-TINF23F/Team2-BaSyx-WebClient](#)
- (21) [F.05 Verbesserte Nutzerführung für die Submodelle · Issue #38 · DHBW-TINF23F/Team2-BaSyx-WebClient](#)
- (22) [Implementation of help buttons · Issue #36 · DHBW-TINF23F/Team3-BaSyx-Editor-Plugin](#)
- (23) [UI: API-Call Logs · Issue #71 · DHBW-TINF23F/Team1-BaSyx-Security-Plugin](#)
- (24) Erweiterung der Produktklassen im Bereich 4-04 (Kfz-Elektrik) um eine Property *PrimaryPartType*, wurde als Change Request zur ECLASS-Version 16 vom ECLASS e.V. angenommen.
- (25) „Qualifier-Initiative“ bei der IDTA, AdHoc-WG mit F. Scherschlich, M.Hoffmeister und C.Bornträger. Erarbeitung einer Vorschlagsliste zur Ergänzung der *IDTA GUIDELINE: How to create a Submodel Template Specification* [22].

### 3.7 AP 12.7 - Dokumentation und Schulung

In diesem AP stand die Erstellung von Implementierungsleitfäden im Vordergrund, die für Dritte als „Muster-Lösung“ bzw. „Kochrezept“ nachnutzbar sind. Eine Bereitstellung als webbasierte Online-Dokumentation ist über das VWS4LS-Github-Repository zugreifbar.

Die Ergebnisse dieses APs sind vorwiegend in dieser Abschlussdokumentation des TP12 und TP 13 enthalten, insbesondere die folgenden Kapitel:

- AP 12.2 - Datenmodellierung
- AP 13.4 – Semantische Durchgängigkeit
- AP 13.5 – Meta-Level AAS Designer
- AP 13.6 – XITASO Mnestix Viewer

## 4 TP13 – Standards und Middleware

Im Teilprojekt 13 „Standards und Middleware“ des Projekts VWS4LS wurden Lösungsansätze für die Anwendung von Branchenstandards und Normen zur Beschreibung von Informationsfragmenten in der Verwaltungsschale erarbeitet. Ein besonderer Fokus liegt dabei auf der Analyse und dem Vergleich von bereits etablierten Lösungen wie Eclipse BaSyx, dem AAS-Designer sowie MNESTIX, um die Einsatzmöglichkeiten der VWS in der industriellen Praxis zu bewerten.

Darüber hinaus werden vorhandene semantische Beschreibungen, wie beispielsweise der VEC und ECLASS, auf ihre Kompatibilität und ihren Mehrwert für die Verwaltungsschale geprüft. Ein weiterer zentraler Aspekt des Teilprojekts ist die Untersuchung, inwieweit Aspektmodelle und VWS-Submodelle für verschiedene Technologien – wie EDC und VWS – geeignet sind und ob beide Ansätze parallel oder jeweils nur einer davon eingesetzt werden kann.

Abschließend wird der aktuelle Entwicklungsstand der OPC-UA Companion Specification für die Wiring Harness Industrie dargestellt, um einen umfassenden Überblick über den Stand der Standardisierung und Interoperabilität in diesem Bereich zu geben.

### 4.1 AP 13.1 – Abgleich von Aspekt- und Teilmodellen

Eine wesentliche Herausforderung für den interoperablen Datentransfer im Catena-X Automotive Network (Catena-X) ist die Bereitstellung von Informationen in Form von Aspektmodellen.

Im Projektverlauf wurden zwei wesentliche Erkenntnisse gewonnen:

1. Für den Zugriff auf Verwaltungsschalen über Catena-X ist es nicht zwangsweise notwendig die Teilmodelle der Verwaltungsschale in Aspektmodelle für Catena-X zu transformieren. Stattdessen kann über die Control Plane des EDC die Zugriffskontrolle zum Verwaltungsschalenserver geregelt werden. Anschließend kann ein direkter Zugriff auf die Teilmodelle erfolgen.
2. Im Februar 2025 wurde zwischen IDTA und Catena-X eine gemeinsame Arbeitsgruppe «Guideline: How to use SAMM as semantic definition for Asset Administration Shell» geformt, um ein ganzheitliches Regelwerk zu erarbeiten, mit dem Interoperabilität zwischen Aspekt- und Teilmodellen hergestellt werden soll.

Aus diesen Gründen hat das Konsortium entschieden keine parallelen Forschungsaktivitäten hinsichtlich des Abgleichs von Aspekt- und Teilmodellen zu starten.

### 4.2 AP 13.2 – Freigabe der Teilmodelle „Process Parameters“

Im Rahmen von AP 13.2 wurde das Teilmodell «Bill of Process» (IDTA 02031) zu «Process Parameters» umbenannt, in zwei Teile aufgegliedert, inhaltlich klarer formuliert und im Kontext einer umfassenden Prozessbeschreibung - in Form eines Prozess-Teilmodells - eingeordnet.

Die ursprüngliche Bezeichnung «Bill of Process» war missverständlich und legte eine thematische Nähe zum Teilmodell «Hierarchical Structures enabling Bills of Material» (IDTA 02011) nahe. Diese thematische Nähe ist jedoch nicht vorhanden, weshalb das Teilmodell für ein besseres Verständnis umbenannt wurde.

Im weiteren Verlauf wurde deutlich, dass das Teilmodell in zwei Teilen veröffentlicht werden muss. Dieser Schritt ist notwendig, weil die Parameter auf Typ- und Instanz-Ebene beschrieben werden. Da die Verwaltungsschalen für Typen und Instanzen eigene Teilmodelle benötigen, sind hierfür entsprechend zwei Teilmodelle notwendig. Diese wurden entsprechend als Teil 1 für Typen und Teil 2 für Instanzen aufgegliedert:

- IDTA 02031-1 Process Parameters, Part 1: Type
- IDTA 02031-1 Process Parameters, Part 2: Instance

Die Teilmodelle wurden am 24.04.2025 in den IDTA-Freigabeprozess gegeben, mit dem Ziel einer Freigabe innerhalb der Laufzeit des Forschungsprojektes am 08.06.2025 zu erreichen.

### 4.3 AP 13.3 – OPC UA Begleitstandard "Wire Harness Manufacturing"

Der im Rahmen von TP1 erarbeitete OPC UA Begleitstandard «Wire Harness Manufacturing» (OPC 40570) wurde am 24.02.2025 in der Version 1.0 veröffentlicht und umfasst zunächst Prozesse im Schneidraum. Um einen ganzheitlichen Begleitstandard zu erreichen, wurden Prozesse für Prüfmaschinen beschrieben.

Hierfür wurde ein mehrstufiges Vorgehen gewählt:

1. Prüfprozesse identifizieren
2. Use Cases und Prozessparameter definieren
3. Abgleich mit Domänenexperten
4. Parameterabgleich mit VEC
5. Übergabe an VDMA-Arbeitsgruppe

#### 4.3.1 Prüfprozesse identifizieren

Auf Basis der Prozessliste aus TP3 wurde zunächst die Liste der Prüfprozesse extrahiert und Kernprozesse für den Begleitstandard ausgewählt. In Tabelle 4-1 sind die Prüfprozesse mit Erläuterungen aufgeführt. Es handelt sich hierbei um eine unvollständige Liste der Prüfprozesse, die für die Leitungssatzherstellung üblich sind.

Prüfprozess	Beschreibung
<b>Kapazitätsmessung</b>	Bei der Kapazitätsmessung werden die im Leitungssatz verbauten Kondensatoren auf Vorhandensein und Größe überprüft.
<b>Farberkennung</b>	Die Farberkennung dient im Wesentlichen der Erkennung von Leitungsfarben oder farblich kodierten Anbauteilen.
<b>Durchmesser</b>	Durchmesserprüfungen werden in der Regel durch geometrische Formen realisiert und dienen beispielsweise der Überprüfung von Ringkabelschuhen.
<b>Dioden</b>	Bei der Dioden-Prüfung werden im Leitungssatz verbaute Dioden in Durchgangs- und Sperr-Richtung überprüft.
<b>Lichtleiter</b>	Die Lichtleiter-Prüfung dient der Bewertung der Dämpfung des Lichtsignals im Leitungssatz, um schwache Lichtsignale zu erkennen.
<b>Verbindungstest</b>	Beim Verbindungstest werden alle elektrischen Verbindungen im Leitungssatz überprüft, um Kurzschlüsse, Unterbrechungen und vertauschte Leitungen zu erkennen.
<b>Pin-Position</b>	Bei der Pin-Positions-Prüfung werden die Stecktiefen der Terminals im Steckergehäuse überprüft, um unzureichende oder zu weit herausstehende Kontakte zu erkennen.
<b>Präsenz</b>	Die Präsenzerkennung wird eingesetzt, um Anbauteile und Kodierungen von Leitungssatzkomponenten zu erkennen.
<b>Relais</b>	Bei der Relaisprüfung werden im Leitungssatz verbaute Relais auf Funktion und Schaltverhalten überprüft.
<b>Airbag-Brücke</b>	Airbag-Brücken sind elektrische Brücken im Stecker der Airbag-Zündpille, die ein versehentliches Auslösen durch statische Entladungen verhindern sollen. Bei der Prüfung wird die Brücke im geschlossenen und geöffneten Zustand überprüft.
<b>Thermistor</b>	Die Thermistor-Prüfung wird für temperaturveränderliche Widerstände eingesetzt. Hierbei wird der Temperaturkennwert des Widerstands mit einem Vergleichswert verglichen.

<b>Leckage-Test</b>	Der Leckage-Test findet Anwendung bei Steckergehäusen, für die besondere Anforderungen an Dichtungen und Wasser- bzw. Luftdurchlässigkeit gelten. Hierbei wird die Dichtheit mittels Über- oder Unterdruckkontrolle nachgewiesen.
<b>Bestückungskontrolle</b>	Die Bestückungskontrolle findet Anwendung bei der Montage von Sicherungsboxen. Hierbei werden die bestückten Sicherungen, Relais und andere Anbauteile auf Vorhandensein, Steckrichtung und Lesbarkeit der Beschriftung überprüft.
<b>Höhenkontrolle</b>	Die Höhenkontrolle wird in Kombination mit der Bestückungskontrolle durchgeführt und überprüft ob die bestückten Sicherungen, Relais und andere Anbauteile vollständig gesteckt wurden.
<b>Schraubmontage</b>	Der Schraubprozess ist sowohl ein Montage- als auch ein Prüfprozess. Hierbei werden kritische Verschraubungen nach der Richtlinie VDI/VDE 2862 « <i>Mindestanforderungen zum Einsatz von Schraubsystemen und -Werkzeugen - Anwendungen in der Automobilindustrie</i> » durchgeführt und protokolliert. Zusätzlich werden die Schraubpositionen, Schraubparameter und ggf. elektrische Verbindungen überwacht.

Tabelle 4-1: Liste von Prüfprozessen mit Erläuterungen

Da eine realistische Standardisierung für alle Prüfprozesse innerhalb der Projektlaufzeit nicht möglich ist, wurden für die weiteren Arbeiten am Begleitstandard die folgenden Prüfprozesse ausgewählt:

- Verbindungstest
- Bestückungskontrolle
- Höhenkontrolle
- Höhenkontrolle

Es handelt sich hierbei um elementare Prüfprozesse, für die erhebliche Informationsbedarfe bestehen und somit wesentliche Anforderungen an den OPC UA Begleitstandard stellen.

#### 4.3.2 Use Cases und Prozessparameter definieren

Für die ausgewählten Prüfprozesse wurden im nächsten Schritt die Use Cases und die wichtigsten Prozessparameter definiert. Hierbei sind zunächst nur diejenigen Use Cases und Parameter betrachtet worden, die für die jeweiligen Prozesse unerlässlich sind und somit den größten Nutzen erzeugen. Weitere Parameter werden für zukünftige Versionen des Begleitstandards vorbehalten. Die vollständige Liste der Use Cases und Parameter ist dem Anhang zu entnehmen.

Abbildung 4-1 gibt einen Überblick über die allgemeine Struktur der Prüfprozesse, wie sie als erster Entwurf definiert wurden. Die Details sind in den folgenden Kapiteln näher beschrieben.

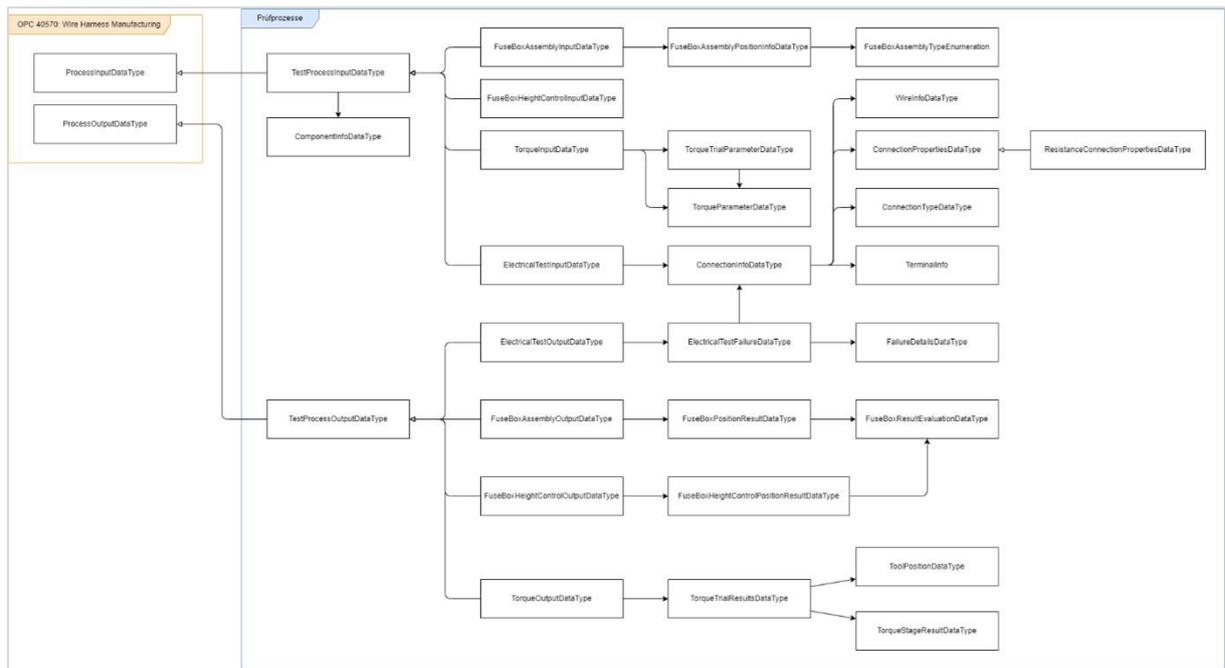


Abbildung 4-1: Struktur der Prüfprozesse für den OPC UA Begleitstandard

#### 4.3.2.1 Prüfprozesse

Der allgemeine Teil bezieht sich auf Parameter und Use Cases, die für alle Prüfprozesse gleichermaßen gültig sind. Hierbei geht es in erster Linie um Parameter bezüglich der zu prüfenden Komponenten, sowie des Gesamtergebnisses. Die Anforderungen sind in Tabelle 4-2 dargestellt.

Prüfprozesse	
<b>Name</b>	Prüfprozesse
<b>Ziel</b>	Kontrolle und Rückverfolgbarkeit von Prüfprozessen sicherstellen.
<b>Beschreibung</b>	Der Client möchte... <ul style="list-style-type: none"> <li>die zu prüfenden Komponenten festlegen</li> <li>wissen, welcher Benutzer und in welcher Schicht eine Prüfung durchgeführt wurde.</li> </ul> wissen, wie häufig eine Prüfung wiederholt wurde.

Tabelle 4-2: Use Cases für Prüfprozesse

#### 4.3.2.2 Verbindungstest

Beim Verbindungstest werden alle elektrischen Verbindungen im Leitungssatz überprüft, um Kurzschlüsse, Unterbrechungen und vertauschte Leitungen zu erkennen. In Tabelle 4-3 sind die Use Cases für den Prozess aufgeführt und beschrieben.

Elektrische Verbindung	
<b>Name</b>	Elektrische Verbindung
<b>Ziel</b>	Kontrolle und Überprüfung des elektrischen Durchgangs, der ordnungsgemäßen Installation und der Eigenschaften von Kabelbäumen.
<b>Beschreibung</b>	Der Client möchte... <ul style="list-style-type: none"> <li>die elektrische Durchgängigkeit und weitere Eigenschaften des Leitungssatzes überprüfen.</li> <li>steuern, ob eine Kurzschlussprüfung erfolgen soll.</li> <li>den Grenzwert für die Kurzschlussprüfung festlegen.</li> </ul>

	<ul style="list-style-type: none"> <li>• wissen, ob eine Komponente im Leitungssatz vorhanden ist.</li> <li>• wissen, ob eine Komponente bestimmte Eigenschaften aufweist.</li> <li>• wissen, ob eine Leitung fehlt.</li> <li>• wissen, ob eine Leitung am falschen Terminal angeschlossen ist.</li> <li>• wissen, ob zwei Leitungen kurzgeschlossen sind.</li> <li>• wissen, ob zwei Leitungen vertauscht wurden.</li> </ul> <p>wissen, was der genaue Grund für eine Fehler ist.</p>
--	--

Tabelle 4-3: Verbindungstest Use Cases

#### 4.3.2.3 Bestückungskontrolle

Die Bestückungskontrolle findet Anwendung bei der Montage von Sicherungsboxen. Hierbei werden die bestückten Sicherungen, Relais und andere Anbauteile auf Vorhandensein, Steckrichtung und Lesbarkeit der Beschriftung überprüft. In Tabelle 4-4 sind die Use Cases für den Prozess aufgeführt und beschrieben.

Bestückungskontrolle	
<b>Name</b>	Bestückungskontrolle
<b>Ziel</b>	Kontrolle und Überprüfung der korrekten Bestückung einer Sicherungsbox
<b>Beschreibung</b>	<p>Der Client möchte...</p> <ul style="list-style-type: none"> <li>• eine Sicherungsbox bestücken</li> <li>• steuern, ob eine leere Position überprüft werden soll.</li> <li>• steuern, ab welchem Schwellenwert eine Komponente als bestückt erkannt wird.</li> <li>• wissen, ob die korrekte Komponente in einer Position bestückt wurde.</li> <li>• wissen, welche Komponente in einer Position bestückt ist.</li> <li>• wissen, mit welcher Konfidenz eine Komponente erkannt wurde.</li> <li>• wissen, ob eine Komponente fehlt.</li> <li>• wissen, ob eine Komponente an der falschen Position bestückt wurde.</li> <li>• wissen, ob eine Komponente in der falschen Drehrichtung bestückt wurde.</li> <li>• wissen, ob eine Komponente auf einer Position bestückt wurde, die leer sein sollte.</li> </ul> <p>wissen, ob eine Beschilderung fehlt.</p>

Tabelle 4-4: Bestückungskontrolle Use Cases

#### 4.3.2.4 Höhenkontrolle

Die Höhenkontrolle wird in Kombination mit der Bestückungskontrolle durchgeführt und überprüft ob die bestückten Sicherungen, Relais und andere Anbauteile vollständig gesteckt wurden. In Tabelle 4-5 sind die Use Cases für den Prozess aufgeführt und beschrieben.

Höhenkontrolle	
<b>Name</b>	Höhenkontrolle
<b>Ziel</b>	Kontrolle und Überprüfung der korrekten Einbauposition von Komponenten einer Sicherungsbox
<b>Beschreibung</b>	<p>Der Client möchte...</p> <ul style="list-style-type: none"> <li>• die Einbauposition von Komponenten einer Sicherungsbox überprüfen.</li> <li>• steuern, ob leere Positionen überprüft werden sollen.</li> <li>• wissen, ob eine Komponente nicht vollständig gesteckt wurde.</li> </ul> <p>wissen, ob eine Komponente ungleichmäßig gesteckt wurde.</p>

Tabelle 4-5: Höhenkontrolle Use Cases

#### 4.3.2.5 Schraubmontage

Der Schraubprozess ist sowohl ein Montage- als auch ein Prüfprozess. Hierbei werden kritische Verschraubungen nach der Richtlinie VDI/VDE 2862 «*Mindestanforderungen zum Einsatz von Schraubsystemen und -Werkzeugen - Anwendungen in der Automobilindustrie*» durchgeführt und protokolliert. Zusätzlich werden die Schraubpositionen, Schraubparameter und ggf. elektrische Verbindungen überwacht. In Tabelle 4-6 sind die Use Cases für den Prozess aufgeführt und beschrieben.

Schraubmontage	
<b>Name</b>	Schraubmontage
<b>Ziel</b>	Kontrolle und Überprüfung der korrekten Verschraubung von sicherheitsrelevanten Schraubverbindungen.
<b>Beschreibung</b>	<p>Der Client möchte...</p> <ul style="list-style-type: none"> <li>• Schraubvorgänge an sicherheitsrelevanten Schraubverbindungen durchführen.</li> <li>• die Anzahl der Wiederholungen mit einer Schraube oder Mutter begrenzen.</li> <li>• wissen, ob ein Schraubvorgang mit den richtigen Einstellungen durchgeführt wurde.</li> <li>• wissen, welches Schraubwerkzeug für den Schraubvorgang verwendet wurde.</li> <li>• wissen, welche ID zum Schraubvorgang gehört.</li> <li>• wissen, wie häufig eine Schraube oder Mutter verschraubt wurde.</li> <li>• die Ergebnisse jeder Schraubstufe wissen.</li> <li>• wissen, an welcher Position die Verschraubung durchgeführt wurde.</li> <li>• die Schraubprogramme für eine Position festlegen.</li> </ul> <p>die Losschraubprogramme für eine Position festlegen.</p>

Tabelle 4-6: Schraubmontage Use Cases

#### 4.3.3 Abgleich mit Domänenexperten

Die erarbeiteten Use Cases und Prozessparameter wurden im nächsten Schritt weiteren Domänenexperten zur Kommentierung und Ergänzung zur Verfügung gestellt. Hierzu haben die Unternehmen Adaptronic und Cirris der Komax Gruppe in Zusammenarbeit mit dem Konsortialpartner Komax Testing den Abgleich durchgeführt.

Das Ergebnis des Abgleichs ist eine Liste der Use Cases und Prozessparameter, die aus Sicht der Domänenexperten notwendig sind.

#### 4.3.4 Abgleich mit VEC

Nach dem Abgleich mit den Domänenexperten wurden die Prozessparameter mit dem VEC-Standard abgeglichen, wie es auch für die vorhandenen Prozesse des Begleitstandards erfolgt ist. Hierfür wurden die Parameter zunächst auf relevante Klassen und Eigenschaften des VEC-Standards eingegrenzt, um anschließend die benötigten Strukturen zu extrahieren.

Eine wesentliche Erkenntnis bei der Anwendung des VEC-Standards ist die Möglichkeit Prüfprozesse auf der Grundlage von unterschiedlichen Merkmalen abzubilden. Beispielsweise können einfache elektrische Prüfprozesse, die lediglich die Verbindung zwischen zwei Komponenten erfassen, bereits auf Grundlage des logischen Modells des Leitungssatzes durchgeführt werden. Komplexere Anforderungen, hingegen, erfordern detailliertere Informationen aus dem physischen Modell. Das stellt besondere Herausforderungen an Maschinenhersteller und VEC-Datenquellen, die im Rahmen der Projektlaufzeit dieses Forschungsprojektes nicht weiter betrachtet werden konnten.

Auch die Zuordnung von VEC-Merkmalen zu den Prozessparametern stellte eine besondere Herausforderung dar, die nicht abschließend gelöst wurde. Beispielsweise lassen sich Merkmale aus dem VEC nicht immer eindeutig auf die heute üblichen Prozessmerkmale übertragen. Mögliche Ansätze für die

Auflösung dieser Herausforderungen sind Änderungen an den heute üblichen Prozessparametern, sowie die Erweiterung des VEC-Standards, um Spezifikationslücken zu schließen.

#### 4.3.5 Übergabe an die VDMA-Arbeitsgruppe «Wire Harness Manufacturing»

Die mit den Domänenexperten und VEC abgeglichenen Prüfprozesse wurden an die VDMA-Arbeitsgruppe «Wire Harness Manufacturing» für die weitere Standardisierung im Begleitstandard übergeben.

### 4.4 AP 13.4 – Semantische Durchgängigkeit

Eine elementare Voraussetzung für digitale Interoperabilität ist die Nutzung von semantischen Identifikationsmerkmalen und eindeutigen Bezeichnern über die gesamte digitale Wertkette.

Eindeutige Bezeichner sind erforderlich, um auf eine Verwaltungsschale und ihre Teilmodelle zu verweisen. Eindeutige Bezeichner werden zudem für den Verweis auf externe semantische Informationen verwendet. Die im Folgenden beschriebenen Kennzeichnungsschemata sind für die Verwendung in der Verwaltungsschale von entscheidender Bedeutung.

#### 4.4.1 Modellierung von Bereichswerten

Die „[Range-Definition der VWS](#)“ ist ähnlich wie die Property aufgebaut und als „experimental“ klassifiziert. Problematisch ist, dass damit kein Ist- / bzw. Sollwert abgebildet werden können, sondern lediglich eine Ober- und Untergrenze. Semantische Referenzierbarkeit über *valueId* wird im „Range“-Konstrukt ebenfalls nicht unterstützt.

Siehe dazu auch die Diskussion hier: <https://github.com/admin-shell-io/VWS-specs/issues/450>

Um für eine Property Bereichswerte zu modellieren, sollen geeignete „[Qualifier](#)“ definiert und in einen geeigneten VWS-Standard überführt werden.

Dazu wurde im Rahmen einer adHoc-„Qualifier-Initiative“ bei der IDTA mit F. Scherenschlich, M. Hoffmeister und C. Bornträger eine Vorschlagsliste zur Ergänzung der *IDTA GUIDELINE: How to create a Submodel Template Specification* [22] erarbeitet. Im Wesentlichen geht es dabei um diese vier „TOLERANCE\_...“-Definitionen:

```
{
  "name": "Predicate relation | negative tolerance %",
  "qualifier": {
    "kind": "ConceptQualifier",
    "semanticId": {
      "type": "ExternalReference",
      "keys": [
        {
          "type": "GlobalReference",
          "value": "0112/2///61360_4#AAF444"
        }
      ]
    }
  },
  "type": "PredicateRelation",
  "value": "TOLERANCE_REL_NEG_10",
  "valueId": null
},
{
  "name": "Predicate relation | positive tolerance %",
  "qualifier": {
    "kind": "ConceptQualifier",
    "semanticId": {
      "type": "ExternalReference",
      "keys": [
        {
          "type": "GlobalReference",
          "value": "0112/2///61360_4#AAF445"
        }
      ]
    }
  },
  "type": "PredicateRelation",
  "value": "TOLERANCE_REL_POS_11",
  "valueId": null
},
{
  "name": "Predicate relation | negative tolerance abs",
  "qualifier": {
    "kind": "ConceptQualifier",
    "semanticId": {
```

```

    "type": "ExternalReference",
    "keys": [
      {
        "type": "GlobalReference",
        "value": "http://www.prostep.org/ontologies/ecad/2024/03/vec#toleranceLowerBoundary"
      }
    ]
  },
  "type": "PredicateRelation",
  "value": "TOLERANCE_ABS_NEG_12",
  "valueId": null
}
},
{
  "name": "Predicate relation | positive tolerance abs",
  "qualifier": {
    "kind": "ConceptQualifier",
    "semanticId": {
      "type": "ExternalReference",
      "keys": [
        {
          "type": "GlobalReference",
          "value": "http://www.prostep.org/ontologies/ecad/2024/03/vec#toleranceUpperBoundary"
        }
      ]
    }
  },
  "type": "PredicateRelation",
  "value": "TOLERANCE_ABS_POS_13",
  "valueId": null
}
},
}
}

```

Listing 4-1: Vorschlag für die Definition von Qualifiern

Die Qualifier-Definitionen in dieser Datei des VWSXPE sollen konsolidiert werden in der *IDTA GUIDELINE: How to create a Submodel Template Specification* [22]:

<https://github.com/eclipse-VWSpe/package-explorer/blob/main/src/VWSxPackageExplorer/qualifier-presets.json>

Die [DIN SPEC 92000](#) und bestehende IEC-Definitionen sollen dabei beachtet werden, z.B.

[https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/PropertiesAllVersions/0112-2---61360\\_4%23AAF445](https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/PropertiesAllVersions/0112-2---61360_4%23AAF445)

[https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/PropertiesAllVersions/0112-2---61360\\_4%23AAF575](https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/PropertiesAllVersions/0112-2---61360_4%23AAF575)

#### 4.4.2 UUID / GUID

**UUID** (Universally Unique Identifier<sup>34</sup>) ist eine 128-Bit-Kennung (16 Byte), die vom Open Systems Interconnection (OSI)-Framework standardisiert und in der RFC 4122 spezifiziert ist. UUIDs sind immer einzigartig, ohne dass eine zentrale Behörde zur Koordination notwendig wäre. Die Einzigartigkeit beruht auf Wahrscheinlichkeit (für V4) oder sorgfältigem Design (für V1, V3, V5), wodurch die Wahrscheinlichkeit von Kollisionen mathematisch ausgeschlossen werden kann.

**GUID** (Globally Unique Identifier) ist eine spezielle Form von UUID, die von Microsoft gestaltet wurde. Im Kontext der Verwaltungsschale können UUID und GUID als austauschbar betrachtet werden.

**Format:** Eine UUID wird in der Regel als 32-stellige Hexadezimalzahl dargestellt, die durch Bindestriche in fünf Gruppen unterteilt ist: 8-4-4-4-12.

**Beispiel:** 550e8400-e29b-41d4-a716-446655440000

**Varianten:** RFC 4122 definiert eine bestimmte „Variante“ (Bits 64-65 auf 10 gesetzt), um sie von anderen 128-Bit-ID-Schemata zu unterscheiden. Die meisten UUIDs folgen dieser Variante.

UUIDs werden oft von VWS-Tools generiert, um sie als VWS- und Teilmodell-IDs zu verwenden. Dies mag in vielen Fällen als schnelle Lösung angemessen sein, kann aber langfristig problematisch werden, da UUIDs nicht *systematisch* und *sprechend* und somit schwierig zu verwalten sind. Insbesondere für die Verwendung in logisch aufgebauten ID-Strukturen sind UUIDs eher nicht geeignet. Daher sollten VWS-Ersteller eine sorgfältige Entscheidung über die Verwendung von UUIDs treffen.

<sup>34</sup> [https://de.wikipedia.org/wiki/Universally\\_Unique\\_Identifier](https://de.wikipedia.org/wiki/Universally_Unique_Identifier)

### 4.4.3 IRDI (ISO 29005-5)

Der *International Registration Data Identifier* (IRDI) ist ein globales Identifikationssystem für Eigenschaften, Werte und Konzepte, siehe Tabelle 4-2. Es ist in ISO 29005-5 und ISO/IEC 11179-6 als ein bewährtes Mittel zur Schaffung handhabbarer eindeutiger Identifikatoren definiert, die über verschiedene Sprachen und IT-Systeme hinweg konsistent bleiben. IRDIs werden in ECLASS-, IEC- und ISO-Normen verwendet.

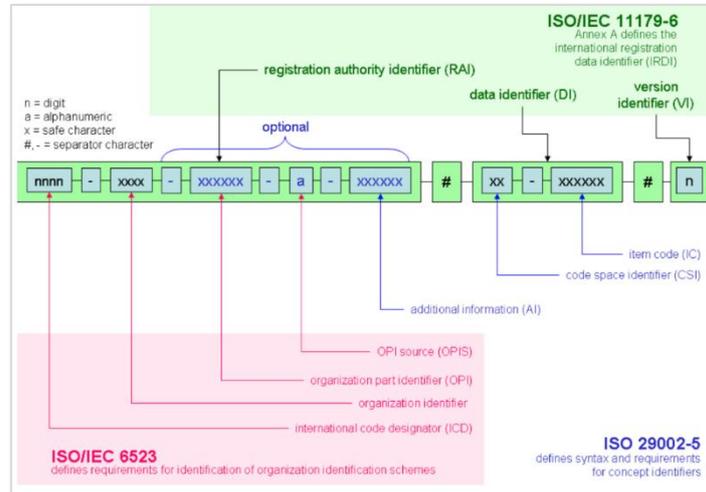


Abbildung 4-2: IRDI-Schema nach ISO 29005-5<sup>35,36</sup>

IRDIs sind ein historisch etablierter Referenzier-Mechanismus, mit dem in der VWS umgegangen werden muss. Da sie aber eine externe Verwaltung benötigen, wird davon abgeraten, neue IRDIs für Elemente in der VWS zu erzeugen.

#### 4.4.3.1 IEC IRDI-Struktur

Eine IEC-CDD folgt dem allgemeinen Format: **ICD/OI/AI#IC#VI**

- **International Code Designator (ICD):** Identifiziert die Registrierungsbehörde (z. B. „0112“ für IEC).
- **Organization Identifier (OI):** Gibt die Organisation innerhalb der Behörde an (z. B. „2“ für IEC).
- **Application Identifier (AI):** Gibt die spezifische Norm an (z. B. „61360\_4“ für IEC 61360-4 DB).
- **Internal Code (IC):** Eindeutiger Code für das Element innerhalb des Wörterbuchs (z. B. „AAA612“).
- **Version Indicator (VI):** Bezeichnet die Version des Eintrags (z. B. „001“).

**Beispiel:** [0112/2///61360\\_4#AAA612#001](#), siehe Tabelle 4-7

0112/2///61360_4#AAA612	
<b>Code:</b>	description
<b>0112/2/</b>	Issuing Agency Code (IEC)
<b>61360_4</b>	IEC 61360 Standard Reference
<b>#AAA612</b>	Unique identifier for a property/class

Tabelle 4-7: Breakdown of IRDI Example (0112/2///61360\_4#AAA612)

<sup>35</sup> [https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS-BMEcat-Guideline-2005\\_1\\_v2\\_1.pdf](https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS-BMEcat-Guideline-2005_1_v2_1.pdf)

<sup>36</sup> <https://reference.opcfoundation.org/Core/Part19/v105/docs/5.3>

#### 4.4.3.2 ECLASS IRDI-Struktur

Eine ECLASS IRDI hat in der Regel das Format: **ICD/OI/CSI#Code#Version**, siehe Tabelle 4-8 und Tabelle 4-9

- **ICD (International Code Designator):** Code für die Registrierungsbehörde, z. B. „0173“ für ECLASS.
- **OI (Organization Identifier):** Identifiziert die Organisation
- **CSI (Code Space Identifier):** Gibt die Art des Strukturelements an (z. B. „01“ für Klassifizierungsklasse, „02“ für Eigenschaft, „07“ für Wert):

Code Space Identifier	
01	Class
02	Property
05	unit of measurement
07	property value

Tabelle 4-8: Excerpt of Code Space Identifiers (CSI) according to ISO 290305-5<sup>27</sup>

- **Code:** Eindeutiger Bezeichner für das spezifische Element (z. B. „27-22-01-01“ für eine Klasse oder „AAB123“ für eine Eigenschaft).
- **Version:** Eine Versionsnummer (z. B. „001“).

0173-1#01-AAA123#001	
Code:	description
0173	ICD code for eCl@ss
1	eCl@ss Office
02	property
ABH173	identifier of class
003	version of class

Tabelle 4-9: Breakdown of IRDI Example (0173-1#02-ABH173#003)

#### Classification classes

**Beispiel:** 0173-1#01-27-22-01-01#001 entspricht 0173-1#01-AAC098#020

Informative Links zu online Elementbeschreibungen können so zusammengesetzt werden:

[https://eclass.eu/eclass-standard/content-suche/show?tx\\_eclassesearch\\_ecsearch%5Bid%5D=27220101](https://eclass.eu/eclass-standard/content-suche/show?tx_eclassesearch_ecsearch%5Bid%5D=27220101)

#### 4.4.4 URI / IRI

Eine **URI** (Uniform Resource Identifier) ist eine Zeichenkette zur Identifizierung einer Ressource und ist durch RFC 3986 standardisiert. Es handelt sich um ein Konzept, das alles umfasst, was benannt oder lokalisiert werden kann, sei es eine Webseite, eine Datei oder eine abstrakte Einheit.

##### Beispiel

`http://example.com/resource/123`

##### Bestandteile

- **Schema** (`http`),
- **Autorität** (`example.com`),
- **Pfad** (`/resource/123`)
- **Abfragen** (`?key=value`) und/oder Fragmente (`#section1`).

##### Untertypen

- URL (Uniform Resource Locator)
- URN (Uniform Resource Name)

Eine **IRI** (Internationalized Resource Identifier) ist eine durch RFC 3987 definierte Erweiterung des URI, die durch Umkodierung auch Nicht-ASCII-Zeichen (z.B. Akzente, chinesische Zeichen) unterstützt. IRIs sind technisch gesehen eine Obermenge von URIs, d.h. jeder URI ist ein IRI, aber nicht andersherum.

##### Beispiel

`http://example.com/资源/123`

wird codiert als

`https://xn--example-xta.com/%E8%B5%84%E6%BA%90/123`

Die VWS benötigt weltweit eindeutige, maschinenlesbare und interoperable Identifikatoren. IRIs erfüllen diese Aufgabe, da sie die URIs (Uniform Resource Identifiers) um internationale Zeichen erweitern und sich durch die folgenden Merkmale an den globalen Geltungsbereich von Industrie 4.0 anpassen:

- **Globale Einzigartigkeit:** IRIs nutzen Namensräume (z. B. Domännennamen), um sicherzustellen, dass sich zwei Assets nicht überschneiden, auch nicht in verschiedenen Organisationen.
- **Internationalisierung:** IRIs erlauben Nicht-ASCII-Zeichen (z. B. `http://工厂.cn/设备/123` für eine chinesische Fabrik), was für multinationale Lieferketten entscheidend ist.
- **Auflösbarkeit:** HTTP-basierte IRIs können auf eine Ressource (z.B. einen VWS-Server) verweisen, was den direkten Abruf von Daten ermöglicht.
- **Standardisierung:** IRIs entsprechen den Webstandards (RFC 3987) und den Praktiken des sog. „Semantic Web“<sup>37</sup>, wodurch VWS mit breiteren Ökosystemen wie OPC UA oder Linked Data kompatibel ist.

In der VWS-Metamodellspezifikation [13] werden IRIs ausdrücklich als primärer Bezeichnertyp sowohl für das Asset als auch für die VWS selbst empfohlen und bieten wesentliche administrative Vorteile:

**Namensraum-Kontrolle:** Das Schema und die Domäne fungieren als Namensraum, sodass Organisationen oder Systeme ihre eigenen Bezeichner ohne zentrale Koordinierungsstelle definieren können.

**Erweiterbarkeit:** URIs sind flexibel, man kann einen Pfad, eine Abfrage oder ein Fragment hinzufügen, um die Identität zu verfeinern.

Für die Anwendung in der VWS wird empfohlen, die Verwendung von Nicht-ASCII-Sonderzeichen in URIs/IRIs zu vermeiden.

<sup>37</sup> [https://de.wikipedia.org/wiki/Semantic\\_Web](https://de.wikipedia.org/wiki/Semantic_Web)

#### 4.4.5 Semantische Datenbanken

Eine semantische Referenz ist ein Link zu einem externen Standard oder einer Ontologie, die die Bedeutung eines Datenelements innerhalb einer VWS definiert. Diese Referenzen gewährleisten Interoperabilität, Konsistenz und Automatisierbarkeit zwischen Industrie 4.0 Komponenten.

Für technische Daten in Industrieanlagen ist ein generischer Rahmen zur Strukturierung der Informationen erforderlich, um standardisierte Vokabulare und Industriestandards zur Definition und Verknüpfung von Komponentenattributen bereitzustellen. Es gibt eine Reihe von Industriestandards für semantische Referenzen:

- **ECLASS:** Ein branchenübergreifender Standard mit Schwerpunkt auf Produktidentifikation mit detaillierten technischen Eigenschaften, der in Europa weit verbreitet ist. Er ist sehr granular und unterstützt mehrere Domänen.

<https://eclass.eu/en/eclass-standard/search-content/search>

- **ETIM** (Elektrotechnisches Informationsmodell): Ein standardisiertes Klassifizierungssystem hauptsächlich für Elektro- und HLK-Produkte. Konzentriert sich auf technische Produktdaten für die Elektro-, Gebäude- und Installationsbranche. Beliebt in Europa, insbesondere bei Herstellern, Großhändlern und Auftragnehmern für den Austausch von Produktdaten. Ähnlich wie ECLASS bietet es Klassen, Merkmale und Werte, ist aber stärker auf die elektrotechnischen und verwandten Branchen spezialisiert. Es wird von der Organisation ETIM International verwaltet.

<https://prod.etim-international.com/class>,  
<https://etimapi.etim-international.com/>

- **GPC** (Globale Produktklassifikation): Ein Produktklassifizierungssystem, das von GS1 für den globalen Handel entwickelt wurde. Es deckt Konsumgüter, Industrieprodukte und Dienstleistungen ab, wobei der Schwerpunkt auf dem Einzelhandel und dem Handel liegt. Wird in Verbindung mit GS1-Standards (z. B. Barcodes) für die Effizienz der Lieferkette verwendet.

<https://gpc-browser.gs1.org/>

- **Electropedia:** Von der IEC herausgegebene Online-Terminologiedatenbank, enthält alle Begriffe und Definitionen des Internationalen Elektrotechnischen Vokabulars (IEV), das in den IEC 60050-Schriften veröffentlicht ist. Sie enthält mehr als 22 000 terminologische Einträge in englischer und französischer Sprache, gegliedert nach Themenbereichen, mit entsprechenden Begriffen in verschiedenen anderen Sprachen: Arabisch, Chinesisch, Dänisch, Deutsch, Finnisch, Italienisch, Japanisch, Koreanisch, Kroatisch, Mongolisch, Niederländisch, Norwegisch, Polnisch, Portugiesisch, Russisch, Schwedisch, Serbisch, Slowakisch, Slowenisch, Spanisch, Tschechisch, Türkisch und Ukrainisch (die Abdeckung variiert je nach Fachgebiet).

<https://electropedia.org/>

- **IEC-CDD:** Konzentriert sich auf elektrotechnische und industrielle Bereiche, basiert auf IEC 61360-Normen und legt den Schwerpunkt auf die technische Merkmalsbeschreibung.

<https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/SearchFrameset>,  
<https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TreeFrameset>

- **VEC** (Vehicle Electric Container) ist ein offener Standard, der im Rahmen von prostep ivip und VDA entwickelt wurde, um elektrische und elektronische Systeme in Fahrzeugen zu beschreiben, z. B. Kabelbäume, Komponenten und Anschlussmöglichkeiten. Es ist ein XML-basiertes Datenmodell, stellt aber auch eine Ontologie bereit, d.h. ein formalisiertes Vokabular mit Klassen, Eigenschaften und Beziehungen. Seine Elemente können unter Anwendung der Prinzipien des semantischen Webs über URIs referenziert werden.

<https://ecad.prostep.org/ontologies/2024/03/vec>

Einzelne Referenzsysteme bieten meist nur spezialisierte semantische Definitionen. Es kann daher auch sinnvoll sein, mehrere Referenzsysteme in Kombination zu verwenden.

#### 4.4.5.1 ECLASS

ECLASS ist ein international anerkanntes Klassifizierungssystem, welches einen standardisierten Rahmen für die Beschreibung von Produkten und Dienstleistungen in allen Branchen bietet. Es ermöglicht Unternehmen, Herstellern und Lieferanten eine konsistente Identifizierung von Produktklassen und -eigenschaften über verschiedene Länder, Sprachen und Geschäftssysteme hinweg.

Der ECLASS-Standard wird kontinuierlich erweitert, wobei neue IRDIs über eine Content-Development-Plattform vorgeschlagen und geprüft werden.

Der Datenbestand kann über eine native Webseite eingesehen werden, siehe Abbildung 4-3: <https://e-class.eu/en/e-class-standard/search-content/>:

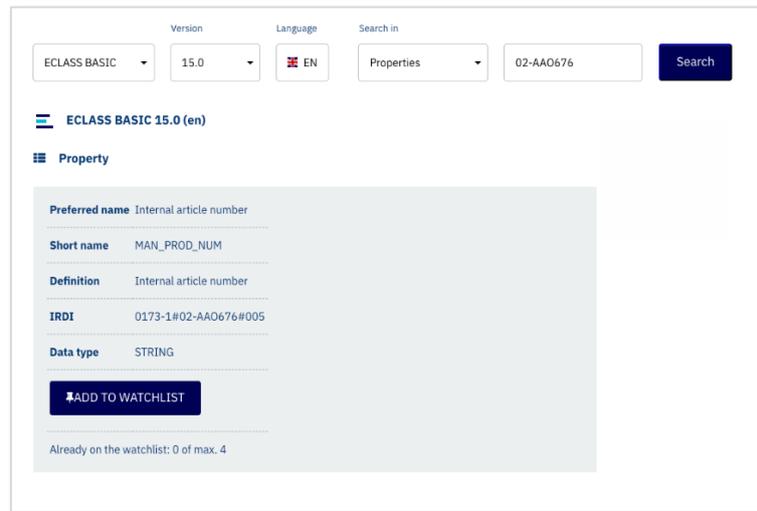


Abbildung 4-3: ECLASS - Webbasierte Contentsuche und -anzeige

Die Nutzung von ECLASS-Daten unterliegt den Vertragsbedingungen des ECLASS e.V. Mitglieder zahlen Beiträge, die je nach Unternehmensgröße variieren, und erhalten Zugang zu den Daten, auch über eine webbasierte REST-API. Nicht-Mitglieder müssen ebenfalls Lizenzen erwerben, um IRDIs rechtmäßig zu nutzen. Eine unbefugte Nutzung ohne Lizenz verstößt gegen die Nutzungsbedingungen und kann rechtliche Konsequenzen nach sich ziehen. Es gibt auch die Möglichkeit, ECLASS IRDIs über ein Pay-per-IRDI-Modell zu lizenzieren.

Die IDTA hat eine Best-Practice-Guideline zur Anwendung von ECLASS im VWS-Kontext veröffentlicht [16].

#### 4.4.5.2 IEC-CDD

Das *Common Data Dictionary* (CDD) der Internationalen Elektrotechnischen Kommission (IEC) verwendet den International Registration Data Identifier (IRDI) zum Referenzieren von Eigenschaften, Klassen und Werten und gewährleistet so die Interoperabilität über Branchen, digitale Zwillinge und Lieferketten hinweg. IEC-Normen sind in der Industrieautomation, in Energiesystemen, in der Elektronik und in der Fertigung weit verbreitet.

Die Nutzung von IEC CDD IRDIs ist grundsätzlich kostenfrei, Nutzer müssen jedoch die IEC-Nutzungsbedingungen und die zugrunde liegenden Standards einhalten, insbesondere bei kommerziellen Anwendungen oder Integrationen in Systeme. Die Datenbank ist über eine Weboberfläche öffentlich zugänglich, eine netzwerkbasierende Schnittstelle über eine REST-API wurde bislang jedoch noch nicht realisiert.

Der Zugang über die native Weboberfläche ist im Folgenden kurz beschrieben:

**Schritt 1:** Aufruf der Webseite: <https://cdd.iec.ch/cdd/common/iec61360-7.nsf/TreeFrameset>

**Schritt 2:** Auswahl eines geeigneten IEC-Standards, z.B. „IEC61360-4“ (siehe Abbildung 4-4):

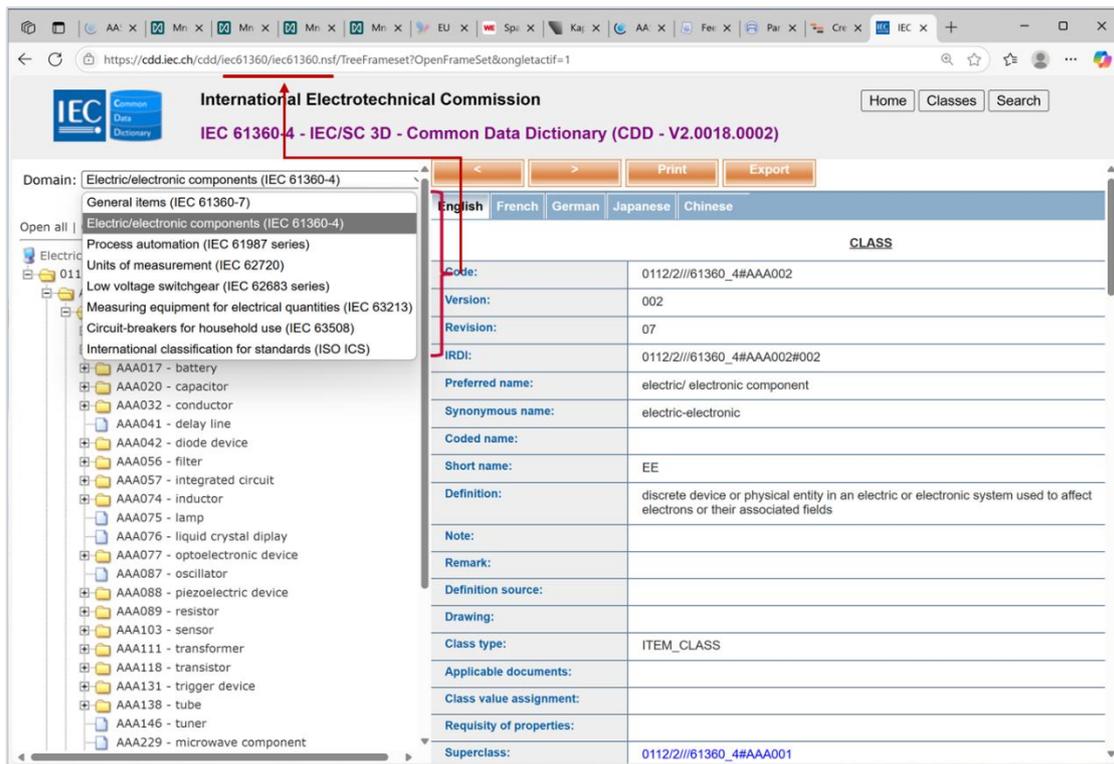


Abbildung 4-4: IEC CDD: Auswahl des Standards

**Schritt 3:** Durchsuchen des ausgewählten Baums manuell über <https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TreeFrameset> und nach der passenden Klasse und/oder dem passenden Attribut suchen, siehe Abbildung 4-5. Alternativ kann eine Textsuche über <https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/SearchFrameset> durchgeführt werden. Wenn Sie z. B. die IEC-ID für „Temperaturtyp“ ([https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/PropertiesAllVersions/0112-2--61360\\_4%23AAE683](https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/PropertiesAllVersions/0112-2--61360_4%23AAE683)) finden möchten, scrollen Sie entweder auf der Website nach unten oder führen Sie eine Textsuche über das entsprechende Suchwort durch. Klicken Sie auf das Suchergebnis, das Ihrer Meinung nach am besten zu Ihrer Suche passt.

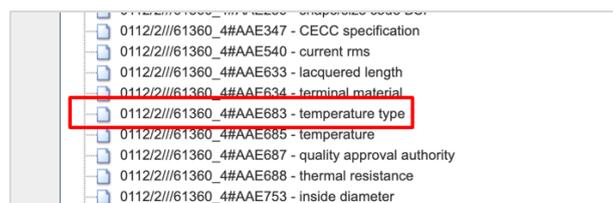


Abbildung 4-5: Suche nach IEC-IRDIs

**Schritt 4:** Auswahl des IEC IRDI für die gewählte Property, siehe Abbildung 4-6

English	French	German	Japanese	Chinese
<b>PROPERTY</b>				
<b>Code:</b>	0112/2///61360_4#AAE685			
<b>Version:</b>	001			
<b>Revision:</b>	06			
<b>IRDI:</b>	0112/2///61360_4#AAE685#001			
<b>Preferred name:</b>	temperature			
<b>Synonymous name:</b>				
<b>Symbol:</b>	@T			
<b>Synonymous symbol:</b>				

Abbildung 4-6: IEC-IRDI für eine Property

**Schritt 5:** Das ausgewählte IRDI kann nun als semantische Referenz an entsprechender Stelle (z.B. als *semanticId*, *referredSemanticId*, *valueId*) in der VWS eingefügt werden. Siehe Abbildung 4-7 als Beispiel die Verwendung für ProductClassId

Abbildung 4-7: ProductClassId mit IEC-IRDI Referenz

#### 4.4.5.2.1 Vordefinierte Daten aus dem IEC-CDD

Typisch anwendbare Attribute für spezielle Themen wie Farben, Materialien, Schutzklassen, etc. finden sich im IEC-CDD in gut ausdefinierter Form. Im Folgenden eine auszugsweise Auflistung.

### Colour properties

Applicable properties:	Enumeration code list:
<a href="#">0112/2///61360 4#AAF250 - insulation colour code</a> <a href="#">0112/2///61360 4#AAH065 - housing colour code</a> <a href="#">0112/2///61360 7#CBA018 - IEC colour code of item</a>	<a href="#">0112/2///61360 7#CEA012 - IEC colour codes:</a> N.A., BK, BN, RD, OG, GN, YE, BU, VT, GY, WH, PK, GD, TQ, SR, GNYE, BKBN, BKRD, BKOG, BKGN, BKVT, BKGY, BKWH, BKPK, BKGD, BKTQ, BKSR, BRRD, BROG, BRBU, BRVT, BRGY, BRWH, BRPK, BRGD, BRTK, BRSR, RDOG, RDBU, RDVT, RDGY, RDWH, RDPK, RDGD, RDTQ, RDSR, OGBU, OGVY, OGGY, OGWH, OGPk, OGGD, OGTQ, OGSr, BUVT, BUGY, BUWH, BUPK, BUGD, BUTQ, BUSR, VTGY, VTWH, VTPK, VTGD, VTTQ, VTSR, GYWH, GYPK, GYGD, GYTQ, GYSR, WHPK, WHGD, WHTQ, WHSR, PKGD, PKTQ, PKSR, GDTQ, GDSR, TQSR, OTHERS BG, BK, BL, BN, BZ, GN, GY, IV, NC, OR, PK, RD, TN, VT, WT, YL
<a href="#">0112/2///61360 4#AAF128 - package colour</a>	

Tabelle 4-10: IEC Colour properties

### Conductor properties

Applicable properties:	Enumeration code list:
<a href="#">0112/2///61360 4#AAF243 - conductor configuration</a>	BRAID, BUNCH, LITZ, SOLID, STRAND, TINSEL
<a href="#">0112/2///61360 4#AAJ018 - sealing class</a>	DUSTP, OPEN, SEAL
<a href="#">0112/2///61360 4#AAH056 - body insulation material</a>	CER, GLS, PLA
<a href="#">0112/2///61360 4#AAF248 - insulating material</a>	ECTFE, ENAM, E/TFE, FEP, PA, PAPER, PE, PFA, POLY, PP, PTFE, PUR, PVC, RUBBER, TEXTILE, UP
<a href="#">0112/2///61360 4#AAF241 - conductive material</a>	Al, Cu, CuCd, CuCdCr, CuCr, CuNi, CuSn, CuZn, Fe/Cu
<a href="#">0112/2///61360 4#AAF240 - conductor finish</a>	Ag, Ni, Sn
<a href="#">0112/2///61360 4#AAR025 - contact material</a>	Ag, AgCdO, AgCdO/Au, AgNi, AgNi/Au, AgPd, AgPd/Au, AgSnO2, AgSnO2/Au, AgW, Ag/Au, AuAg, PdCu, PdNi, Rh, Rh/Au, W
<a href="#">0112/2///61360 4#AAE355 - contact body material</a>	BeCu, Cu, CuSn, CuZn, Ni, PCuSn
<a href="#">0112/2///61360 4#AAE350 - contact finish</a>	Ag, Au, CuZn, Ni, PCuSn, Pd, Sn, Zn
<a href="#">0112/2///61360 4#AAE351 - housing material</a>	CER, DAP, MET, PA, PC, PLA, PPOX, PTFE
<a href="#">0112/2///61360 4#AAH005 - housing finish</a>	Ag, Au, Cr, ELOX, LAC, Ni, PLA, RAW, RUB, Sn, Zn
<a href="#">0112/2///61360 4#AAE634 - terminal material</a>	AgPd, NiSn
<a href="#">0112/2///61360 4#AAH028 - terminal finish</a>	Ag, Au, Cr, Ni, Pd, RAW, Sn

Tabelle 4-11: IEC Conductor properties

## Mechanical properties

Applicable properties:	Enumeration code list:
<a href="#">0112/2///61360 4#AAH011 - designation of IP protection</a> <a href="#">0112/2///61360 7#CBA025 - IP code</a> <a href="#">0112/2///61987#ABA558 - degree of protection (IP)</a>	<a href="#">0112/2///61360 7#CEA015</a> IP00, IP01, IP02, IP03, IP04, IP05, IP06, IP07, IP08, IP10, IP11, IP12, IP13, IP14, IP15, IP16, IP17, IP18, IP20, IP21, IP22, IP23, IP24, IP25, IP26, IP27, IP28, IP30, IP31, IP32, IP33, IP34, IP35, IP36, IP37, IP38, IP40, IP41, IP42, IP43, IP44, IP45, IP46, IP47, IP48, IP50, IP51, IP52, IP53, IP54, IP55, IP56, IP57, IP58, IP60, IP61, IP62, IP63, IP64, IP65, IP66, IP67, IP68, IP69, IPX1, IPX2, IPX3, IPX4, IPX5, IPX6, IPX7, IPX8, IP1X, IP2X, IP3X, IP4X, IP5X, IP6X

Tabelle 4-12: IEC Mechanical Properties



### [0112/2///62683#ACC066 - Installation, mounting and dimensions](#)

- [0112/2///63213#KEA336 - max. height of the device](#)
- [0112/2///63213#KEA337 - max. width of the device](#)
- [0112/2///63213#KEA337 - max. width of the device](#)
- [0112/2///63213#KEA321 - portable device](#)
- [0112/2///63213#KEA322 - dimension of panel mounted device](#)
- [0112/2///63213#KEA323 - number of modular spacings](#)
- [0112/2///63213#KEA324 - housing device for DIN rail mounting](#)
- [0112/2///62683#ACE801 - height of the device](#)
- [0112/2///62683#ACE802 - width of the device](#)
- [0112/2///62683#ACE803 - length of the device](#)
- [0112/2///62683#ACE804 - mounting onto standard rail](#)
- [0112/2///62683#ACE808 - product mass](#)

## Power supply properties

- [0112/2///62683#AC531 - supply voltage limit](#)

## Identification properties



### [0112/2///61987#ABC269 - Identification](#)

- [0112/2///61987#ABA565 - name of manufacturer](#)
- [0112/2///61987#ABP463 - URI of company logo](#)
- [0112/2///61987#ABP462 - country of origin](#)
- [0112/2///61987#ABB064 - name of supplier](#)
- [0112/2///61987#ABA566 - type name of product](#)
- [0112/2///61987#ABP464 - family name of product](#)
- [0112/2///61987#ABA567 - model name of product](#)
- [0112/2///61987#ABA300 - code of product](#)
- [0112/2///61987#ABA950 - order code of product](#)
- [0112/2///61987#ABA581 - article number](#)
- [0112/2///61987#ABA587 - GTIN code](#)
- [0112/2///61987#ABA301 - national stock number](#)
- [0112/2///61987#ABP643 - device version](#)
- [0112/2///61987#ABA601 - software version](#)
- [0112/2///61987#ABA601 - software version](#)
- [0112/2///61987#ABA926 - hardware version](#)
- [0112/2///61987#ABB062 - fabrication number](#)
- [0112/2///61987#ABA951 - serial number](#)
- [0112/2///61987#ABN590 - URI of product instance](#)
- [0112/2///61987#ABN591 - URI of manufacturer](#)
- [0112/2///61987#ABP465 - URI of supplier](#)
- [0112/2///61987#ABB757 - date of manufacture](#)
- [0112/2///61987#ABB757 - date of manufacture](#)

### 4.4.5.3 VEC

Der Vehicle Electric Container (VEC)<sup>38</sup> ist ein Beispiel für ein Industriestandard-Datenmodell, das für den Austausch von elektrischen Systeminformationen in der Automobil- und Transportbranche entwickelt wurde. Er wird von ProSTEP iViP entwickelt und gepflegt, einem Konsortium, das sich auf die Interoperabilität im technischen Datenaustausch konzentriert. Die Verwendung von VEC-Definitionen ist grundsätzlich kostenfrei, erfordert jedoch die Einhaltung der VDA- und prostep ivip-Nutzungsbedingungen sowie der technischen Spezifikationen.

VEC bietet ein strukturiertes Format für die Darstellung und den Austausch von elektrischen Kabelbaumdaten, einschließlich Komponenten, Verbindungen, Signalen, Geometrien und Metadaten. Seine Aufgabe ist es, die nahtlose Kommunikation zwischen verschiedenen CAD- und PLM-Systemen (Product Lifecycle Management) zu ermöglichen. VEC ist in der VDA-Empfehlung 4968 und der ProSTEP iViP-Empfehlung PSI21 definiert, in Form eines standardisierten Informationsmodells, eines Datenverzeichnisses, sowie eines XML-Schemas und einer Ontologie in "RDF 1.1 Turtle"-Syntax, die im VWS wie folgt genutzt werden kann:

**Schritt 1:** Öffnen der Datei: <https://ecad-wiki.prostep.org/specifications/vec/v210/vec-2.1.0-ontology.ttl>

**Schritt 2:** Drücken Sie Strg+F und suchen Sie nach dem gewünschten Begriff. Wenn z. B. nach Definitionen zur Temperatur gesucht wird, einfach nach „Temperatur“ suchen und die Trefferliste auf Eignung analysieren.

**Schritt 3:** eines der Ergebnisse ist unten dargestellt. Die relevante VEC-Definition für die Suche wäre `vec:TemperatureInformation`, siehe Abbildung 4-8.

```
vec:TemperatureInformation
  rdfs:type          owl:Class;
  rdfs:comment      "Defines valid temperature ranges for a general technical part. A part can have different valid temperature ranges in different scenarios e.g. operating temperature, storage temperature, processing temperature, environment temperature. A class can own multiple TemperatureInformations but must not have more than one TemperatureInformations of the same TemperatureType.\n"@en;
  rdfs:label        "TemperatureInformation"@en .

vec:ConnectorLockingState_Closed
  rdfs:type          vec:ConnectorLockingState owl:NamedIndividual;
```

Abbildung 4-8: Suche nach Temperaturinformationen im VEC-Modell

Für die Verwendung innerhalb der VWS muss eine VWS-taugliche ID-Bildung definiert werden, z. B. in Form von IRIs (Internationalized Resource Identifier):

```
http://www.prostep.org/ontologies/ecad/2024/03/vec#TemperatureType
http://www.prostep.org/ontologies/ecad/2024/03/vec#InsulationSpecification
```

Listing 4-2: Referenzbeispiele zur Klassendefinitionen

```
http://www.prostep.org/ontologies/ecad/2024/03/vec#PrimaryPartType_Wire
http://www.prostep.org/ontologies/ecad/2024/03/vec#PrimaryPartType_PluggableTerminal
http://www.prostep.org/ontologies/ecad/2024/03/vec#TemperatureType_AmbientTemperature
```

Listing 4-3: Referenzbeispiel für Enumerationswerte

```
http://www.prostep.org/ontologies/ecad/2024/03/vec#itemVersionCompanyName
http://www.prostep.org/ontologies/ecad/2024/03/vec#partVersionPrimaryPartType
http://www.prostep.org/ontologies/ecad/2024/03/vec#partVersionPartNumber
http://www.prostep.org/ontologies/ecad/2024/03/vec#partVersionPreferredUseCase
http://www.prostep.org/ontologies/ecad/2024/03/vec#insulationSpecificationBaseColor
http://www.prostep.org/ontologies/ecad/2024/03/vec#insulationSpecificationMaterial
http://www.prostep.org/ontologies/ecad/2024/03/vec#conductorSpecificationCrossSectionArea
```

Listing 4-4: Referenzbeispiele für Properties

<sup>38</sup> <https://ecad.prostep.org/ontologies/2024/03/vec>

#### 4.4.6 Klassifizierungssysteme für Einheiten

**IEC 62720** bietet Definitionen für physikalische Einheiten, die auf den SI-Einheiten basieren und für den Datenaustausch in Lieferketten oder IoT-Systemen geeignet sind. Die IRDI-Referenzierungscodes können in der VWS über *semanticId* für einfache Einheitenreferenzierung verwendet werden. Die IEC 62720 ist auch in der IEC-CDD repräsentiert, und die Definitionen damit indirekt über URI-Verknüpfungen zugreifbar.

**ECLASS** bietet IRDI-Referenzcodes für physikalische SI-Einheiten an, verweist dabei aber grundsätzlich auf IEC 62720 als Definitionsquelle (siehe Abbildung 4-9).

<b>Bevorzugte Benennung</b>	millimetre
<b>Name</b>	mm
<b>Abkürzung</b>	mm
<b>Definition</b>	0,001-fold of the SI base unit metre
<b>SINotation</b>	mm
<b>DINNotation</b>	mm
<b>ECECode</b>	MMT
<b>IECClassification</b>	0112/2///62720#UAA862
<b>IRDI</b>	0173-1#05-AAA480#005

Abbildung 4-9: Datensatz zu einer Einheitsdefinition in ECLASS

Zur Anwendung von ECLASS und Einheitsdefinitionen in *Concept Descriptions* hat die IDTA eine Best-Practice-Guideline<sup>39</sup> veröffentlicht [16] und dort die folgenden wesentlichen Zuordnungen festgelegt:

- `<id>` soll mit der IRDI der Property befüllt werden. Die Referenz auf die *ConceptDescription* wird dann im Submodel oder der Property über *semanticId* gesetzt.
- `<definition>` soll mit dem Text aus der ECLASS-Definition der Property befüllt werden
- `<unit>` soll den Namen der Einheit aus der ECLASS-Definition enthalten (z.B. „mm“)
- `<unitId>` soll mit der IRDI der Einheit befüllt werden (z.B. „0173-1#05-AAA480“)

Da es sich bei ECLASS ein proprietäres kostenpflichtiges Referenzsystem handelt und keine Verlinkung auf eine frei im Internet verfügbare Definitionsquelle möglich ist, wird empfohlen, eine frei verfügbare und kostenlose Alternative zu verwenden, die diese Einschränkungen nicht hat.

**QUDT** (Quantities, Units, Dimensions, and Types)<sup>40</sup> ist ein weit verbreitetes und kostenfrei verfügbares Open-Source-Referenzierungssystem für physikalische Einheiten, Größen und Datentypen, das speziell für die semantische Modellierung und den Einsatz in digitalen Systemen entwickelt wurde. QUDT bietet eine detaillierte Ontologie für physikalische Größen, Einheiten, Dimensionen und Datentypen, die auf Standards wie dem SI-System, IEC 62720, ISO 80000 und anderen basiert. QUDT ist als RDF/OWL-Ontologie (Resource Description Framework / Web Ontology Language) implementiert, was es ideal für semantische Web-Anwendungen und digitale Zwillinge basierend auf der VWS macht. Ergänzend zu den IRDIs der IEC 62720 sind mit QUDT präzise, maschinenlesbare Verknüpfungen über URIs möglich (z. B. <http://qudt.org/vocab/unit/W> für Watt), sowie in die zugrundeliegende RDF-Ontologie (z. B. <http://qudt.org/vocab/unit#W> für Watt). In der folgenden Tabelle 4-13 findet sich eine Gegenüberstellung von Referenz-IDs für ausgewählte Einheiten.

<sup>39</sup> [https://industrialdigitaltwin.org/wp-content/uploads/2024/10/2024-10\\_IDTA\\_ECLASS\\_Semantic\\_Transport\\_ECLASS\\_in\\_VWS\\_1.0.pdf](https://industrialdigitaltwin.org/wp-content/uploads/2024/10/2024-10_IDTA_ECLASS_Semantic_Transport_ECLASS_in_VWS_1.0.pdf)

<sup>40</sup> <https://qudt.org>

Physikalische Einheit	ECLASS-Referenz	IEC-Referenz	QUDT-Referenz
Spannung (Volt)	0173-1#05-AAA153	<a href="http://qudt.org/vocab/unit/V">0112/2///62720#UAA296</a>	<a href="http://qudt.org/vocab/unit/V">http://qudt.org/vocab/unit/V</a>
Stromstärke (Ampere)	0173-1#05-AAA220	<a href="http://qudt.org/vocab/unit/A">0112/2///62720#UAA101</a>	<a href="http://qudt.org/vocab/unit/A">http://qudt.org/vocab/unit/A</a>
Leistung (Watt)	0173-1#05-AAA282	<a href="http://qudt.org/vocab/unit/W">0112/2///62720#UAA306</a>	<a href="http://qudt.org/vocab/unit/W">http://qudt.org/vocab/unit/W</a>
Temperatur (Kelvin)	0173-1#05-AAA064	<a href="http://qudt.org/vocab/unit/K">0112/2///62720#UAA185</a>	<a href="http://qudt.org/vocab/unit/K">http://qudt.org/vocab/unit/K</a>
Temperatur (Celsius)	0173-1#05-AAA567	<a href="http://qudt.org/vocab/unit/DEG_C">0112/2///62720#UAA033</a>	<a href="http://qudt.org/vocab/unit/DEG_C">http://qudt.org/vocab/unit/DEG_C</a>
Temperatur (Fahrenheit)	0173-1#05-AAA739	<a href="http://qudt.org/vocab/unit/DEG_F">0112/2///62720#UAA039</a>	<a href="http://qudt.org/vocab/unit/DEG_F">http://qudt.org/vocab/unit/DEG_F</a>
Distanz (Millimeter)	0173-1#05-AAA480	<a href="http://qudt.org/vocab/unit/Millim">0112/2///62720#UAA862</a>	<a href="http://qudt.org/vocab/unit/Millim">http://qudt.org/vocab/unit/Millim</a>
Distanz (Meter)	0173-1#05-AAA551	<a href="http://qudt.org/vocab/unit/M">0112/2///62720#UAA726</a>	<a href="http://qudt.org/vocab/unit/M">http://qudt.org/vocab/unit/M</a>
Kraft (Newton)	0173-1#05-AAA561	<a href="http://qudt.org/vocab/unit/N">0112/2///62720#UAA235</a>	<a href="http://qudt.org/vocab/unit/N">http://qudt.org/vocab/unit/N</a>
Masse (Kilogramm)	0173-1#05-AAA731	<a href="http://qudt.org/vocab/unit/KiloGM">0112/2///62720#UAA594</a>	<a href="http://qudt.org/vocab/unit/KiloGM">http://qudt.org/vocab/unit/KiloGM</a>
Masse (Gramm)	0173-1#05-AAA728	<a href="http://qudt.org/vocab/unit/GM">0112/2///62720#UAA465</a>	<a href="http://qudt.org/vocab/unit/GM">http://qudt.org/vocab/unit/GM</a>
Fläche (mm <sup>2</sup> )	0173-1#05-AAA295	<a href="https://qudt.org/vocab/unit/Millim2">0112/2///62720#UAA871</a>	<a href="https://qudt.org/vocab/unit/Millim2">https://qudt.org/vocab/unit/Millim2</a>
Stücke (Pieces)	unknown	<a href="https://qudt.org/vocab/quantitykind/Count">0112/2///62720#UAD146</a>	<a href="https://qudt.org/vocab/quantitykind/Count">https://qudt.org/vocab/quantitykind/Count</a>

Tabelle 4-13: Referenzbeispiele für physikalische Einheiten mit ECLASS, IEC und QUDT

Wie im Best-Practice-Guideline der IDTA [16] beschrieben, muss in der VWS der Umweg über die IEC61360 Data Descriptions bzw. Concept Descriptions gegangen werden, um einem Property eine Einheit zuweisen zu können. Dort müssen zusätzlich zur *referenceId* mindestens die Einheitenbezeichner oder -symbole hinterlegt werden, damit die Viewer-Tools diese anzeigen können, siehe Abbildung 4-10.

MLP	ConceptDescription
"Connect_tech" → Crimping	<b>Referable:</b>
Prop "CONNECT_WIRE_SIZE" = 0.12 - 0.14 [Square Millimetre] @[GREATER_EQUAL_1=0.3	idShort: vec_terminalCurrentInformationCurrentRange
Prop "Contact_family" = MLK 1,2	<b>HasExtension:</b>
Prop "CONNECT_TERMINAL_SIZE" = 1.2 x 0.6 [Millimeter]	<b>Identifiable:</b>
Prop "CONNECT_SURFACE_LAMINA" = Ag	id: https://ecad.prostep.org/ontologies/2024/03/vec#terminalCurrentInformationC
Prop "CONNECT_OPERATING_TEMPERATURE" = 170 [degreeCelsius]	id (Base64): aHR0cHM6Ly9lY2FkLnByb3NOZXhAub3JnL29udG9sb2dpZmVvMjAyNC8wMy92Zl
Prop "CONNECT_MATING_FORCE_TERMINAL" = 2.5 [Newton]	<b>HasDataSpecification (records of embedded data specification):</b>
Prop "CONNECT_CURRENT_CARRYING_CAPACITY_AT_80DEGREE" = 7 [Ampere]	<b>dataSpec.[0] / Reference:</b>
Prop "CONNECT_UNMATING_FORCE_TERMINAL" = 1.0 [Newton]	dataSpec.[0]: (GlobalReference) http://admin-shell.io/DataSpecificationTemplates/DataSpecif
SMC "FurtherInformation" (2 elements)	<b>dataSpec.[0] / Content:</b>
	<b>Data Specification Content IEC61360:</b>
	preferredName: [en] Ampere
	unit: Ampere
	unitId: (GlobalReference) 0112/2///62720#UAA101
	referredSem.Id: (GlobalReference) https://cdd.iec.ch/cdd/iec62720/iec6
	sourceOfDef: https://qudt.org/2.1/vocab/unit#A
	symbol: A
	dataType: REAL_MEASURE
	definition: [en] SI unit of electric current defined by taking the fixed numerical value of the
	[de] SI-Einheit des elektrischen Stroms, die definiert wird, indem der feste nume
	levelType: <input type="checkbox"/> min <input checked="" type="checkbox"/> max <input type="checkbox"/> nom <input type="checkbox"/> typ
	<b>Property</b>
	valueType: xs:double
	value: 7

Abbildung 4-10: Beispiel zur Einheiten-Referenzierung in einer Concept Description

#### 4.4.7 Klassifizierungssysteme für Dokumente

Das SM *HandoverDocumentation* nutzt die im Folgenden kurz beschriebenen Klassifizierungssysteme für Dokumente, siehe Abbildung 4-11.



Abbildung 4-11: Klassifizierung des Dokuments „Drawing“ im SM *HandoverDocumentation*

##### 4.4.7.1 VDI 2770

VDI 2770 Blatt 1:2020-04 definiert Mindestanforderungen an digitale Herstellerinformationen für die Prozessindustrie, ist jedoch branchenübergreifend anwendbar. Der Zweck ist, die digitale Dokumentation von Maschinen und Anlagen zu standardisieren, insbesondere für die Übergabe von Herstellerinformationen an Betreiber.

In der SMC *DocumentClassificationItem* ist der Wert von *ClassificationSystem* auf *VDI2770 Blatt 1:2020* zu setzen und der Wert von *ClassId* auf einen der Zahlencodes aus der nachstehenden Tabelle 4-14 einzustellen:

ClassID	ClassName (EN)	ClassName (DE)
01-01	Identification	Identifikation
02-01	Technical specification	Technische Spezifikation
02-02	Drawings, plans	Zeichnungen, Pläne
02-03	Assemblies	Bauteile
02-04	Certificates, declarations	Zeugnisse, Zertifikate, Bescheinigungen
03-01	Commissioning, de-commissioning	Montage, Demontage
03-02	Operation	Bedienung
03-03	General safety	Allgemeine Sicherheit
03-04	Inspection, maintenance, testing	Inspektion, Wartung, Prüfung
03-05	Repair	Instandsetzung
03-06	Spare parts	Ersatzteile
04-01	Contract documents	Vertragsunterlagen

Tabelle 4-14: *DocumentClassification* according to VDI 2770 Blatt 1: 2020

##### 4.4.7.2 IEC 61355

Die IEC 61355-1:2008 befasst sich mit der Klassifikation und Kennzeichnung von Dokumenten für Anlagen, Systeme und Ausrüstungen. Sie bietet ein standardisiertes System zur Strukturierung und Kategorisierung technischer Dokumentationen, um deren Verwaltung, Auffindbarkeit und Austausch zu erleichtern. Der Fokus liegt auf Dokumenten in der Industrie, insbesondere in der Elektrotechnik, Anlagenbau und Prozessindustrie.

Die IEC 61355-Datenbank<sup>41</sup> kann als direkte Referenz für die Klassifikation von Dokumentenarten (DCC-Codes) und Metadaten dienen, um eine einheitliche Dokumentenverwaltung zu gewährleisten, z.B.

- Data sheet (DA): [https://std.iec.ch/iec61355/iec61355.nsf/\(SysSymbolEN\)/D00178](https://std.iec.ch/iec61355/iec61355.nsf/(SysSymbolEN)/D00178)
- Operating manual (DC): [https://std.iec.ch/iec61355/iec61355.nsf/\(SysSymbolEN\)/D00034](https://std.iec.ch/iec61355/iec61355.nsf/(SysSymbolEN)/D00034)
- Requirement specification (EC): [https://std.iec.ch/iec61355/iec61355.nsf/\(SysSymbolEN\)/D00035](https://std.iec.ch/iec61355/iec61355.nsf/(SysSymbolEN)/D00035)
- Technical Drawing (ED): [https://std.iec.ch/IEC61355/iec61355.nsf/\(sysSymbolEn\)/D00032](https://std.iec.ch/IEC61355/iec61355.nsf/(sysSymbolEn)/D00032)

Im Folgenden eine tabellarische Zusammenstellung der in der IEC 61355 definierten Codes<sup>42</sup>. Der Wert von „ClassificationSystem“ ist auf „IEC 61355-1:2008“ zu setzen und der Wert von „ClassId“ auf einen der Buchstabencodes aus der nachstehenden Tabelle 4-14 einzustellen:

ClassID	ClassName (DE)	ClassName (EN)
A	Dokumentationsbeschreibende Dokumente	Documentation describing documents
AA	Verwaltungstechnische Dokumente	Administrative documents
AB	Listen (Dokumente betreffend)	Lists (regarding documents)
AC	Erläuternde Dokumente (Dokument betreffend)	Explanatory documents (regarding documents)
B	Managementdokumente	Management documents
BB	Berichte	Reports
BC	Schriftwechsel	Correspondence
BD	Projektleitungsdokumente	Project control documents
BE	Ressourcenplanungsdokumente	Resource planning documents
BF	Versand-, Lager- und Transportdokumente	Dispatch, storage and transport documents
BG	Standortplanungs- und Standortorganisationsdokumente	Site planning and site organization documents
BH	Dokumente zum Änderungswesen	Documents regarding changes
BS	Objektschutzdokumente	Security documents
BT	Schulungsdokumente	Training specific documents
C	Vertragliche und nicht-technische Dokumente	Contractual and non-technical documents
CA	Anfrage-, Kalkulations- und Angebotsdokumente	Inquiry, calculation and offer documents
CB	Genehmigungsdokumente	Approval documents
CC	Vertragliche Dokumente	Contractual documents
CD	Bestell- und Lieferdokumente	Order and delivery documents
CE	Rechnungsdokumente	Invoice documents
CF	Versicherungsdokumente	Insurance documents
CG	Gewährleistungsdokumente	Warranty documents
CH	Gutachten	Expertises
D	Dokumente mit allgemeiner technischer Information	General technical information documents
DA	Datenblätter	Data sheets
DB	Erläuternde Dokumente	Explanatory documents
DC	Anleitungen und Handbücher	Instructions and manuals
DD	Technische Berichte	Technical reports
DE	Kataloge, Werbeschriften	Catalogues Advertising documents
DF	Technische Veröffentlichungen	Technical publications
E	Dokumente für technische Anforderungen und Auslegung	Technical requirement and dimensioning documents

<sup>41</sup> <https://std.iec.ch/iec61355/iec61355.nsf>

<sup>42</sup> [https://en.wikipedia.org/wiki/IEC\\_61355](https://en.wikipedia.org/wiki/IEC_61355)

<b>EA</b>	Dokumente über gesetzliche Anforderungen	Legal requirement documents
<b>EB</b>	Normen und Richtlinien	Standards and regulations
<b>EC</b>	Technische Spezifikations- / Anforderungsdokumente	Technical specification / requirement documents
<b>ED</b>	Dimensionierungsdokumente	Dimensioning documents
<b>F</b>	Funktionsbeschreibende Dokumente	Function describing documents
<b>FA</b>	Funktionsübersichtsdokumente	Functional overview documents
<b>FB</b>	Fließschemata	Flow diagrams
<b>FC</b>	Dokumente der MMS-Gestaltung (Mensch-Machine-Schnittstelle)	MMI layout documents (MMI = man-machine interface)
<b>FE</b>	Funktionsbeschreibungen	Function descriptions
<b>FF</b>	Funktionsschaltpläne	Function diagrams
<b>FP</b>	Signalbeschreibungen	Signal descriptions
<b>FQ</b>	Einstellwertdokumente	Setting value documents
<b>FS</b>	Schaltkreisdokumente	Circuitry documents
<b>FT</b>	Softwarespezifische Dokumente	Software specific documents
<b>L</b>	Ortsbeschreibende Dokumente	Location documents
<b>LA</b>	Erschließungs- und Vermessungsdokumente	Exploitation and survey documents
<b>LB</b>	Erdbau- und Fundamentbaudokumente	Earthwork and foundation work documents
<b>LC</b>	Rohbaudokumente	Building carcass documents
<b>LD</b>	Dokumente, die Orte an Standorten beschreiben	On-site location documents
<b>LH</b>	Orte in Gebäuden (Schiffen, Flugzeugen, etc.) beschreibende Dokumente	In-building location documents (also applied for ships, aircraft, etc.)
<b>LU</b>	Orte in/auf Einrichtungen beschreibende Dokumente	In/on-equipment location documents
<b>M</b>	Verbindungsbeschreibende Dokumente	Connection describing documents
<b>MA</b>	Verbindungsbezogene Dokumente	Connection documents
<b>MB</b>	Verkabelungs- und Rohrleitungsdokumente	Cabling or piping documents
<b>P</b>	Objektlisten	Object listings
<b>PA</b>	Materiallisten	Material lists
<b>PB</b>	Teilelisten	Parts lists
<b>PC</b>	Stücklisten	Item lists
<b>PD</b>	Produktlisten und Produkttypenlisten	Product lists and product type lists
<b>PF</b>	Funktionslisten	Function lists
<b>PL</b>	Ortslisten	Location lists
<b>Q</b>	Qualitätsmanagementdokumente und sicherheitsbeschreibende Dokumente	Quality management documents; safety-describing documents
<b>QA</b>	Qualitätsmanagementdokumente	Quality management documents
<b>QB</b>	Sicherheitsbeschreibende Dokumente	Safety-describing documents
<b>QC</b>	Qualitätsnachweisdokumente	Quality verifying documents
<b>T</b>	Dokumente zur Beschreibung geometrischer Formen	Geometry-related documents
<b>TA</b>	Entwurfszeichnung	Planning drawings
<b>TB</b>	Konstruktionszeichnungen	Construction drawings
<b>TC</b>	Fertigungs- und Errichtungszeichnungen	Manufacturing and erection drawings
<b>TL</b>	Anordnungszeichnung	Arrangement documents
<b>W</b>	Betriebliche Protokolle und Aufzeichnungen	Operation records
<b>WA</b>	Einstellwertdokumente	Set point documents
<b>WT</b>	Logbücher	Logbooks

Tabelle 4-15: Document classification according to IEC 61355

## 4.5 AP 13.5 – Meta-Level AAS Designer

Im Zuge des Teilprojekts 12 erfolgte die begleitende Weiterentwicklung des AAS-Designers der Fa. Meta-level, um Komponenten-VWS anwenderfreundlicher anlegen, editieren und validieren zu können. Dabei wurde auch ein erstes Anwenderhandbuch für den AAS-Designer erstellt<sup>43</sup>.

Die im Folgenden beschriebenen Features wurden im Kontext des TP12 angeregt und implementiert.

### 4.5.1 Import von semantischen IDs aus Referenzkatalogen

Zur Sicherstellung der semantischen Interoperabilität sind die Verwendungen von allgemein bekannten semantischen IDs aus Referenzkatalogen wie ECLASS von entscheidender Bedeutung. Um die Zuweisung von solchen IDs für den Anwender so einfach wie möglich zu gestalten, wurde die Möglichkeit geschaffen, IDs direkt bei der Erstellung aus einer Auswahlliste zu wählen und zu filtern. Spezifisch für den Einsatz in der Leitungssatzfertigung wurde zudem die Ontologie vom VEC als Referenzsystem hinzugefügt.

Für die folgenden Elemente können Semantische IDs nun direkt ausgewählt werden:

#### Properties

Für *Properties* (*Eigenschaft: valueId*) können Wertereferenzen aus Referenzkatalogen importiert werden, siehe Abbildung 4-12. Abbildung 4-13 zeigt den Auswahldialog für mögliche Wertereferenzen aus dem VEC-Katalog.

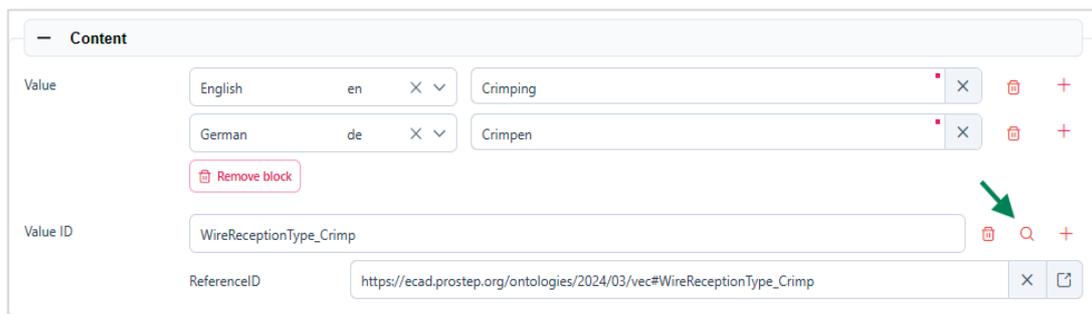


Abbildung 4-12: Suche nach Wertereferenzen in Referenzkatalogen

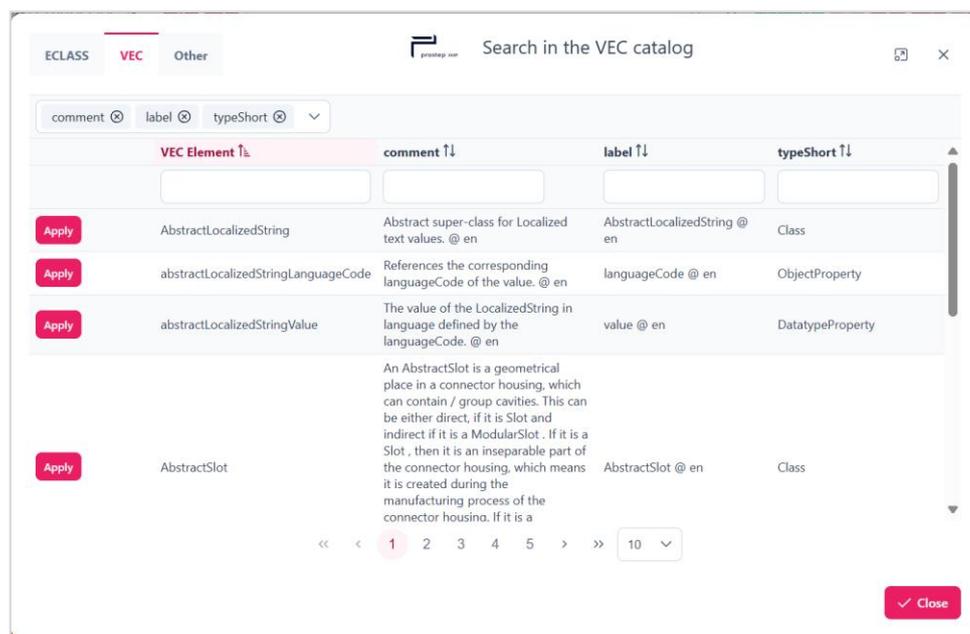


Abbildung 4-13: Import von Wertereferenzen aus der VEC-Ontologie

<sup>43</sup> [https://github.com/VWS4LS/vws4ls-subproject-results/tree/main/TP12/VWS4LS\\_AAS\\_Designer\\_Manual.pdf](https://github.com/VWS4LS/vws4ls-subproject-results/tree/main/TP12/VWS4LS_AAS_Designer_Manual.pdf)

## Concept Descriptions

In *ConceptDescriptions* (Eigenschaft: *unitId*) können Einheitenreferenzen aus Referenzkatalogen zugewiesen und Inhalte daraus importiert werden (ECLASS, QUDT, SI-Units). In Abbildung 4-14 ff. werden die unterschiedlichen Auswahlmöglichkeiten für Einheiten aus den Referenzkatalogen dargestellt.

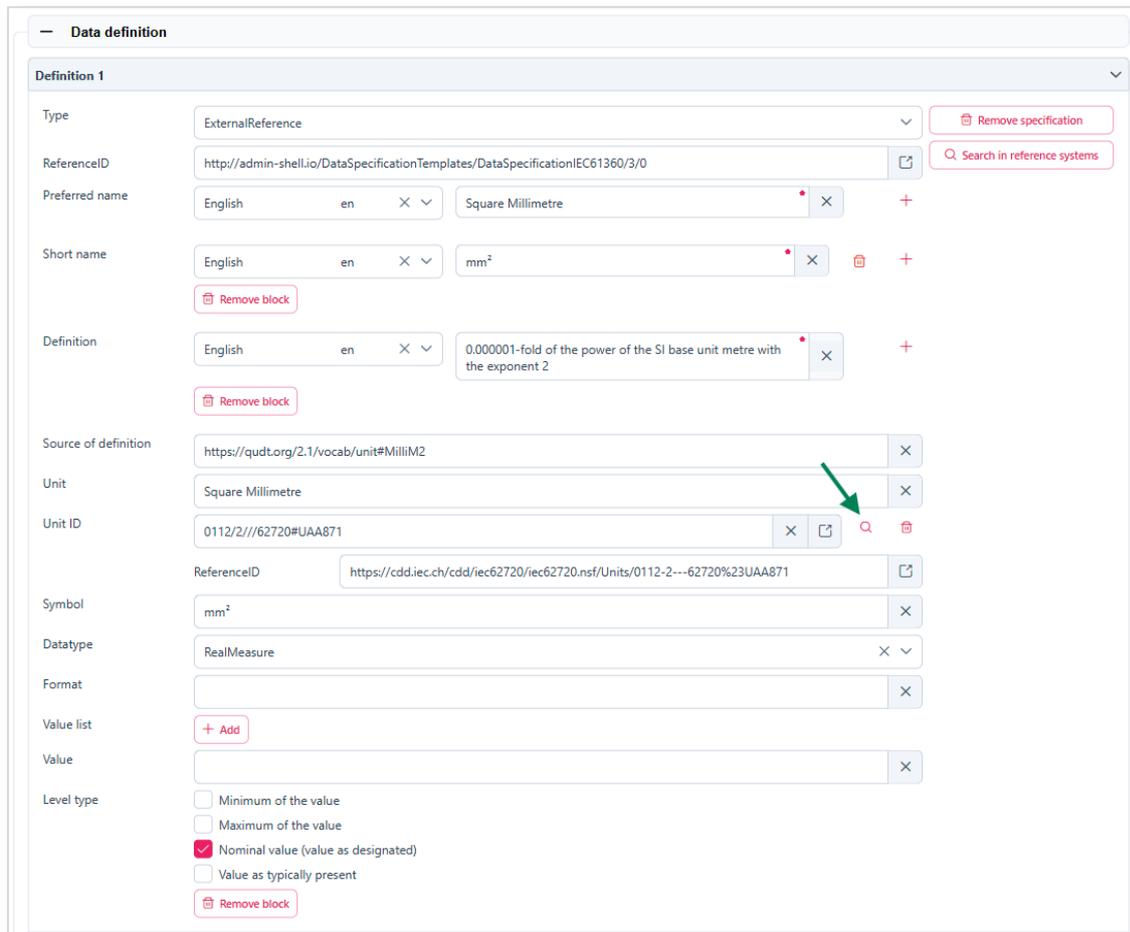
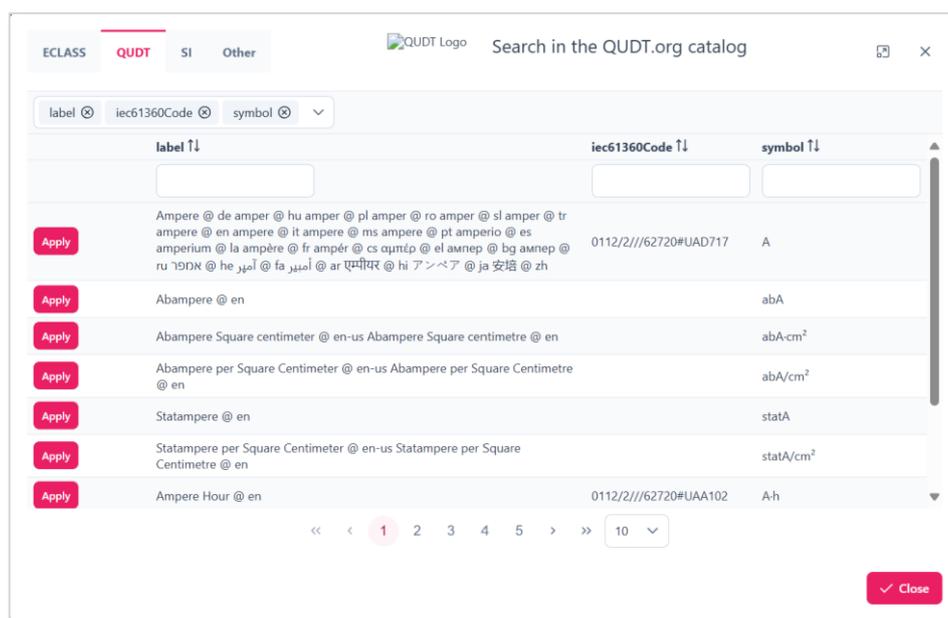


Abbildung 4-14: Suche nach Einheitenreferenzen in Referenzkatalogen



label	iec61360Code	symbol
Ampere @ de ampere @ hu amper @ pl amper @ ro amper @ sl amper @ tr ampere @ en ampere @ it ampere @ ms ampere @ pt amperio @ es amperium @ la ampère @ fr ampér @ cs amptér @ el amnep @ bg amnep @ ru ампер @ he אמפר @ fa آمپر @ ar एम्पीयर @ hi आम्पेरा @ ja アンペア @ zh 安培	0112/2///62720#UAD717	A
Abampere @ en		abA
Abampere Square centimeter @ en-us Abampere Square centimetre @ en		abA-cm <sup>2</sup>
Abampere per Square Centimeter @ en-us Abampere per Square Centimetre @ en		abA/cm <sup>2</sup>
Statampere @ en		statA
Statampere per Square Centimeter @ en-us Statampere per Square Centimetre @ en		statA/cm <sup>2</sup>
Ampere Hour @ en	0112/2///62720#UAA102	A-h

Abbildung 4-15: Import von Einheiten aus QUDT.org

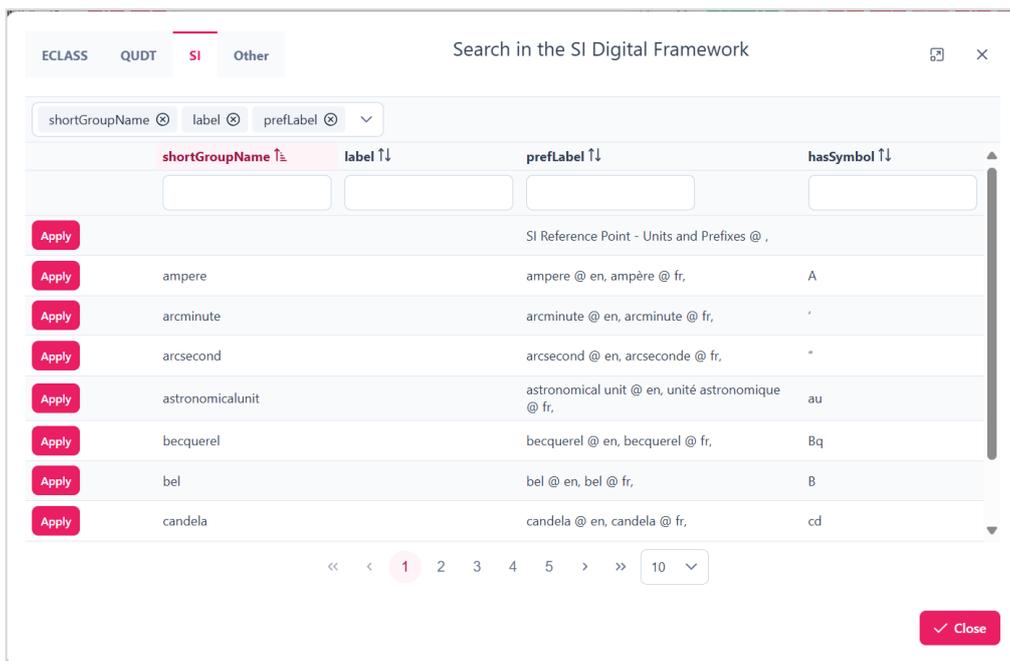


Abbildung 4-16: Import von Einheiten aus SI-Units

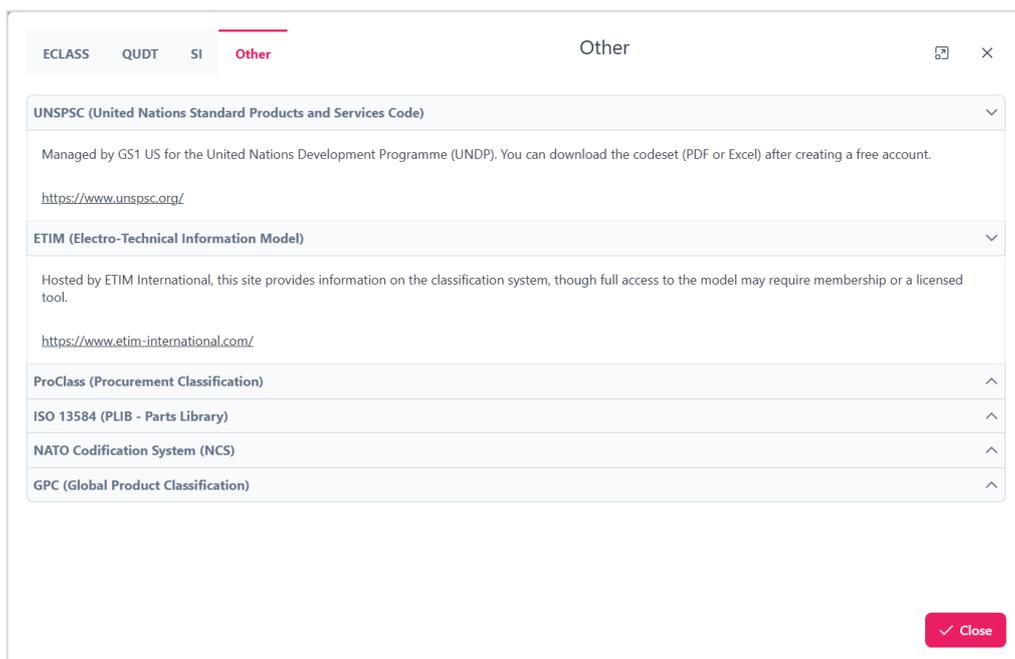


Abbildung 4-17: Verweise auf weitere Referenzkataloge

### 4.5.2 Handover Documentation-Wizard

Die Bereitstellung von Übergabedokumentationen in Verwaltungsschalen ist einer der grundlegendsten Funktionen, die durch Anwender auf verständliche Weise genutzt werden. Um den Einstieg so einfach wie möglich zu gestalten, bietet es sich an, die Dokumentenbereitstellung direkt über die Weboberfläche zu ermöglichen.

Hierfür wurde die vereinfachte Anlage von Elementen im Teilmodell *HandoverDocumentation hinzugefügt*. So ist es jetzt möglich Dokumente direkt über die Weboberfläche hochzuladen und mit semantischen Informationen anzureichern. Abbildung 4-18 zeigt exemplarisch die Eingabemaske für die Neuanlage von Dokumenten.

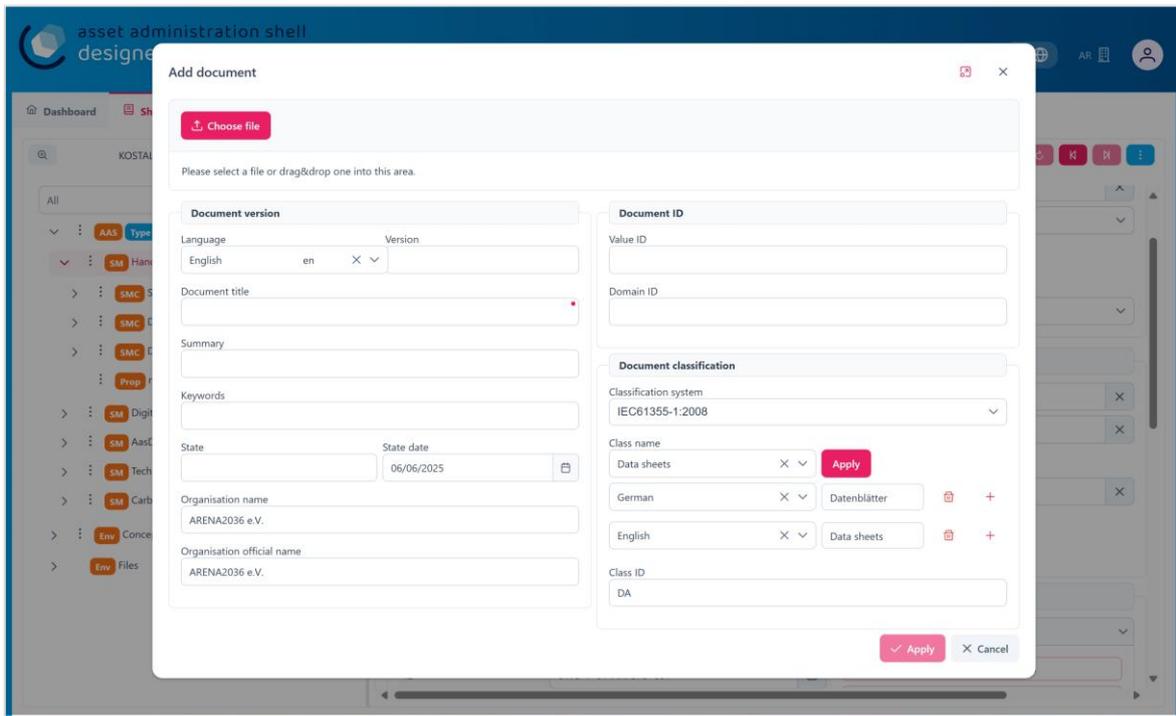


Abbildung 4-18: Wizard HandoverDocumentation im AAS-Designer

### 4.5.3 Auswahl des ContentType

Bei angehängten oder verlinkten Dateireferenzen ist es wichtig, den korrekten Dateityp (mime type) anzugeben, siehe Abbildung 4-19. Deshalb wurde ein Feature implementiert, welches anhand der Dateinamenserweiterung einen passenden MimeType<sup>44</sup> vorauswählt, z.B.:

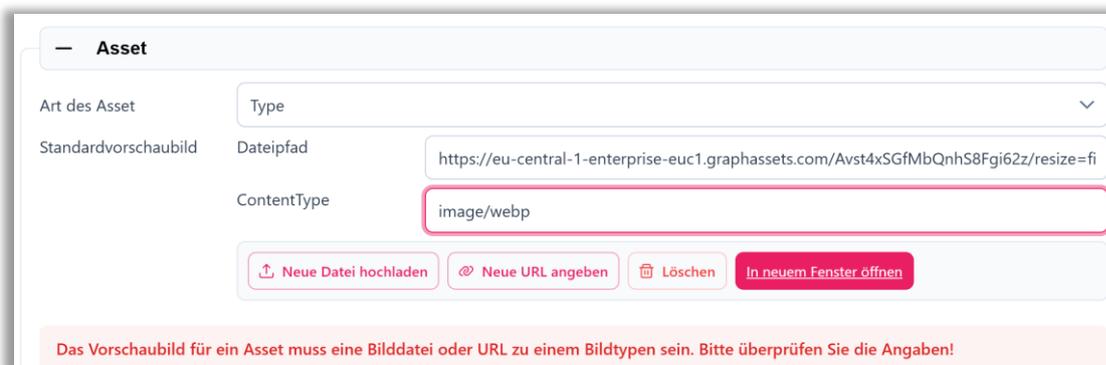


Abbildung 4-19: ContentType

<sup>44</sup> <https://www.iana.org/assignments/media-types/media-types.xhtml>

#### 4.5.4 ID-Validation

Die konsistente Befüllung und Pflege von IDs in einer Verwaltungsschale ist ein fehlerträchtiger Prozess, insbesondere wenn dieser manuell ausgeführt wird und keine geeignete Toolunterstützung vorhanden ist. Für den AAS-Designer wurde deswegen eine Konsistenzprüfung mit dem Feature "Validate IDs" ergänzt. Abbildung 4-20 zeigt exemplarisch die Ergebnisse eines solchen Prüfvorgangs.

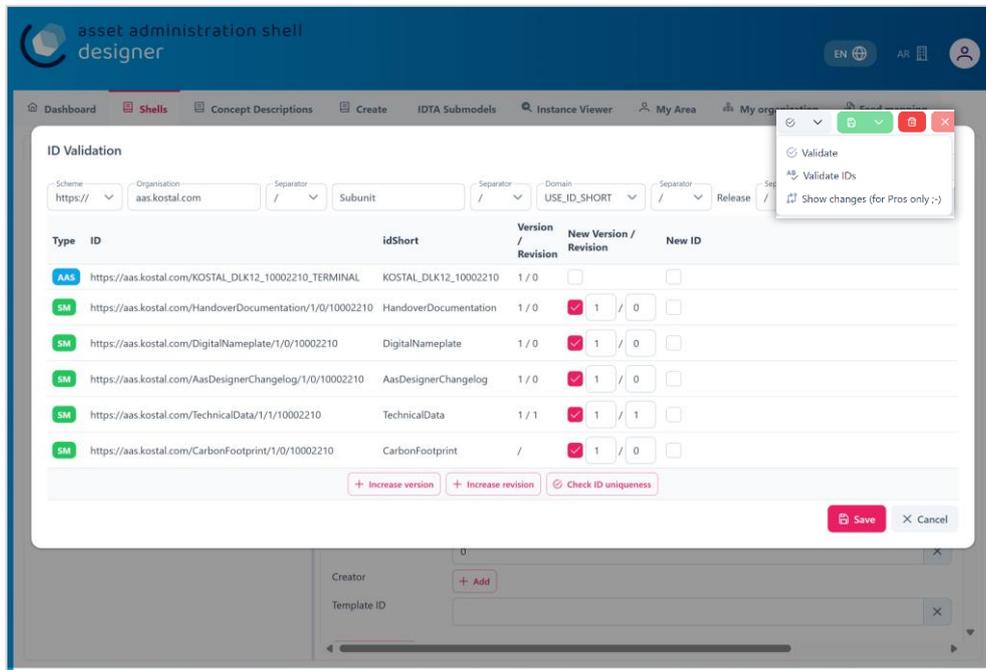


Abbildung 4-20: Ergebnis einer ID-Validierung im AAS-Designer

#### 4.5.5 Auswahllisten für Produkthierarchien

Produkthierarchien sind per Definition mehrstufig und insbesondere bei umfangreichen Verwaltungsschalen-Repositories ohne Hilfsmittel nur schwer zu verwalten. Um die Handhabung zu vereinfachen, wurde der AAS-Designer um die Möglichkeit zum Verwalten von Eigenschaften der Produkthierarchie im *DigitalNameplate* (*ManufacturerProductRoot*, *ManufacturerProductFamily*, *ManufacturerProductDesignation*) ergänzt. Die folgende Abbildung 4-21 zeigt exemplarisch eine Definition der Produktfamilien.

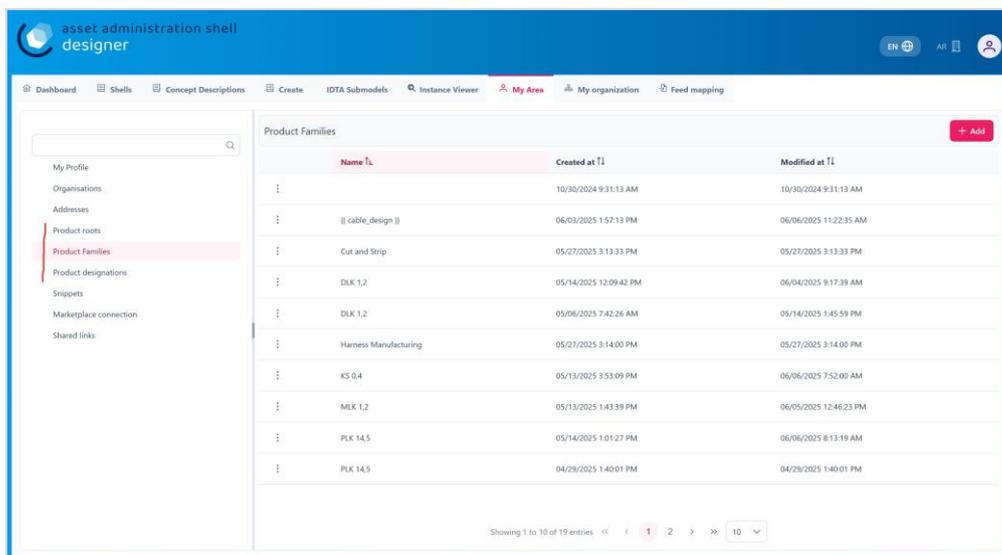


Abbildung 4-21: Vorauswahllisten Produkthierarchie im AAS-Designer

#### 4.5.6 ConceptDescription-Management

Um die semantische Eindeutigkeit zu gewährleisten, werden eine Reihe von *ConceptDescriptions* verwendet. Um die Einrichtung neuer Werte und den damit verbundenen Suchaufwand für *ConceptDescriptions* zu minimieren, muss die Auffindbarkeit sichergestellt werden.

Hierfür wurde die Verwaltung von *ConceptDescriptions* im AAS-Designer deutlich verbessert, indem diese im ausgewählte AAS-Repository über eine such- und sortierbare Tabelle bearbeitet und verwaltet werden können. Abbildung 4-22 zeigt exemplarisch eine Liste von *Concept Descriptions* mit den entsprechenden Suchschaltflächen und Sortioptionen im AAS-Designer.

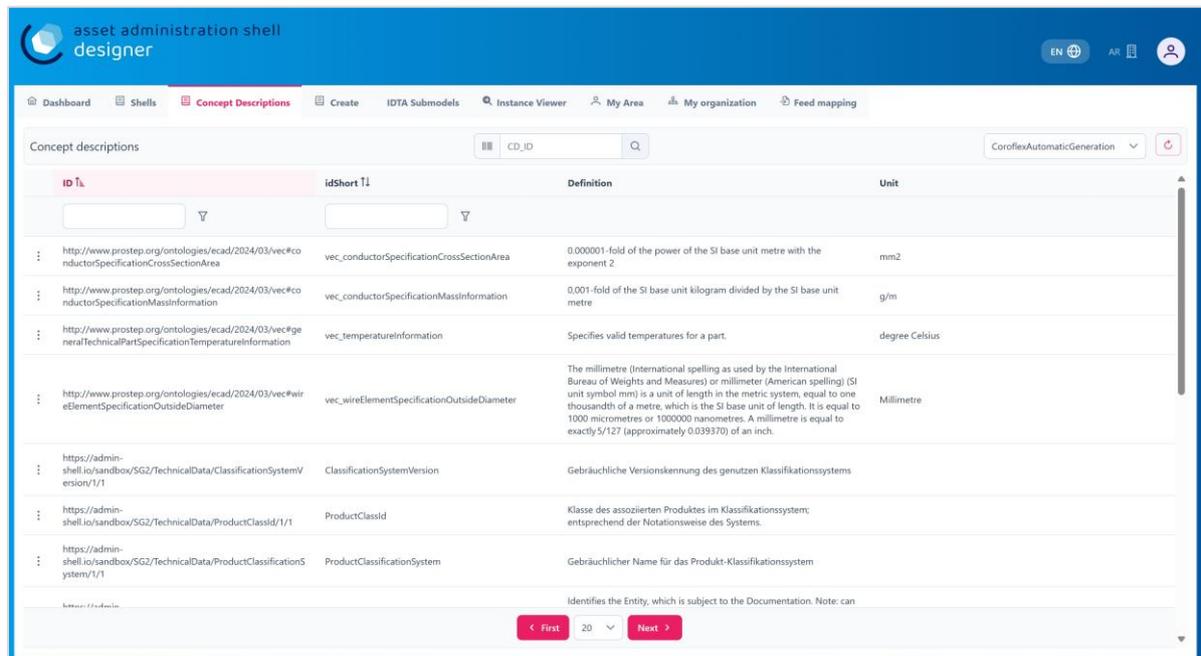


Abbildung 4-22: *ConceptDescriptions* im AAS-Designer

## 4.6 AP 13.6 – XITASO Mnestix Viewer

Im Zuge des Teilprojekts 12 erfolgte die begleitende Weiterentwicklung des Eclipse-basierten OpenSource-VWS-Viewers „Mnestix“<sup>45</sup> um für Produktkataloge üblichen Such- und Filterfunktionen – insbesondere die Weiterentwicklung einzelner Viewer-Features zur optimalen Darstellung der Submodelle „*TechnicalData*“, „*HandoverDocumentation*“ und „*DigitalNameplate*“ in einem ansprechenden Produktkatalog-Erscheinungsbild. Der Mnestix-Viewer wurde daher auf vereinfachte Darstellung für den Anwendungsfall „Produktkatalog“ optimiert (siehe auch Kapitel 0).

Die komplette Dokumentation, wie beispielsweise die Konfiguration eines sicheren Logins und eines Rollen-basierten Zugriffssystem<sup>46</sup>, findet sich [im respektiven GitHub Repository](#)<sup>47</sup>. Der Quellcode für das in TP12 entstandene Artefakt findet sich auf dem „Branch“ „1.5.0-product-catalog“ des GitHub Repositories<sup>48</sup>.

### 4.6.1 Funktionsweise

Technologisch gesehen ist der Mnestix AAS Viewer kein ausschließlich browserbasierter AAS-Client, d.h. der Webbrowser kommuniziert nicht direkt über die REST-API mit dem AAS-Server, sondern indirekt über das sog. Backend-for-Frontend. Dies schränkt potenziell die Anwendungsfreiheit ein, da nicht ohne Vorkonfiguration des Backends auf beliebige AAS-Server zugegriffen werden kann.

Dieser Ansatz ermöglicht es Unternehmen zu spezifizieren, auf welche Verwaltungsschalen die Nutzer am Ende Zugriff haben sollen. So können bspw. auch VWS Repositories angebunden werden, welche nur im internen Firmennetz erreichbar sind und ohne direkt die API zur Verfügung zu stellen. Zudem werden so Probleme mit „Cross-Origin Resource Sharing (CORS)“ umgangen, um den Nutzern einen möglichst einfachen Umgang mit der Software zu ermöglichen.

Produkte haben eine weltweit eindeutige **assetId**, die einer oder mehreren Anlagenverwaltungsschalen zugeordnet werden kann. Für viele Anwendungsszenarien ist es sinnvoll, zusätzlich einen DNS-Zugang über die **assetId** einzurichten und damit die Datenabfrage aus herkömmlichen Browseranwendungen zu ermöglichen, siehe Abbildung 4-23.

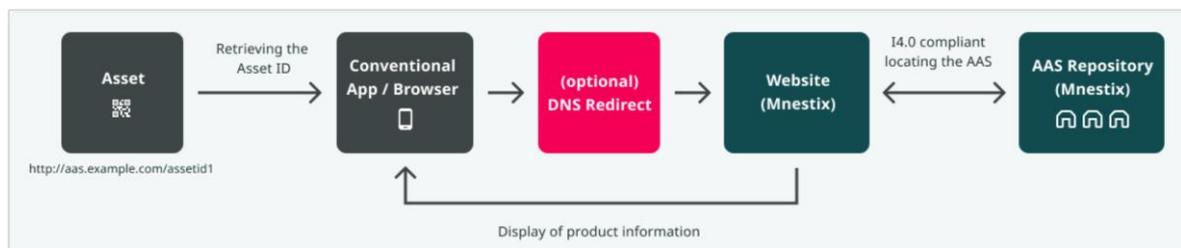


Abbildung 4-23: VWS Repository, Asset ID und DNS

Zum Beispiel wird die **assetId** <https://asset.coroplast.de/9-2611-35> umgeleitet nach <https://vws4ls.dev.mnestix.xitaso.net/asset/http%3A%2F%2Fasset.coroplast.de%2F9-2611-35>.

Dort wird die **aaSId** [https://aas.coroplast.de/COROFLEX\\_9\\_2611\\_FHLR2GCB2G\\_35](https://aas.coroplast.de/COROFLEX_9_2611_FHLR2GCB2G_35) ermittelt und im **Viewer-Link** Base64-codiert übergeben: <https://vws4ls.dev.mnestix.xitaso.net/en/product/aHR0cHM6Ly9hYXMuY29yb3BsYXN0LmRIL0NPUk9GTE-VYXzlfMjYxMV9GSExSMkdDQjJHXzM1>

<sup>45</sup> <https://github.com/eclipse-mnestix/mnestix-browser>

<sup>46</sup> <https://github.com/eclipse-mnestix/mnestix-browser/wiki/Role-Based-Access-Control>

<sup>47</sup> <https://github.com/eclipse-mnestix/mnestix-browser/blob/1.5.0-product-catalog/wiki/product-catalog.md>

<sup>48</sup> <https://github.com/eclipse-mnestix/mnestix-browser/tree/1.5.0-product-catalog>

## 4.6.2 Bedienung

Die Bedienung des Mnestix-Produktkatalog-Viewers ist im Folgenden erläutert, wobei die folgenden grundsätzlichen Bedienelemente im Mnestix-Viewer zur Verfügung stehen (siehe Abbildung 4-24):

1. Hamburger-Menü: Auswahl zwischen **Dashboard**, **AAS-Liste**, **Templates** und **Einstellungen**
2. Breadcrumb mit Produkthierarchie und anwählbaren Bestandteilen
3. Sprachauswahl
4. Umschaltung zwischen **AAS View** und **Product View**.
5. AASX-Datei herunterladen

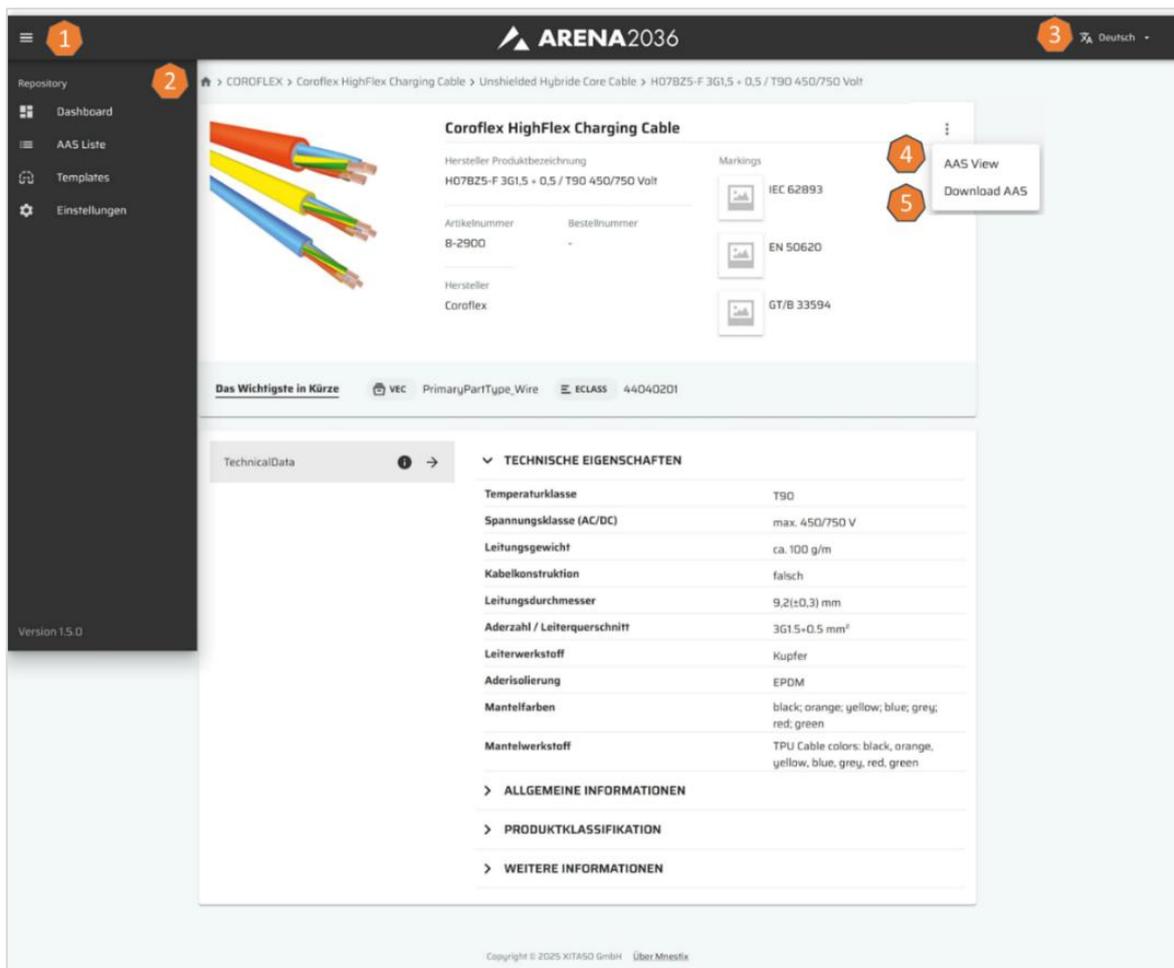
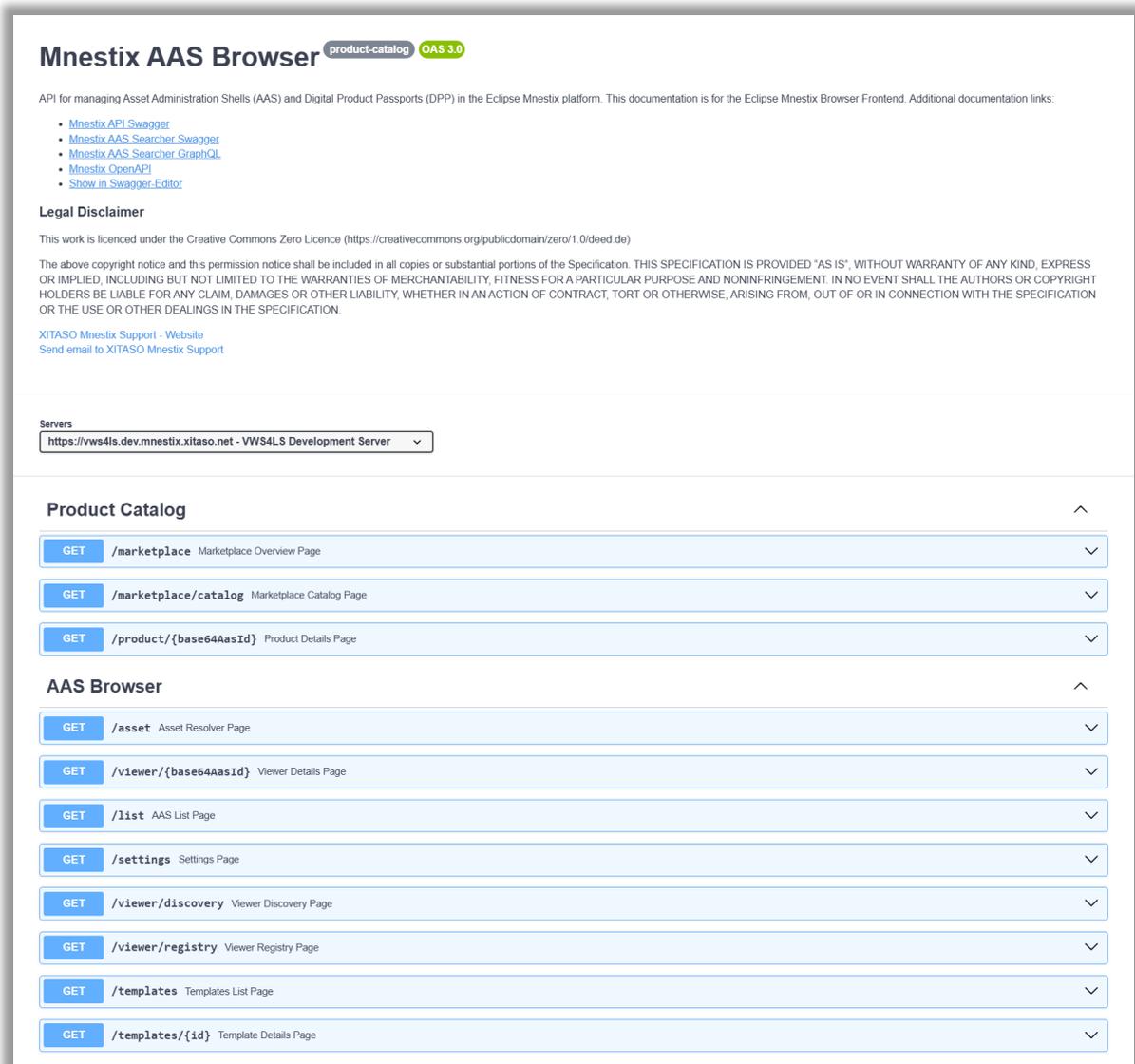


Abbildung 4-24: Mnestix Bedienelemente

Eine [openapi-Spezifikation](#)<sup>49</sup> wurde für Mnestix erstellt, siehe [im Swagger-Editor](#)<sup>50</sup> in Abbildung 4-25: Mnestix API



**Mnestix AAS Browser** product-catalog OAS 3.0

API for managing Asset Administration Shells (AAS) and Digital Product Passports (DPP) in the Eclipse Mnestix platform. This documentation is for the Eclipse Mnestix Browser Frontend. Additional documentation links:

- [Mnestix API Swagger](#)
- [Mnestix AAS Searcher Swagger](#)
- [Mnestix AAS Searcher GraphQL](#)
- [Mnestix OpenAPI](#)
- [Show in Swagger-Editor](#)

**Legal Disclaimer**

This work is licenced under the Creative Commons Zero Licence (<https://creativecommons.org/publicdomain/zero/1.0/deed.de>)

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. THIS SPECIFICATION IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SPECIFICATION OR THE USE OR OTHER DEALINGS IN THE SPECIFICATION.

XITASO Mnestix Support - Website  
Send email to XITASO Mnestix Support

**Servers**

**Product Catalog** ^

- GET </marketplace> Marketplace Overview Page
- GET </marketplace/catalog> Marketplace Catalog Page
- GET </product/{base64AasId}> Product Details Page

**AAS Browser** ^

- GET </asset> Asset Resolver Page
- GET </viewer/{base64AasId}> Viewer Details Page
- GET </list> AAS List Page
- GET </settings> Settings Page
- GET </viewer/discovery> Viewer Discovery Page
- GET </viewer/registry> Viewer Registry Page
- GET </templates> Templates List Page
- GET </templates/{id}> Template Details Page

Abbildung 4-25: Mnestix API

<sup>49</sup> <https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP12/mnestix-openapi.yaml>

<sup>50</sup> <https://editor.swagger.io/?url=https://raw.githubusercontent.com/VWS4LS/vws4ls-subproject-results/refs/heads/main/TP12/mnestix-openapi.yaml>

### 4.6.3 Einstellungen

Die Einstellungen für Mnestix werden im Bereich Settings (Abbildung 4-26) vorgenommen. Hier lassen sich unter anderem Vorgaben für die Bildung von global eindeutigen IDs festlegen, sodass diese automatisch erzeugt werden. Diese Einstellungen werden in dem, beim Start von Mnestix hinterlegten Repository gespeichert, weswegen andere Tools u.U. diese VWSn anzeigen. Sie sind an dem Prefix „https://mnestix.com/aas/“ zu erkennen. Bei Änderungen kann es zu Problemen in Mnestix kommen.

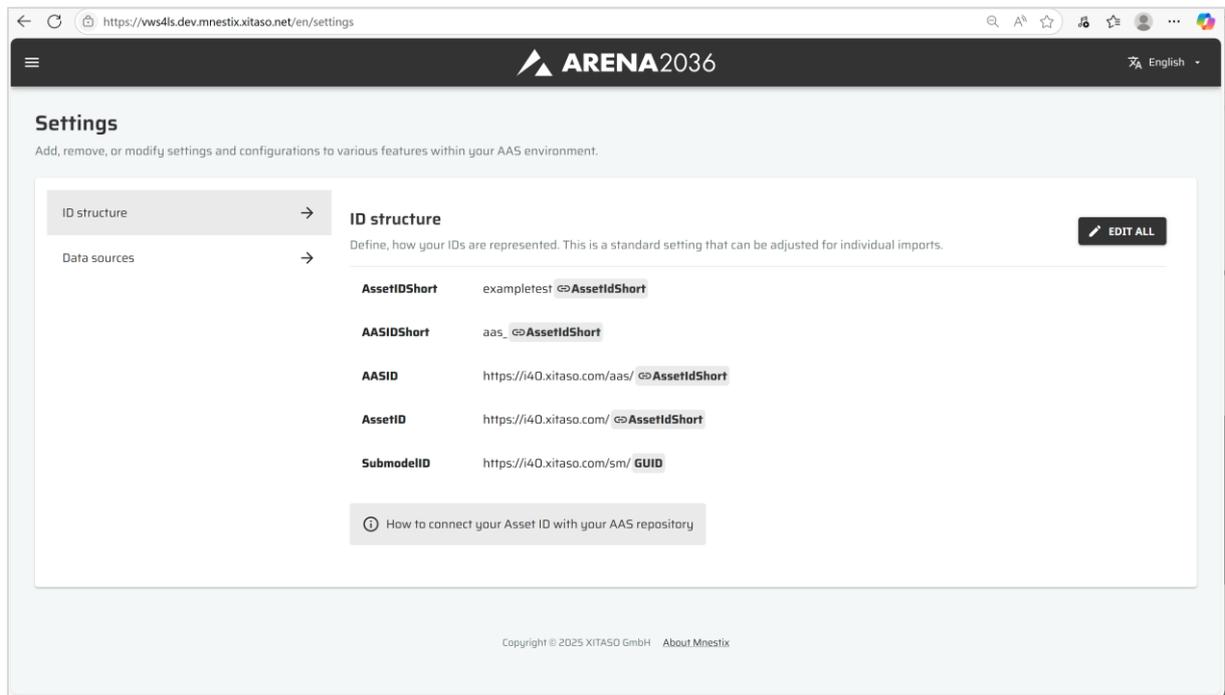


Abbildung 4-26: Mnestix Konfiguration Überblick

Nach betätigen von „EDIT ALL“ werden die Felder editierbar, Einstellungen werden nach betätigen von „SAVE ALL“ gespeichert, siehe Abbildung 4-27.

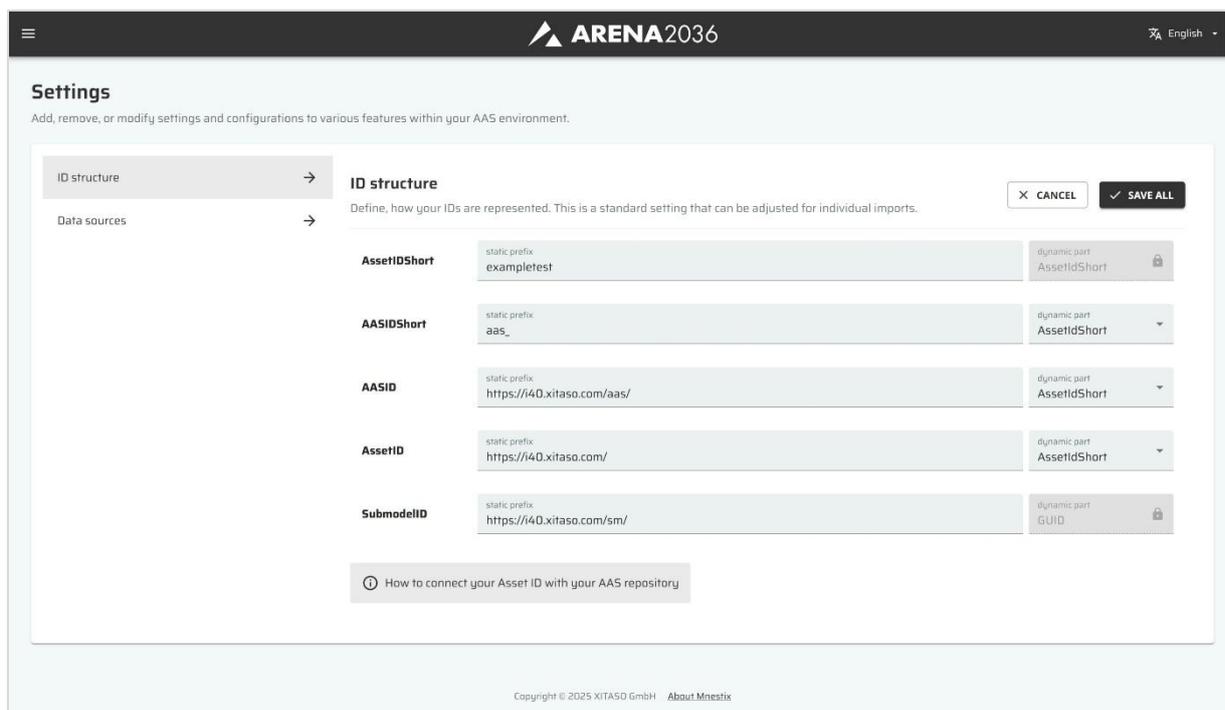
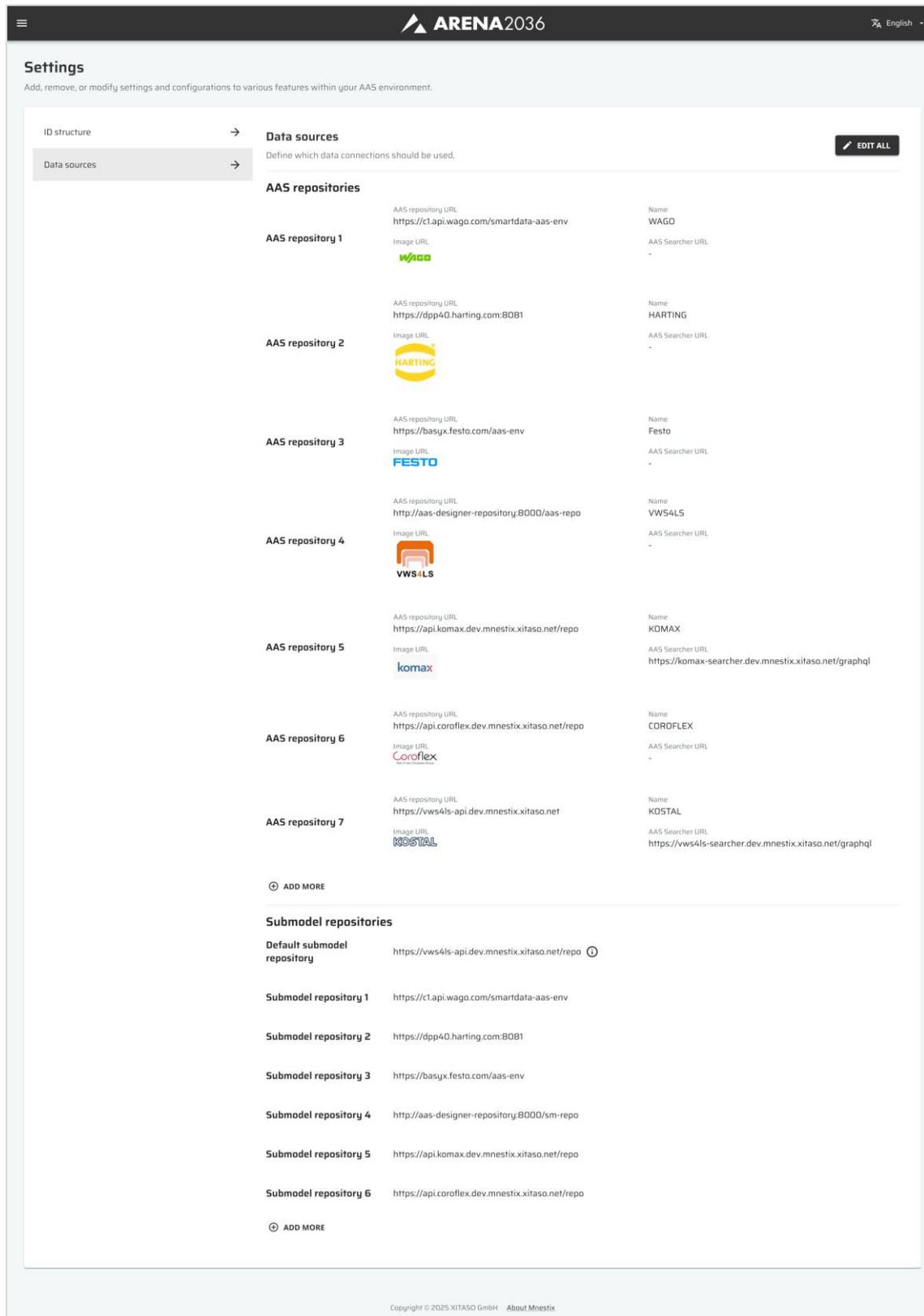


Abbildung 4-27: Mnestix Konfiguration Edit Mode

### 4.6.3.1 Datenquellen

Im Konfigurationsdialog „Data sources“ lassen sich die AAS-Repositories konfigurieren, die im Dashboard und der AAS-Liste zur Auswahl stehen sollen, siehe Abbildung 4-28.



The screenshot shows the 'Settings' page in the ARENA2036 application, specifically the 'Data sources' configuration section. The page title is 'Settings' with a subtitle 'Add, remove, or modify settings and configurations to various features within your AAS environment.' The 'Data sources' section is active, showing a list of AAS repositories and submodel repositories.

AAS repository	AAS repository URL	Name	AAS Searcher URL
AAS repository 1	https://c1.api.wago.com/smartdata-aas-env Image URL: 	WAGO	-
AAS repository 2	https://dpp40.harting.com:8081 Image URL: 	HARTING	-
AAS repository 3	https://basyx.festo.com/aas-env Image URL: 	Festo	-
AAS repository 4	http://aas-designer-repository.8000/aas-repo Image URL: 	VWS4LS	-
AAS repository 5	https://api.komax.dev.mnestix.xitaso.net/repo Image URL: 	KOMAX	https://komax-searcher.dev.mnestix.xitaso.net/graphql
AAS repository 6	https://api.coroflex.dev.mnestix.xitaso.net/repo Image URL: 	COROFLEX	-
AAS repository 7	https://vws4ls-api.dev.mnestix.xitaso.net Image URL: 	KOSTAL	https://vws4ls-searcher.dev.mnestix.xitaso.net/graphql

Below the AAS repositories, there is an 'ADD MORE' button and a section for 'Submodel repositories'. The 'Default submodel repository' is set to https://vws4ls-api.dev.mnestix.xitaso.net/repo. Other submodel repositories are listed with their respective URLs.

Copyright © 2025 XITASO GmbH About Mnestix

Abbildung 4-28: Mnestix Konfiguration für „Data Sources“

#### 4.6.4 Dashboard

Dies ist der einsteigsdialog für den normalen Anwender, in dem entweder direkt ein Herstellerkatalog ausgewählt oder per Chatbot-Funktion eine katalogübergreifende Produktsuche durchgeführt werden kann, siehe Abbildung 4-29.

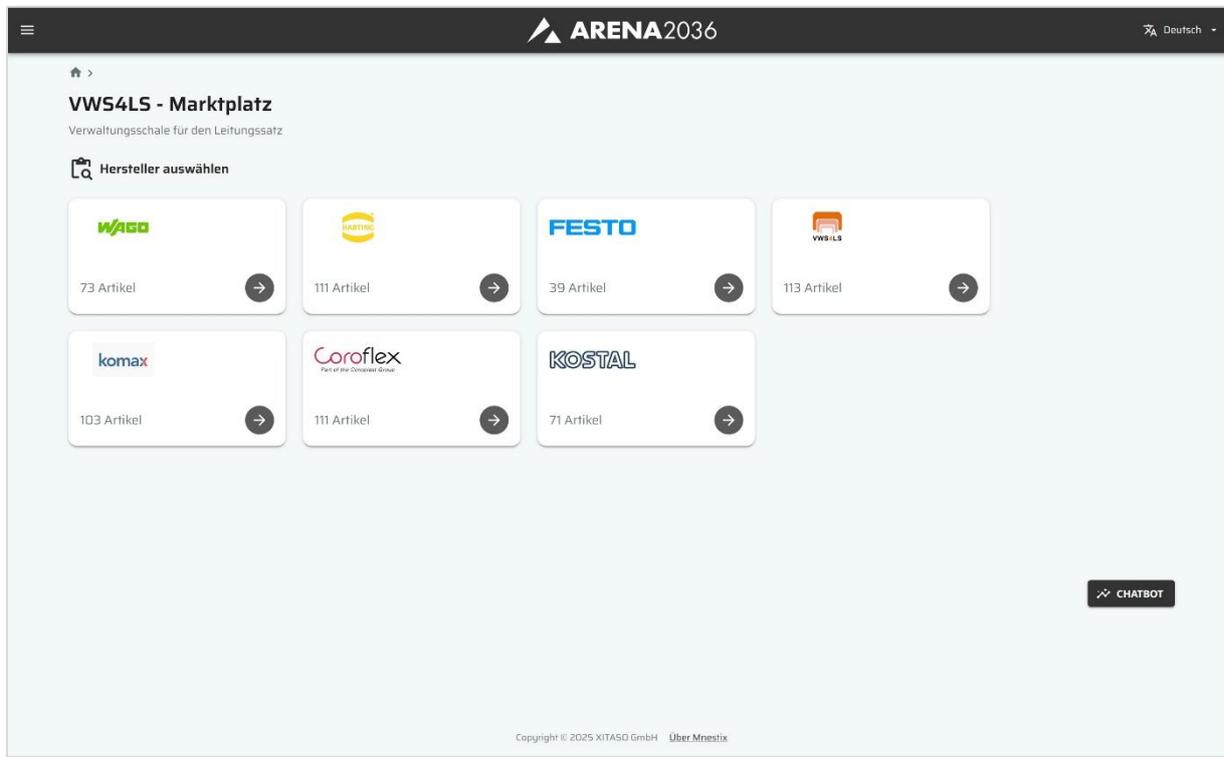


Abbildung 4-29: Mnestix Dashboard

#### 4.6.5 Produktfilterung

Nach Auswahl eines Herstellerkatalogs kann in diesem Dialog innerhalb des ausgewählten Katalogs mit detaillierten Filtern sowie Tabellensortierung die Produktsuche gezielt verfeinert werden, siehe Abbildung 4-30.

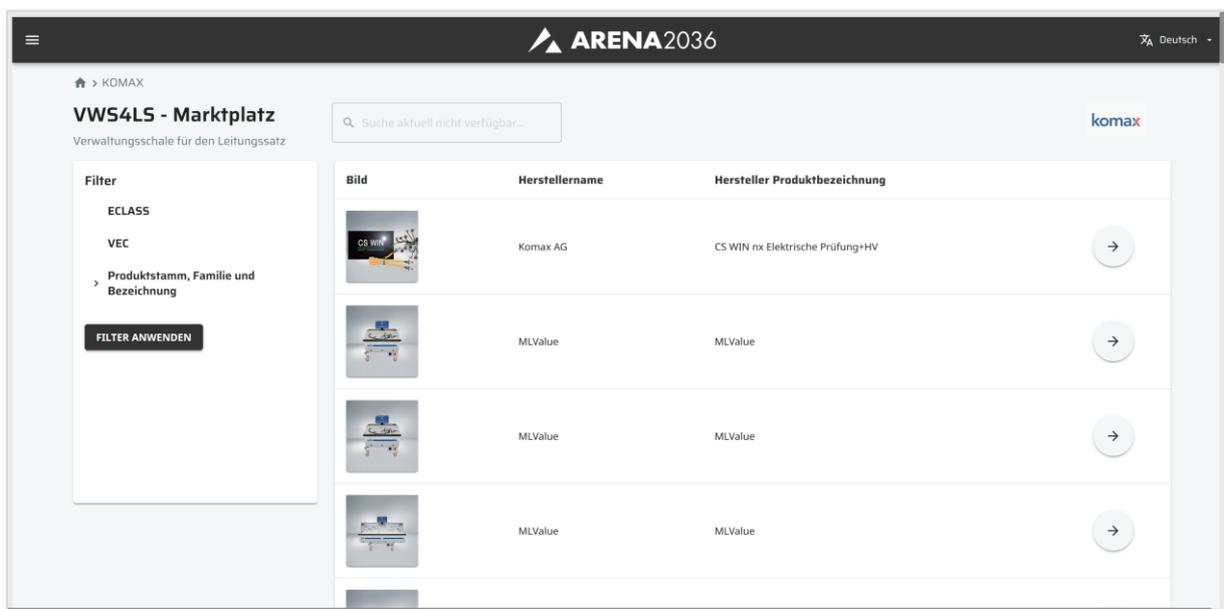
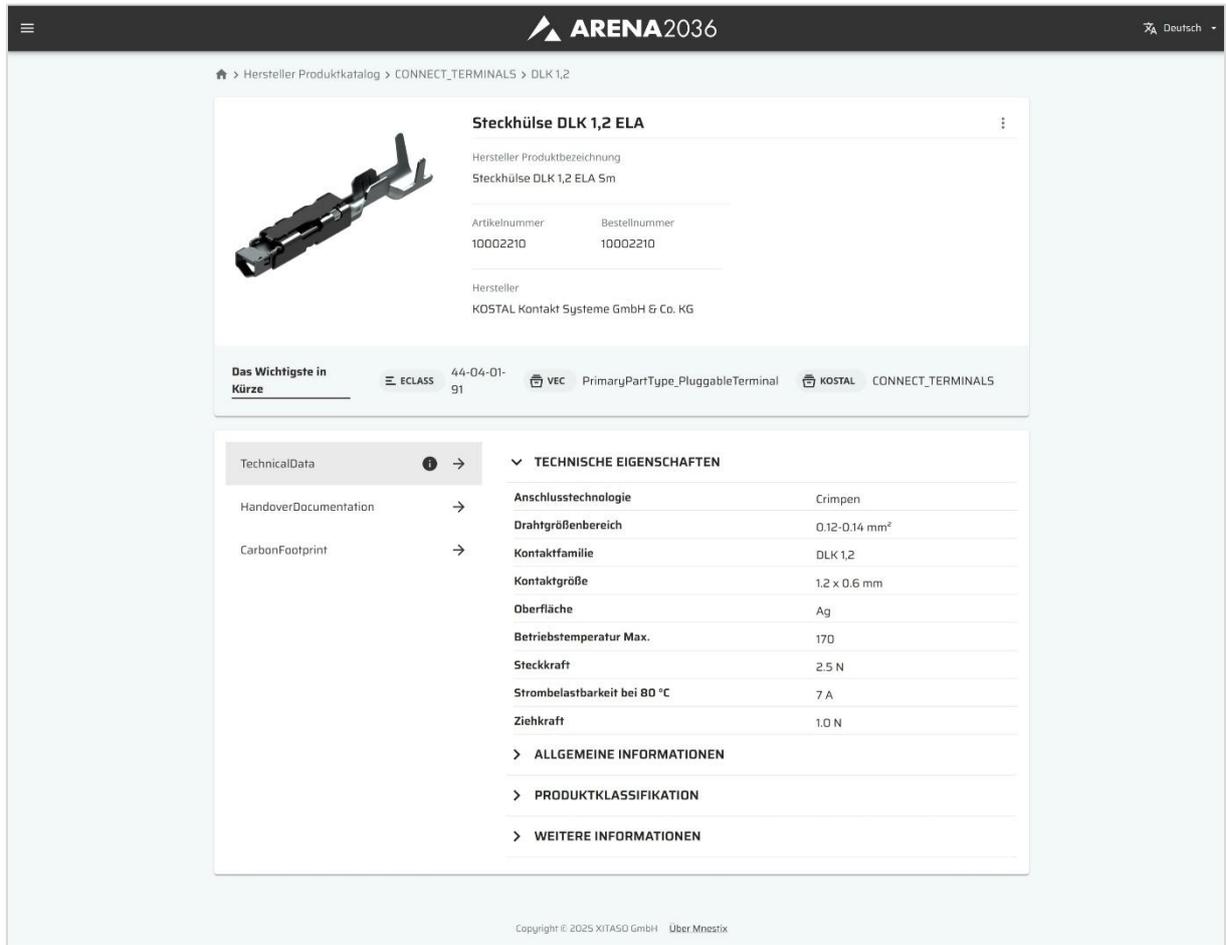


Abbildung 4-30: Mnestix Produktfilterung

## 4.6.6 Produktanzeige

In diesem Dialog wird das final ausgewählte Produkt angezeigt und ggf. die Interaktionselemente zu einem e-Shop bereitgestellt, siehe Abbildung 4-31.



The screenshot shows the product page for 'Steckhülse DLK 1,2 ELA' in the ARENA2036 system. The page includes a product image, a title, manufacturer information, and technical specifications.

**Steckhülse DLK 1,2 ELA**

Hersteller Produktbezeichnung  
Steckhülse DLK 1,2 ELA 5m

Artikelnummer	Bestellnummer
10002210	10002210

Hersteller  
KOSTAL Kontakt Systeme GmbH & Co. KG

**Das Wichtigste in Kürze**

ECLASS 44-04-01-91 VEC PrimaryPartType\_PluggableTerminal KOSTAL CONNECT\_TERMINALS

**TECHNISCHE EIGENSCHAFTEN**

Anschluss-technologie	Crimpen
Drahtgrößenbereich	0.12-0.14 mm²
Kontaktfamilie	DLK 1,2
Kontaktgröße	1,2 x 0,6 mm
Oberfläche	Ag
Betriebstemperatur Max.	170
Steckkraft	2,5 N
Strombelastbarkeit bei 80 °C	7 A
Ziehkraft	1,0 N

ALLGEMEINE INFORMATIONEN  
PRODUKTKLASSIFIKATION  
WEITERE INFORMATIONEN

Copyright © 2025 XITASO GmbH Über Mnestix

Abbildung 4-31: Mnestix Produktanzeige

## 4.7 AP 13.7 – BaSyx

Eclipse BaSyx™ ist eine Middleware für Verwaltungsschalen und stellt neben Verwaltungsschalen-Repositories weitere Funktionalitäten für den Zugriff und Integration von Verwaltungsschalen in die Systemlandschaften zur Verfügung. Beispielsweise ein Web-Frontend für die Visualisierung von Verwaltungsschalen und eine REST API für den Datenzugriff auf die Verwaltungsschalen.

### 4.7.1 Anpassungen in Eclipse BaSyx

Im Rahmen des Forschungsvorhabens wurden Anpassungen an der BaSyx Plattform vorgenommen, die im Folgenden kurz erläutert werden.

#### 4.7.1.1 Basic Authentication für das BaSyx Web UI

Zur Verbesserung der Sicherheit wurde im Web UI von Eclipse BaSyx eine Basic Authentication<sup>51</sup> eingeführt. Nutzernamen und Passwörter können nun als Konfigurationsparameter direkt im UI konfiguriert werden, was die Handhabung im Betrieb vereinfacht, und eine einheitliche Zugangskontrolle ermöglicht.

#### 4.7.1.2 Query Language

Die Funktionalität des Systems wurde durch die Implementierung der Query-Endpunkte erweitert. Diese basieren auf Elasticsearch<sup>52</sup> und erlauben eine performante und flexible Abfrage von Daten. Dadurch wird insbesondere die Suche innerhalb großer Datenmengen deutlich effizienter.

#### 4.7.1.3 Design-Anpassungen im Web UI

Nach Rückmeldung von Projektmitgliedern wurde das Design des Web UIs überarbeitet. Es wurde ein neues, übersichtlicheres Hauptmenü eingeführt, welches das einfache Einbinden von Modulen ermöglicht hat. Ebenso wurden die Einstellungsseiten verbessert und Konfigurationen, etwa der verlinkten Repository- und Registry-URLs, in eigene Menüs zusammengefasst. Hilfestellungen wurden verbessert und klarer strukturiert, um die Nutzbarkeit des Web UI zu verbessern. Zudem wurden die Kontextmenüs vereinfacht, was die Bedienbarkeit insgesamt erhöht.

#### 4.7.1.4 RBAC-Verbesserungen

Zur Optimierung der rollenbasierten Zugriffskontrolle (RBAC – Role-Based Access Control) wurde ein Update der eingesetzten Keycloak<sup>53</sup>-Version durchgeführt. Zusätzlich wurde die teilmodellbasierte RBAC-Konfiguration verbessert, um eine präzisere und flexiblere Rechtevergabe zu ermöglichen. Dies trägt wesentlich zur Sicherheit und Skalierbarkeit der Anwendung bei.

### 4.7.2 BaSyx UI – Modul-Feature

Das Eclipse BaSyx™ Web UI ist eines der gängigen Werkzeuge zur Interaktion mit Verwaltungsschalen. Es bietet neben der Darstellung der Rohdaten aus der Verwaltungsschale auch die Möglichkeit eigene Visualisierungen für Teilmodelle modular anzudocken. Es ist somit eine geeignete Anwendung für verschiedene Nutzergruppen, die dedizierte Oberflächen für ihre jeweiligen Anwendungsfälle benötigen.

Für die Umsetzung anwendungsfallbezogener Applikationen müssen aktuell jedoch neue Applikationen entwickelt werden. Dies resultiert in einem hohen zeitlichen und finanziellen Aufwand. Grundfunktionen wie das Abfragen von Verwaltungsschalen und Teilmodellen sowie zugehörige Methoden müssen für jede Anwendung neu implementiert werden. Außerdem steigt die Komplexität für den Endnutzer, da dieser potenziell mit einer Vielzahl heterogener Anwendungen konfrontiert wird.

Um den Prozess der Erstellung und Nutzung von AAS-Applikationen zu homogenisieren, wurde im Rahmen von Teilprojekt 14 das sogenannte Modul-Feature konzipiert und umgesetzt. Module erlauben es eigene Anwendungen als Teil des BaSyx Web UIs zu realisieren. Diese können dynamisch dem Modul-

---

<sup>51</sup> [https://en.wikipedia.org/wiki/Basic\\_access\\_authentication](https://en.wikipedia.org/wiki/Basic_access_authentication)

<sup>52</sup> <https://www.elastic.co/>

<sup>53</sup> <https://www.keycloak.org/>

Ordner hinzugefügt werden und sind anschließend im Hauptmenü des BaSyx UIs verfügbar (vgl. Abbildung 4-32). Eine vollständige Dokumentation für die Entwicklung eigener Module ist im BaSyx Wiki<sup>54</sup> zu finden.

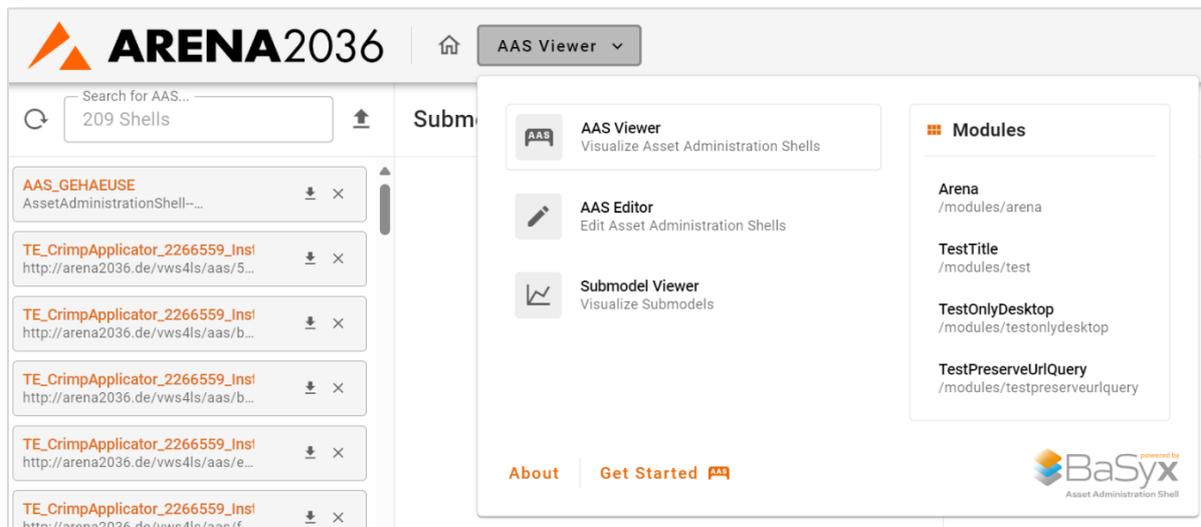


Abbildung 4-32 Hauptmenü zur Auswahl verfügbarer Module

Module werden wie das Web UI selbst in Vue.js<sup>55</sup> entwickelt und können somit auf sämtliche Hilfsfunktionen und Composables (reaktive Methoden) aus der Kernanwendung zugreifen. Das bedeutet, dass Funktionalitäten - wie das Abfragen von Verwaltungsschalen oder die Anzeige von Teilmodellen wie „Hierarchical Structures enabling Bills of Materials“ - nicht neu implementiert werden müssen, sondern einfach wiederverwendet werden können. Das reduziert den Entwicklungsaufwand und führt zu einheitlichen Darstellungen für dieselben Konzepte. Auch für den Endnutzer des Moduls führt dies zu einer homogenen Darstellung und bekannte Interaktionsmuster.

Als Teil des Projektes wurden mehrere Module mit dem Modul-Feature umgesetzt. Diese werden in den Kapiteln „4.7.3 AAS-Editor“, „5.1.1 BaSyx UI – Leitungssatzherstellung (Demonstrator UI)“ und „0 Modul für Änderungsmanagement“ im Detail beschrieben.

### 4.7.3 BaSyx UI – AAS Editor

Die Eclipse BaSyx™ Web UI ist eines der gängigen Werkzeuge zur Darstellung von Verwaltungsschalen. Es bietet neben der reinen Betrachtung von Verwaltungsschalen auch die Möglichkeit Werte einzelner Teilmodell-Elemente zu ändern. Es ist somit ein geeignetes Werkzeug, um Verwaltungsschalen zu sichten und Einsteiger an das Thema Verwaltungsschale heranzuführen.

Zum Erstellen neuer Verwaltungsschalen muss heute jedoch häufig auf Expertenwerkzeuge wie der Eclipse AASX Package Explorer™ oder Entwickler-Bibliotheken wie aas-core-works<sup>56</sup> zurückgegriffen werden. Diese Werkzeuge und Bibliotheken erfordern jedoch ein relativ gutes Vorwissen und Erfahrung und sind für Neueinsteiger weniger geeignet.

Um der Übergang vom reinen Sichten zum Erstellen von Verwaltungsschalen für Neueinsteiger einfacher zu gestalten, wurden im Rahmen von TP12 und TP13 neue Anforderung an das Eclipse BaSyx™ Web UI gestellt, siehe Abbildung 3-33.

Als Kernanforderung soll es möglich sein, ohne tiefere Kenntnisse, eine neue Verwaltungsschale zu erstellen und mit neuen Teilmodellen anzureichern. Hierfür muss es auch möglich sein neue Teilmodell-Elemente zu erstellen. Mittels geführter Dialoge sollen die wesentlichen Eigenschaften vom Benutzer eingefordert werden.

<sup>54</sup> <https://wiki.basyx.org/>

<sup>55</sup> <https://vuejs.org/>

<sup>56</sup> <https://github.com/aas-core-works>

Das Team vom Fraunhofer IESE hat diese Anforderungen aufgenommen und eine erste Version mit dem Eclipse BaSyx™ Web UI v2-250305 veröffentlicht. Weitere Ergänzungen sind in der Version v2-250417 hinzugekommen. Eine vollständige Dokumentation ist im BaSyx Wiki<sup>57</sup> zu finden.

Der AAS-Editor ist nun Teil des Eclipse BaSyx™ Web UI und kann über den Ansichtsmodus "AAS-Editor" aktiviert werden. Wenn der Modus aktiviert wurde, sind eine Reihe von neuen Funktionen über die Benutzeroberfläche verfügbar.

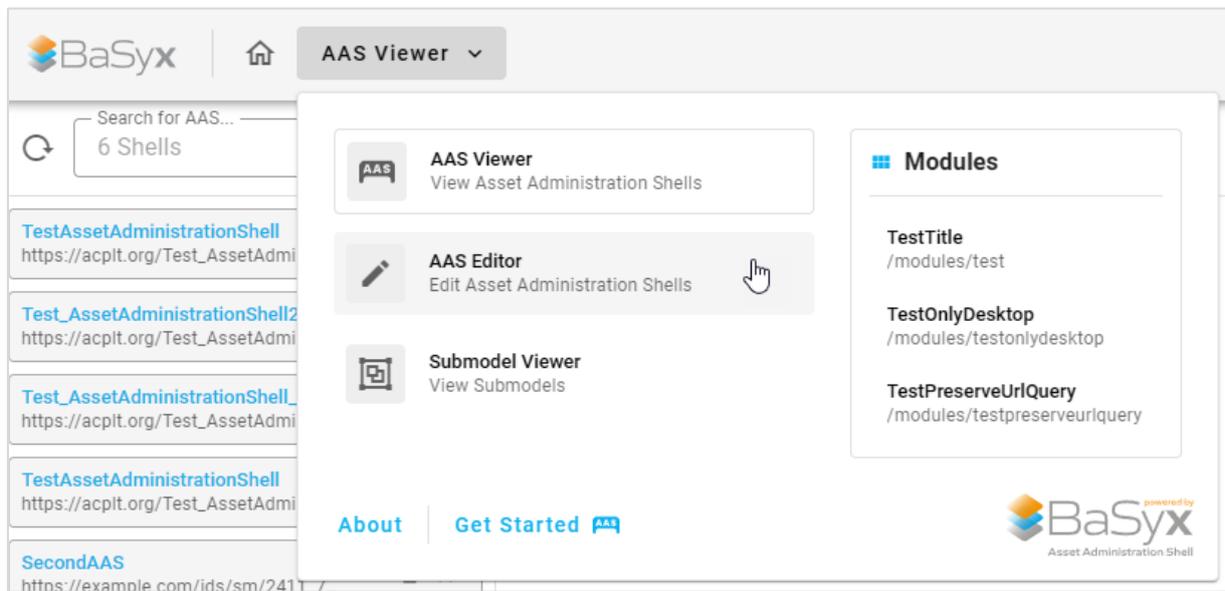


Abbildung 4-33: Umschalten zum AAS-Editor

#### 4.7.3.1 Verwaltungsschale erstellen

Über das Kontextmenü neben der Suchschaltfläche kann eine neue Verwaltungsschale erstellt werden, siehe in Abbildung 4-34.

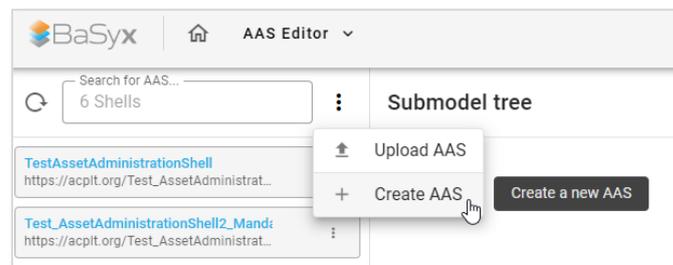


Abbildung 4-34: Neue Verwaltungsschale erstellen

Es erscheint ein Dialog, über den die wichtigsten Parameter für die Verwaltungsschale eingegeben bzw. ausgewählt werden können. Einige Inhalte werden automatisch erzeugt und vorbelegt, sodass die Anwender sich auf die inhaltlichen Werte konzentrieren können.

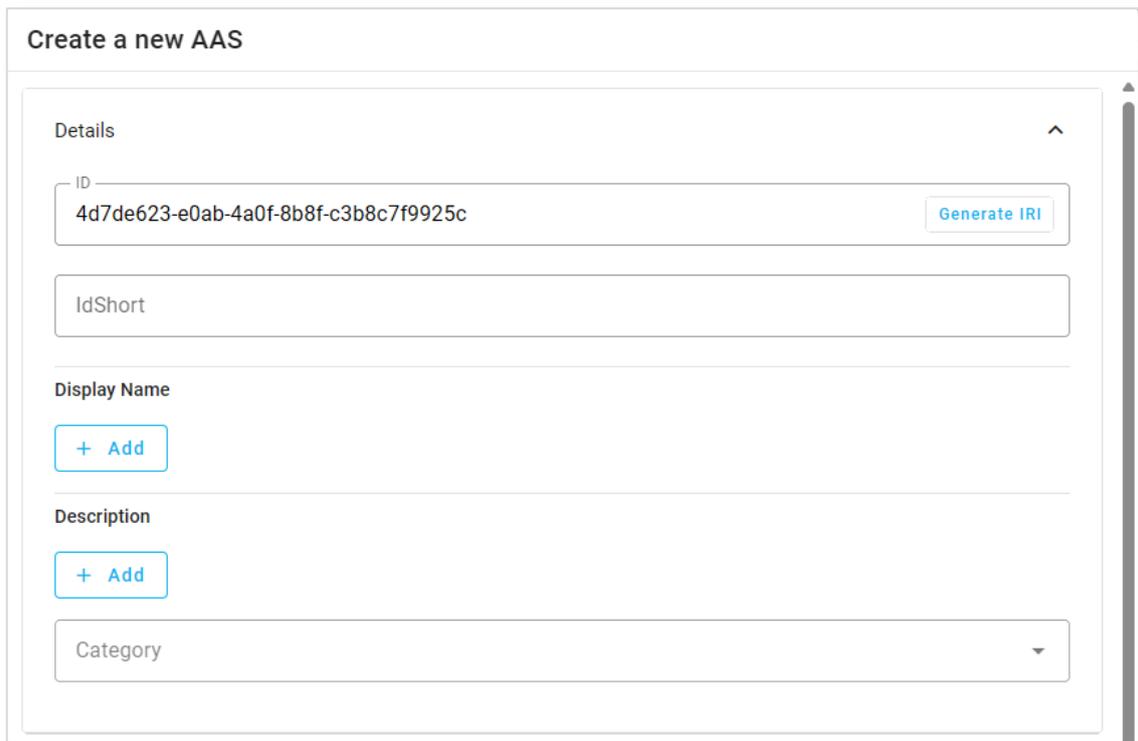
Der Dialog ist in mehrere Kategorien aufgegliedert:

#### Details

Im Detailbereich (vgl. Abbildung 4-35) werden zunächst die Kopfdaten der Verwaltungsschale festgelegt. Hierzu gehören die global eindeutige Verwaltungsschalen ID, sowie die menschenlesbare *idShort*, womit die Verwaltungsschale in der Auswahlliste erscheint. Die Verwaltungsschalen ID kann über die Schaltfläche **Generate IRI** neu erzeugt werden.

<sup>57</sup> <https://wiki.basysx.org/>

Der Anzeigename (engl. Display Name) und die Beschreibung (engl. Description) mitsamt Kategorie sind optional und können nach Bedarf hinzugefügt werden.



The screenshot shows a web form titled "Create a new AAS". The "Details" section is expanded and contains the following fields:

- ID:** A text input field containing the value "4d7de623-e0ab-4a0f-8b8f-c3b8c7f9925c". To the right of the field is a blue button labeled "Generate IRI".
- IdShort:** An empty text input field.
- Display Name:** A section with a blue button labeled "+ Add".
- Description:** A section with a blue button labeled "+ Add".
- Category:** A dropdown menu with the text "Category" and a downward arrow.

Abbildung 4-35: Anreicherung mit Informationen im Bereich „Details“

### Administrative Information

Im Bereich *Administrative Information* (vgl. Abbildung 4-36) werden Informationen über die Version und den Ersteller hinterlegt. Diese Eingaben sind optional und können für den einfachen Einstieg vernachlässigt werden.

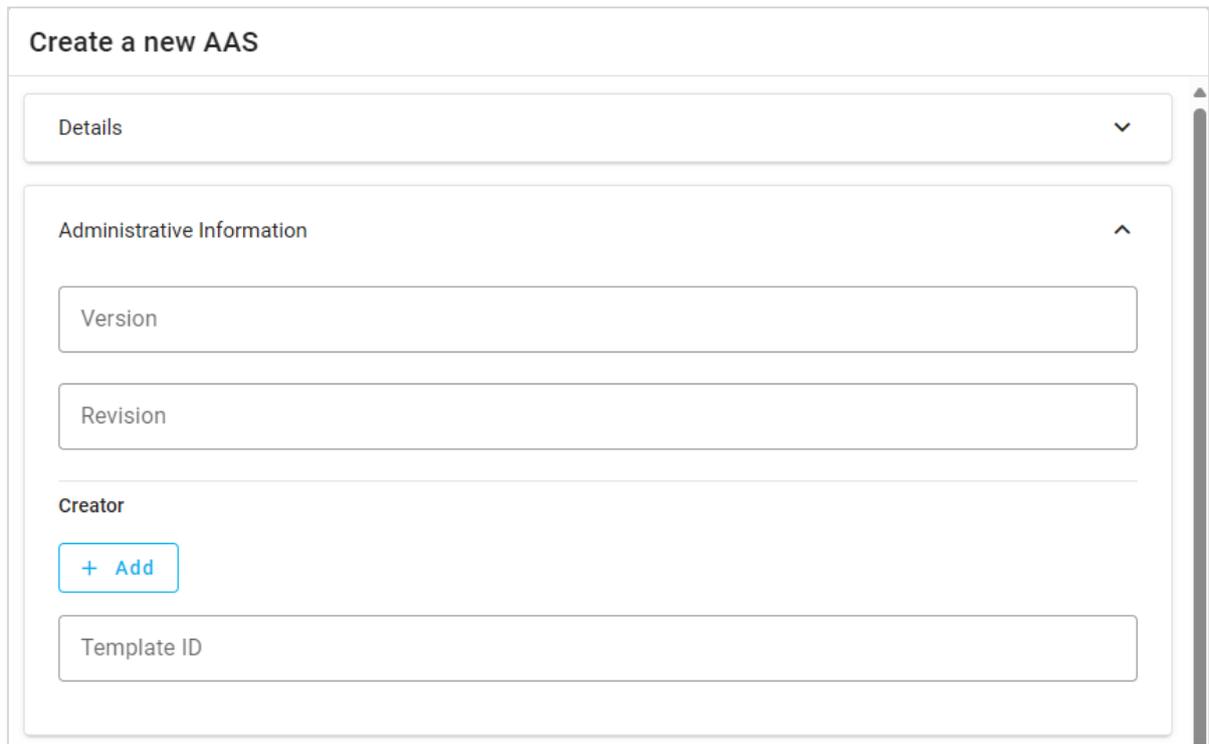


Abbildung 4-36: Administrative Information

### Derivation

Der Bereich *Derivation* (vgl. Abbildung 4-37) ist aktuell noch nicht implementiert und wird in einer zukünftigen Version unterstützt. In diesem Bereich wird es möglich sein eine Ableitungsbeziehung (derivedFrom) zu einer anderen Verwaltungsschale herzustellen. Diese Funktionalität ist u.a. für die Verwendung in Typ-Instanz-Beziehungen notwendig.

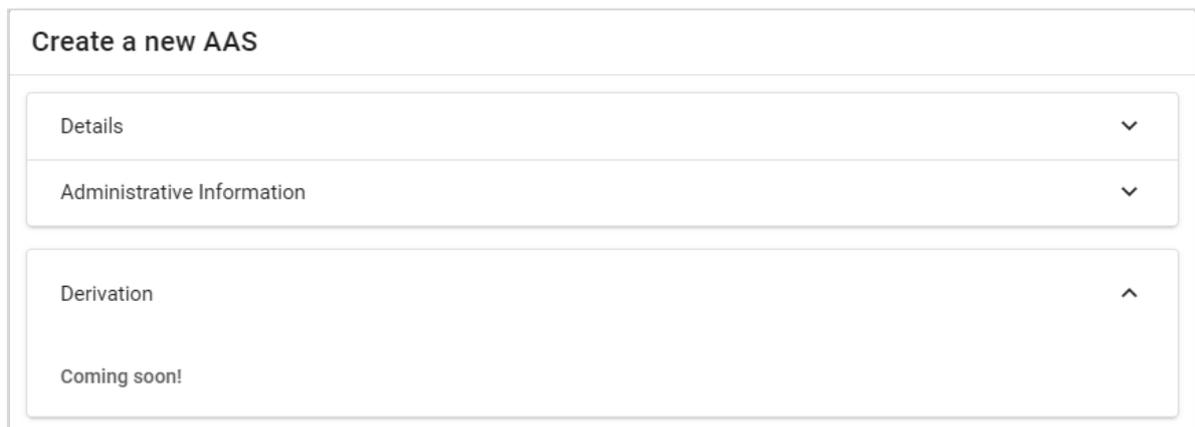


Abbildung 4-37: Derivation

### Asset

Im Bereich *Asset* (vgl. Abbildung 4-38) werden Informationen zum (physikalischen) Asset hergestellt. Hierfür ist zunächst die Angabe der Art des Assets (engl. Asset Kind) notwendig. Also, ob es sich um eine Instanz oder einen Typ handelt.

Zudem die Angabe einer global eindeutigen Asset ID, die auch über die Schaltfläche **Generate IRI** erzeugt werden kann und den Asset Typ (engl. Asset Type).

Optional kann hier auch noch eine Asset-Abbildung in Form einer angehängten Bilddatei (FILE) oder URL auf eine Bilddatei hinterlegt werden.

### Create a new AAS

Details ▼

Administrative Information ▼

Derivation ▼

Asset ▲

Asset Kind  
Instance ▼

Global Asset ID Generate IRI

Asset Type

Default Thumbnail

NONE
FILE
URL

CANCEL SAVE

Abbildung 4-38: Asset

Wenn alle Angaben vollständig sind, kann die Verwaltungsschale über die Schaltfläche **Save** gespeichert werden und erscheint in der Liste der Verwaltungsschalen. Abbildung 4-39 zeigt exemplarisch die Details einer erzeugte Verwaltungsschale.

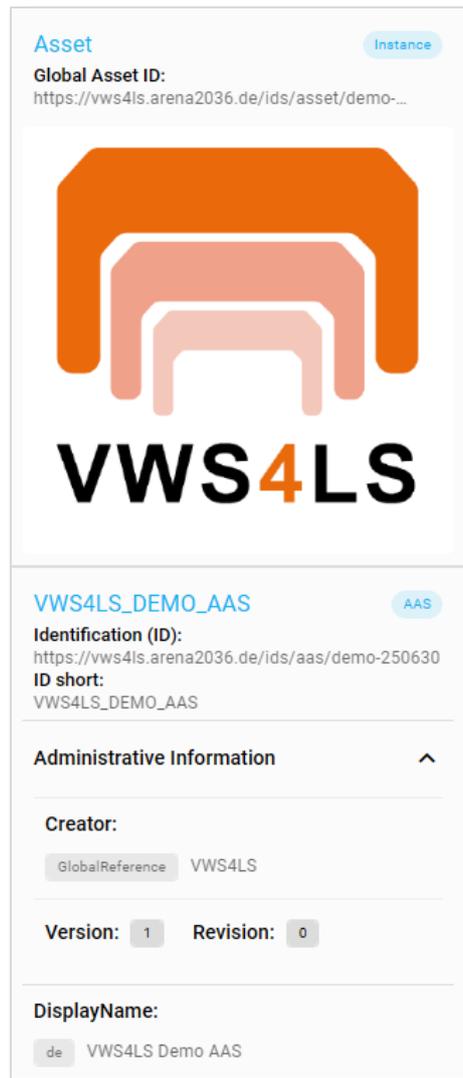


Abbildung 4-39: VWS4LS Demo VWS

Die Verwaltungsschale ist somit vollständig erstellt. Über das Kontextmenü (vgl. Abbildung 4-40) kann eine Verwaltungsschale bearbeitet oder ergänzt werden.

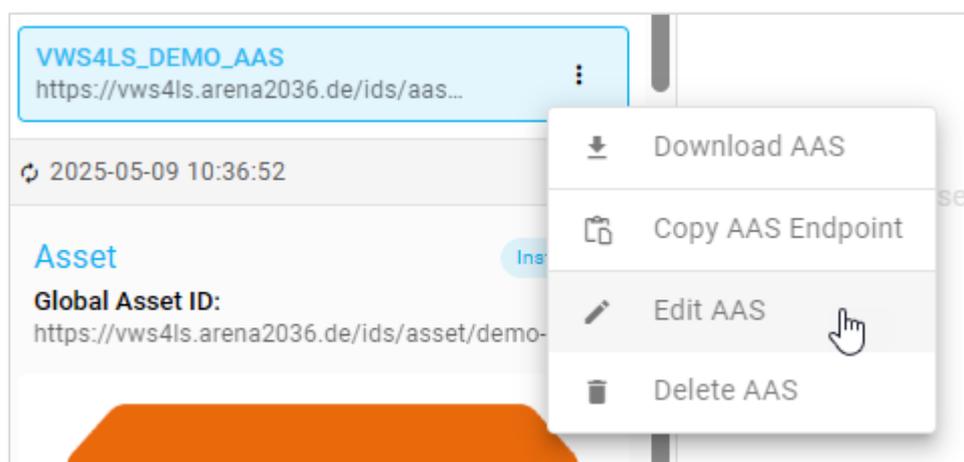


Abbildung 4-40: Kontextmenü zum Bearbeiten einer Verwaltungsschale

#### 4.7.3.2 Teilmodell erstellen

Um eine Verwaltungsschale mit Inhalten zu füllen, sind Teilmodelle erforderlich. Diese können nach der Auswahl der Verwaltungsschale über das Kontextmenü im Inhaltsbereich erstellt werden (vgl. Abbildung 4-41).

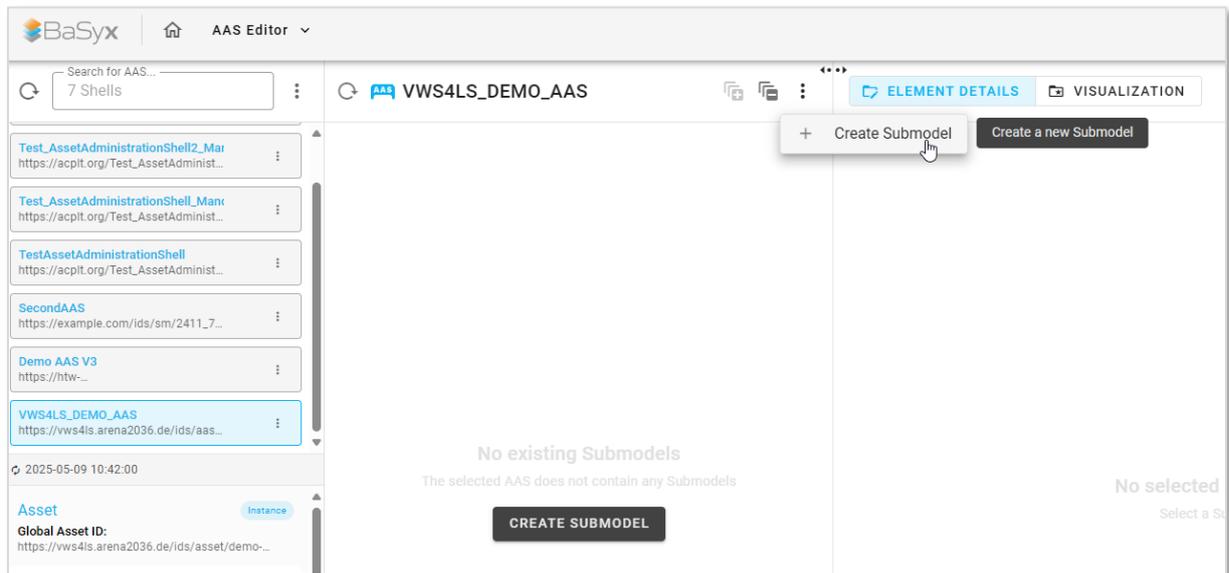


Abbildung 4-41: Neues Teilmodell erstellen

Ähnlich wie beim Erstellen neuer Verwaltungsschalen erscheint ein Dialog, in dem diverse Parameter eingegeben werden können. Einige Inhalte werden automatisch erzeugt und vorbelegt, sodass die Anwender sich auf die inhaltlichen Werte konzentrieren können.

Der Dialog ist in mehrere Kategorien aufgegliedert:

##### Details

Im Detailbereich (vgl. Abbildung 4-42) werden zunächst die Kopfdaten des Teilmodells festgelegt. Hierzu gehören die global eindeutige Teilmodell ID, sowie die menschenlesbare *IdShort*, womit das Teilmodell in der Auswahlliste erscheint. Die Teilmodell ID kann über die Schaltfläche **Generate IRI** neu erzeugt werden.

Der Anzeigename (engl. Display Name) und die Beschreibung (engl. Description) mitsamt Kategorie sind optional und können nach Bedarf hinzugefügt werden.

### Create a new Submodel

**Details** ^

ID  Generate IRI

IdShort

Modelling Kind  ▼

---

**Display Name**

+ Add

---

**Description**

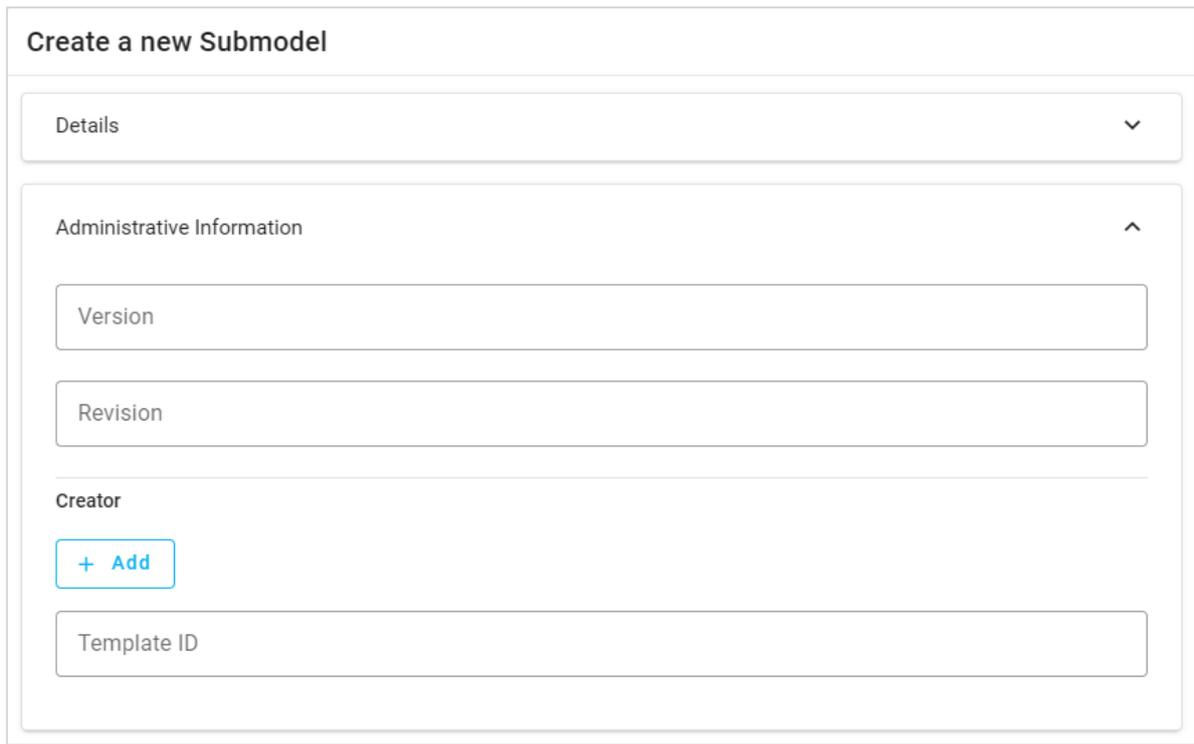
+ Add

Category  ▼

Abbildung 4-42: Submodel Editor - Details

### Administrative Information

Im Bereich *Administrative Information* (vgl. Abbildung 4-43) werden Informationen über die Version und den Ersteller hinterlegt. Diese Eingaben sind optional und können für den einfachen Einstieg vernachlässigt werden.



**Create a new Submodel**

Details ▾

Administrative Information ▲

Version

Revision

Creator

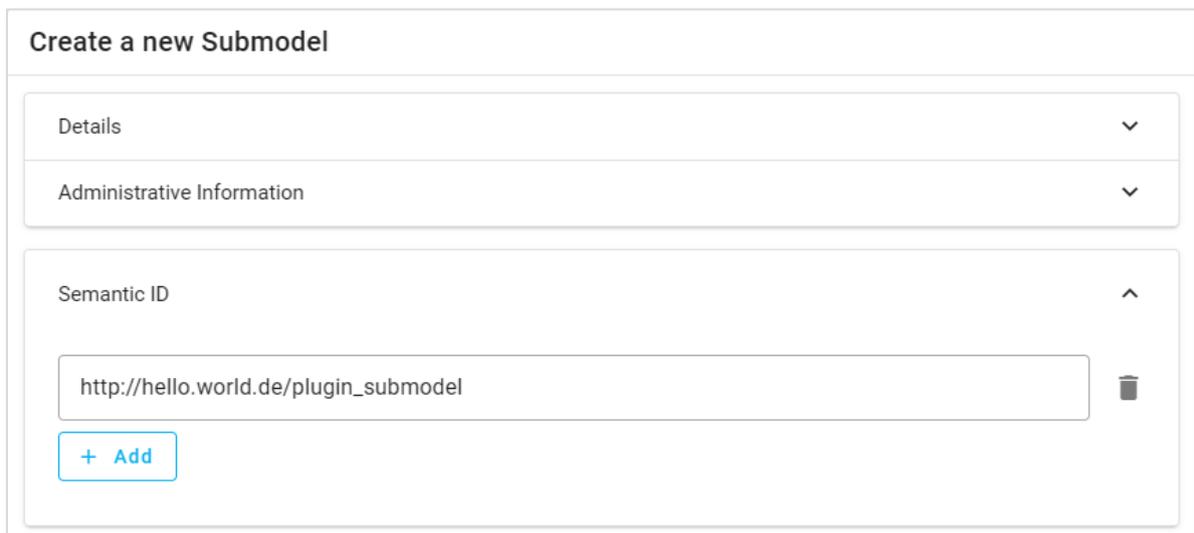
+ Add

Template ID

Abbildung 4-43: Submodel Editor - Administrative Information

### Semantic ID

Der Bereich Semantic ID (vgl. Abbildung 4-44) definiert den semantischen Identifier für das Teilmodell. Entsprechende Identifier sind in den Teilmodell-Vorlagen der IDTA<sup>58</sup> zu finden.



**Create a new Submodel**

Details ▾

Administrative Information ▾

Semantic ID ▲

http://hello.world.de/plugin\_submodel

+ Add

Abbildung 4-44: Submodel Editor - Semantic ID

### Data Specification

Der Bereich *Data Specification* (vgl. Abbildung 4-45) ist aktuell noch nicht implementiert und wird in einer zukünftigen Version unterstützt.

<sup>58</sup> <https://industrialdigitaltwin.org/content-hub/teilmodelle>

### Create a new Submodel

Details	▼
Administrative Information	▼
Semantic ID	▼
Data Specification	▲
Coming soon!	

Abbildung 4-45: Submodel Editor - Data Specification

Wenn alle Angaben vollständig sind, kann das Teilmodell über die Schaltfläche **Save** gespeichert werden und erscheint in der Liste der Teilmodelle. Abbildung 4-46 zeigt exemplarisch die Details eines erstellten Teilmodells.

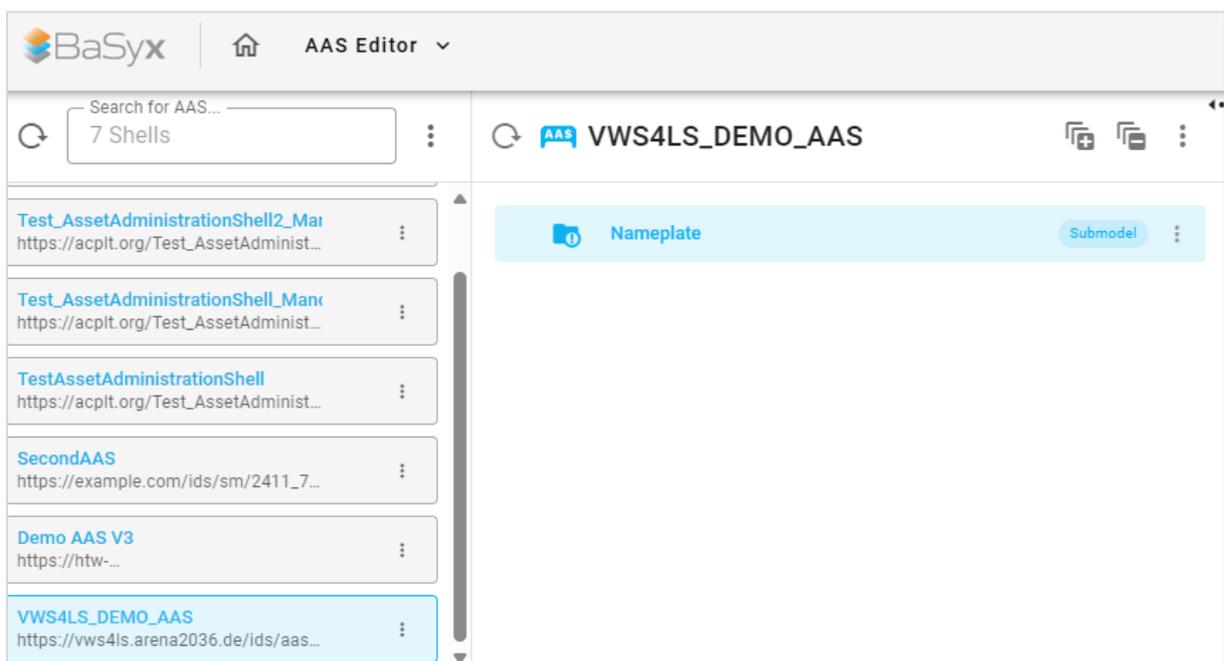


Abbildung 4-46: Erstelltes Teilmodell

Neben dem Erstellen von Teilmodellen ermöglicht der AAS Editor auch das Bearbeiten existierender Teilmodelle. Hierfür kann über das Teilmodell über das Kontextmenü (Abbildung 4-47) bearbeitet bzw. ergänzt werden.

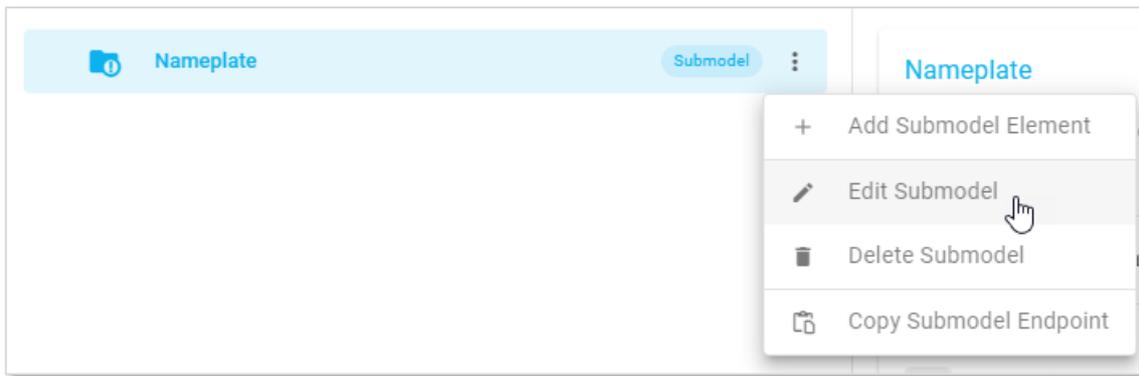


Abbildung 4-47: Teilmodell Kontextmenü

Die erstellten Teilmodelle sind zunächst ohne Inhalt und müssen im Weiteren mit Teilmodellelementen angereichert werden.

#### 4.7.3.3 Teilmodell-Element erstellen

Die Inhalte von Teilmodellen werden mittels Teilmodell-Elementen dargestellt. Diese können im AAS Editor über das Kontextmenü des Teilmodells (siehe Abbildung 4-48) hinzugefügt werden.

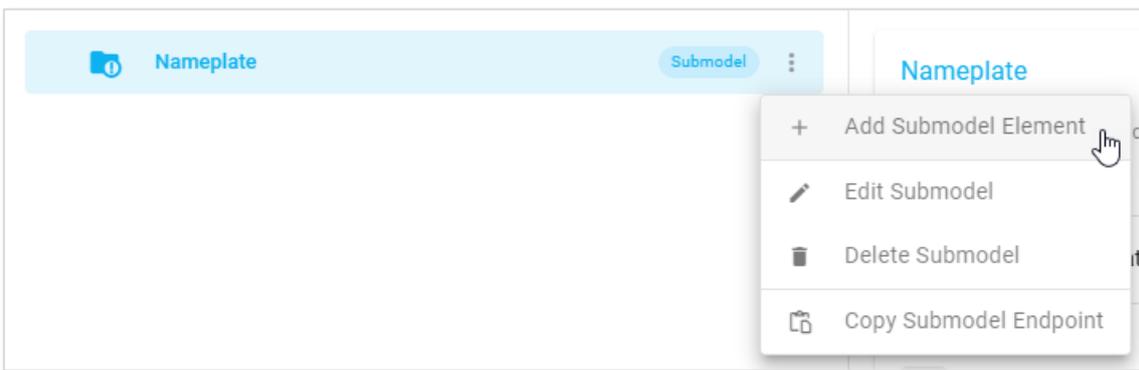


Abbildung 4-48: Teilmodell-Element zu einem Teilmodell hinzufügen

Dafür muss zunächst der Datentyp des Elements ausgewählt werden (Abbildung 4-49)



Abbildung 4-49: Submodel Element - Type

In der aktuellen Version werden die folgenden Datentypen unterstützt:

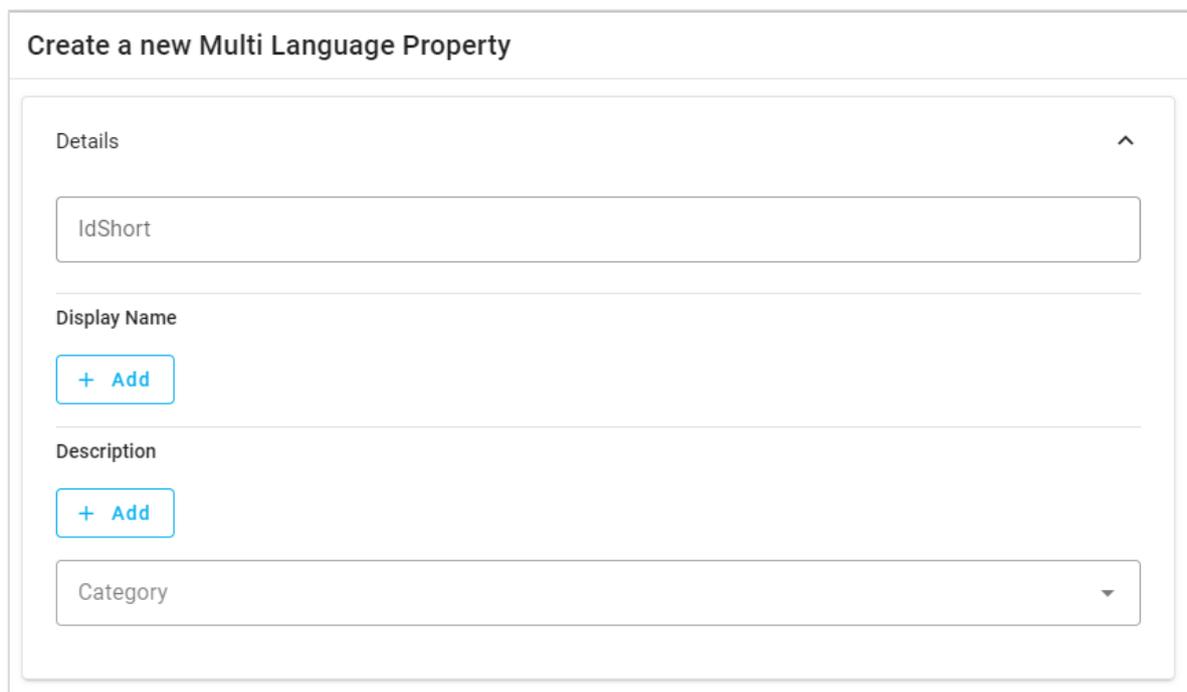
- Property
- SubmodelElementCollection
- SubmodelElementList
- MultiLanguageProperty
- Range
- File

Mit diesen Optionen können bereits viele Teilmodelle vollständig mit Inhalten gefüllt werden. Im Weiteren wird der Ablauf anhand einer *MultiLanguageProperty* beschrieben.

Nachdem der Datentyp mit der Schaltfläche **Next** bestätigt wurde, erscheint die Eingabemaske für die Inhalte. Abhängig vom gewählten Datentyp können sich die angezeigten Inhalte unterscheiden.

### Details

Im Bereich *Details* (vgl. Abbildung 4-50) werden die allgemeinen Informationen des Elements festgelegt. Dazu gehören die *IdShort*, welche im jeweiligen Kontext eindeutig sein muss, sowie optional der Anzeigename (engl. Display Name) und die Beschreibung (engl. Description).

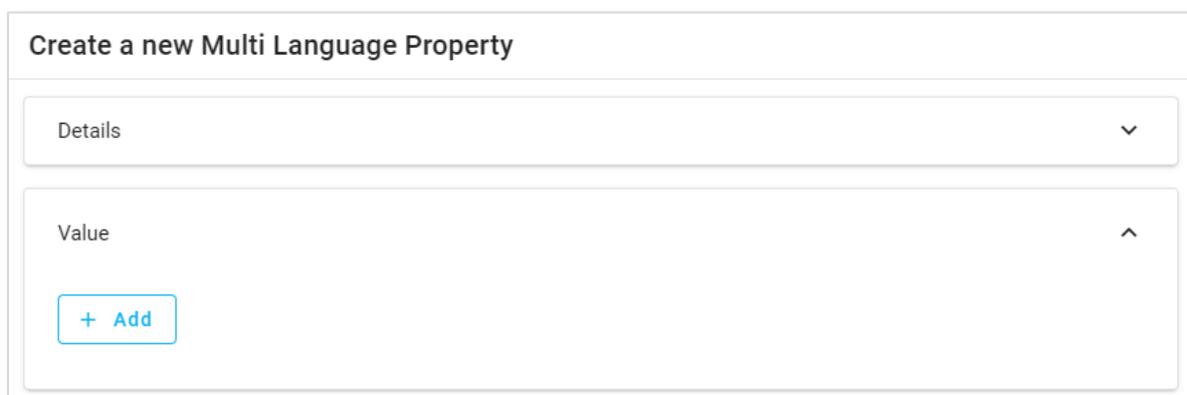


The screenshot shows a web form titled "Create a new Multi Language Property". The "Details" section is expanded, showing a text input field for "IdShort". Below it, the "Display Name" section has a "+ Add" button. The "Description" section also has a "+ Add" button. At the bottom, there is a dropdown menu for "Category".

Abbildung 4-50: Submodel Element – Details

### Value

Im Bereich *Value* (vgl. Abbildung 4-51) wird der Merkmalswert des Elements eingetragen. Je nach gewähltem Datentyp können sich die Inhalte stark unterscheiden.



The screenshot shows the same "Create a new Multi Language Property" form, but the "Value" section is expanded. The "Details" section is collapsed. The "Value" section contains a "+ Add" button.

Abbildung 4-51: Submodel Element – Value

### Semantic ID

Im Bereich *Semantic ID* (vgl. Abbildung 4-52) wird der semantische Identifier für das Merkmal festgelegt. Dieser Wert ist essenziell für den interoperablen Datenaustausch mit dritten.

### Create a new Multi Language Property

Details ▼

Value ▼

Semantic ID ^

+ Add

Abbildung 4-52: Submodel Element - Semantic ID

### Data Specification

Der Bereich *Data Specification* (vgl. Abbildung 4-53) ist aktuell noch nicht implementiert und wird in einer zukünftigen Version unterstützt.

### Create a new Multi Language Property

Details ▼

Value ▼

Semantic ID ▼

Data Specification ^

Coming soon!

CANCEL SAVE

Abbildung 4-53: Submodel Element - Data Specification

Wenn alle Angaben vollständig sind, kann das Element über die Schaltfläche **Save** gespeichert werden und erscheint in der Liste der Teilmodell-Elemente. Abbildung 4-54 zeigt exemplarisch die Details eines erstellten Elements.

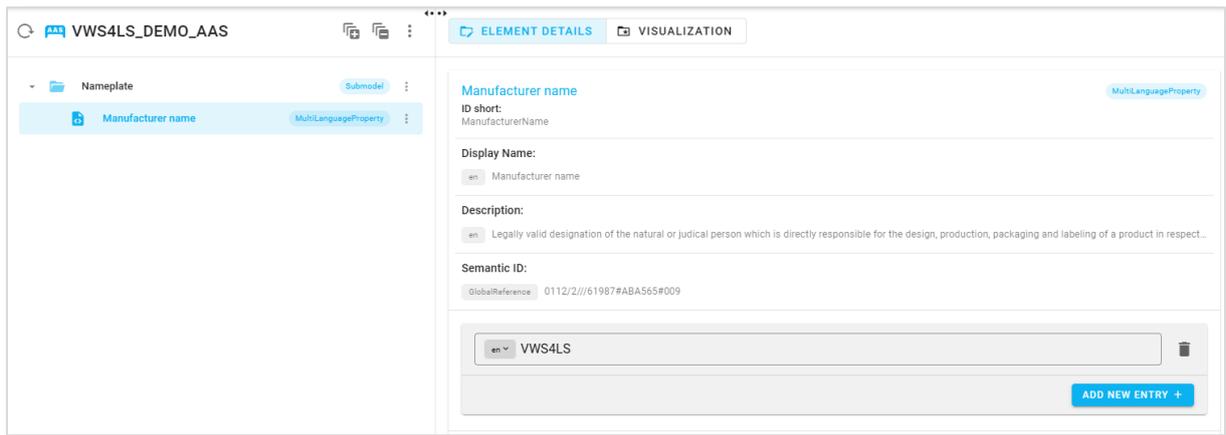


Abbildung 4-54: Erstelltes Submodel Element

Neben dem Erstellen von Teilmodell-Elementen ermöglicht der AAS-Editor auch das Bearbeiten existierender Elemente. Hierfür kann das jeweilige Element über das Kontextmenü (siehe Abbildung 4-55) bearbeitet, gelöscht oder kopiert.

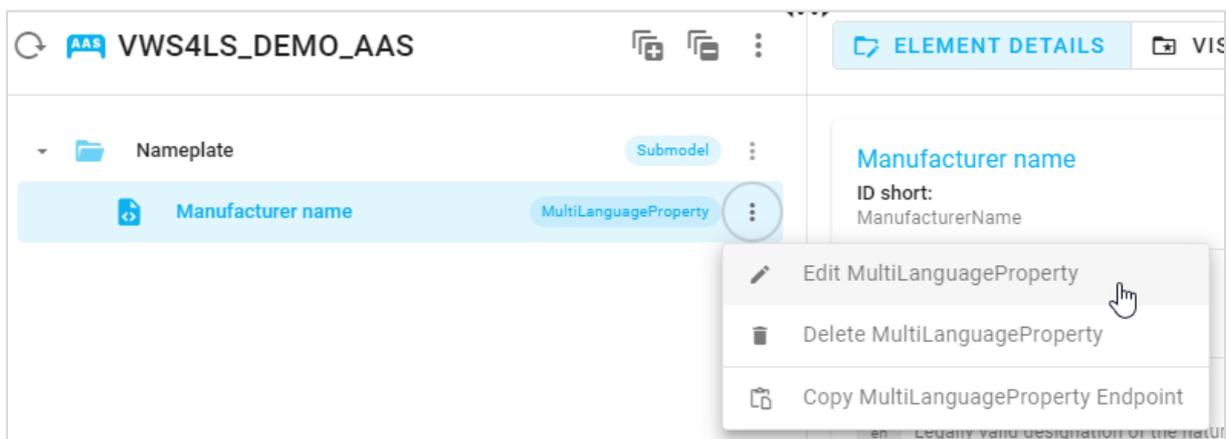


Abbildung 4-55: Submodel Element Kontextmenü

## 5 TP14 – Demonstrator

Im Teilprojekt 14 „Demonstrator“ des Projekts VWS4LS steht die praktische Validierung und Erprobung der im Projekt entwickelten Lösungen im Mittelpunkt. Ziel ist es, die verschiedenen Ansätze und Konzepte, die in den vorangegangenen Teilprojekten erarbeitet wurden, in einer realitätsnahen Umgebung umzusetzen und auf ihre Praxistauglichkeit zu überprüfen. Der Aufbau des Demonstrators orientiert sich dabei im Wesentlichen an der in der Ergebnisdokumentation 1 beschriebenen Architektur, mit einigen Anpassungen: So kommen unter anderem eine Crimp-Maschine, eine BaSyx-Umgebung mit einem neu entwickelten Frontend sowie eine Komax Alpha-Simulation zum Einsatz.

Die Schwerpunkte des TPs liegen darauf, die Usability des BaSyx-Frameworks für den potenziellen Nutzer entlang des Produktionsprozesses Anwenderfreundlicher zu gestalten. Die einzelnen Handlungsschritte sollen dargestellt, nachvollziehbarer und die notwendigen Informationen pro Prozessschritt aggregiert werden. Darüber hinaus, wurde ein Proof of Concept entwickelt, welches sich dem Änderungsmanagement und wie dieses umgesetzt werden kann widmet. Hierbei wird demonstriert, wie Änderungen im System effizient erkannt, verwaltet und nachvollzogen werden können. Durch die Integration dieser Komponenten wird sichergestellt, dass alle relevanten Lösungsansätze des Projekts in einem umfassenden Demonstrator abgebildet und getestet werden können.

Damit leistet das Teilprojekt einen entscheidenden Beitrag zur Überführung der theoretischen Konzepte in die industrielle Anwendung und schafft die Grundlage für eine spätere Implementierung im realen Produktionsumfeld.

### 5.1 Workflow-gesteuerte Auftragsabwicklung

Der Demonstrator bildet einen kompletten Fertigungsauftrag in Form eines digital vernetzten Ablaufes nach. Im Kern handelt es sich um eine Java-basierte Anwendung, die auf dem Spring-Boot-Framework aufsetzt. Über eine schlanke REST-Schnittstelle können externe Akteure Aufträge anlegen, verhandeln, bestätigen und schließlich abwickeln. Die Geschäftslogik jedes Auftrags wird dabei durch ein BPMN-Prozessmodell beschrieben, das zur Laufzeit von einer Workflow-Engine ausgeführt und von speziell zugeschnittenen Worker-Modulen unterstützt wird.

Der Ablauf beginnt typischerweise damit, dass ein Client einen neuen Produktionsauftrag initiiert. Dazu übergibt er lediglich die Produkt-ID an das System. Daraufhin lädt die Anwendung die passende VWS des Produkttyps, generiert individuelle Identifikatoren für Auftrag, Charge und Lot und legt diese VWSen in der BaSyx-Umgebung an. Anschließend startet ein Verhandlungsprozess: Das System nimmt die gewünschte Stückzahl entgegen, schreibt sie in das Submodell des Auftrags und leitet über einen kurzen Aufruf an einen Node-RED-Flow einen Verhandlungsdialog zwischen den verfügbaren Maschinen ein. In diesem Verhandlungsszenario werden die offered- mit den required-Capabilities abgeglichen. Ziel ist es ein Match zwischen den produktseitigen required-Capabilities und der Maschinen seitigen offered-Capabilities zu ermöglichen.

Sobald eine Maschine ausgewählt wurde, spielt die Anwendung nun das zugehörige BPMN-Modell in die Workflow-Engine ein und startet eine frische Prozessinstanz. Zu diesem Zeitpunkt wechselt das System in die Phase der Asset-Bestätigung. Schritt für Schritt werden Maschine, Werkzeug, Terminal und Kabel geprüft, indem der Client jeweils die gescannte Kennung einsendet. Die Kennung ist in diesem Fall ein QR-Code, der die jeweilige Asset-ID des zu verwendenden Produktes beinhaltet. Das System vergleicht die Werte mit denen im Auftrags-Submodell und bestätigt oder verwirft die Eingabe.

Sind alle Prüfungen erfolgreich, geht der Auftrag in die Ausführung über. Die Worker-Module greifen einzelne Aufgaben aus dem BPMN-Modell ab, setzen Werte in Submodellen oder publizieren Statusmeldungen auf einen internen Nachrichtendienst. Gleichzeitig aktualisieren sie den Fortschritt in einer eigenen Status-VWS, sodass Bedienende und externe Systeme den Prozess in nahezu Echtzeit verfolgen können.

Wenn während der Ausführung ein Fehler auftritt oder ein Testlauf erneut gestartet werden soll, steht ein Reset-Endpoint zur Verfügung. Dieser beendet alle untergeordneten Prozesse, stoppt die Worker und setzt das System in einen definierten Ausgangszustand zurück. Dadurch lässt sich schnell wieder ein neues Szenario beginnen, ohne manuell aufräumen zu müssen.

Für jede Entität, seien es Maschinen, Materialien oder der Produktionsauftrag selbst, existiert eine Shell, in der statische Informationen und aktuelle Mess- oder Statuswerte gleichermaßen abgelegt sind. Ein

eigenes Submodell dient als zentrale Drehscheibe für Laufzeitdaten wie den internen Systemstatus oder den Schlüssel der gerade laufenden Prozessinstanz. Dank der semantischen Kennzeichnungen können externe Anwendungen gezielt auf einzelne Elemente zugreifen, ohne den vollständigen Pfad kennen zu müssen.

Ein REST-Controller bündelt alle Schnittstellen, ein Handler verwaltet Zustände und verteilt sie auf die entsprechenden Submodelle, während Hilfsklassen das Anlegen und Aktualisieren von Shell-Elementen übernehmen. Worker-Module stellen die Verbindung zur Workflow-Engine her; jede Worker-Klasse ist für einen klar umrissenen Aufgabentyp verantwortlich.

Externe Dienste ergänzen die Laufzeitumgebung. Eine BaSyx-Instanz beherbergt die Verwaltungsschalen, eine Workflow-Engine führt die BPMN-Modelle aus, ein Nachrichtendienst transportiert Prozess- und Prüfeignisse, und ein Node-RED-Flow vereinfacht die Kopplung einzelner Schritte. All diese Dienste laufen eigenständig, werden aber über Adressen konfiguriert, die die Anwendung beim Start einliest.

Die Anwendung wird in der Regel containerisiert betrieben. Ein typischer Aufbau umfasst die BaSyx-Umgebung, die Workflow-Engine und das Backend selbst in separaten Containern, optional ergänzt durch einen Nachrichtendienst. Dadurch lässt sich das System leicht auf einem Entwicklungsrechner, in einer Laborumgebung oder in der Cloud starten.

Wer das System in eine bestehende IT-Landschaft integrieren möchte, geht im Wesentlichen wie folgt vor: Zunächst werden die in den VWSen vergebenen Identifikatoren in das eigene Stammdaten- oder Master-Data-System übernommen. Anschließend wird ein Gateway eingerichtet, das den Verkehr zu den REST-Endpunkten und zum Nachrichtendienst lenkt. Schließlich werden betriebliche Auftragsnummern auf die IDs des Demonstrators gemappt, sodass vorhandene Tools die Abläufe auswerten können. Sobald dieser Rahmen steht, kann die Produktion in kleinen Schritten auf die VWS-basierte Steuerung umgestellt oder in einer Sandbox für Schulungen und Demonstrationen genutzt werden.

Damit bietet der Demonstrator eine praxisnahe, zugleich überschaubare Grundlage, um digitale Zwillinge in der Fertigung kennenzulernen und eigene Erweiterungen auszuprobieren – ohne sich sofort in Details wie Portnummern, Pfadstrukturen oder Konfigurationsdateien vertiefen zu müssen.

### 5.1.1 BaSyx UI – Leitungssatzherstellung (Demonstrator UI)

Im Rahmen von TP14 wurde ein Modul entwickelt, das die Abbildung und Nachverfolgung eines Produktionsprozesses für ein Halbprodukt, konkret ein Kabel mit gecrimptem Terminal, ermöglicht. Ziel war es, den gesamten Ablauf von der Produktauswahl bis zur Qualitätskontrolle digital abzubilden und über die VWS-gestützte Infrastruktur steuerbar zu machen.

Das Modul wurde als eigenständiges BaSyx UI-Modul umgesetzt und nutzt das zuvor beschriebene Modul-Feature. Verwendet werden insbesondere bereits bestehende Funktionen des Web UIs wie die Abfrage von Verwaltungsschalen und Teilmodellen, sowie die Filterung nach spezifischen Semantischen IDs und Short-IDs. In der visuellen Umsetzung werden insbesondere die Plugins für „Bills of Materials“ und „Time Series Data“ des BaSyx UIs wiederverwendet. Im Folgenden sind die einzelnen Prozessschritte als Teil des Moduls beschrieben.

#### **Schritt 1: Produktauswahl**

Der Benutzer wählt ein konkretes Produkt aus, das produziert werden soll. In unserem Anwendungsfall handelt es sich um ein Kabel mit definiertem Querschnitt, das als Halbprodukt klassifiziert ist, siehe Abbildung 5-1.

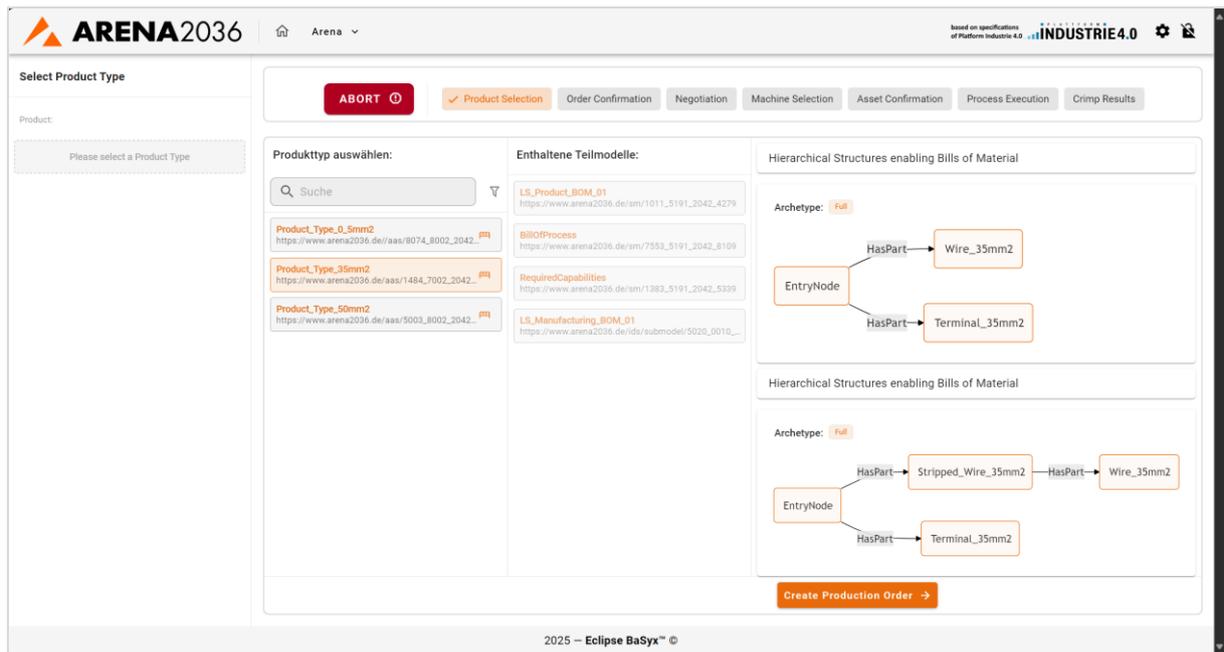


Abbildung 5-1: BaSyx-Modul VWS4LS Demonstrator - Produktauswahl

## Schritt 2: Auftragsbestätigung

Im nächsten Schritt, siehe Abbildung 5-2 wird die Auftragsgröße festgelegt. Für Demonstrationszwecke wird ein einfacher Auftrag mit einer Charge (engl. Batch), einem Los (engl. Lot) und einem Produkt abgebildet.

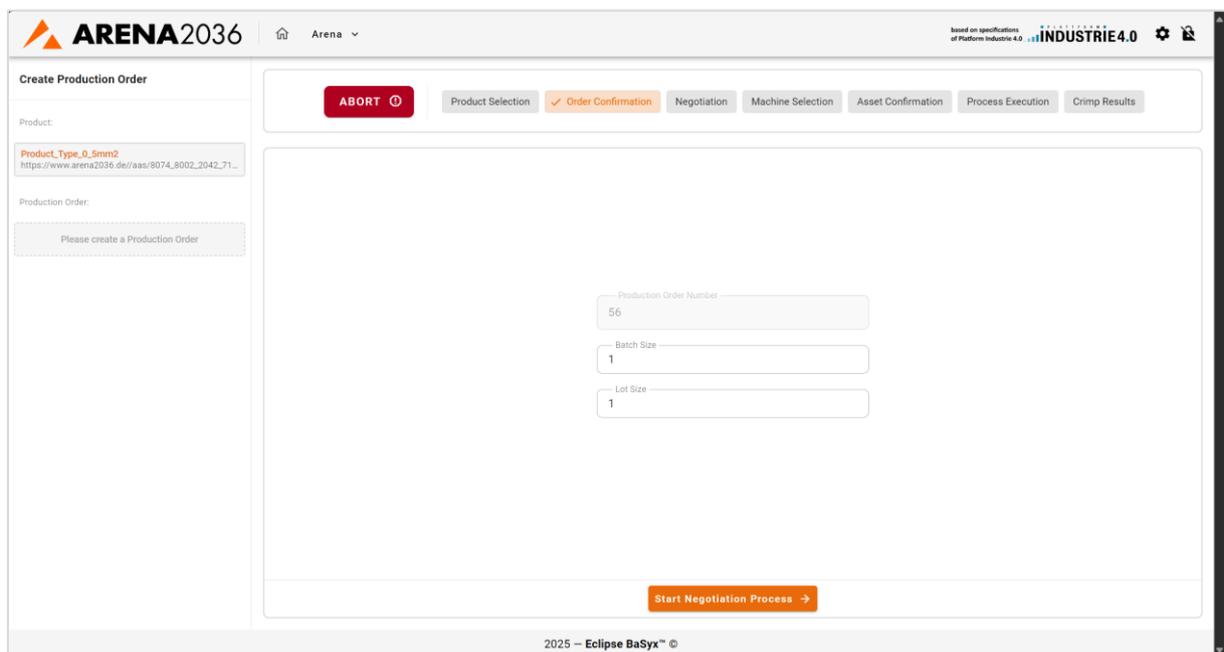


Abbildung 5-2: BaSyx-Modul VWS4LS Demonstrator - Auftragsdaten

## Schritt 3: Maschinenverhandlung

Auf Basis der Produktspezifikation wird ein Produktionsprozess abgeleitet. Eine Flowable-Engine wird genutzt, um eine Verhandlung über mögliche Produktionsressourcen (Maschinen) anzustoßen, siehe Abbildung 5-3. In dieser Phase geben verschiedene virtuelle Produktionssysteme Gebote ab.

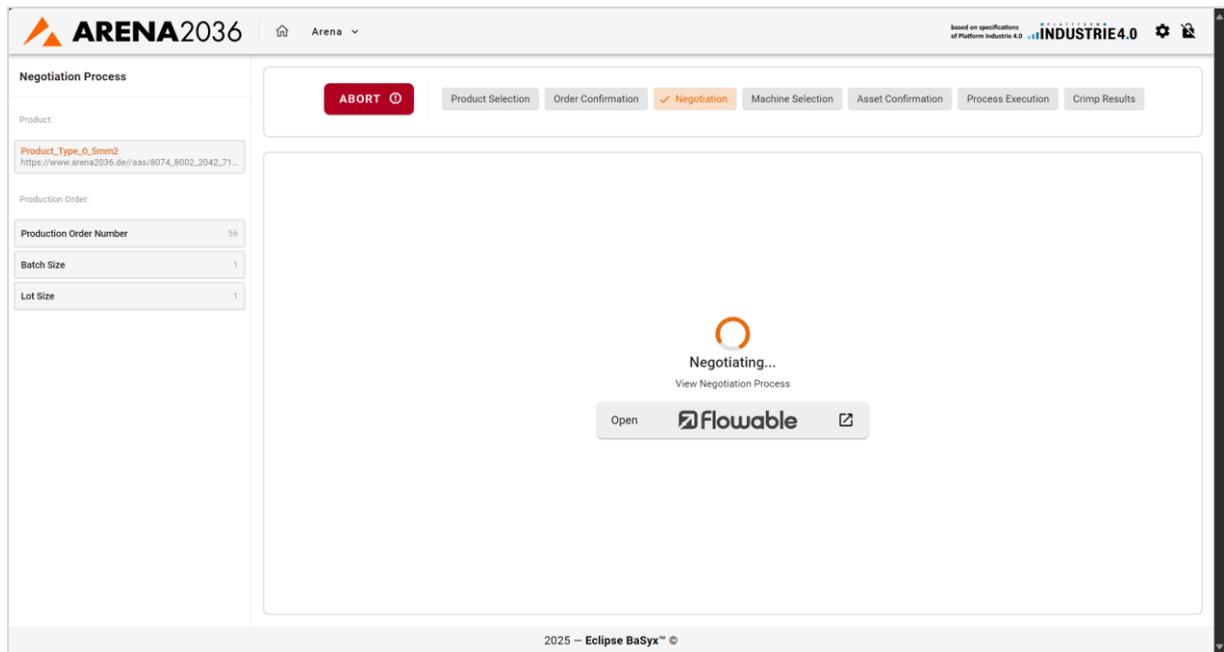


Abbildung 5-3: BaSyx-Modul VWS4LS Demonstrator – Verhandlungsprozess

#### Schritt 4: Maschinenauswahl

Die Ergebnisse der Verhandlung werden dem Nutzer präsentiert. Er kann nun eine geeignete Maschine auswählen, auf der der Produktionsprozess durchgeführt werden soll, siehe Abbildung 5-4.

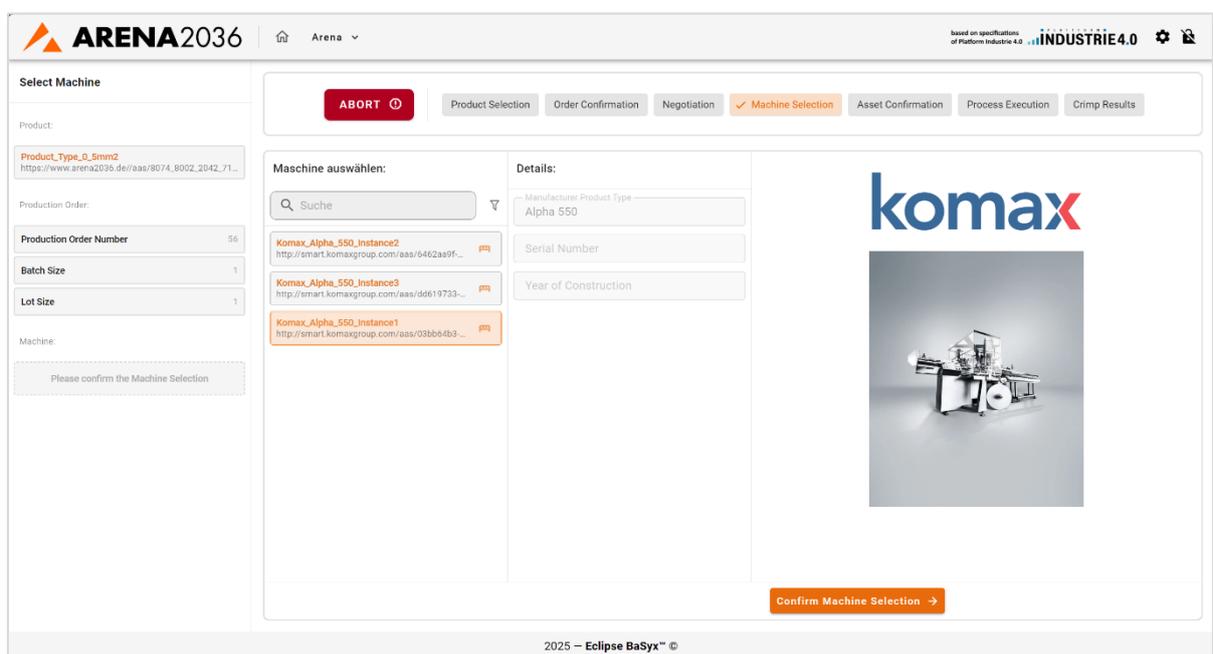


Abbildung 5-4: BaSyx-Modul VWS4LS Demonstrator - Maschinenauswahl

#### Schritt 5: Asset-Bestätigung per Scan

Vor dem Produktionsstart werden alle relevanten Assets durch QR-Code-Scans bestätigt, siehe Abbildung 5-5. Dazu zählen:

- Die reale Produktionsmaschine
- Das gewählte Kabel (Rohmaterial)

- Das passende Terminal
- Das eingesetzte Werkzeug (z. B. Crimpwerkzeug)

Die Validierung dient der Sicherstellung, dass alle für die Produktion benötigten Komponenten korrekt bereitgestellt wurden.

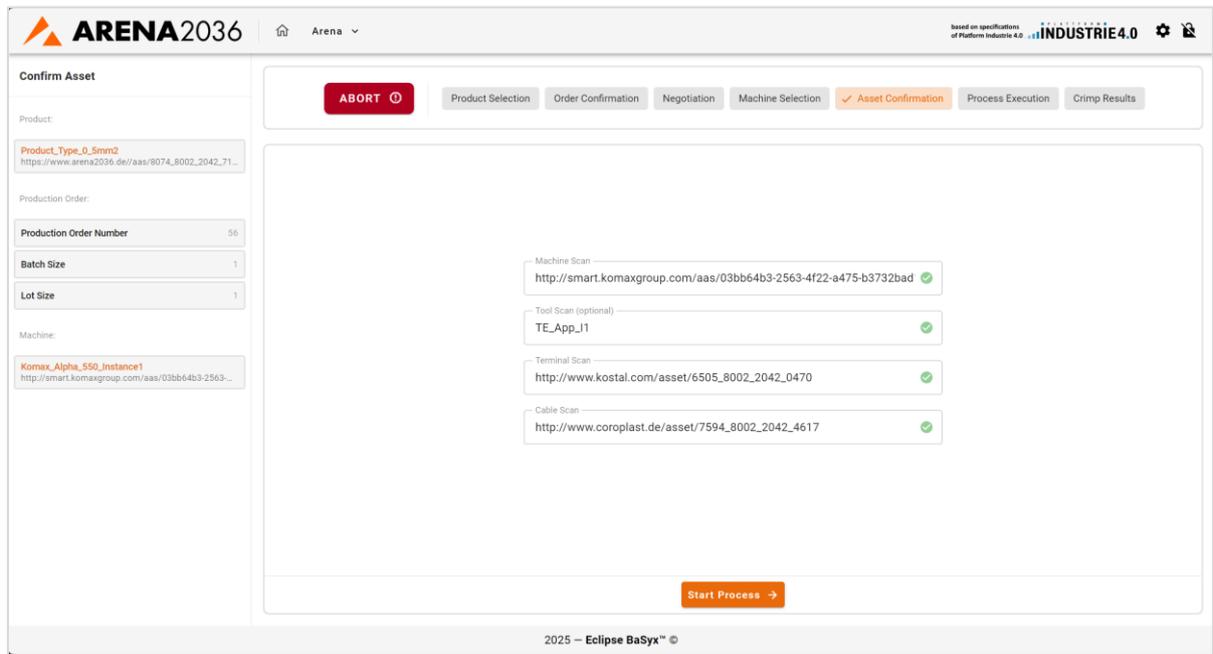


Abbildung 5-5: BaSyx-Modul VWS4LS Demonstrator - Asset Identifikation (Spezifische Asset ID)

### Schritt 6: Prozessdurchführung

Der Produktionsprozess wird auf Basis eines BPMN-Diagramms visualisiert, siehe Abbildung 5-6. Das Diagramm wird mit *bpmn.io* eingebettet und zeigt stets den aktuellen Zustand des Prozesses an, z. B. „Schneiden“, „Abisolieren“ oder „Crimpen“.

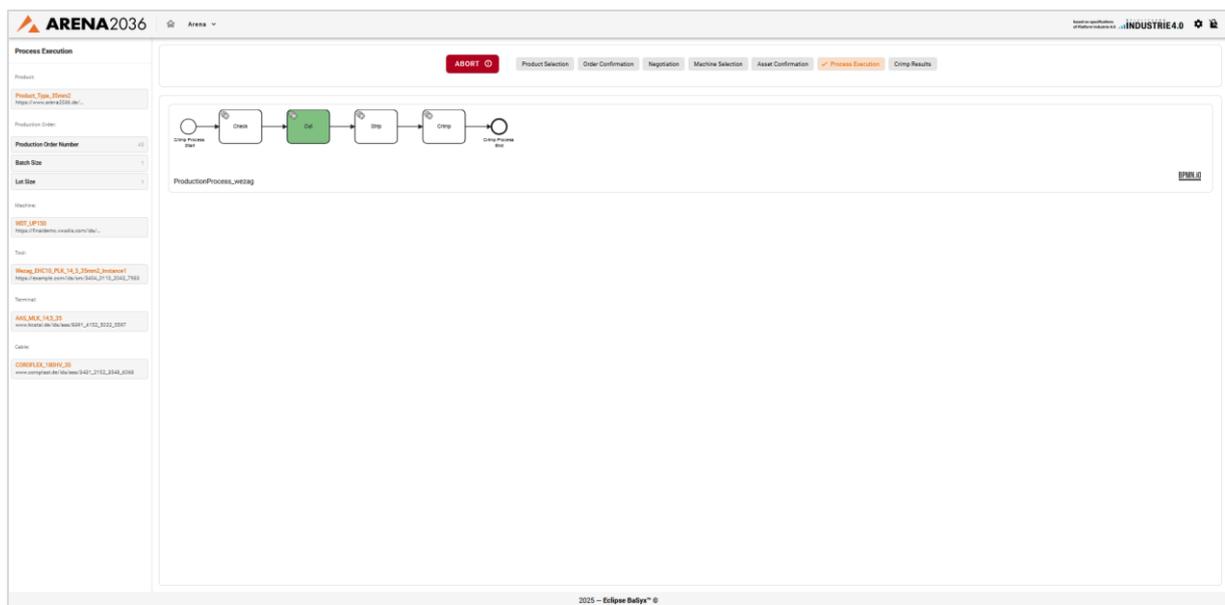


Abbildung 5-6: BaSyx-Modul VWS4LS Demonstrator - Prozessdurchführung

### Schritt 7: Qualitätsprüfung der Crimpung

Nach dem Crimpvorgang wird die sogenannte Crimpkurve als Qualitätsmerkmal ausgewertet. Die Kurve wird in einem Diagramm visualisiert und mit definierten Grenzwerten verglichen, siehe Abbildung 5-7. Der Benutzer erhält eine Rückmeldung darüber, ob der Produktionsschritt innerhalb der Qualitätsparameter liegt.



Abbildung 5-7: BaSysx-Modul VWS4LS Demonstrator – Qualitätsprüfung der Crimpung

## 5.2 Änderungsmanagement Einleitung

Ziel des VWS4LS-Demonstrators ist es, verschiedene Anwendungsfälle und Einsatzmöglichkeiten der Verwaltungsschale (VWS) zu veranschaulichen. Einer dieser Anwendungsfälle ist das Änderungsmanagement.

*Hinweis:* Das Änderungsmanagement bezieht sich in diesem Fall auf die Benachrichtigung, wenn Änderungen in einer VWS und deren Infrastruktur vorgenommen werden. Die entsprechenden VWS-Elemente müssen durch das Event-Element der VWS explizit abonniert werden.

Als mögliche Events werden

- die Erstellung neuer,
- das Update – also die Änderung der Werte der Attribute bereits existierender
- oder das Löschen von

Teilmodellen, Teilmodellelementen, Verwaltungsschalen und Concept Descriptions betrachtet. Diese möglichen Events werden im Dokument als **Create, Update und Delete** bezeichnet. Hierbei sind *create* und *delete* dem Strukturevent zuzuweisen. Weiterhin gibt es noch andere Arten von Events, wie Security- und Runtime-Events als auch Alarmer, die jedoch in dem VWS4LS-Demonstrator nicht näher betrachtet werden. Abbildung 5-8 stellt einen Baum von VWS-Entitäten dar, von der VWS-Umgebung bis hin zu einem Teilmodellelement, und zeigt die ihnen zugeordneten Eventtypen mit Beispielen. Innerhalb einer einzelnen VWS können strukturelle Änderungen (z. B. Erstellung und Löschung) oder die Aktualisierung von VWS-Entitäten wie Teilmodellen, Teilmodellelementen, VWS-Metadaten und Teilmodell-Metadaten abonniert werden. Die Spezifikation [1] erwähnt auch die Benachrichtigung über die Erstellung und Löschung einer gesamten VWS sowie über infrastrukturbezogene Ereignisse, die außerhalb einer bestimmten VWS auftreten, wie z. B. Alarmer, Sicherheitsvorfälle (z. B. Logging-Ereignisse, Zugriffsverletzungen, Unstimmigkeiten zwischen Rollen und Rechten, Denial-of-Service-Versuche) und Lebenszyklus-Ereignisse von Infrastrukturkomponenten (z. B. Booten, Herunterfahren, Out-of-Memory-Zustände).

Das derzeitige Metamodell definiert das Event Element als Teilmodellelement, wodurch dessen Implementierung implizit auf das Innere einer VWS, genauer gesagt auf ein Teilmodell, beschränkt ist. Es

kann rein logisch gesehen daher nur Ereignisse beobachten, die innerhalb dieser bestimmten VWS auftreten. Die Beobachtung der Erstellung einer VWS-Instanz ist logisch gesehen nicht möglich: In welchem Teilmodell würde man ein Event-Element platzieren, das die Existenz einer VWS überwacht, die entweder noch nicht existiert? Das gleiche Problem gilt für die in Abbildung 5-8 gezeigte Sicherheits-, Alarm- und Infrastruktur, da solche Event Elemente typischerweise nicht dem durch eine einzelne VWS repräsentierten Asset zuzuordnen sind. Diese Einschränkung könnte nur durch die Instanziierung einer VWS überwunden werden, die die VWS-Umgebung selbst repräsentiert. Eine Entscheidung, die diskutiert werden muss. Diese Probleme motivieren eine Neupositionierung des Event Elements innerhalb des VWS-Metamodells. Da die Abbildung auf Englisch ist, wurde für den Begriff VWS die englische Abkürzung AAS (Asset Administration Shell) genutzt.

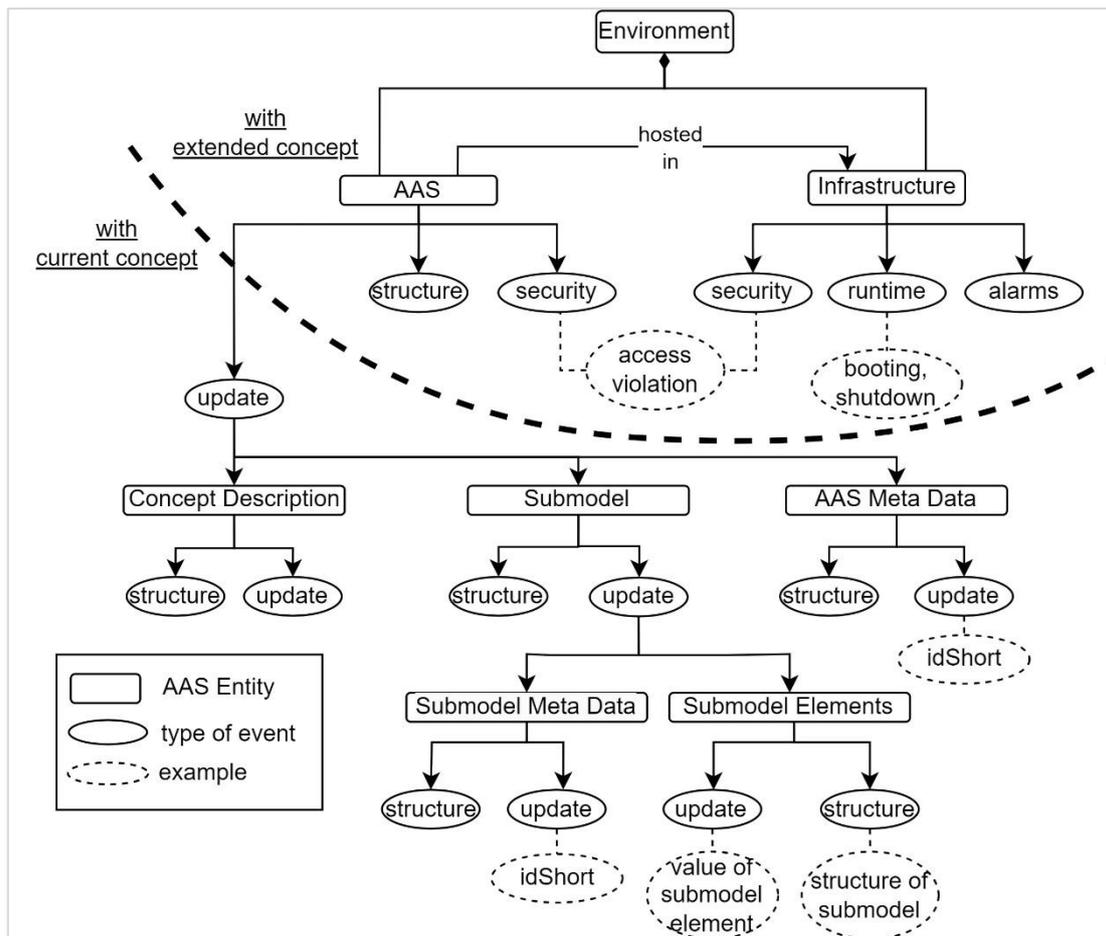


Abbildung 5-8: VWS-Entitäten und ihre Eventtypen

Das entwickelte technische Konzept wurde im VWS4LS-Demonstrator prototypisch umgesetzt und evaluiert. Über die entwickelte Weboberfläche (siehe Kapitel 5.1.1) können VWS-Entitäten, im folgenden „Element“ genannt, abonniert und über Änderungen der abonnierten Elemente informiert werden. Je nach Szenario, sollen diese Änderungen vom Nutzer angenommen oder abgelehnt werden dürfen. Das allgemeine Anwendungsfalldiagramm dazu ist in Abbildung 5-9 zu sehen.

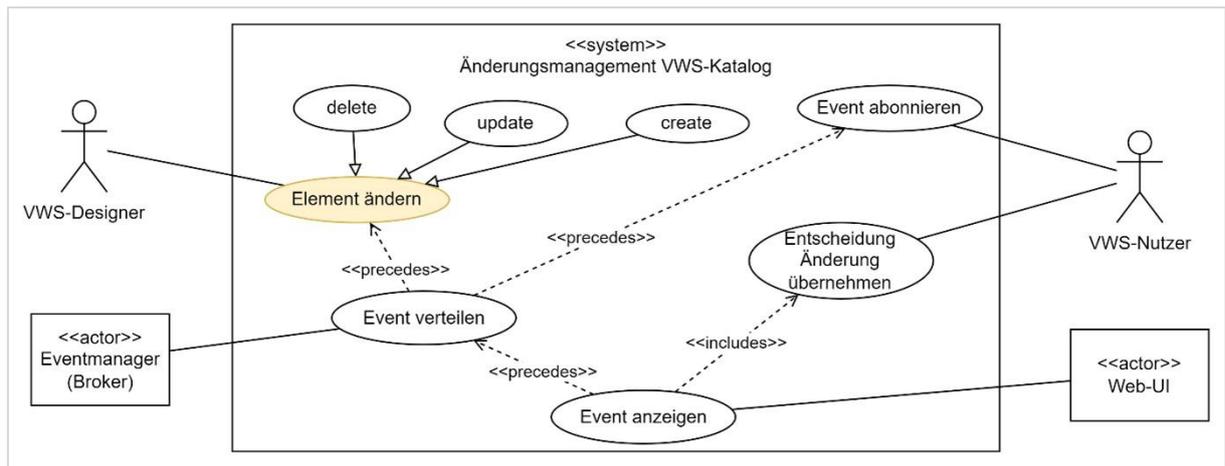


Abbildung 5-9: Anwendungsfalldiagramm Änderungsmanagement

Ziel dieses Anwendungsfalls ist es nicht nachgeschaltete Automatismen, nach einer Änderung auszuführen, sondern lediglich über eine erfolgte Änderung zu informieren. Solche Automatismen beziehen sich im Änderungsmanagement der Leitungssatzentwicklung beispielsweise auf Entscheidungs- und Anpassungsvorgänge, wenn z. B. die Länge einer Leitung geändert werden muss. Das ist nicht Teil des hier definierten Änderungsmanagements.

Aus dem allgemeinen Anwendungsfall und den zuvor in Abbildung 5-8 definierten VWS-Eventtypen „Structure“ (*create*, *delete*) und „Update“, wurden drei Szenarien entwickelt, die im Folgenden beschrieben sind.

### 5.2.1 Szenario 1: Abonnieren eines Teilmodellelements (update)

In diesem Szenario wird sich auf das Abonnieren einer Änderung eines Teilmodellelements fokussiert. D.h. der Abonnent möchte informiert werden, sobald sich der Wert eines Teilmodellelements verändert. Dies kann die Metainformation des Teilmodellelements betreffen oder aber den Wert des Elements selbst. Das Szenario ist in Abbildung 5-10 dargestellt. Als Voraussetzung wird angenommen, dass der VWS-Designer bereits eine VWS mit ihren Teilmodellen erstellt hat (1). Über einen Event-Broker abonniert der VWS-Designer anschließend ein Teilmodellelement (2). Sobald eine Applikation den Wert dieses Teilmodellelements ändert (3), wird der VWS-Designer durch den Broker über diese Änderung informiert (4).

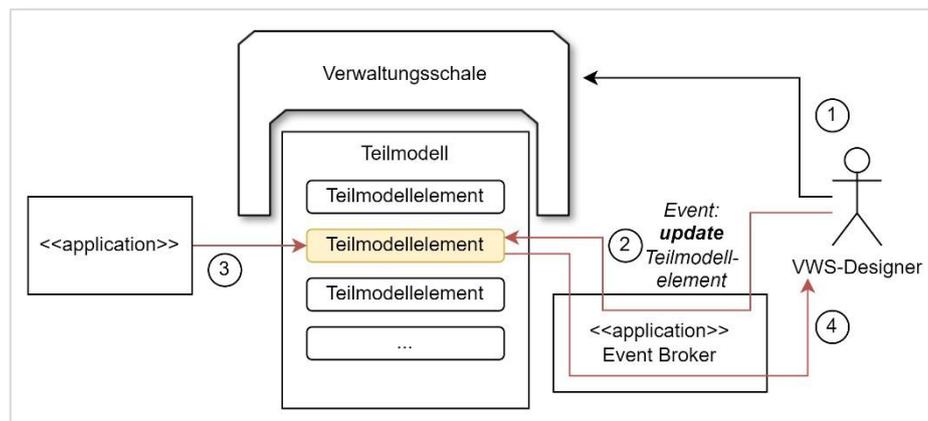


Abbildung 5-10: allgemeines Szenario - Abonnieren eines Teilmodellelements (update)

#### Bezug zu dem VWS4LS-Demonstrator

Bezogen auf den VWS4LS-Demonstrator nimmt der Produktionsleiter die Rolle des VWS-Designers ein und die externe Applikation wird von einem Manufacturing Execution System (MES) eingenommen,

siehe Abbildung 5-11. Der Produktionsleiter legt eine Auftrags-VWS an (1) und abonniert den Auftragsstatus (2). Ein Mitarbeiter führt die notwendigen Produktionsschritte aus und trägt deren Ergebnisse in das MES ein. Dieses wiederum aktualisiert die Auftrags-VWS und ändert unter anderem auch das Teilmodellelement „Produktionsstatus“. Sobald eine Änderung vorgenommen wird, informiert der Broker den Produktionsleiter über die aktuelle Änderung mit Hilfe einer Pop-Up Benachrichtigung.

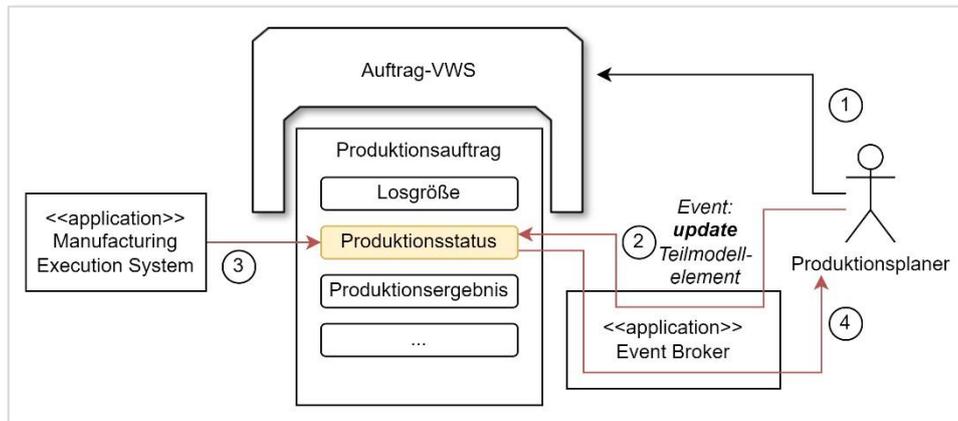


Abbildung 5-11: VWS4LS Szenario - Abonnieren eines Teilmodellelements (update)

### 5.2.2 Szenario 2: Abonnieren eines neuen Teilmodells in einer VWS (create)

Dieses Szenario behandelt die Nachverfolgung von neu hinzugefügten Teilmodellen zu einer bekannten VWS (s. Abbildung 5-12). Es wird vorausgesetzt, dass bereits sowohl eine originale als auch eine Kopie der originalen VWS vorliegen. Die originale VWS wird vom VWS-Designer erzeugt und die Kopie vom VWS-Nutzer verwendet.

Um über ein neu hinzugekommenes Teilmodell informiert zu werden, wählt der VWS-Abonnent zuerst den Eventtyp *create* und danach die VWS, in der die Teilmodelle beobachtet werden sollen aus (1). Dies erledigt er mit Hilfe einer Weboberfläche, die in der Abbildung nicht dargestellt ist. Danach wird im Broker ein Event angelegt, das auslöst, sobald ein neues Teilmodell in der originalen VWS erstellt wurde (2). Sobald ein neues Teilmodell durch den VWS-Designer in der originalen VWS hinzugefügt wird, sendet der Event Broker eine Benachrichtigung an den VWS-Nutzer.

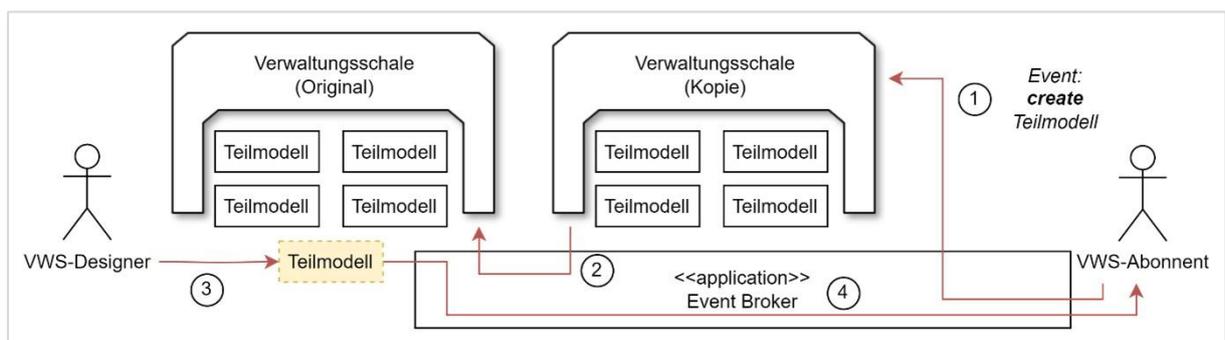


Abbildung 5-12: allgemeines Szenario - Abonnieren eines neuen Teilmodell in einer VWS (create)

#### Bezug zu dem VWS4LS-Demonstrator

Bezogen auf den VWS4LS-Demonstrator kann in diesem Szenario der Komponentenkatalog eines Konfektionärs herangezogen werden. Die originalen VWS der Komponentenhersteller (VWS-Designer) liegen in diesem Katalog als Kopien und haben aber noch eine Referenz zu der originalen VWS. Der Konfektionär kann bei Durchsicht der Komponenten die VWS auswählen, zu denen er benachrichtigt werden möchte, sobald ein neues Teilmodell hinzugefügt wird (2). Sobald dann die abonnierte Änderung eintritt (3), wird der Konfektionär mittels einer Pop-Up Benachrichtigung über die Änderung informiert (4). Danach kann der Konfektionär entscheiden, ob er dieses neue Teilmodell in seine VWS-Kopie

übernehmen möchte oder nicht. Das angewandte Szenario ist mit Ausnahme des letzten Schritts, der Änderungsübernahme, in Abbildung 5-13 dargestellt.

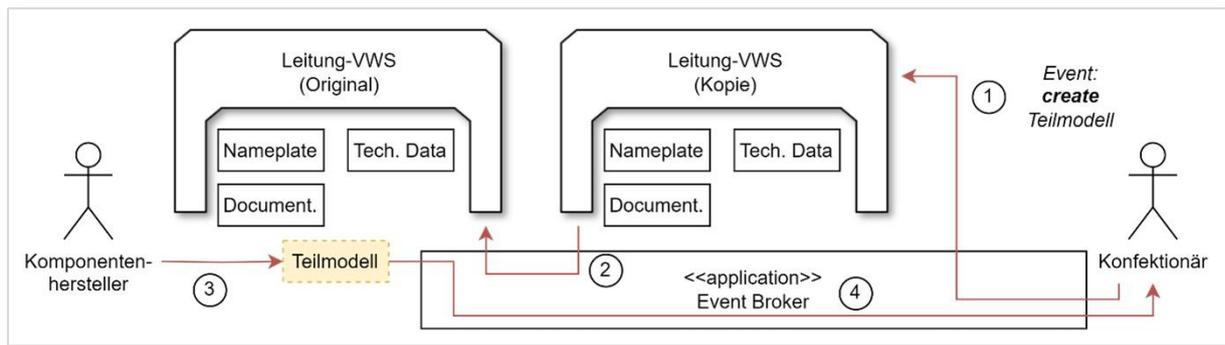


Abbildung 5-13: VWS4LS Szenario - Abonnieren eines neuen Teilmodell in einer VWS (create)

### 5.2.3 Szenario 3: Abonnieren einer neuen VWS (create)

In diesem Szenario (siehe Abbildung 5-14) soll der Abonnent über das Hinzufügen einer neuen VWS informiert werden. Dazu abonniert der Abonnent über den Broker ein Event, dass mit dem Hinzufügen einer neuen VWS in einem Repository auslöst (1). Sobald der VWS-Designer eine neue VWS in das Repository hochlädt (2), informiert der Broker den Abonnenten über die Änderung (3).

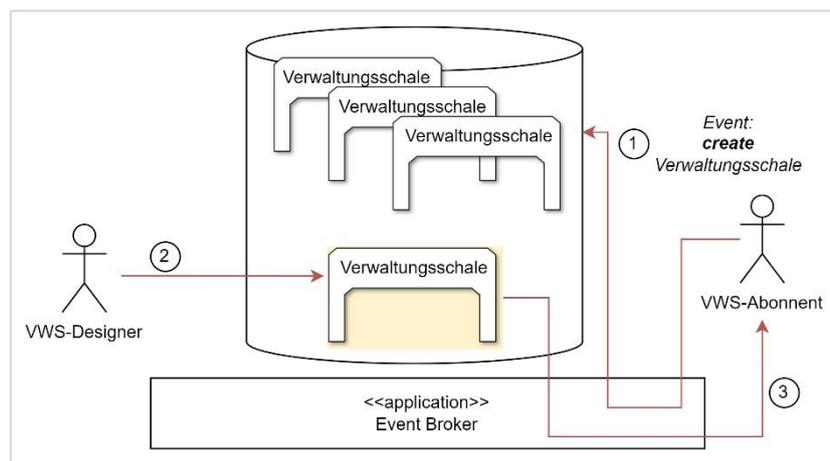


Abbildung 5-14: allgemeines Szenario - Abonnieren einer neuen VWS (create)

#### Bezug zu dem VWS4LS-Demonstrator

Bezogen auf den VWS4LS-Demonstrator könnte es ein Repository mit Leitungssatz-VWS für einen definierten OEM geben, siehe Abbildung 5-15. Der OEM kann dann die Benachrichtigung über neu hinzugefügte VWS mit Hilfe des Eventtyps „create“ und dem zu beobachtenden Repository abonnieren (1). Sobald ein Leitungssatz hergestellt wurde, lädt der Konfektionär die zugehörige VWS in das Repository des OEM (2) und der Broker sendet dem OEM eine Pop-Up Benachrichtigung (3).

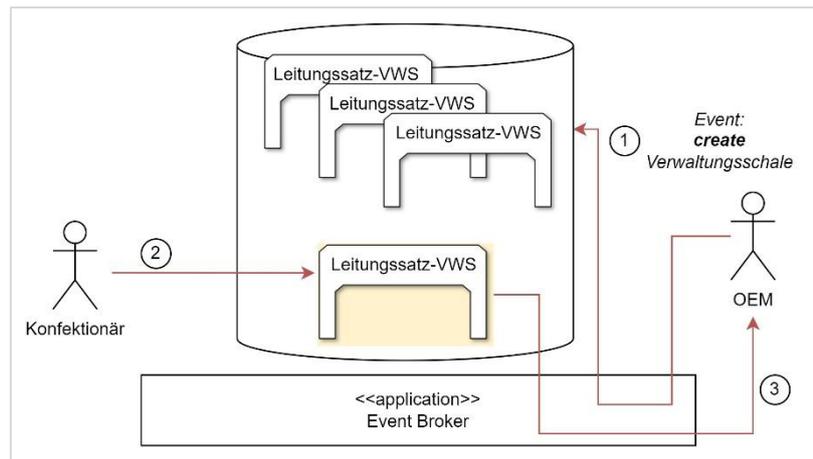


Abbildung 5-15: VWS4LS Szenario - Abonnieren einer neuen VWS (create)

Das Ziel der Arbeit ist es zu beschreiben, wie diese Änderungsarten mithilfe des Event-Elements aus der Metamodell-Spezifikation übertragen werden können. Dazu gehören unter anderem folgende Fragen:

- Wie wird das "Abonnieren" mithilfe des Event-Elements modelliert?
- Wie werden diese Änderungen technisch übertragen?
- Welche Infrastrukturkomponenten sind dafür notwendig?
- Wie spielen diese Infrastrukturkomponenten und Rollen (VWS-Nutzer, VWS-Owner, VWS-Designer) zusammen, also wie sieht die Systemarchitektur aus?
- Wie ist der Inhalt der mithilfe des Event-Elements übertragenen Informationen abgebildet?

#### 5.2.4 Rollen und ihre Interaktionen

Die hier dargestellten Abläufe und Rollen fokussieren sich auf die Modellierung und Abbildung der oben eingeführten Szenarien mithilfe der VWS.

Um ein Asset in seiner VWS zu beschreiben, ist die Rolle des VWS-Designers erforderlich. Dieser ist dafür verantwortlich, Änderungen im Lebenszyklus eines Assets in entsprechende Änderungen der zugehörigen VWS bzw. Teilmodelle (TM) zu überführen. Die Änderungen können dabei durch eine Person oder eine Software durchgeführt werden.

Der Ablauf für das Änderungsmanagement ist in Abbildung 5-16 schematisch für das Abonnieren einer einzelnen Verwaltungsschale dargestellt und erfolgt für alle anderen VWS-Elemente in gleicher Weise.

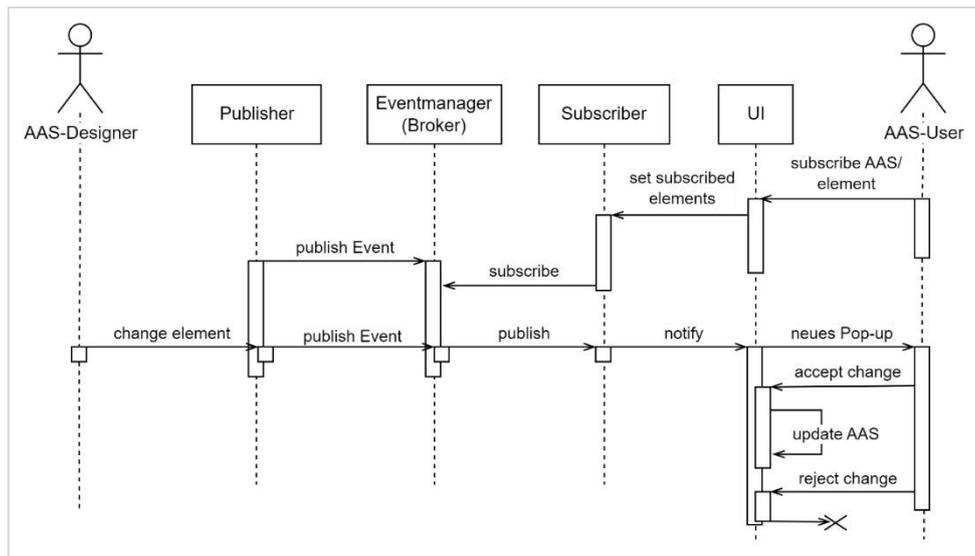


Abbildung 5-16: Sequenzdiagramm des allgemeinen Änderungsmanagements

### 5.3 Anforderungsanalyse

Zu Beginn wurde eine Anforderungsanalyse durchgeführt, um die Ziele für die zu entwickelnden Konzepte und die darauffolgende Implementierung festzulegen.

Wie im vorherigen Abschnitt beschrieben werden vier Aspekte betrachtet:

1. VWS-Elemente, die abonniert werden können,
2. VWS-Elemente, die abonniert werden können,
3. Architektur, in der das Event umgesetzt wird und
4. Technologie, mit der Events versendet werden.

#### 5.3.1 Die Arten der zu abonnierenden VWS-Elemente

Aus dem Anwendungsfall und den abgeleiteten Szenarien ergeben sich die in Abbildung 5-17 dargestellten Elemente der VWS, die der Nutzer bezüglich des Änderungsmanagements abonnieren kann. Dies ist eine Untermenge der in Abbildung 5-8 dargestellten VWS-Elemente und Eventtypen, da nicht alle Möglichkeiten im Projekt aufgrund der zeitlichen Begrenzung umgesetzt werden konnten. Tabelle 5-1 schlüsselt hierbei die einzelnen Eventtypen, die abonniert werden können (*create*, *update*, *delete*) mit ihren Anforderungen (Tabelle 5-4) für die gezeigten Elemente auf. Hierbei wurde der Eventtyp „Structure“ in die zwei Subtypen *create* und *delete* aufgeschlüsselt.

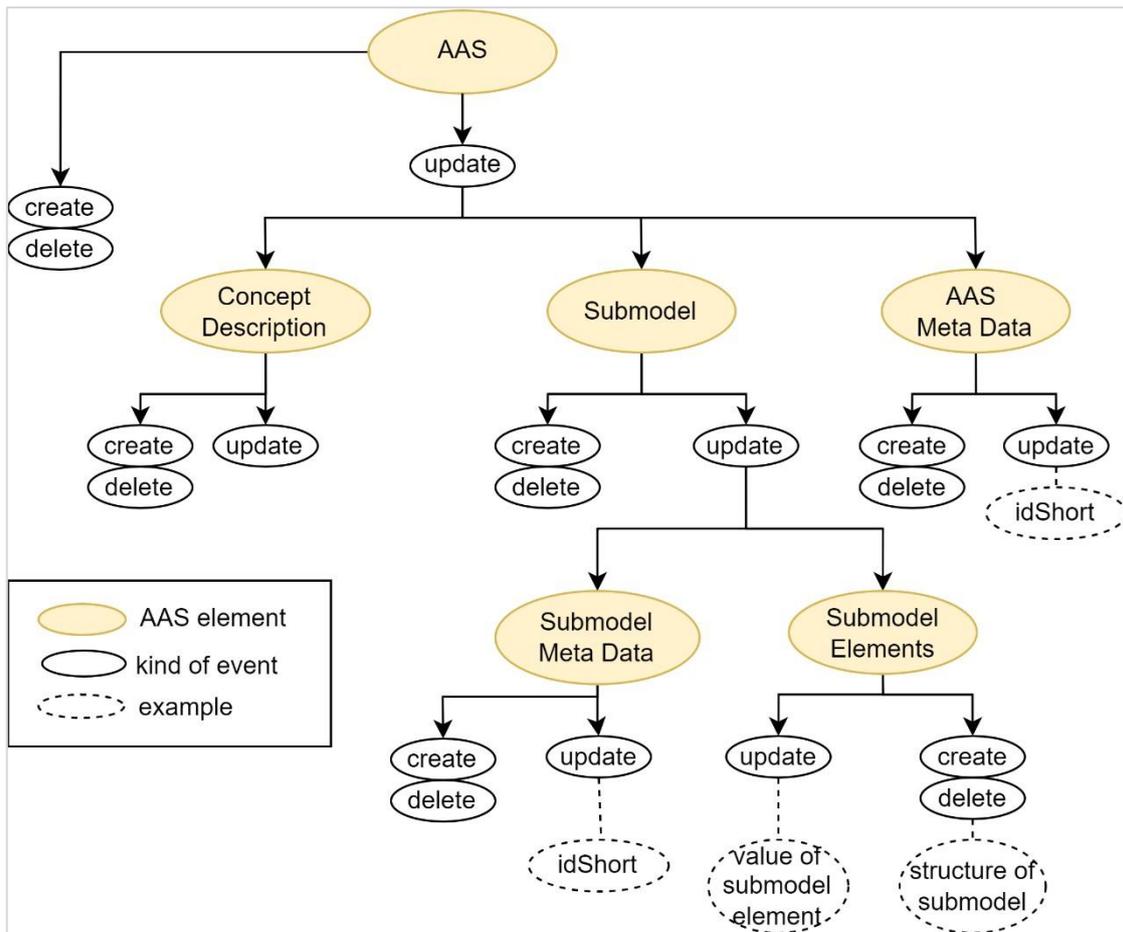


Abbildung 5-17: Varianten der zu abonnierenden VWS-Elemente mit ausgewählten Eventtypen

Die konkrete Anwendung des Event-Elements auf die in Abbildung 5-17 gezeigten VWS-Elemente erfolgt in Kapitel 5.3.4.

Lfd. Nr.	Element	Event-typ	Beschreibung	Bezug zu Anforderungsnummer (Tabelle 4)
1	VWS	create	Benachrichtigung über das Hinzufügen einer neuen VWS.	<a href="#">#AF 1</a> , <a href="#">#AF 15</a>
2	VWS	delete	Benachrichtigung über das Löschen einer bestehenden VWS.	<a href="#">#AF 2</a> , <a href="#">#AF 3</a>
3	VWS	update	Benachrichtigung über ein geändertes Element in der VWS (z.B. Metainformation, Teilmodell, Concept Description)	<a href="#">#AF 2</a> , <a href="#">#AF 4</a> , <a href="#">#AF 5</a>
4	VWS Metainformation	update, create, delete	Benachrichtigung über eine geänderte Metainformation. <u>Eingrenzung:</u> Wird eine einzelne Metainformation geändert, werden alle Metainformationen der VWS als update gesendet.	<a href="#">#AF 2</a> , <a href="#">#AF 6</a> , <a href="#">#AF 7</a>
5	Concept Description	create	Benachrichtigung über das Hinzufügen einer neuen Concept Description <b>in einer VWS</b> .	<a href="#">#AF 1</a> , <a href="#">#AF 8</a> , <a href="#">#AF 15</a>
7	Concept Description	delete	Benachrichtigung über das Löschen einer Concept Description <b>aus einer VWS</b> .	<a href="#">#AF 3</a> , <a href="#">#AF 8</a>

9	Concept Description	update	Benachrichtigung über die Aktualisierung einer Concept Description. <u>Eingrenzung:</u> Wenn ein Element in einer Concept Description geändert wird, erfolgt immer ein update der gesamten Concept Description.	<a href="#">#AF 2</a> , <a href="#">#AF 8</a> , <a href="#">#AF 9</a>
10	Teilmodell	create	Benachrichtigung über das Hinzufügen eines neuen Teilmodells <b>in einer VWS</b> .	<a href="#">#AF 1</a> , <a href="#">#AF 8</a> , <a href="#">#AF 15</a>
12	Teilmodell	delete	Benachrichtigung über das Löschen eines Teilmodells <b>aus einer VWS</b> .	<a href="#">#AF 3</a> , <a href="#">#AF 8</a>
14	Teilmodell	update	Benachrichtigung über ein geändertes Element in einem Teilmodell (z.B. Metainformation, Teilmodellinhalt).	<a href="#">#AF 2</a> , <a href="#">#AF 5</a>
15	Teilmodell Metainformation	update, create, delete	Benachrichtigung über eine geänderte Metainformation. <u>Eingrenzung:</u> Wird eine einzelne Metainformation geändert, werden alle Metainformationen dem Teilmodell als update gesendet.	<a href="#">#AF 2</a> , <a href="#">#AF 6</a> , <a href="#">#AF 7</a>
16	Teilmodellelement	create	Benachrichtigung über das Hinzufügen eines neuen Teilmodellelements in einem Teilmodell.	<a href="#">#AF 1</a> , <a href="#">#AF 15</a>
17	Teilmodellelement	delete	Benachrichtigung über das Löschen eines Teilmodellelements in einem Teilmodell.	<a href="#">#AF 3</a>
18	Teilmodellelement	update	Benachrichtigung über ein geändertes Teilmodellelement in einem Teilmodell. Das können sowohl geänderte Metainformationen (z. B. idShort) als auch der geänderte Wert des Elements selbst sein.	<a href="#">#AF 2</a> , <a href="#">#AF 5</a>

Tabelle 5-1: VWS-Elemente und ihre zu abonnierenden Operationen

### 5.3.2 Spezifizierung des Event-Elements

In Abschnitt 5.2 wurde beschrieben, welche Änderungen abonniert werden können, sowie welche Anforderungen sich daraus ergeben. In diesem Abschnitt wird analysiert, welche Anforderungen an das Event-Element zum Abonnieren von Änderungen vorliegen. Zur Einführung wird das im Metamodell der VWS [1] als experimentell gekennzeichnete *BasicEventElement* und der *EventPayload* erklärt.

Das Event-Element kann sich in seinem Typ unterscheiden. In [1] sind bereits erste Eventtypen aufgeführt. Die für das Änderungsmanagement verwendeten sind:

- Strukturelle Änderungen von Verwaltungsschalen
- Aktualisierungen von Teilmodellelementen und deren Attribute

Mit den strukturellen Änderungen können die Eventtypen *create* und *delete* in Verbindung gebracht werden. Für die Aktualisierungen von VWS-Elementen (s. Abbildung 5-17) bietet sich die Änderungsart *update* an. Zusätzlich zu diesen und den anderen definierten Events können laut [1] benutzerdefinierte Events spezifiziert werden. Diese müssen jedoch mit einer global eindeutigen semantischen ID versehen werden. Dies gilt ebenso für die bereits aufgezählten Eventtypen aus [1]. Sie haben ebenfalls noch keine definierte semantische ID ([#AF 12](#))

Elemente, die durch ein Event abonniert werden können, sind wie in der Abbildung 5-17 dargestellt, Verwaltungsschalen, Concept Descriptions, Teilmodelle und alle Teilmodellelemente sowie die Metainformationen von Verwaltungsschalen und Teilmodellen.

*Hinweis:* Concept Descriptions werden in der Spezifikation [1] nicht als zu abonnierendes Element genannt ([#AF 11](#)).

### 5.3.2.1 Spezifikation BasicEventElement

Das Event-Element selbst ist, wie in Abbildung 5-18 zu sehen, spezifiziert.

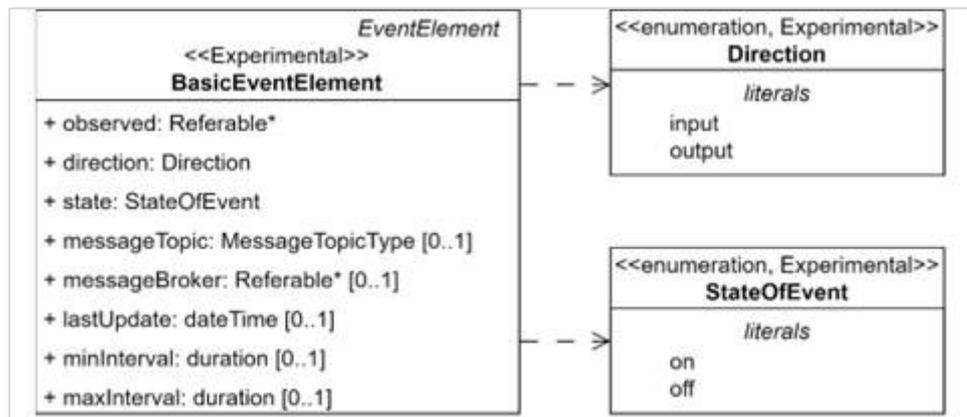


Abbildung 5-18: Spezifikation Event-Element [1]

Die aufgeführten Attribute des Event-Element haben folgende Bedeutung:

- **observed:** Referenz auf ein referenzierbares Element.
- **direction:** Definition der Richtung des Events (eingehend oder ausgehend).
- **state:** Status des Events (An und Aus).
- **messageTopic:** Eindeutige Bezeichnung des Events für die Nachrichtenübertragung mit maximal 255 Zeichen.
- **messageBroker:** Referenz auf eine Applikation, die die Eventnachrichten entgegennimmt und verteilt.
- **lastUpdate:** Letzter Zeitstempel an dem das letzte Event gesendet bzw. empfangen wurde.
- **minInterval:** Minimale Dauer zwischen dem Senden bzw. Empfangen von Events.
- **maxInterval:** Maximale Dauer zwischen dem Senden von Events, wenn kein anderer Trigger eingegangen ist. Dies gilt nur für ausgehende Events.

Es ist zu sehen, dass die Beschreibung des Events technologieunabhängig ist.

Durch die Verwendung typischer MQTT-Begriffe in der Spezifikation des *BasicEventElement* entsteht der Eindruck, dass die Nutzung von MQTT mit einem MQTT-Broker und MQTT-Topics für die Übertragung von Events in der Spezifikation der VWS implizit vorausgesetzt wird. Dies ist jedoch nicht korrekt, da dies in der Spezifikation nicht eindeutig festgelegt ist. Andere Technologien sind somit nicht ausgeschlossen. Allerdings verwenden alternative Technologien oft Konzepte, die dem MQTT-Broker und den Topics ähneln. Ein Beispiel hierfür ist eine HTTP/REST-Schnittstelle, bei der öffentliche Server als Analogie zu einem Broker fungieren und Endpunkte als Entsprechung zu den Topics dienen, an die Events gesendet werden. ([#AF 13](#))

### 5.3.2.2 Spezifikation EventPayload

Der Inhalt des Events ist durch den *EventPayload* in Abbildung 5-19 definiert. Die Attribute haben die folgende Bedeutung:

- **source:** Referenz auf das zugehörige Event-Element
- **sourceSemanticId:** Semantische ID des Event-Elements, bevorzugt als externe Referenz. ([#AF 12](#))

- **observableReference**: Referenz auf das zu beobachtende Element, welches das Attribut *observed* in der Beschreibung des *BasicEventElement* ist.
- **observableSemanticId**: Semantische ID des zu beobachtenden Elements.
- **topic**: Deckungsgleich mit dem *messageTopic* des *BasicEventElement*.
- **subjectId**: Externe Referenz auf das Subjekt, dass die Erzeugung des Events bzw. Payloads vorgenommen hat.
- **timestamp**: Zeitstempel, an dem das Event ausgelöst wurde.
- **payload**: Spezifisch auf das Event angepasster Inhalt als Blob Element. ([#AF 10](#))

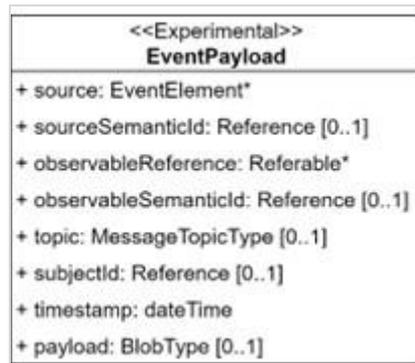


Abbildung 5-19: Spezifikation EventPayload [1]

Die ausführliche Spezifikation des Event-Elements, angepasst auf den Anwendungsfall des Änderungsmanagements, erfolgt in Kapitel 5.3.2.

### 5.3.3 Integration des Events in einen Systementwurf

In diesem Abschnitt wird betrachtet, wie das Event-Element im Kontext der VWS implementiert werden kann. In der aktuellen Spezifikation des VWS-Metamodells [1] ist es als Teilmodellelement gekennzeichnet. Damit ist es laut Spezifikation nur in Teilmodellen nutzbar. Deshalb wird zu Beginn die Datenarchitektur eines Teilmodells betrachtet, in dem Event-Elemente gespeichert werden (siehe Abschnitt 5.3.3.1) können. Eine ausführliche Beschreibung der Integration des derzeit spezifizierten Event-Elements ist in dem gemeinsam veröffentlichten Whitepaper [2] des Projekts DAVID nachzulesen.

Ein anderes Konzept, das in diesem Projekt entwickelt werden soll, sieht die Abwandlung der Metamodell-Spezifikation [1] vor. Hierbei wird das Event-Element nicht als Teilmodellelement definiert. Vielmehr wird es wie eine VWS oder ein Teilmodell mit deren Repository und Registry als eigenständiges Element betrachtet. Hierbei wird in Abschnitt 5.3.3.2 der Fokus auf die Systemarchitektur gelegt. Die Ausarbeitung des zweiten Konzepts erfolgt in Kapitel 5.4.

In beiden Konzepten ist zu beachten, dass ein Event-Element laut den Anforderungen aus Tabelle 4 die Benachrichtigung über Änderungen von Verwaltungsschalen über Concept Descriptions bis hin zu einzelnen Teilmodellelementen ermöglichen soll.

#### 5.3.3.1 Datenarchitektur

Wird das Event-Element der Spezifikation [1] folgend als Teilmodellelement betrachtet, ergeben sich zwei Optionen für dessen Integration [2].

##### 5.3.3.1.1 Option 1: Event im zugehörigen Teilmodell

Bei dieser Option wird das zu einem Teilmodellelement zugehörige Event-Element im gleichen Teilmodell gespeichert, in dem auch das zu abonnierende Teilmodellelement liegt. Abbildung 5-20 zeigt schematisch das Konzept.

Hinweis: Bei Befolgung dieses Schemas wird das Abonnieren von Änderungen an Verwaltungsschalen, deren Metainformationen und Concept Descriptions ausgeschlossen.

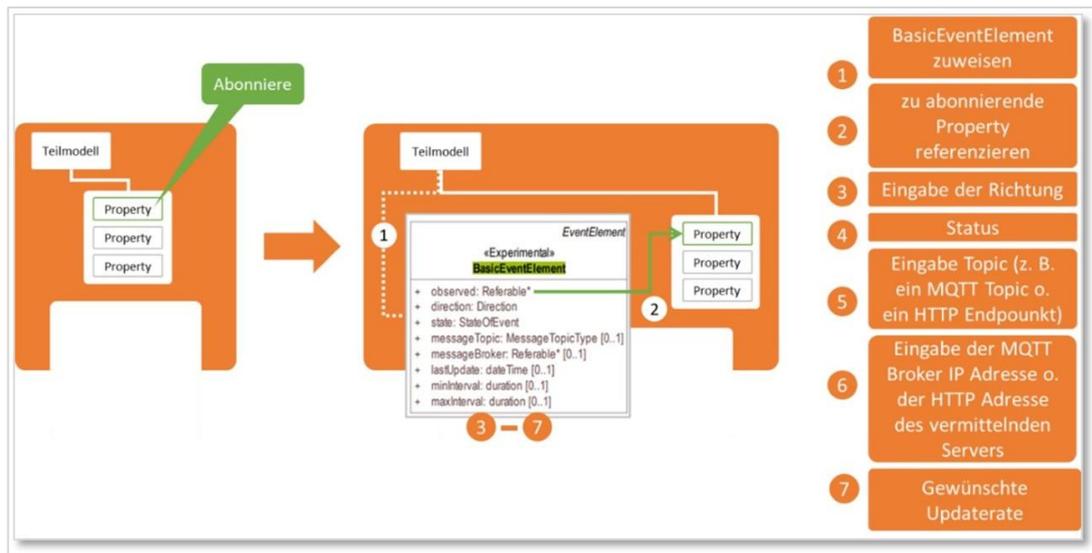


Abbildung 5-20: Abonnieren durch Hinzufügen des BasicEventElement zu einem Teilmodell

### 5.3.3.1.2 Option 2: Ein Teilmodell nur für Events

In dieser Option ist es vorgesehen, dass ein Teilmodell zur Sammlung aller Events für eine VWS genutzt wird, siehe Abbildung 5-21. Hier können Events für die Änderungsnachverfolgung sowohl an der VWS selbst, in der das Teilmodell liegt, Concept Descriptions sowie Teilmodellen und ihren Teilmodellelementen der spezifischen VWS definiert werden. Auch das Abonnieren von Metainformationen ist mit dieser Option möglich.

**Hinweis:** Bei Befolgung dieses Schemas wird das Abonnieren von Informationen über das Hinzufügen bzw. Löschen einer Verwaltungsschale ausgeschlossen.

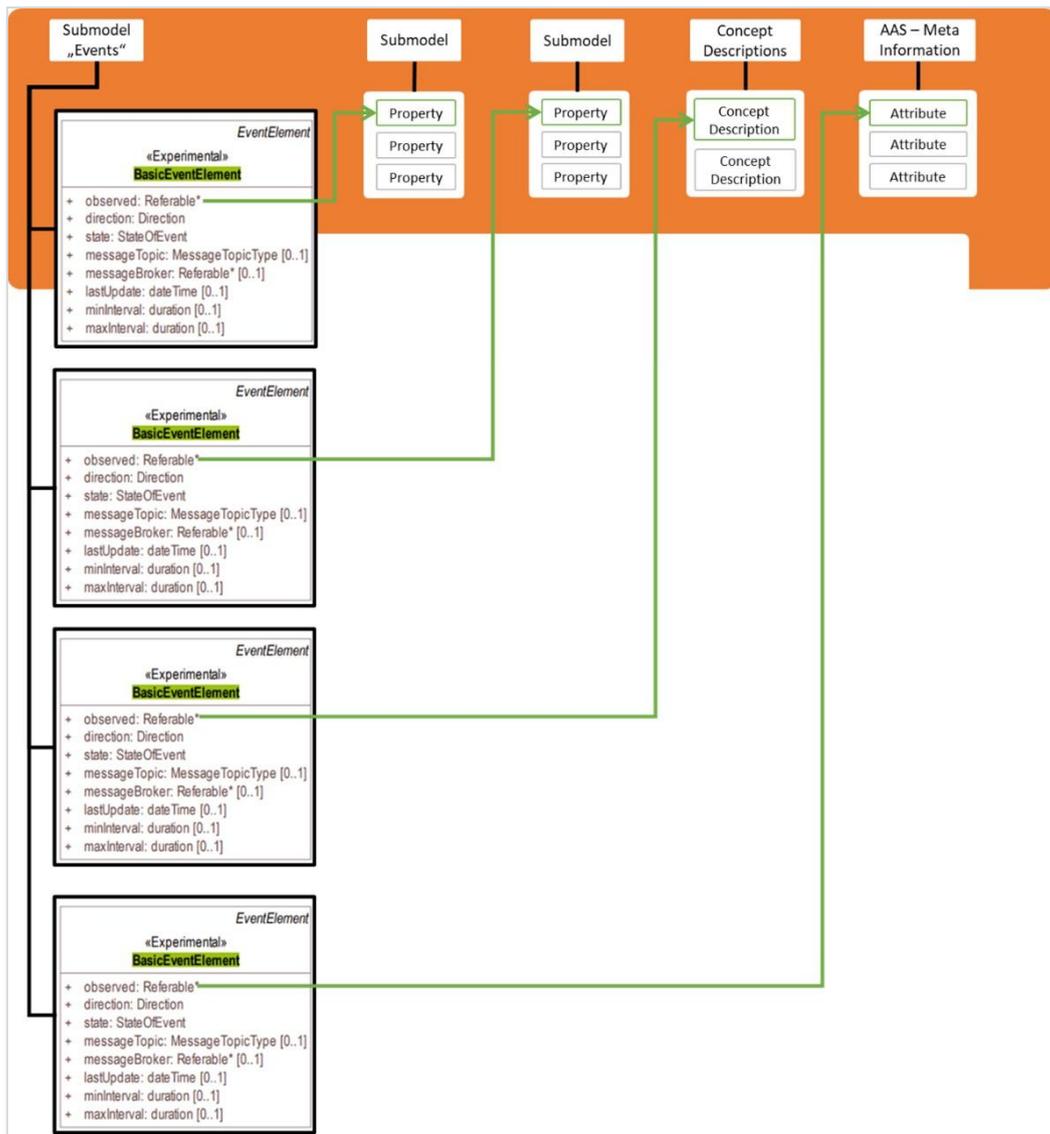


Abbildung 5-21: Abonnieren durch Sammlung von EventElements in spezifischem Teilmodell

Die Vor- und Nachteile beider Optionen sind in Tabelle 5-2 zusammengefasst.

Vorteile	Nachteile
Durch die Nutzung von Sicherheitsmechanismen wie RBAC oder ABAC sind nur die Event-Elemente zum Abonnieren sichtbar, die für den Benutzer der VWS freigegeben sind.	Es können nur die Elemente als Events abonniert werden, die in einer VWS modelliert sind. Dadurch kann #AF 2, die Benachrichtigung z. B. über eine neu hinzugefügte VWS, nicht realisiert werden.
	Schlecht skalierbar mit steigender Anzahl an Verwaltungsschalen

Tabelle 5-2: Vor- und Nachteile der Integration des Event-Elements als Teilmodellelement

### 5.3.3.2 Systemarchitektur

Nachdem das Event-Element im vorhergehenden Abschnitt als Teilmodellelement betrachtet wurde, sieht das folgende Konzept vor, das Event-Element wie eine VWS oder ein Teilmodell gemäß Abbildung 5-22 zu spezifizieren.

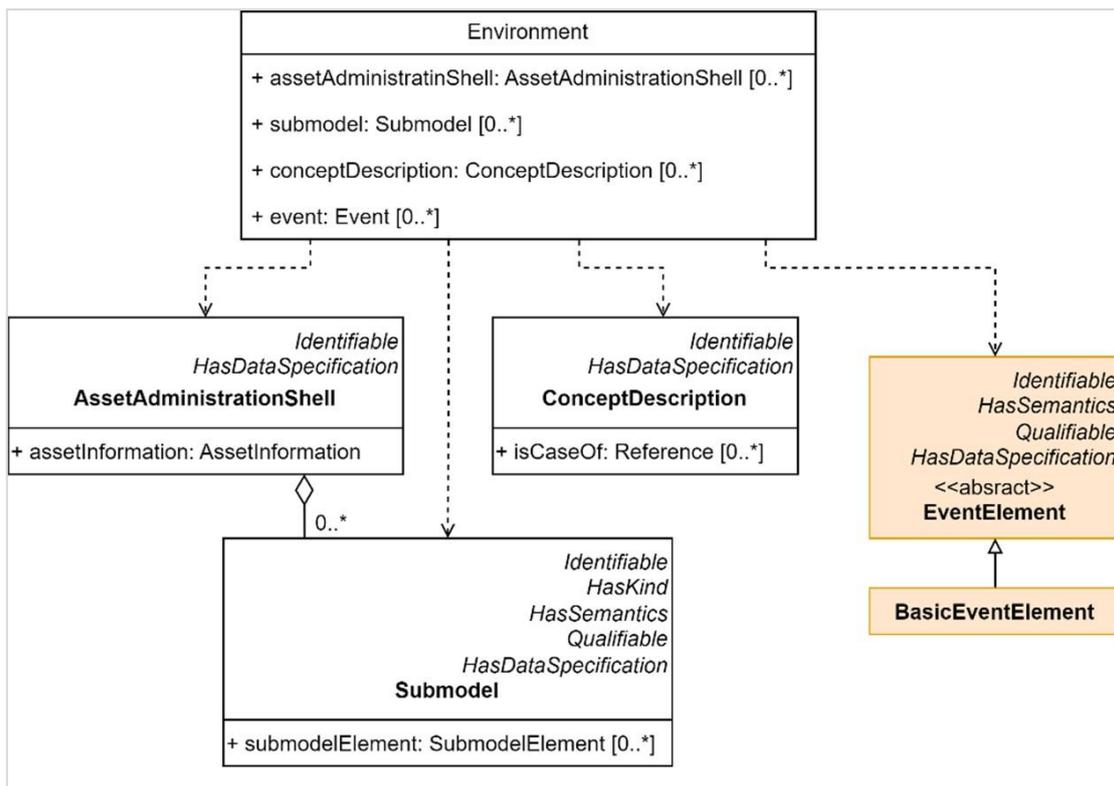


Abbildung 5-22: Neue Spezifizierung des Event-Elements in der VWS-Umgebung

Dieses Konzept bedingt die Definition neuer Infrastrukturkomponenten (Repository und Registry), wie sie es auch bereits für VWS, Teilmodelle und Concept Descriptions gibt. Das bedeutet, dass die aktuelle Spezifikation des VWS-Metamodells [1] geändert und für die Event-Registry, die die Endpunkte zu den Event-Elementen beinhaltet, ein Descriptor mit Metainformationen spezifiziert werden muss. Die Events selbst werden in einem bzw. mehreren separaten Repositories gehostet. Über ein Eventing Feature, das das Repository nutzt, können die Nachrichten des Events, der sogenannte Event Payload, zwischen Teilnehmern versendet werden. Das allgemeine Konzept ist in Abbildung 5-23 zu sehen. Die genaue Spezifizierung erfolgt in Kapitel 5.5. Notwendige Anpassungen am definierten Event-Element (#AF 17) und die Definition der Schnittstellen der Infrastrukturkomponenten (#AF 18) sind weitere Anforderungen.

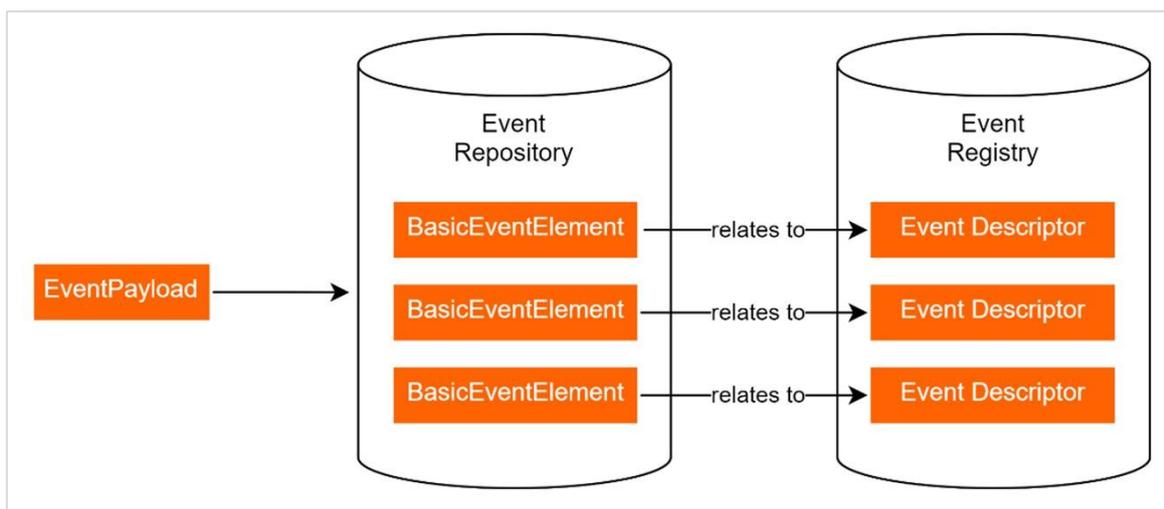


Abbildung 5-23: Infrastrukturkomponenten für das Event

Die Vor- und Nachteile des Konzepts sind in Tabelle 5-3 aufgeführt.

Vorteile	Nachteile	Vorteile	Nachteile
Vereinfachtes Logging von Events in einem separat dafür erschaffenen Eventing Feature.	Zunehmende Komplexität des Environments und der Interaktionen zwischen den Komponenten.	Vereinfachtes Logging von Events in einem separat dafür erschaffenen Eventing Feature.	Zunehmende Komplexität des Environments und der Interaktionen zwischen den Komponenten.
Anwendung des Discovery-Service für vorhandene Events.	Die Spezifizierung neuer Schnittstellen wird notwendig.	Anwendung des Discovery-Service für vorhandene Events.	Die Spezifizierung neuer Schnittstellen wird notwendig.
Zentrale Verwaltung aller Events in einer geschlossenen Umgebung.		Zentrale Verwaltung aller Events in einer geschlossenen Umgebung.	

Tabelle 5-3: Vor- und Nachteile der Integration des Event-Elements als eigenständiges Element

### 5.3.4 Übertragungstechnologie

Für die Übertragung der Events wird sich im Rahmen dieses Projekts auf die Nutzung von MQTT und HTTP/REST fokussiert. Aus dem Aufbau des Event-Elements aus Abschnitt 5.3.2 ist durch die Benennung der Attribute bereits ersichtlich, dass MQTT als ein Protokoll in Betracht zu ziehen ist.

Soll HTTP/REST für die Übertragung genutzt werden, müssen keine Änderungen an der Spezifikation des Event-Elements vorgenommen werden. Die Attributnamen sind in diesem Fall so zu verstehen, dass das Attribut *messageBroker* die http-Adresse der Applikation enthält, die die Events entgegennimmt und weiterleitet. Das *messageTopic* ist immer noch die eindeutige Bezeichnung für das Event selbst.

### 5.3.5 Anforderungsliste

In Tabelle 5-4 sind aus Sicht des Demonstrators die Anforderungen an das Event-Element selbst, die Systemarchitektur und die Benutzerschnittstelle aufgelistet, die sich aus den vier zuvor betrachteten Aspekten ergeben haben.

AF. Nr.	Kategorie	Anforderung
1	Event	Definition von Events ohne Kenntnis von dem referenzierten Element.
2	Event	Definition von Events abhängig von einem vorher bekannten VWS-Element.
3	Systemarchitektur	Events, die direkt einem Element zugeordnet sind, müssen nach dem Löschen des Elements ebenfalls gelöscht werden.
4	Benutzerschnittstelle	In der Nutzeroberfläche muss das Abonnieren von Concept Descriptions und Teilmodellen mit ihren unterlagerten Elementen durch eine Anwahl der gewünschten Elemente möglich sein. Grund dafür ist, dass die eben genannten Elemente unabhängig von der VWS in eigenen Repositories gespeichert werden. Zusammenhang zu <a href="#">#AF 14</a> .
5	Systemarchitektur	Werden Events zu Elementen abonniert, die noch unterlagerte Elemente beinhalten können (z.B. abonnieren eines Teilmodells mit seinen Inhalten), dann müssen automatisch die Events der unterlagerten Elemente abonniert werden.
6	Benutzerschnittstelle	In der Nutzeroberfläche muss die Anwahl von Metainformationen für ein Abo möglich sein.

7	Systemarchitektur	Metainformationen werden nur im gesamten aktualisiert.
8	Systemarchitektur	Das Nutzertool muss die Verlinkung von Teilmodellen und Concept Descriptions zu VWS kennen, um Events korrekt an die Nutzeroberfläche weiterleiten zu können.
9	Systemarchitektur	Concept Descriptions werden nur im gesamten aktualisiert.
10	Event	Der Inhalt einer Änderung eines Elements muss als Nachrichteninhalte im Event übersendet werden. Dies erfolgt in dem spezifizierten JSON-Datenmodell des VWS-Metamodells entsprechend dem abonnierten Element.
11	Event	Der Umfang von Events soll auch die Concept Descriptions enthalten.
12	Event	Die Eventtypen <i>create</i> , <i>update</i> und <i>delete</i> benötigen für das jeweilige VWS-Element eine eindeutige semantische ID.
13	Event	Event muss für MQTT und HTTP/REST anwendbar sein.
14	Benutzerschnittstelle	In der Nutzeroberfläche soll nach bestimmten verfügbaren Eventtypen gefiltert werden können. Zusammenhang mit <a href="#">#AF 12</a> .
15	Systemarchitektur	In der Nutzeroberfläche muss angewählt werden können, welche Elemente (VWS, Teilmodell, Teilmodellelement) von dem Event-Typ <i>create</i> immer automatisch abonniert werden sollen. Zusammenhang zu <a href="#">#AF 16</a> .
16	Systemarchitektur	Events vom Typ <i>create</i> müssen nach Anwahl in der Nutzeroberfläche automatisch vom Empfänger abonniert werden.
17	Systemarchitektur	VWS4J muss um Event-Element erweitert werden, wenn eine Anpassung des bereits spezifizierten Event-Elements erfolgt.
18	Systemarchitektur	Die REST-API muss um Operationen zum Auslösen des Events erweitert werden.

Tabelle 5-4: Anforderungsliste

## 5.4 Konzept

In den folgenden Abschnitten werden die erarbeiteten Lösungen für die zuvor genannten Konzeptideen und deren Anforderungen vorgestellt. Der Fokus wird daraufgelegt, das Event-Element als eigenständiges Element im Metamodell der VWS zu betrachten (s. Abschnitt 5.3.3.1.2). Das hat zur Folge, dass die Spezifikation des Event-Elements angepasst und entsprechende Infrastrukturkomponenten für die Verwaltung und Bereitstellung der Event-Elemente definiert werden müssen.

Der erste Abschnitt beschäftigt sich mit der Systemarchitektur für das Event-Element. Daran anschließend erfolgt die Definition des neu hinzugekommenen Event Descriptors sowie die Beschreibung über notwendige Anpassungen des *BasicEventElements*. Danach werden in Abschnitt 5.5 die semantischen IDs für die verwendeten Eventtypen definiert. Mit diesen grundlegenden Voraussetzungen wird die Anwendung, d.h. die Beschreibung des Events mit seinem Descriptor und dem *EventPayload* auf einzelne VWS-Elemente gezeigt. In Abschnitt 5.5.3 erfolgt die Spezifikation der Schnittstellen für die neu definierten Infrastrukturkomponenten.

### 5.4.1 Systemarchitektur

In Abbildung 5-24 ist die technologieneutrale Systemarchitektur für das Event als eigenständiges Element des VWS-Metamodells zu sehen. Zentrales Element bildet das Eventing-Feature, das die Verteilung der *EventPayloads* ermöglicht, während das Event Repository und die Event Registry die Event-Elemente und -Descriptor speichern. Das Eventing-Feature selbst besteht aus einem Orchestration-Layer, das die Kommunikation zwischen den Subkomponenten orchestriert. Dazu zählt die Konfiguration des Event Payload Transformers und des Event Routers durch die Events im Event Repository. Der Event Payload Transformer empfängt über einen externen Observer die stattfindenden Änderungen

in einem VWS Repository. Diese Nachrichten werden noch nicht im spezifizierten *EventPayload* übertragen, sodass der Event Payload Transformer für verschiedene Observer die Umwandlung in den *EventPayload* vornimmt. Diese *EventPayloads* werden anschließend an den Event Router weitergeleitet. Der Event Router erzeugt mit seiner eigenen Konfiguration die Message Broker, die in den Event-Elementen spezifiziert sind. Mit dem Empfang des *EventPayloads*, verteilt der Event Router den Payload auf den richtigen Message Broker. Von außen kann nun ein Subscriber ein Event-Element aus dem Event-Repository lesen und mit Hilfe des darin beschriebenen Endpunktes des Message Brokers sich direkt mit dem jeweiligen Broker verbinden und die *EventPayloads* direkt empfangen. Die Schnittstellenspezifikation der Event-Registry und des Event-Repository erfolgt in Abschnitt 5.5.3.

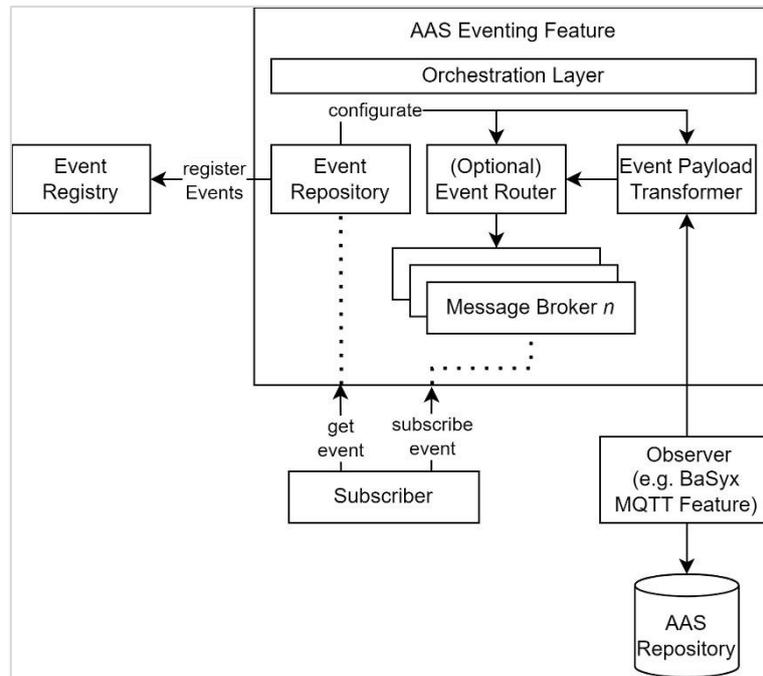


Abbildung 5-24: Systemarchitektur für das Event

## 5.5 Definition der Event-Metamodellelemente

In der allgemeinen Systemarchitektur aus Abbildung 5-22 sind die drei wichtigen Elemente zur Realisierung des Events ersichtlich. Der *EventPayload* beinhaltet die Nachricht, die über das *BasicEventElement* gesendet werden soll. Der *EventDescriptor* wiederum enthält Metainformationen und den Endpunkt des *BasicEventElements*. Alle drei Modellelemente werden im Folgenden detailliert betrachtet und wenn notwendig neu definiert.

Begonnen wird mit dem *EventDescriptor* (s. Abbildung 5-25). Er soll unter anderem semantische Informationen über die Art des Event-Elements beinhalten. Dies geschieht über die *semanticId* und *supplementalSemanticId*. (Die genaue Definition der *semanticId* erfolgt in Abschnitt 5.3.3) Des Weiteren muss der Endpunkt, administrative Informationen und Benennungen des *BasicEventElements* erfolgen. Die genannten Punkte sind bereits alle im *SubmodelDescriptor* aus [3] spezifiziert, weshalb diese Struktur für den *EventDescriptor* übernommen wurde.

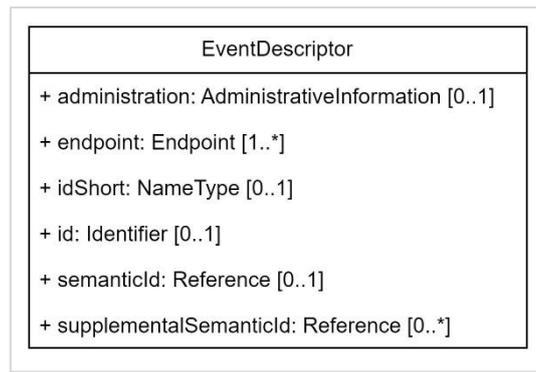


Abbildung 5-25: VWS4LS Definition EventDescriptor

In Abbildung 5-26 erfolgt die neue Definition des *BasicEventElement*. Da es in dieser Lösung nicht als Teilmodellelement betrachtet wird, sind Anpassungen notwendig. Das *EventElement* muss die gleichen Metainformationen erhalten, wie ein Teilmodell, da es nun als eigenständiges Element neben dem Teilmodell betrachtet wird. Das heißt, das *EventElement* erbt von den Klassen *Identifiable*, *HasSemantics*, *Qualifiable* und *HasDataSpecification*. Die Elemente im abgeleiteten *BasicEventElement* bleiben bis auf eine Änderung der Attribute „observed“ und „messageBroker“ von „Referable“ auf „Reference“ gleich. Der Grund für die Änderung liegt darin, dass im Falle des Eventtyps „create“ für eine neue VWS, das zu beobachtende Environment verlinkt werden muss. Dies geht nicht über das Element „Referable“.

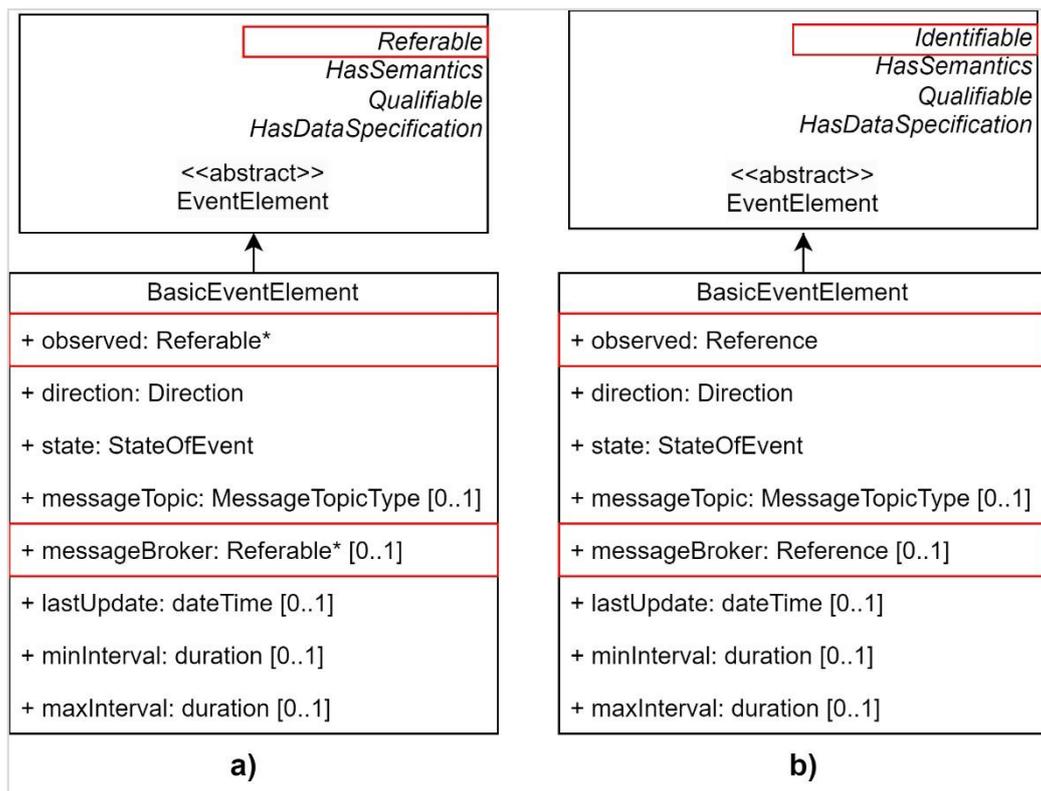


Abbildung 5-26: a) derzeitige EventElement Spezifikation und b) VWS4LS Spezifikation

Der *EventPayload* wird ebenfalls einer kleinen Veränderung unterzogen. Der Typ der „observableReference“ wird von „Referable“ zu „Reference“ geändert, siehe Abbildung 5-27.

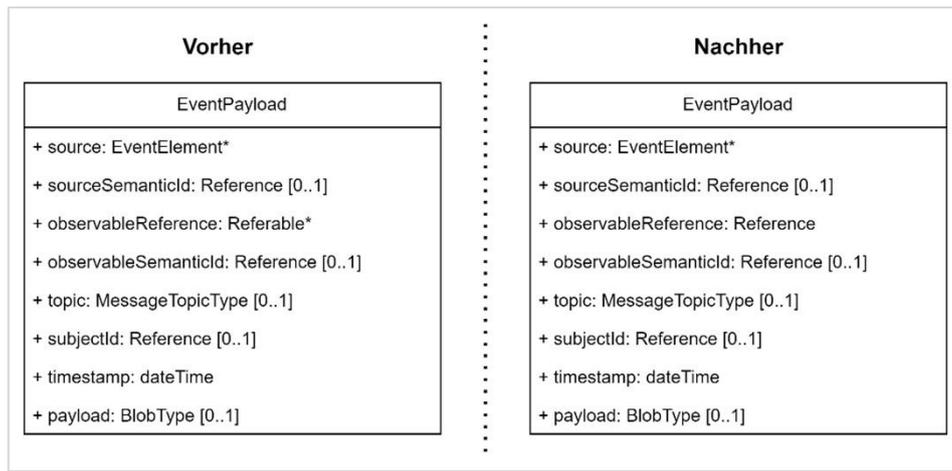


Abbildung 5-27: VWS4LS Definition EventPayload

### 5.5.1 Festlegung der Eventtypen

Für die Definition der semantischen ID von Eventtypen, wurde ein Schema entwickelt, das die Eventtypen, die in diesem Dokument spezifizierten Änderungsarten und die zu abonnierenden VWS-Elemente einfließen lässt. Zur Erläuterung des Schemas werden zuerst die Kategorien der drei genannten Informationen eingeführt.

Erste Eventtypen sind, wie bereits erwähnt, in der Spezifikation des VWS-Metamodells [1] aufgelistet. Diese Auflistung ist in Abbildung 5-28 zu sehen, wobei die für den Anwendungsfall relevanten Eventtypen rot eingrahmt sind. Im Projekt werden für die strukturellen Änderungen Structural Change (SC) verwendet und für die Updates von bestehenden Elementen das Kürzel „JEE“ (Update Existing Element). Die Definition der Kürzel für die anderen Eventtypen wird hier nicht vorgenommen.

Group	Direction	Motivation / Conditions
Structural changes of the Asset Administration Shell	Out	<ul style="list-style-type: none"> <li>• CRUD15 of Submodels, Assets, SubmodelElements, etc.</li> </ul>
	In	<ul style="list-style-type: none"> <li>• Detect updates on parent/type/derivedFrom Asset Administration Shell</li> </ul>
Updates of properties and dependent attribute	Out	<ul style="list-style-type: none"> <li>• update of values of SubmodelElements</li> <li>• time-stamped updates and time series updates</li> <li>• explicit triggering of an update event</li> </ul>
Operation element of Asset Administration Shell	Out	<ul style="list-style-type: none"> <li>• monitoring of (long-lasting) execution of OperationElement and updating events during execution</li> </ul>
Monitoring, conditional, calculated events	Out	<ul style="list-style-type: none"> <li>• e.g. when voiding some limits (e.g. stated by Qualifiers with expression semantics)</li> </ul>
Infrastructure events	Out	<ul style="list-style-type: none"> <li>• Booting, shutdown, out of memory, etc. of software entity of respective Referable (Asset Administration Shell, Submodel)</li> </ul>
Repository events	In/ Out	<ul style="list-style-type: none"> <li>• Change of semantics of IRDIs (associated concept definition)</li> </ul>
Security events	Out	<ul style="list-style-type: none"> <li>• logging events</li> <li>• access violations, unfitting roles and rights, denial of service, etc.</li> </ul>
Alarms and events	Out	<ul style="list-style-type: none"> <li>• alarms and events management analog to distributed control systems (DCS)</li> </ul>

Abbildung 5-28: Kategorisierung von Eventtypen nach [1]

Die Änderungsarten wurden bereits mehrfach erwähnt und umfassen **Create, Update und Delete**.

Die VWS-Elemente, deren Änderungen abonniert werden können, wurden ebenfalls zuvor definiert (s. Abbildung 5-8).

Aus den Kategorien der drei Informationen, ergibt sich die folgende Tabelle 5-5 an Kombinationsmöglichkeiten für die semantische ID. Es werden einheitlich englische Begriffe verwendet

Eventtyp	Änderungsart	VWS-Element
Structural Change)	Create	VWS
		Submodel
		ConceptDescription
Update Existing Element	Delete Update	VWS
		VWSMetainformation
		ConceptDescription
		Submodel
		SubmodelMetainformation
		SubmodelElementMetainformation

Tabelle 5-5: Kategorisierung für die semantische ID eines Events

Kombiniert man die Zeilen der drei Spalten nachfolgendem Schema, ergibt sich daraus die jeweilige semantische ID.

```

<Identifizier> ::= <Namespace>” /Event/” <Eventtyp>”/”<Änderungsart>”/”<VWS-Element>
<Namespace> ::= https://www.arena2036.de
<Eventtyp> ::= “SC” | “UEE”
<Änderungsart> ::= “Create” | “Update” | “Delete”
VWS-Element ::= “VWS” | “VWSMetainformation” | ... (siehe Tabelle 5)

```

Listing 5-1: Darstellung der Semantischen ID

Als ein Beispiel lautet die semantische ID des Events für den Update eines bestehenden Teilmodellelements wie folgt:

```
„https://www.arena2036.de/Event/UEE/Update/SubmodelElement“
```

Listing 5-2: Beispielhafte Darstellung der Semantischen ID

## 5.5.2 Modellierung des Event-Elements

Am Beispiel der Szenarien aus Kapitel 5.2 wird in den folgenden Abschnitten die zugehörige Umsetzung des Events beispielhaft gezeigt. Davor wird jedoch die allgemeine Anwendung der einzelnen Attribute des *BasicEventElements* und des *EventPayloads* nach den entwickelten Konzepten definiert.

Das Event-Element selbst ist, wie in Abbildung 5-29 zu sehen, erweitert worden.

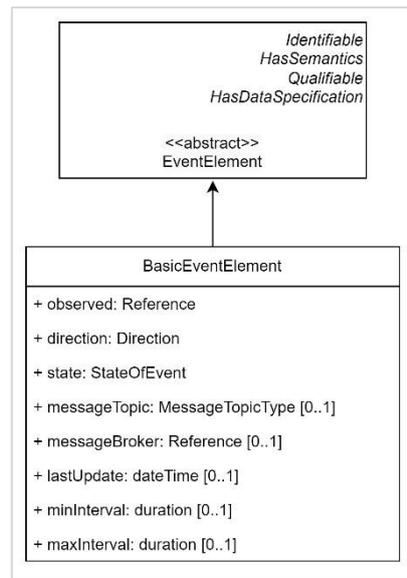


Abbildung 5-29: VWS4LS Definition EventElement und BasicEventElement

Die aufgeführten Attribute des Event-Element haben folgende Bedeutung:

- **observed:** Referenz auf ein Element des VWS-Metamodells. Sollen alle Änderungen innerhalb einer VWS beobachtet werden, wird in diesem Fall eine Referenz auf eine VWS gelegt. Sollen alle Änderungen innerhalb eines Teilmodells beobachtet werden, wird das entsprechende Teilmodell referenziert. Damit wird die Erzeugung eines EventElements für jedes zu beobachtende Element, wie z.B. einer Property, vermieden. Für die Entscheidung kann der Baum mit den Eventtypen aus Kapitel 5.3 herangezogen werden. Alles, was in diesem Baum z. B. unter dem update einer VWS definiert ist, wird in dem Event berücksichtigt. Damit kann die Detailtiefe des *EventElements* vom Nutzer bestimmt werden.
- **direction:** Definition der Richtung des Events (eingehend oder ausgehend) gemäß [2].
- **state:** Status des Events (An und Aus).
- **messageTopic:** Eindeutige Bezeichnung des Events für die Nachrichtenübertragung mit maximal 255 Zeichen.
- **messageBroker:** Referenz auf einen Endpunkt einer Applikation, die die Eventnachrichten entgegennimmt und verteilt. Hier wird vorgeschlagen, auf ein „standalone“ Teilmodell zu verweisen, das keiner VWS zugeordnet ist. Die Beschreibung des messageBrokers kann über das bereits veröffentlichte Teilmodell „Asset Interfaces Description“, im speziellen über „EndpointMetadata“ erfolgen. Dies bietet sich an, da das Teilmodell die Beschreibung der Endpunkte sowohl für http/REST als auch für MQTT anbietet.
- **lastUpdate:** Letzter Zeitstempel an dem das letzte Event gesendet bzw. empfangen wurde.
- **minInterval:** Minimale Dauer zwischen dem Senden bzw. Empfangen von Events.
- **maxInterval:** Maximale Dauer zwischen dem Senden von Events, wenn kein anderer Trigger eingegangen ist. Dies gilt nur für ausgehende Events.

Der Inhalt des Events ist durch den *EventPayload* in Abbildung 5-30 definiert und wurde nicht abgeändert. Lediglich die Nutzung der Attribute geschieht im Konzept anhand folgender Definitionen:

- **source:** Referenz auf die eindeutige ID des zugehörigen Event-Elements. Im Gegensatz zur bisherigen Spezifikation wird hier das Element „Reference“ anstatt „Referable“ genutzt. Das erlaubt auch im Fall der Benachrichtigung über neue VWS in einem Repository, das Repository zu verlinken.

- **sourceSemanticId**: Semantische ID des Event-Elements, definiert durch Kapitel 5.3.2.
- **observableReference**: Referenz auf das sich geänderte Element. Im Fall, dass im *BasicEventElement* eine ganze VWS beobachtet werden soll und sich nun eine einzelne Property geändert hat, wird hier durch den Publisher die Referenz auf die Property gelegt. Somit wird im „payload“ Attribut nicht die gesamte VWS übertragen, sondern nur das geänderte Element.
- **observableSemanticId**: Semantische ID des geänderten Elements.
- **topic**: Deckungsgleich mit dem *messageTopic* des *BasicEventElement*.
- **subjectId**: Externe Referenz auf das Subjekt, dass die Erzeugung des Events bzw. Payloads vorgenommen hat. Hier wird vorgeschlagen, den Akteur mit einer eindeutigen IRI anzugeben, der das Element geändert hat. Beispielsweise kann dies die IRI eines Unternehmens, wie in den Beispielen der Komponentenhersteller, oder einer Applikation, wie einem MES, sein.
- **timestamp**: Zeitstempel, an dem das Event ausgelöst wurde.
- **payload**: Spezifisch auf das geänderte Element angepasster Inhalt als serialisierter JSON-String. Im Falle einer strukturellen Änderung in einem Teilmodell, wird das gesamte Teilmodell übergeben. Sollte kein Payload angegeben werden, kann der Inhalt des Elements über die *observableReference* und einem Zugriff auf den Server, der das Element hostet, abgerufen werden. Das ist vor allem sinnvoll, wenn Files in einem Teilmodellelement gespeichert werden, die nicht als JSON-String übertragen werden können.

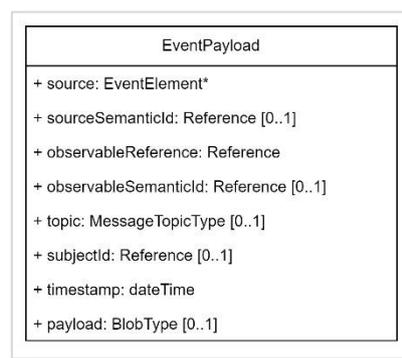


Abbildung 5-30: VWS4LS Definition EventPayload

### 5.5.2.1 Update von Teilmodellelementen

Abbildung 5-31 zeigt eine exemplarische Modellierung für das 3. Szenario aus Kapitel 5.2, dass im VWS4LS-Demonstrator umgesetzt wurde. In diesem Fall soll eine Benachrichtigung erfolgen, sobald der Wert eines bestimmten Teilmodellelements verändert wurde.

Das Attribut „observed“ des *BasicEventElements* verweist auf das Referable des abonnierten Teilmodellelements, zum Beispiel eine Property. Das Attribut „HasSemantics“ des *BasicEventElements* gibt an, über welche Art von Änderung benachrichtigt werden soll. Hierbei wird die definierte semantische ID aus Kapitel 5.3.2 verwendet und ist im Fall des Updates eines Teilmodellelements folgende:

```
„https://www.arena2036.de/Event/UEE/Update/SubmodelElement“
```

Listing 5-3: Beispielhafte Umsetzung

Als „source“ des *EventPayload* wird das *Identifiable* des zugehörigen *BasicEventElement* definiert. Das Attribut „sourceSemanticId“ entspricht dabei exakt dem Attribut „HasSemantics“ des *BasicEventElements*. Das Attribut „observableReference“ enthält in diesem Beispiel analog zum „observed“ des *BasicEventElements* das *Referable* der überwachten Property. Im Falle, dass im „observed“ des *BasicEventElements* die Updates einer VWS beobachtet werden sollen, kann im Attribut „observableReference“ die Referenz auf das Teilmodellelement erfolgen, wenn sich nur dies geändert hat. Die „observableSemanticId“ enthält die semantische ID der beobachteten Property. Das Attribut „Topic“ übernimmt denselben Wert wie das Attribut „messageTopic“ des *BasicEventElement*.

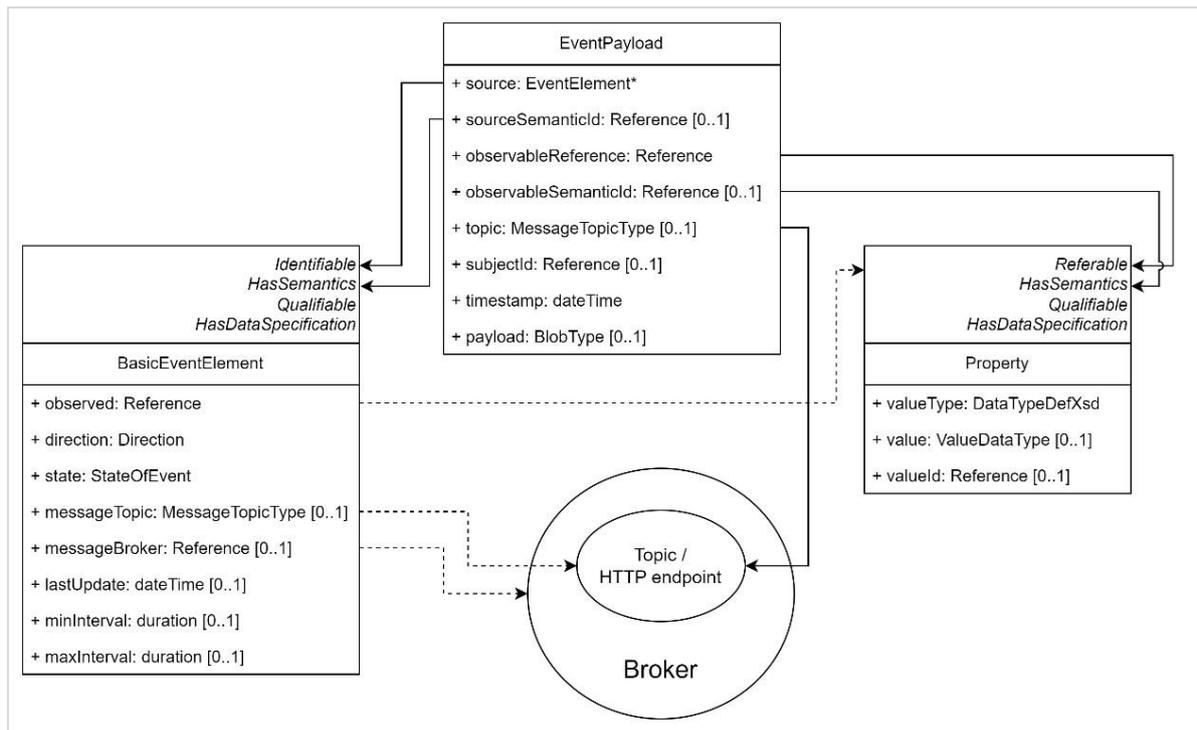


Abbildung 5-31: Modellierung update-Event für ein Teilmodellelement (Szenario 1)

### 5.5.2.2 Create von Teilmodellen

Abbildung 5-32: Modellierung create-Event für neue Teilmodelle in einer VWS (Szenario 2) zeigt die exemplarische Modellierung des 2. Szenarios aus Kapitel 5.2, und zwar dem Abonnieren neu zu einer VWS hinzugefügten Teilmodellen.

Das Attribut „observed“ des *BasicEventElement* verweist auf die aasId der überwachten VWS, die auf ein neues Teilmodell hin beobachtet werden soll. Das Attribut „HasSemantics“ hat den Wert:

„<https://www.arena2036.de/Event/SC/Create/Submodel1>“

Als „source“ des *EventPayload* wird das *Identifiable* des zugehörigen *BasicEventElements* definiert. Das Attribut „sourceSemantic“ entspricht dabei exakt dem Attribut „HasSemantics“ des *BasicEventElements*. Das Attribut „observableReference“ im *EventPayload* verweist in diesem Fall auf das neu hinzugefügte Teilmodell. Die „observableSemanticID“ des „EventPayload“ enthält die semantische ID des neuen Teilmodells. Das Attribut „topic“ übernimmt denselben Wert wie das Attribut „messageTopic“ des *BasicEventElements*. Der „payload“ des Events ist das neu hinzugefügte Teilmodell.

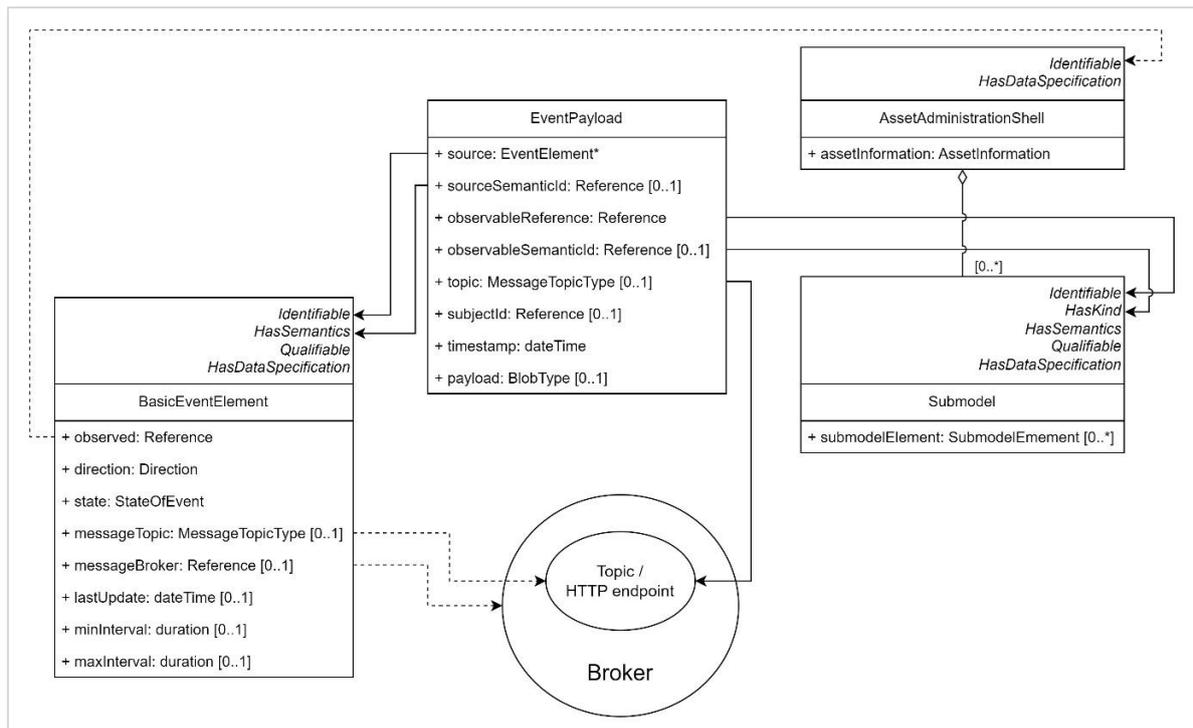


Abbildung 5-32: Modellierung create-Event für neue Teilmodelle in einer VWS (Szenario 2)

### 5.5.2.3 Create von VWS

Abbildung 5-33 zeigt eine exemplarische Modellierung eines Events für die Erstellung (create) einer VWS aus dem 3. Szenario (s. Kapitel 5.2). Das Attribut „observed“ des *BasicEventElement* verweist auf das zu beobachtende VWS-Environment. Das Attribut „HasSemantics“ gibt an, über welche Art von Änderung das jeweilige *BasicEventElement* benachrichtigen soll, in diesem Fall:

„<https://www.arena2036.de/Event/SC/Create/VWS>“

Listing 5-4: Beispielhafte Umsetzung

Als Quelle im *EventPayload* wird das *Identifiable* des *BasicEventElements* angegeben und die „source-SemanticId“ enthält den gleichen Wert wie die semanticID des *BasicEventElements*. Das Attribut „observableReference“ enthält die Referenz (*Identifiable*) auf die neue VWS. Die „observableSemanticId“ ist in diesem Fall leer, da eine VWS keine semanticID besitzt. Als „payload“ wird die neue VWS in JSON-Serialisierung übergeben. Ist kein Payload angefügt, kann über die „observableReference“ die VWS mittels eines Zugriffs auf das Environment vollständig heruntergeladen werden.

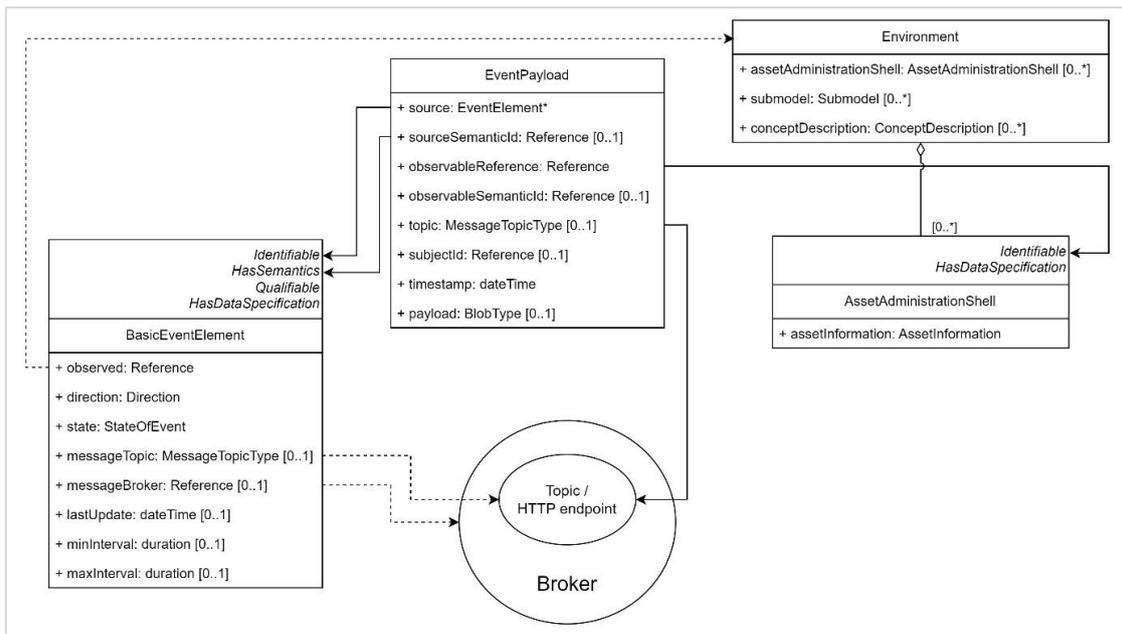


Abbildung 5-33: Modellierung *creat-Events* für neue VWS in definiertem Environment (Szenario 3)

### 5.5.3 Spezifikation der Infrastrukturkomponenten-Schnittstellen

In Abschnitt 5.4.1 wurde die technologieneutrale Systemarchitektur für das Event-Element als eigenständige Komponente vorgestellt. Zur Umsetzung der vorgestellten Anwendungsfälle erfolgt die Spezifikation der Schnittstellen von Event-Registry und -Repository in Anlehnung an die API-Spezifikation der VWS [3] in englischer Sprache. Die zur Verfügung stehenden Operationen orientieren sich dabei an den Operationen des VWS- und Teilmodell-Registries, bzw. -Repositories.

#### 5.5.3.1 Event-Registry-Interface

Die Event-Registry bietet über die in Tabelle 5-6 angegebenen Operationen einen zur VWS- oder Teilmodell-Registry vergleichbaren Funktionsumfang. Die enthaltenen Deskriptoren können in ihrer Gesamtheit oder einzeln per Event ID (siehe Definition von *EventElement* in Abschnitt 5.3.2) aufgerufen werden. Neue Deskriptoren können durch eine zusätzliche Operation angelegt werden, bereits vorhandene ersetzt oder gelöscht werden.

Operation Name	Description
<b>GetAllEventDescriptors</b>	Returns all Event Descriptors
<b>GetEventDescriptorById</b>	Returns a specific Event Descriptor
<b>PostEventDescriptor</b>	Creates a new Event Descriptor, i.e., registers an Event
<b>PutEventDescriptorById</b>	Replaces an existing Event Descriptor, i.e., replaces registration information
<b>DeleteEventDescriptorById</b>	Deletes an Event Descriptor, i.e., deregisters an Event

Tabelle 5-6: Event-Registry-Interface

Für den Zugriff per HTTP/REST ergeben sich entsprechend unterschiedliche URIs, um die Operationen umzusetzen. Diese Anfragen können durch GET, POST, PUT und DELETE umgesetzt werden und sind in Tabelle 5-7 angegeben.

Operation Name	URI
<b>GetAllEventDescriptors</b>	GET <code>http://&lt;host&gt;:&lt;port&gt;/event-descriptors</code>
<b>GetEventDescriptorById</b>	GET <code>http://&lt;host&gt;:&lt;port&gt;/event-descriptors/&lt;descriptor-id&gt;</code>
<b>PostEventDescriptor</b>	POST <code>http://&lt;host&gt;:&lt;port&gt;/event-descriptors &lt;EventDescriptor&gt;</code>
<b>PutEventDescriptorById</b>	PUT <code>http://&lt;host&gt;:&lt;port&gt;/event-descriptors/&lt;descriptor-id&gt; &lt;EventDescriptor&gt;</code>
<b>DeleteEventDescriptorById</b>	DELETE <code>http://&lt;host&gt;:&lt;port&gt;/event-descriptors/&lt;descriptor-id&gt;</code>

Tabelle 5-7: Event-Registry-HTTP-REST-URIs

Die Ein- und Ausgabeparameter der Operationen werden in Tabelle 5-8 bis Tabelle 5-12 angegeben.

semanticId https://arena2036.de/event/API/GetAllEventDescriptors				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
limit	Maximum size of result set	no	nonNegativeInteger	1
cursor	Position from which to resume the result listing	no	string	1
<b>Output Parameters</b>				
statusCode	Status Code	yes	StatusCode	1
payload	List of Event Descriptors	no	EventDescriptor	1..*

Tabelle 5-8: Parameter GetAllEventDescriptors

semanticId https://arena2036.de/event/API/GetEventDescriptorById				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
identifier	The descriptors unique id (Base64)	yes	Identifier	1
<b>Output Parameters</b>				
statusCode	Status Code	yes	StatusCode	1
payload	Requested Event Descriptor	no	Event-Descriptor	1

Tabelle 5-9: Parameter GetEventDescriptorById

semanticId https://arena2036.de/event/API/PostEventDescriptor				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
eventDescriptor	Object containing the events id and endpoint	yes	EventDescriptor	1
<b>Output Parameters</b>				
statusCode	Status Code	yes	StatusCode	1
payload	Created Event Descriptor	yes	EventDescriptor	1

Tabelle 5-10: Parameter PostEventDescriptor

semanticId https://arena2036.de/event/API/PutEventDescriptorById	
--	--

Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>eventDescriptor</b>	Object containing the events id and endpoint	yes	EventDescriptor	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1
<b>payload</b>	Replaced Event Descriptor	no	EventDescriptor	1

Tabelle 5-11: Parameter PutEventDescriptorById

semanticId <a href="https://arena2036.de/event/API/DeleteEventDescriptorById">https://arena2036.de/event/API/DeleteEventDescriptorById</a>				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>eventIdentifier</b>	The descriptors unique id (Base64 encoded)	yes	Identifier	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1

Tabelle 5-12: Parameter DeleteEventDescriptorById

### 5.5.3.2 Event-Repository-Interface

Der Funktionsumfang des Event-Repositories orientiert sich an den Operationen von VWS- und Teilmodell-Repository. Die zur Verfügung stehenden Interfaces werden in Tabelle 5-13 angegeben. Die hinterlegten Events können gruppenweise nach IdShort, bzw. SemanticId oder SupplementalSemanticId, oder einzeln per ID ausgelesen werden. Neue Events können angelegt, vorhandene ersetzt oder gelöscht werden.

Operation Name	Description
<b>GetAllEvents</b>	Returns all Events
<b>GetEventById</b>	Returns a specific Event
<b>GetAllEventsBySemanticId</b>	Returns all Events with a specific SemanticId
<b>GetAllEventsByIdShort</b>	Returns all Events with a specific idShort
<b>PostEvent</b>	Creates a new Event. The id of the new Event must be set in the payload.
<b>PutEventById</b>	Replaces an existing Event
<b>DeleteEventById</b>	Deletes an Event

Tabelle 5-13: Event-Repository-Interface

Für den Zugriff per HTTP/REST ergeben sich entsprechend unterschiedliche URIs, um die Operationen umzusetzen. Diese Anfragen können durch GET, POST, PUT und DELETE umgesetzt werden und sind in Tabelle 5-14 bis Tabelle 5-21 angegeben.

Operation Name	URI
<b>GetAllEvents</b>	GET http://<host>:<port>/event
<b>GetEventById</b>	GET http://<host>:<port>/event/<event-id>
<b>GetAllEventsBySemanticId</b>	GET http://<host>:<port>/event?semanticId=<semanticId>

<b>GetAllEventsByldShort (experimental)</b>	GET http://<host>:<port>/event?idShort=<idShort>
<b>PostEvent</b>	POST http://<host>:<port>/event <EventElement>
<b>PutEventByld</b>	PUT http://<host>:<port>/event/<event-id> <EventElement>
<b>DeleteEventByld</b>	DELETE http://<host>:<port>/event/<event-id>

Tabelle 5-14: Event-Repository-HTTP-REST-URIs

semanticId https://arena2036.de/event/API/GetAllEvents				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>serializationModifier</b>	Format of the response	yes	SerializationModifier	1
<b>limit</b>	Maximum size of result set	no	nonNegativeInteger	1
<b>cursor</b>	Position from which to resume the result listing	no	string	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1
<b>payload</b>	List of Events	no	EventElement	1..*

Tabelle 5-15: Parameter GetAllEvents

semanticId https://arena2036.de/event/API/GetEventByld				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>id</b>	The events unique id (Base64)	yes	Identifier	1
<b>serializationModifier</b>	Format of the response	yes	SerializationModifier	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1
<b>payload</b>	Requested Event Element	yes	EventElement	1

Tabelle 5-16: Parameter GetEventByld

semanticId https://arena2036.de/event/API/GetAllEventsBySemanticId				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>semanticId</b>	Id of the semantic definition (Base64 encoded)	yes	Reference	1
<b>serializationModifier</b>	Format of the response	yes	SerializationModifier	1
<b>limit</b>	Maximum size of result set	no	nonNegativeInteger	1
<b>cursor</b>	Position from which to resume the result listing	no	string	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1
<b>payload</b>	Requested Event Elements	no	EventElement	1..*

Tabelle 5-17: Parameter GetAllEventsBySemanticId

semanticId https://arena2036.de/event/API/GetAllEventsByldShort				
---	--	--	--	--

Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>idShort</b>	IdShort of the event (Base64)	yes	NameType	1
<b>serializationModifier</b>	Format of the response	yes	SerializationModifier	1
<b>limit</b>	Maximum size of result set	no	nonNegativeInteger	1
<b>cursor</b>	Position from which to resume the result listing	no	string	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1
<b>payload</b>	Requested Event Elements	no	EventElement	1..*

Tabelle 5-18: Parameter GetAllEventsByIdShort (experimental)

semanticId <a href="https://arena2036.de/event/API/PostEvent">https://arena2036.de/event/API/PostEvent</a>				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>event</b>	Event Element Object	yes	EventElement	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1
<b>payload</b>	The created Event Element	yes	EventElement	1

Tabelle 5-19: Parameter PostEvent

semanticId <a href="https://arena2036.de/event/API/PutEventById">https://arena2036.de/event/API/PutEventById</a>				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>event</b>	Event Element Object	yes	EventElement	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1
<b>payload</b>	The replaced Event Element	no	EventElement	1

Tabelle 5-20: Parameter PutEventById

semanticId <a href="https://arena2036.de/event/API/DeleteEventById">https://arena2036.de/event/API/DeleteEventById</a>				
Name	Description	Mand.	Type	Cardinality
<b>Input Parameters</b>				
<b>id</b>	The events unique id (Base64)	yes	Identifier	1
<b>Output Parameters</b>				
<b>statusCode</b>	Status Code	yes	StatusCode	1

Tabelle 5-21: Parameter DeleteEventById

## 5.6 BaSyx UI – Änderungsmanagement

Ein weiteres Eclipse BaSyx™ Web UI Modul, welches im Rahmen von TP14 entstanden ist, dient der Abbildung des Änderungsmanagements. Ziel ist die Benachrichtigung von Nutzern über Änderungen in Verwaltungsschalen. Das können beispielsweise Auftragsverwaltungsschalen oder Produktverwaltungsschalen sein. Produkt meint dabei den Leitungssatz. Im Anwendungsfall wird beispielhaft die Änderung des Produktionsstatus betrachtet.

Anforderung an das Modul ist die Möglichkeit einzelne Datenpunkte und zusammenhängende Datenobjekte (Teilmodelle) zu Abonnieren. Abonnierte Elemente werden dann auf Änderungen überwacht. Erfolgt eine Änderung in einem abonnierten Element, wird dies dem Abonnenten über das Änderungsmanagement-Modul mitgeteilt.

Das Modul selbst ist in zwei Hauptfunktionalitäten unterteilt. Es kann zum einen über den „Abonnieren“ Reiter auf die Oberfläche gewechselt werden, um neue Elemente (*Properties*, *SubmodelElementCollections*, ...) zu abonnieren. Zum anderen kann per Click auf den Reiter „Änderungen“ eine Liste aller Änderungsbenachrichtigungen angezeigt werden. Hier werden chronologisch alle Nachrichten gezeigt, die nicht bereits als gelesen bestätigt wurden.

Zunächst wird der Prozess des Abonnierens von Elementen detailliert (siehe Abbildung 5-34). Das Modul zeigt zunächst eine Liste aller registrierten Verwaltungsschalen des Systems (linke Spalte). Der Anwender kann aus der Liste eine Verwaltungsschale nach Belieben auswählen. Erfolgt eine Auswahl, zeigt das Modul dem Nutzer alle enthaltenen Teilmodelle (mittlere Spalte). Aus den Teilmodellen können *Properties* aber auch ganze *SubmodelElementCollections* (verschachtelte *Properties*) angehakt werden (rechte Spalte).

Nach dem Anhaken aller für den Anwender relevanten Elemente kann dieser über die Schaltfläche „Abonnements aktualisieren“ eine Anfrage der Abonnements an das Event-Feature versenden. Das Event-Feature speichert die angefragten Abonnements in einem Event-Repository, welches ein HTTP/REST Server ist und der ähnlich wie ein Verwaltungsschalenserver Event Elemente als Datenobjekte speichert. In diesem Fall werden Abonnements persistiert. Die Anfrage erfolgt an den /subscribe-Endpunkt des Event-Features. Als Anfrageinhalt werden „Subscription-Objekte“ mitgegeben. Diese enthalten jeweils vier wesentliche Eigenschaften:

- (1) Eine „*EventID*“ als eindeutige Identifizierung eines Event-Elements,
- (2) Eine „*ElementID*“ für die Zuordnung zu einem Datenpunkt in einer Verwaltungsschale,
- (3) Eine „*SemanticID*“, welche angibt, um welchen Typ von Element es sich handelt und
- (4) Eine „*SubmodellID*“, für das Teilmodell mit dem zu beobachtenden Element.

Bei erfolgreicher Erstellung eines Abonnements von Elementen einer Verwaltungsschale, werden die angelegten „Subscription-Objekte“ zurückgegeben. Somit hat der Anwender Einblick, welche Elemente er bereits abonniert hat. Diese Information kann jederzeit ebenfalls eingesehen werden, wenn auf den „Abonnieren“-Reiter auf der Hauptseite des Moduls navigiert wird. Dort wird zu Beginn eine Anfrage an das Event-Feature gesendet, welche Elemente abonniert wurden. Ein Löschen von Abonnements ist ebenfalls möglich durch das Entfernen des Hakens neben dem betreffenden Element der ausgewählten Verwaltungsschale.

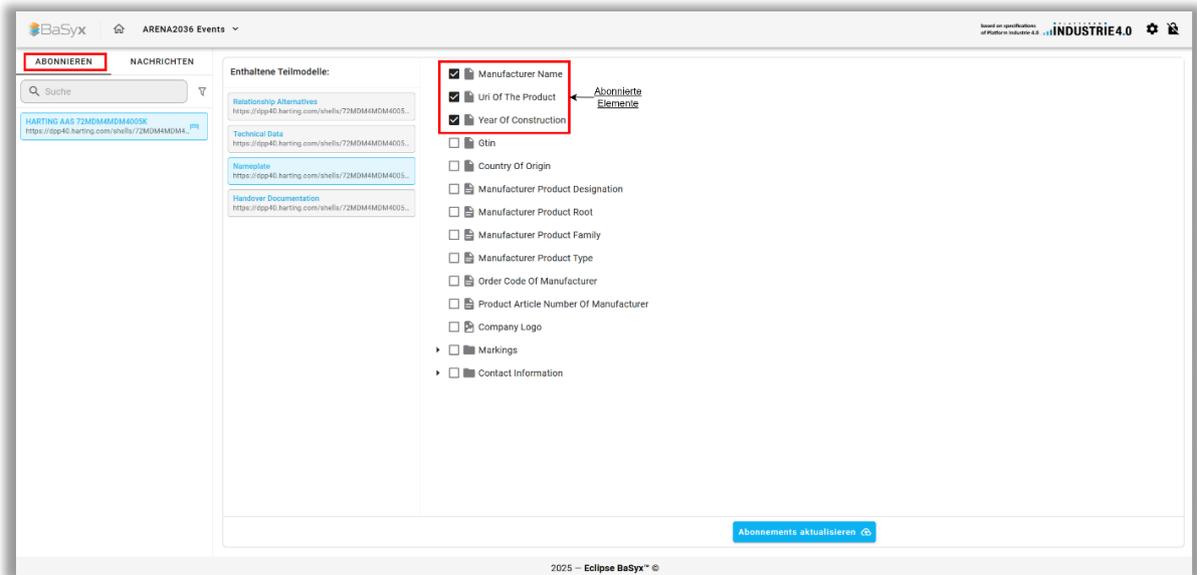


Abbildung 5-34: BaSyx UI - Änderungsmanagement abonnieren

Im Weiteren werden die Änderungsbenachrichtigungen betrachtet (siehe Abbildung 5-35). Auch hier wird zunächst eine Anfrage an das Event-Feature gesendet, auf die es mit einer Liste aller ungelesenen Event Payloads antwortet. Diese werden grafisch im Modul unterteilt nach ungelesenen, also neuen Event Payloads bzw. Nachrichten und bereits gelesenen Nachrichten (vgl. Abbildung 3-3). Nachrichten können durch den Nutzer angewählt werden. Erfolgt dies, wird im Zentrum der Oberfläche die neue Änderung dargestellt. Dies erfolgt bisher prototypisch im JSON-Format. Angewählte und damit auch angesehene Nachrichten werden automatisch als gelesen markiert. Im Hintergrund erfolgt eine Anfrage an den Server, die den Status des Event Elements auf „gelesen“ aktualisiert. Ein neu Laden der Seite führt dazu, dass bereits gelesene Nachrichten ausgeblendet werden. Als Nachrichtentitel in der linken Spalte wird bisher die ID des Event Payloads angezeigt. Dies kann jedoch abgeändert werden und Informationen aus dem verlinkten Event Element können hier eingesetzt werden. Im Event Payload ist die Referenz auf das zu beobachtende Element zu sehen. Im Beispiel aus Abbildung 5-35 ist dies die URI des Produkts, die zuvor in Abbildung 5-34 abonniert wurde. Der Payload wird als byte[] übertragen und kann entsprechend decodiert werden.

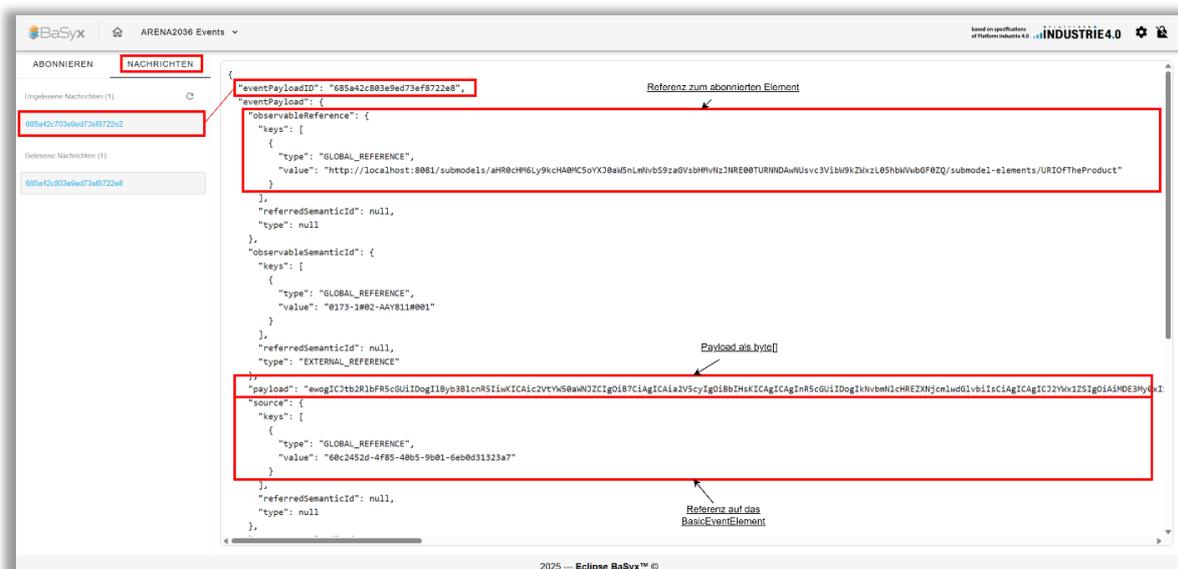


Abbildung 5-35: BaSyx UI - Änderungsmanagement Nachrichten

## 5.7 Zusammenfassung

Ein Ziel in VWS4LS war die nähere Untersuchung und Anwendung des als experimentell gekennzeichneten Event-Elements der VWS. Dafür wurden drei Anwendungsszenarien definiert, anhand derer die Nutzung des Event-Elements gezeigt werden sollten. Anhand der Szenarien wurden Anforderungen an das Event-Element selbst, der Systemarchitektur, in der es integriert ist und dessen Nutzung durch eine Weboberfläche definiert. Diese Anforderungen wurden in einem Konzept umgesetzt, das das Event-Element nicht mehr als Teilmodellelement sieht, sondern als eigenständiges Element neben der VWS und den Teilmodellen. In Folge dieses Konzepts wurde die Definition neuer Infrastrukturkomponenten notwendig, die ähnliche Funktionen wie die bereits existierenden Infrastrukturkomponenten (z. B. VWS-Repository und -Registry) bereitstellen. Als Grundlage für die Implementierung wurde das Eclipse BaSyx Framework genutzt und um eine Event-Registry und ein Event-Repository erweitert. Der Code hierfür ist im [VWS4LS GitHub](#)<sup>59</sup> zu finden. Für die Implementierung des *EventPayload* Transfers wurde ein Eventing Feature definiert, das die Sammlung und Verteilung gemäß der neu spezifizierten Event-Elemente der VWS ermöglicht. Aufgrund der kurzen Zeitspanne zur Untersuchung dieses Themas, wurde nur ein Anwendungsszenario, das Abonnieren eines Updates von Teilmodellelementen, prototypisch umgesetzt.

Ergebnisse dieses Projekts sind zum einen in das Whitepaper des Projekts DAVID mit eingeflossen. Zum anderen wurde vor Abschluss des Projekts ein Beitrag [„Towards an AAS Event Mechanism: A proposal to extend the AAS specification“](#)<sup>60</sup> bei der „30th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)“ eingereicht. Die Annahme oder Ablehnung war zum Zeitpunkt der Beendigung des Projekts noch nicht bekannt.

---

<sup>59</sup> <https://github.com/VWS4LS/vws4ls-event-infrastructure>

<sup>60</sup> <https://github.com/VWS4LS/vws4ls-subproject-results/upload/main/TP14>

## 6 Ausblick

Die Ergebnisse von VWS4LS betten sich in ein Umfeld von Forschungs-, Standardisierungs- und Umsetzungsaktivitäten ein. Im nachfolgend dargestellten Strategiehaus lassen sich die inhaltlichen und organisatorischen Bezüge darstellen, siehe Abbildung 6-1. Insbesondere wird erkennbar, wo noch Bedarfe (Baustellen) zur weiteren Forschungs- und Entwicklungsarbeit bestehen.

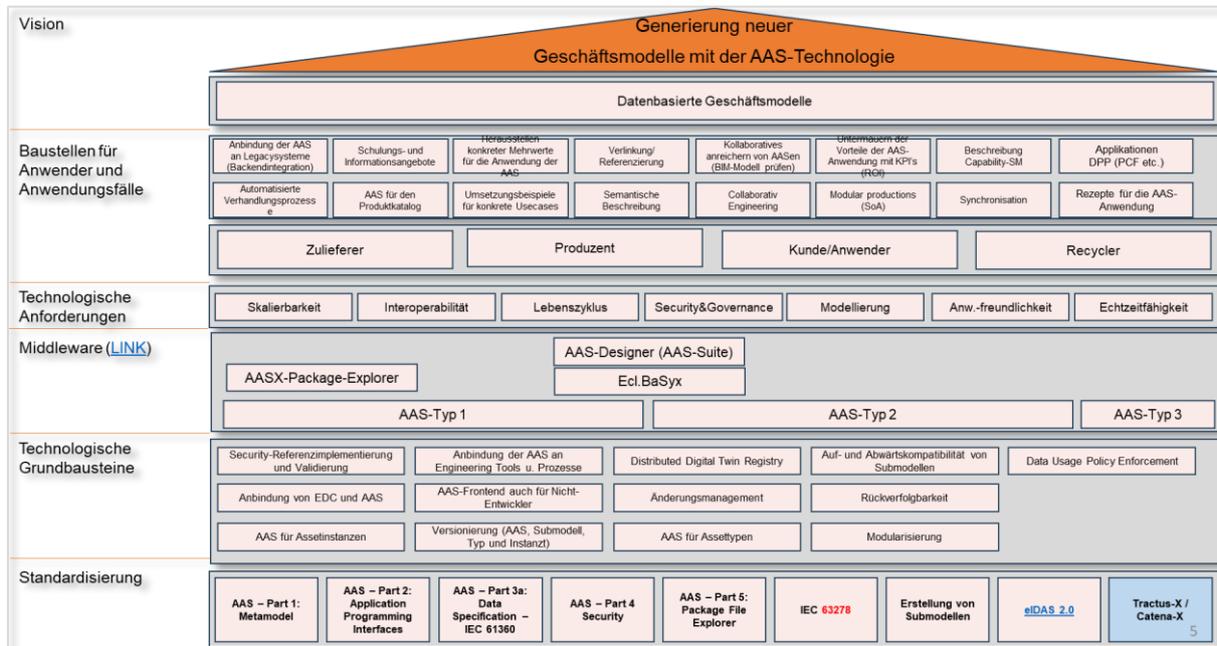


Abbildung 6-1: ARENA2036 AAS-Strategiehaus

### 6.1 Offene Themenfelder und Ansätze zu weiterführenden Arbeiten

Im weiteren Verlauf des Kapitels, wird auf einige Bedarfe, die in Abbildung 6-1 dargestellt sind, detaillierter eingegangen. Wichtig zu erwähnen ist, dass diese Bedarfe aus dem Projektkontext heraus definiert wurden, und somit bestimmte Teilbereiche abdecken und keinen Anspruch auf Vollständigkeit besitzen.

#### 6.1.1 DataSpecification – Templates

Für *ConceptDescriptions* ist derzeit nur das Template „DataSpecificationIec61360“ definiert und es existiert eine Guideline, wie ECLASS-Referenzen darauf gemappt werden können [16]. Der Modellierungsansatz bspw. von vielen VEC-Strukturen wird durch dieses IEC61360-Template allerdings nicht optimal unterstützt. Die Definition und Standardisierung weiterer DataSpecification-Templates und zugehöriger Guidelines für die bessere Unterstützung anderer Modellierungsstandards wie VEC wäre daher wünschenswert. Ebenso die Bereitstellung von standardisierten *Concept Descriptions* als Bibliotheken in global verfügbaren Repositories.

#### 6.1.2 Events

„Events“ als wichtige Datenraum-Technologie sind auf Basis der VWS noch nicht industrietauglich standardisiert. Es existieren vielmehr unterschiedliche Lösungsansätze, die in Forschungsprojekten untersucht und verifiziert werden. Als wesentlicher Beitrag aus dem Projekt VWS4LS wurde im TP 14 „Demonstrator“ wurde dazu ein mögliches Konzept entwickelt und prototypisch implementiert. Dieser Ansatz kann nun zur weiteren Standardisierung eingebracht werden.

#### 6.1.3 Software als Produkt

Bisher fokussierte die Modellierung von Produkten mit der VWS auf typischen Hardwareprodukten. Für die Modellierung von softwarebasierten Produkten wären noch geeignete Konzepte zu definieren.

#### 6.1.4 Services als Produkt

Bisher fokussierte die Modellierung von Produkten mit der VWS auf typischen Hardwareprodukten. Für die Modellierung von Dienstleistungsangeboten (Services) wären noch geeignete Konzepte zu definieren.

#### 6.1.5 VWS-Generierung mit KI

Die inhaltliche Ausgestaltung von Verwaltungsschalen ist meist eine aufwendige Recherchearbeit zur Ermittlung der relevanten technischen Merkmale. Der hohe Aufwand resultiert daraus, dass existierende Produkte in verschiedenen Dateien unterschiedlicher Formate beschrieben sind (z. B. .pdf, .xls, ...). Die korrekte Serialisierung in der Verwaltungsschale ist fehleranfällig, und bei einer großen Anzahl von Komponenten ist eine automatisierte Generierung unerlässlich. Erste Ansätze zur KI-gestützten Erzeugung und Prüfung von Verwaltungsschalen sind vielversprechend und könnten hier ansetzen. Ein Beispiel für eine solche Initiative ist der IOSB FA<sup>3</sup>ST CreAltor<sup>61</sup>. Auch die Anwendung herkömmlicher KI-Chatbots mit geeigneten Large Language Models (LLMs) ist denkbar.

#### 6.1.6 Produktkonfigurator

Ein relevanter Sonderfall in Abgrenzung zu seriengefertigten und lagerhaltigen Produktvarianten ist die Konfiguration, Bestellung und Fertigungsbeauftragung von variantenreichen Produkten (z.B. Leitungen mit frei wählbarer Länge, Farbe und Endbestückungen), ggf. auch in Losgröße 1. Für derartige Fälle wäre eine generische Modellierungsmethodik und Konfigurationsoberfläche sinnvoll, welche eine VWS mit den Bestell- und Produktionsdaten erzeugt.

#### 6.1.7 VWS-Qualitätssicherung und Validierung (regelbasiert und mit KI)

Erstellte Verwaltungsschalen müssen toolgestützt auf formelle und inhaltliche Korrektheit geprüft werden. Für die inhaltliche Validierung von Verwaltungsschalen könnte eine KI-Unterstützung für eine wesentliche Effizienzsteigerung und höhere Fehlerfindungswahrscheinlichkeit sorgen.

#### 6.1.8 Produktsuche mit KI

Im TP 12 wurde exemplarisch umgesetzt, wie ein herstellerübergreifender Produktkatalog aussehen kann (siehe Kapitel 3.5). Für einen Anwender wäre es eine wesentliche Verbesserung, wenn er KI-unterstützt die Produktauswahl durchführen könnte. Die klar strukturierten und klassifizierten Daten der Verwaltungsschalen, die über einheitliche Schnittstellen abrufbar sind, erlaubt es KI-Algorithmen wesentlich effektiver zu arbeiten. Für die Optimierung der Produktsuche wären geeignete LLMs sowie UI/UX-Konzepte für die Anwendung auf Produktkatalogen zu definieren.

#### 6.1.9 VWS für e-Commerce

Ein naheliegender nächster Schritt ist die Anbindung des VWS-basierten Produktkatalogs an e-Commerce-Infrastrukturen (e-Shops). Hierzu wurde ein Beitrag „[Utilization of Asset Administration Shells for e-Commerce Applications](#)“<sup>62</sup> bei der „30th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)“ eingereicht.

##### 6.1.9.1 Bereitstellung kommerzieller Daten

Die bislang nicht adressierte Bereitstellung von kommerziellen Datensätzen (d.h. Preise, Verfügbarkeiten, Lieferzeiten) zu den technischen Datensätzen der Produktmodelle über VWS-Technologie muss noch vollständig gelöst werden. Eine erster Vorschlag wurde im oben angeführten ETFA-Beitrag aufbereitet und eine erste prototypische Implementierung erfolgte im Zuge des TP12 mit dem VWS4LS-Marketplace.

##### 6.1.9.2 Abwicklung Bestell- und Lieferprozesse über VWS

Die im neuen Submodell *IDTA 02050-1-0 Purchase Order* vorgeschlagene Abwicklung von Bestell- und Lieferprozessen muss noch finalisiert und umgesetzt werden. Dahingehende Vorschläge wurden im oben angeführten ETFA-Beitrag aufbereitet.

---

<sup>61</sup> <https://www.iosb.fraunhofer.de/de/projekte-produkte/faaast-ecosystem-digitale-zwillinge-aas.html>

<sup>62</sup> <https://github.com/VWS4LS/vws4ls-subproject-results/upload/main/TP12>

### 6.1.9.3 Anbindung an e-Shop

Die bislang nicht adressierte Anbindung eines AAS-basierten Produktkatalogs an ein e-Commerce-Backend muss noch vollständig gelöst und standardisiert werden. Eine erster Vorschlag wurde im oben angeführten ETFA-Beitrag aufbereitet.

## 6.2 Business-Sicht

Die Einführung der Verwaltungsschale (Asset Administration Shell, AAS) als Standardisierungsinitiative im industriellen Umfeld adressiert zentrale Herausforderungen der digitalen Transformation. Ihr Einsatz ermöglicht eine durchgängige, digitale und interoperable Datenhaltung über den gesamten Lebenszyklus industrieller Assets hinweg. Aus Business-sicht ergeben sich daraus folgende wesentliche Vorteile:

### **Effizienzsteigerung durch Standardisierung**

Die AAS schafft eine einheitliche Datenstruktur und ein maschinenlesbares Datenformat. Dadurch werden bisherige Unzulänglichkeiten, wie die Pflege von Daten in unterschiedlichen Tabellen, Datenbanken oder inkompatiblen Systemen, überwunden. Daten werden nach klar definierten Standards aufbereitet und klassifiziert, was die Komplexität reduziert und Medienbrüche vermeidet.

### **Automatisierung und Fehlerreduktion**

Durch die Standardisierung können zahlreiche Routinetätigkeiten automatisiert werden. Prozesse wie das Suchen, Erfassen und Übertragen von Daten erfolgen künftig digital und ohne manuellen Aufwand. Das minimiert Fehlerquellen und beschleunigt Abläufe signifikant. Die Integration von KI und Automatisierungslösungen wird erleichtert.

### **Digitale Transparenz und Nachvollziehbarkeit**

Mit der AAS werden alle relevanten Daten zentral und nachvollziehbar bereitgestellt. Referenzdokumente wie CAD-Dateien oder Wartungsprotokolle sind eindeutig referenzierbar und jederzeit abrufbar. Das verbessert die Datenqualität, ermöglicht eine lückenlose Dokumentation und erleichtert Audits sowie Compliance-Prüfungen.

### **Verbesserte Datenanalyse und Innovationsförderung**

Die durchgängige Verfügbarkeit und Vergleichbarkeit von Daten erlaubt umfassende Analysen und Auswertungen. Unternehmen können ihre Prozesse gezielt überwachen, Optimierungspotenziale identifizieren und Innovationen schneller umsetzen. Die kontinuierliche Verbesserung wird so systematisch unterstützt.

### **Neue Formen der Zusammenarbeit**

Die AAS fördert die digitale Kollaboration innerhalb von Unternehmen und über Unternehmensgrenzen hinweg. Informationen können effizient, sicher und ortsunabhängig ausgetauscht werden. Das stärkt die Zusammenarbeit in Wertschöpfungsnetzwerken, erleichtert die Integration von Partnern und ermöglicht neue Geschäftsmodelle.

### **Ökonomische Vorteile**

Die Standardisierung durch die Verwaltungsschale führt zu einer signifikanten Reduktion von Prozesskosten. Zeitaufwändige, manuelle Tätigkeiten entfallen, Schnittstellen werden vereinfacht und die IT-Landschaft wird harmonisiert. Unternehmen profitieren von höherer Produktivität, geringeren Fehlerquoten und einer gesteigerten Wettbewerbsfähigkeit.

## Literaturverzeichnis

- [1] Industrial Digital Twin Association e.V., „IDTA 01004-3-0-0: Specification of the Asset Administration Shell - Part 4 - Security,“ 13 03 2025. [Online]. Available: <https://admin-shell-io.github.io/aas-specs-antora/IDTA-01004/v3.0/part-4-security.pdf>.
- [2] Wikipedia, „Digitaler Zwilling,“ 2024. [Online]. Available: [https://de.wikipedia.org/wiki/Digitaler\\_Zwilling](https://de.wikipedia.org/wiki/Digitaler_Zwilling).
- [3] Industrial Digital Twin Association e.V., „IDTA 02003-1-2 Generic Frame for Technical Data for Industrial Equipment in Manufacturing,“ August 2022. [Online]. Available: [https://github.com/admin-shell-io/submodel-templates/tree/main/published/Technical\\_Data/1/2](https://github.com/admin-shell-io/submodel-templates/tree/main/published/Technical_Data/1/2).
- [4] Industrial Digital Twin Association e.V., „IDTA 02004-1-2 Handover Documentation,“ March 2023. [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Handover%20Documentation/1/2>.
- [5] Industrial Digital Twin Association e.V., „IDTA 02006-2-0 Digital Nameplate for Industrial Equipment,“ October 2022. [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Digital%20nameplate/2/0>.
- [6] Industrial Digital Twin Association e.V., „IDTA 02005-1-0 Provision of Simulation Models,“ December 2022. [Online]. Available: [https://github.com/admin-shell-io/submodel-templates/tree/main/published/Provision of Simulation Models/1/0](https://github.com/admin-shell-io/submodel-templates/tree/main/published/Provision%20of%20Simulation%20Models/1/0).
- [7] Industrial Digital Twin Association e.V., „IDTA 02026-1-0 Provision of 3D Models,“ June 2024. [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Provision%20of%203D%20Models/1/0>.
- [8] Industrial Digital Twin Association e.V., „IDTA 02013-1-0 Reliability,“ 2022.
- [9] Industrial Digital Twin Association e.V., „IDTA 02023-1-0 Carbon Footprint,“ 2023.
- [10] Industrial Digital Twin Association e.V., „IDTA 02002-1-0 Submodel for Contact Information,“ May 2022. [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Contact%20Information/1>.
- [11] ECLASS e.V., „ECLASS-Standard,“ [Online]. Available: <https://eclass.eu/eclass-standard/content-suche/search>.
- [12] IEC, „IEC 61360-4 - IEC/SC 3D - Common Data Dictionary,“ [Online]. Available: <https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TreeFrameset?OpenFrameSet>.
- [13] Industrial Digital Twin Association e.V., „IDTA 01001-3-0-1: Specification of the Asset Administration Shell Part 1: Metamodel,“ June 2024. [Online]. Available: <https://admin-shell-io.github.io/aas-specs-antora/IDTA-01001/v3.0.1/index.html>.
- [14] IEC, „DIN EN IEC 61406-1 Identifizierungslink Teil 1: Allgemeine Anforderungen,“ 12 2023. [Online]. Available: <https://www.vde-verlag.de/normen/0800916/din-en-iec-61406-1-vde-0810-406-1-2023-12.html>.

- [15] IEC, „DIN EN IEC 61406-2 Identifizierungslink Teil 2: Typen/Modelle, Lose/Chargen, Artikel und Merkmale,“ 12 2024. [Online]. Available: <https://www.vde-verlag.de/normen/0800994/din-en-iec-61406-2-vde-0810-407-2024-12.html>.
- [16] Industrial Digital Twin Association e.V., GUIDELINE: How to transport ECLASS in the Asset Administration Shell, 2024.
- [17] Prostep ivip, „Vehicle Electric Container (VEC),“ prostep ivip, 8 Jan 2024. [Online]. Available: <https://ecad-wiki.prostep.org/specifications/vec/v210/>.
- [18] OPC Foundation, „OPC 40570: OPC UA for the Wire Harness Manufacturing Industry,“ <https://profiles.opcfoundation.org/workinggroup/88>, WiP. [Online]. Available: <https://profiles.opcfoundation.org/document/214>.
- [19] „DIN 72036:2024-06 Straßenfahrzeuge - Automatisierung der Leitungssatzfertigung,“ DIN-Normenausschuss Auto und Mobilität, 06 2024. [Online]. Available: <https://dx.doi.org/10.31030/3521962>.
- [20] ARENA2036 e.V., „VWS4LS Inhaltlicher Abschlussbericht,“ 2024.
- [21] Industrial Digital Twin Association e.V., „IDTA 01002-3-1: Specification of the Asset Administration Shell Part 2: Application Programming Interfaces,“ May 2025. [Online]. Available: <https://industrialdigitaltwin.io/aas-specifications/IDTA-01002/v3.1/index.html>.
- [22] Industrial Digital Twin Association e.V., „GUIDELINE: HOW TO CREATE A SUBMODEL TEMPLATE SPECIFICATION V1.1,“ 2025.
- [23] Industrial Digital Twin Association e.V., „IDTA 02008-1-1 Time Series Data,“ 2023. [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Time%20Series%20Data/1/1>.
- [24] Industrial Digital Twin Association e.V., „IDTA 02011-1-1 Hierarchical Structures enabling Bills of Material,“ 2024 June. [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Hierarchical%20Structures%20enabling%20Bills%20of%20Material/1/1>.
- [25] Industrial Digital Twin Association e.V., „IDTA 02056-1-0 Data Retention Policies,“ June 2024. [Online]. Available: [https://industrialdigitaltwin.org/wp-content/uploads/2024/06/IDTA-02056-1-0\\_Submodel\\_Data-Retention-Policies.pdf](https://industrialdigitaltwin.org/wp-content/uploads/2024/06/IDTA-02056-1-0_Submodel_Data-Retention-Policies.pdf).
- [26] Industrial Digital Twin Association e.V., „IDTA 02017-1-0 Asset Interfaces Description,“ January 2024. [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Asset%20Interfaces%20Description/1/0>.
- [27] Industrial Digital Twin Association e.V., „IDTA 01005-3-0-1: Specification of the Asset Administration Shell Part 5: Package File Format (AASX),“ 2024. [Online]. Available: [https://industrialdigitaltwin.org/content-hub/aasspecifications/idta-01005-3-0-1\\_package\\_file\\_format\\_aasx](https://industrialdigitaltwin.org/content-hub/aasspecifications/idta-01005-3-0-1_package_file_format_aasx).
- [28] Industrial Digital Twin Association e.V., „IDTA 02020-1-0 Capability Description,“ (WiP). [Online]. Available: <https://industrialdigitaltwin.org/content-hub/teilmodelle>.
- [29] Industrial Digital Twin Association e.V., „IDTA 02031-1-0 Bill of Process,“ (WiP). [Online]. Available: <https://industrialdigitaltwin.org/content-hub/teilmodelle>.

- [30] Industrial Digital Twin Association e.V., „IDTA 02048 Predictive Maintenance,“ (WiP). [Online]. Available: <https://interopera.de/wp-content/uploads/2023/07/230705-Predictive-Maintenance-Abschlusspraesentation.pdf>.
- [31] Industrial Digital Twin Association e.V., „IDTA 02051 Purchase Request Notification,“ (WiP). [Online]. Available: <https://interopera.de/wp-content/uploads/2024/02/231113-Abschlusspraesentation-InterOpera-Purchase-Teilmodelle-Liedl.pdf>.
- [32] Industrial Digital Twin Association e.V., „IDTA 02010-1-0 Service Request Notification,“ October, 2023. [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Service%20Request%20Notification/1/0>.
- [33] ARENA2036 e.V., „VWS4LS-Github,“ [Online]. Available: <https://github.com/VWS4LS>.

## Abbildungsverzeichnis

Abbildung 1-1: Matrixstruktur der Teilprojekte in VWS4LS .....	9
Abbildung 1-2: Projektteilnehmer VWS4LS (Fettgedruckt → TP-Leitung).....	10
Abbildung 1-3: Projekttreffen Projektverlängerung.....	10
Abbildung 2-1: Sequenzdiagramm für den Discovery-Workflow.....	15
Abbildung 2-2: Initialer Catena-X EDC Transfer-Prozess .....	16
Abbildung 2-3: Zweite Fassung des Catena-X EDC Transfer-Prozess .....	19
Abbildung 2-4: Finaler Catena-X EDC Transfer-Prozess.....	20
Abbildung 2-5: Architekturübersicht .....	23
Abbildung 2-6: Architekturübersicht mit zwei Identity Providern .....	24
Abbildung 2-7: Architekturübersicht mit zwei Token Variation .....	25
Abbildung 2-8: Deployment-Diagramm .....	27
Abbildung 2-9: Connector Management (links) und -Registration (rechts) .....	27
Abbildung 2-10: Connector Configuration .....	28
Abbildung 2-11: Abschluss Connector Registration .....	28
Abbildung 2-12: Keycloak Admin Console – Client ID + Client Secret .....	37
Abbildung 2-13: Authentifizierungs-Workflow.....	39
Abbildung 2-14: Unauthenticated Access .....	42
Abbildung 2-15: Internal Authentication .....	42
Abbildung 2-16: External Authentication .....	43
Abbildung 2-17: Token Exchange via Gateway .....	43
Abbildung 3-1: VWS in der Leitungssatz-Wertkette .....	44
Abbildung 3-2: Typische Online-Produktkataloge .....	45
Abbildung 3-3: Anwendungsfälle des VWS-basierten Produktkatalogs.....	46
Abbildung 3-4: IDTA 02003-1-2 TechnicalData.....	48
Abbildung 3-5: SM HandoverDocumentation .....	50
Abbildung 3-6: SM DigitalNameplate .....	52
Abbildung 3-7: Logos für Kennzeichen .....	54
Abbildung 3-8: Grammatik für URI Identifier .....	55
Abbildung 3-9: Basic URL Structure.....	56
Abbildung 3-10: Benennung von Submodell-Elementen .....	57
Abbildung 3-11: Versionierung .....	57
Abbildung 3-12: DataSpecification nach IEC61360 .....	60
Abbildung 3-13: AASPE Beispiel DataSpecification nach IEC61360.....	60
Abbildung 3-14: Beispiel zur Einheitendefinition mittels einer Concept Description .....	61
Abbildung 3-15: QUDT-Onlinedarstellung .....	61
Abbildung 3-16: Spezifikation des Vehicle Electric Container.....	62
Abbildung 3-17: VEC-Datenmodell: Description of Parts .....	62
Abbildung 3-18: Beispiel VEC Terminal Datamodel (Quelle: VEC Recommendation V2.1).....	63
Abbildung 3-19: Einheitendefinition in der Concept Description für ein VEC NumericalValue .....	64
Abbildung 3-20: Definition der Einheit in der eingebetteten Datenspezifikation der Property .....	65
Abbildung 3-21: Workshop-Schema.....	67
Abbildung 3-22: Übergabedokumentation im TSK Direct von Komax Testing .....	68
Abbildung 3-23: Produktstruktur Komax Testing.....	69
Abbildung 3-24: Definition eines Typenschildes von Komax Testing.....	71
Abbildung 3-25: Produktstruktur COROFLEX .....	72
Abbildung 3-26: Produktstruktur KOSTAL.....	74
Abbildung 3-27: Verwaltungsschalen von Komax Testing auf der Demo-Umgebung .....	77
Abbildung 3-28 : Kostal Produktverwaltungsschalen .....	78
Abbildung 3-29: Mnestix VWS-Generator .....	79
Abbildung 3-30: Ausschnitt aus Excel Datenquelle.....	79
Abbildung 3-31: Ausschnitt aus verwendeter JSON-Datei.....	80
Abbildung 3-32: Anforderungsworkshop .....	81
Abbildung 3-33: Feedbackworkshop .....	81
Abbildung 3-34: Figma Demonstrator .....	82
Abbildung 3-35: Mnestix Auswahl Produktkatalog .....	82

Abbildung 3-36: Mnestix Produktfilterung.....	83
Abbildung 3-37: Mnestix Produktanzeige.....	84
Abbildung 3-38: ARENA-X Deployment Architektur.....	85
Abbildung 3-39: Ablauf Token-Exchange.....	86
Abbildung 3-40: Ablauf Authorization Code Flow.....	87
Abbildung 3-41: Ablauf Client Credential.....	87
Abbildung 4-1: Struktur der Prüfprozesse für den OPC UA Begleitstandard.....	93
Abbildung 4-2: IRDI-Schema nach ISO 29005-5.....	98
Abbildung 4-3: ECLASS - Webbasierte Contentsuche und -anzeige.....	102
Abbildung 4-4: IEC CDD: Auswahl des Standards.....	103
Abbildung 4-5: Suche nach IEC-IRDIs.....	103
Abbildung 4-6: IEC-IRDI für eine Property.....	103
Abbildung 4-7: ProductClassId mit IEC-IRDI Referenz.....	104
Abbildung 4-8: Suche nach Temperaturinformationen im VEC-Modell.....	106
Abbildung 4-9: Datensatz zu einer Einheitsdefinition in ECLASS.....	107
Abbildung 4-10: Beispiel zur Einheiten-Referenzierung in einer Concept Description.....	108
Abbildung 4-11: Klassifizierung des Dokuments „Drawing“ im SM HandoverDocumentation.....	109
Abbildung 4-12: Suche nach Wertereferenzen in Referenzkatalogen.....	112
Abbildung 4-13: Import von Wertereferenzen aus der VEC-Ontologie.....	112
Abbildung 4-14: Suche nach Einheitenreferenzen in Referenzkatalogen.....	113
Abbildung 4-15: Import von Einheiten aus QUDT.org.....	113
Abbildung 4-16: Import von Einheiten aus SI-Units.....	114
Abbildung 4-17: Verweise auf weitere Referenzkataloge.....	114
Abbildung 4-18: Wizard HandoverDocumentation im AAS-Designer.....	115
Abbildung 4-19: ContentType.....	115
Abbildung 4-20: Ergebnis einer ID-Validierung im AAS-Designer.....	116
Abbildung 4-21: Vorauswahllisten Produkthierarchie im AAS-Designer.....	116
Abbildung 4-22: ConceptDescriptions im AAS-Designer.....	117
Abbildung 4-23: VWS Repository, Asset ID und DNS.....	118
Abbildung 4-24: Mnestix Bedienelemente.....	119
Abbildung 4-25: Mnestix API.....	120
Abbildung 4-26: Mnestix Konfiguration Überblick.....	121
Abbildung 4-27: Mnestix Konfiguration Edit Mode.....	121
Abbildung 4-28: Mnestix Konfiguration für „Data Sources“.....	122
Abbildung 4-29: Mnestix Dashboard.....	123
Abbildung 4-30: Mnestix Produktfilterung.....	123
Abbildung 4-31: Mnestix Produktanzeige.....	124
Abbildung 4-32 Hauptmenü zur Auswahl verfügbarer Module.....	126
Abbildung 4-33: Umschalten zum AAS-Editor.....	127
Abbildung 4-34: Neue Verwaltungsschale erstellen.....	127
Abbildung 4-35: Anreicherung mit Informationen im Bereich „Details“.....	128
Abbildung 4-36: Administrative Information.....	129
Abbildung 4-37: Derivation.....	129
Abbildung 4-38: Asset.....	130
Abbildung 4-39: VWS4LS Demo VWS.....	131
Abbildung 4-40: Kontextmenü zum Bearbeiten einer Verwaltungsschale.....	131
Abbildung 4-41: Neues Teilmodell erstellen.....	132
Abbildung 4-42: Submodel Editor - Details.....	133
Abbildung 4-43: Submodel Editor - Administrative Information.....	134
Abbildung 4-44: Submodel Editor - Semantic ID.....	134
Abbildung 4-45: Submodel Editor - Data Specification.....	135
Abbildung 4-46: Erstelltes Teilmodell.....	135
Abbildung 4-47: Teilmodell Kontextmenü.....	136
Abbildung 4-48: Teilmodell-Element zu einem Teilmodell hinzufügen.....	136
Abbildung 4-49: Submodel Element - Type.....	136
Abbildung 4-50: Submodel Element – Details.....	137
Abbildung 4-51: Submodel Element – Value.....	137

Abbildung 4-52: Submodel Element - Semantic ID .....	138
Abbildung 4-53: Submodel Element - Data Specification .....	138
Abbildung 4-54: Erstelltes Submodel Element .....	139
Abbildung 4-55: Submodel Element Kontextmenü.....	139
Abbildung 5-1: BaSyx-Modul VWS4LS Demonstrator - Produktauswahl .....	142
Abbildung 5-2: BaSyx-Modul VWS4LS Demonstrator - Auftragsdaten.....	142
Abbildung 5-3: BaSyx-Modul VWS4LS Demonstrator – Verhandlungsprozess .....	143
Abbildung 5-4: BaSyx-Modul VWS4LS Demonstrator - Maschinenauswahl .....	143
Abbildung 5-5: BaSyx-Modul VWS4LS Demonstrator - Asset Identifikation (Spezifische Asset ID) ..	144
Abbildung 5-6: BaSyx-Modul VWS4LS Demonstrator - Prozessdurchführung.....	144
Abbildung 5-7: BaSyx-Modul VWS4LS Demonstrator – Qualitätsprüfung der Crimpung.....	145
Abbildung 5-8: VWS-Entitäten und ihre Eventtypen.....	146
Abbildung 5-9: Anwendungsfalldiagramm Änderungsmanagement .....	147
Abbildung 5-10: allgemeines Szenario - Abonnieren eines Teilmodellelements (update).....	147
Abbildung 5-11: VWS4LS Szenario - Abonnieren eines Teilmodellelements (update) .....	148
Abbildung 5-12: allgemeines Szenario - Abonnieren eines neuen Teilmodell in einer VWS (create)	148
Abbildung 5-13: VWS4LS Szenario - Abonnieren eines neuen Teilmodell in einer VWS (create) .....	149
Abbildung 5-14: allgemeines Szenario - Abonnieren einer neuen VWS (create) .....	149
Abbildung 5-15: VWS4LS Szenario - Abonnieren einer neuen VWS (create).....	150
Abbildung 5-16: Sequenzdiagramm des allgemeinen Änderungsmanagements .....	151
Abbildung 5-17: Varianten der zu abonnierenden VWS-Elemente mit ausgewählten Eventtypen.....	152
Abbildung 5-18: Spezifikation Event-Element [1] .....	154
Abbildung 5-19: Spezifikation EventPayload [1].....	155
Abbildung 5-20: Abonnieren durch Hinzufügen des BasicEventElement zu einem Teilmodell .....	156
Abbildung 5-21: Abonnieren durch Sammlung von EventElements in spezifischem Teilmodell .....	157
Abbildung 5-22: Neue Spezifizierung des Event-Elements in der VWS-Umgebung .....	158
Abbildung 5-23: Infrastrukturkomponenten für das Event.....	158
Abbildung 5-24: Systemarchitektur für das Event .....	161
Abbildung 5-25: VWS4LS Definition EventDescriptor .....	162
Abbildung 5-26: a) derzeitige EventElement Spezifikation und b) VWS4LS Spezifikation .....	162
Abbildung 5-27: VWS4LS Definition EventPayload .....	163
Abbildung 5-28: Kategorisierung von Eventtypen nach [1] .....	163
Abbildung 5-29: VWS4LS Definition EventElement und BasicEventElement.....	165
Abbildung 5-30: VWS4LS Definition EventPayload .....	166
Abbildung 5-31: Modellierung update-Event für ein Teilmodellelement (Szenario 1) .....	167
Abbildung 5-32: Modellierung create-Event für neue Teilmodelle in einer VWS (Szenario 2) .....	168
Abbildung 5-33: Modellierung creat-Events für neue VWS in definiertem Environment (Szenario 3)	169
Abbildung 5-34: BaSyx UI - Änderungsmanagement abonnieren .....	175
Abbildung 5-35: BaSyx UI - Änderungsmanagement Nachrichten .....	175
Abbildung 6-1: ARENA2036 AAS-Strategiehaus .....	177

## Tabellenverzeichnis

Tabelle 2-1: Glossar Anbindung an Catena-X.....	13
Tabelle 3-1: Beispieldaten für SMC ProductClassification .....	49
Tabelle 3-2: Datenblatt: Inhalte für SMC DocumentClassification .....	51
Tabelle 3-3: Technische Spezifikation: Inhalte für SMC DocumentClassification.....	51
Tabelle 3-4: Technische Zeichnung: Inhalte für SMC DocumentClassification .....	51
Tabelle 3-5: Betriebsanleitung: Inhalte für SMC DocumentClassification.....	51
Tabelle 3-6: Konformitätserklärung: Inhalte für SMC DocumentClassification .....	51
Tabelle 3-7: Redundante Properties in SM DigitalNameplate und SM TechnicalData .....	52
Tabelle 3-8: Properties für Produkthierarchie.....	53
Tabelle 3-9: Automotive-Spezifikationen.....	54
Tabelle 3-10: Identifikations-Metadaten der VWS.....	55
Tabelle 3-11: Zuordnung der Produkthierarchie bei Komax Testing.....	69
Tabelle 3-12: Zuordnung der Produkthierarchie bei Coroflex .....	72
Tabelle 3-13: COROFLEX Properties .....	73
Tabelle 3-14: KOSTAL: Bildung der Produkthierarchie.....	74
Tabelle 3-15: KOSTAL TechnicalProperties .....	76
Tabelle 3-16: Subdomänen für "arena2036.app" .....	86
Tabelle 4-1: Liste von Prüfprozessen mit Erläuterungen .....	92
Tabelle 4-2: Use Cases für Prüfprozesse .....	93
Tabelle 4-3: Verbindungstest Use Cases.....	94
Tabelle 4-4: Bestückungskontrolle Use Cases.....	94
Tabelle 4-5: Höhenkontrolle Use Cases.....	94
Tabelle 4-6: Schraubmontage Use Cases .....	95
Tabelle 4-7: Breakdown of IRDI Example (0112/2///61360_4#AAA612) .....	98
Tabelle 4-8: Excerpt of Code Space Identifiers (CSI) according to ISO 290305-5 <sup>27</sup> .....	99
Tabelle 4-9: Breakdown of IRDI Example (0173-1#02-ABH173#003).....	99
Tabelle 4-10: IEC Colour properties .....	104
Tabelle 4-11: IEC Conductor properties .....	104
Tabelle 4-12: IEC Mechanical Properties.....	105
Tabelle 4-13: Referenzbeispiele für physikalische Einheiten mit ECLASS, IEC und QUDT .....	108
Tabelle 4-14: DocumentClassification according to VDI 2770 Blatt 1: 2020.....	109
Tabelle 4-15: Document classification according to IEC 61355.....	111
Tabelle 5-1: VWS-Elemente und ihre zu abonnierenden Operationen.....	153
Tabelle 5-2: Vor- und Nachteile der Integration des Event-Elements als Teilmodellelement.....	157
Tabelle 5-3: Vor- und Nachteile der Integration des Event-Elements als eigenständiges Element....	159
Tabelle 5-4: Anforderungsliste.....	160
Tabelle 5-5: Kategorisierung für die semantische ID eines Events.....	164
Tabelle 5-6: Event-Registry-Interface.....	169
Tabelle 5-7: Event-Registry-HTTP-REST-URIs .....	170
Tabelle 5-8: Parameter GetAllEventDescriptors .....	170
Tabelle 5-9: Parameter GetEventDescriptorById .....	170
Tabelle 5-10: Parameter PostEventDescriptor .....	170
Tabelle 5-11: Parameter PutEventDescriptorById .....	171
Tabelle 5-12: Parameter DeleteEventDescriptorById .....	171
Tabelle 5-13: Event-Repository-Interface.....	171
Tabelle 5-14: Event-Repository-HTTP-REST-URIs .....	172
Tabelle 5-15: Parameter GetAllEvents .....	172
Tabelle 5-16: Parameter GetEventById .....	172
Tabelle 5-17: Parameter GetAllEventsBySemanticId .....	172
Tabelle 5-18: Parameter GetAllEventsByIdShort (experimental).....	173
Tabelle 5-19: Parameter PostEvent .....	173
Tabelle 5-20: Parameter PutEventById .....	173
Tabelle 5-21: Parameter DeleteEventById.....	173

## Abkürzungsverzeichnis

American Wire Gauge .....	AWG
Application Identifier .....	AI
Application Programming Interfaces .....	API
Arbeitspaket .....	AP
Atmosphères Explosives .....	ATEX
Attribute-Based-Access-Control .....	ABAC
Bill of Material .....	BoM
Bill of Process .....	BoP
Business Partner Number .....	BPN, BPN
Communauté Européenne .....	CE
Dezentrale ID .....	DID
Digital Nameplate .....	DNP
Digital Product Passport .....	DPP
Digitale Zwillingssaggregat .....	DTA
Digitale Zwillinginstanz .....	DTI
Digitale Zwillingssprototyp .....	DTP
Distributed Digital Twin Registry .....	dDTR
Eclipse-Dataspace Connector .....	EDC
Enterprise Resource Planning .....	ERP
Extract-Transform-Load .....	ETL
Globally Unique Identifier .....	GUID
HTTP Strict Transport Security .....	HSTS
IEC Common Data Dictionary .....	IEC CDD
Internal Code .....	IC
International Code Designator .....	ICD
International Dataspaces Association .....	IDSA
International Registration Data Identifier .....	IRDI
Internationalized Resource Identifier .....	IRI
JSON-Web-Tokens .....	JWT-Token
Kabelbaumliste .....	KBL
Organization Identifier .....	OI
Product Carbon Footprint .....	PCF
Product Information Management .....	PIM
Product Lifecycle Management .....	PLM
Proof-of-Concepts .....	PoC
Role-Based-Access-Control .....	ABAC
Structural Change .....	SC
SubmodelElementCollection .....	SMC
Teilprojekten .....	TP
Top-Level-Domains .....	TLD
Uniform Resource Identifier .....	URI
Vehicle-Electric-Container .....	VEC
Verifiable Credential .....	VC
Verwaltungsschale .....	VWS

## Anhang

Im Folgenden eine Liste der Use Cases und Parameter für den OPC UA Begleitstandard "Wire Harness Manufacturing".

<b>Test System Information</b>	
Name	Test System Information
Objective	General information about software and hardware.
Description	The client connected to the server wants to... (AT) ... the identification of the connected test box. ... the test box type. ... the test box software version. ... when the test box was last calibrated. ... the result of a test box diagnoses. ... the control software version.
Reasons	1. A client wants to know which system it is connected to. 2. The test box type is a detailed identification information. 3. The test box software version is a detailed identification information. 4. A calibration date shows that the test box is working accurately and reliably. So that the client can know that only calibrated test equipment is used for production. 5. Shows that the test box is functioning correctly. Used for maintenance processes to verify that the test box is in good condition. 6. The software version of the control software is a detailed information of the identification.
<b>Status</b>	
Name	Status
Objective	Information about the current test and system status.
Description	The operator wants to know... (AT) ... which UUT is currently being tested. ... when a UUT was tested. ... if the UUT has been removed. ... if the test system is ready for testing. ... if the test system is currently running a test. ... if the test system has released a UUT. ... if a test has been aborted. ... if a test system error has occurred. ... if the control software is connected to a test box.
Reasons	To manage the test system, an MES requires information about the test process. This includes the identification of a UUT (1), when a UUT was tested (2), if a UUT was unlocked or removed (6)(3), if a test was aborted (7), and if the control software is connected to a test box (9). Furthermore, a controlling MES requires status information of the test system: Ready (4), Running (5), and Error (8).
<b>Statistics</b>	
Name	Statistics
Objective	Information about the current test and system status.
Description	The operator wants to know ... (AT) ... how many UUTs were totally tested. ... how many UUTs were tested without errors. ... how many UUTs were tested with errors. ... how many error-free UUTs had required a retest.
Reasons	An MES required the statistics of a test system for its evaluations. Of interest are the number of tested UUTs (1), the number of UUTs that passed (2), the number of UUTs that failed (4), and possibly the number of UUTs that passed but required a retest (4).
<b>Job Management</b>	
Name	Job Management

Objective	Describes the test data.
Description	The user as a client wants to... (AT) ... name a test data set. ... send a single connection description. ... send a single component description. ... send a single parameter set.
Reasons	Test data that are used to test a UUT consists of descriptive single data. An MES wants to provide the test system with a complete test data set (1) or individual sub-areas or single measurements (2)(3)(4).

<b>Control</b>	
Name	Control
Objective	Controls the testing process.
Description	The user as a client wants to... (AT) ... start a test. ... cancel a test. ... release a UUT. ... change the mode. ... connect to a test box. ... load individual test data. ... optionally load a complete test data set. ... specify which UUT should be tested. ... execute a test box diagnosis.
Reasons	If a test system is to be remotely controlled, it must be possible to start (1) and abort (2) a test. Furthermore, a UUT must be unlocked (3) and the mode must be changed (4). A test box should be connectable to the control software (5). Single test data (6) and complete test data sets (7) must be loadable. To ensure a functional test system, a remote control should also be able to check the test system (9).

<b>Results</b>	
Name	Results
Objective	Describes the generated test results.
Description	The user as a client wants to know... (AT) ... the total result of the UUT test. ... all the results generated by the test. ... the currently generated result of each single measurement at runtime. ... the name of the single result. ... in which segment the single result was generated. ... the test points of the single result. ... a description of the single result. ... a nominal value for the single result. ... the plus and minus tolerance for the single result. ... the measured value of the single result. ... the unit of the single result. ... the error code of the single result.
Reasons	After a UUT test, the total result (1) and all generated single results (2) are of interest to an MES. For runtime monitoring, the MES must be able to read each single result at runtime (3). A single result is described by a name (4), its storage location in the test data (5), the test points (6), a possible description (7), a nominal value (8), the tolerance (9), the measured value (10), the unit (11), and an error code (12).

<b>Event Control</b>	
Name	Event Control
Objective	Reports about the testing process.
Description	The client connected to the server wants to get notified about... (AT) ... the occurring messages generated by the controlling software. ... the occurring messages generated by the connected test box. ... the process status.

Reasons	
<b>Test Management</b>	
Name	Test Management
Objective	Ensure control and traceability of wire harness test processes.
Description	<p>The client connected to the server wants to... (KXT)</p> <ul style="list-style-type: none"> <li>... control which components of a wire harness are being tested so that they can adjust the test process to the specific characteristics of a wire harness that needs to be verified, or disable certain tests that aren't required for a particular harness.</li> <li>... know in which shift and by whom a test was performed so that they can calculate performance indicators for specific shifts and operator in order to take corrective actions, like training, for example when certain errors only occur in a particular shift or by a specific operator.</li> <li>... know how many times a test was repeated so that they can correctly calculate the efficiency of a system and take corrective actions when tests are repeated too many times.</li> </ul> <p>(AT)</p> <ul style="list-style-type: none"> <li>... know the batch name.</li> <li>... know the batch size.</li> <li>... know the current batch counter.</li> <li>... know which operator is logged in.</li> <li>... know the name of the used test data set.</li> <li>... know the serial number.</li> <li>... be able to add additional information to the report.</li> <li>... be able to add additional information to the label.</li> <li>... be able to send additional test information.</li> <li>... know the current test system mode.</li> </ul>
Reasons	To manage the test procedure, an MES requires information about the batch to be processed. This includes the batch name (1), the batch size (2), and the current batch counter (3). Other information of interest to the MES is the logged-in operator (4), the name of the current test data set (5), and the current serial number (6). When the test system creates a report or label, an MES often wants to transfer additional information that is to be added to the report or label (7)(8). An MES also wants to be able to provide information relevant to the test procedure (9). If an MES wants to implement full remote control, the current test system mode (10) is relevant.

<b>Tightening</b>	
Name	Tightening
Objective	Control and verify the integrity of tightening operations for safety-critical joints.
Description	<p>The client connected to the server wants to... (KXT)</p> <ul style="list-style-type: none"> <li>... perform tightening operations for safety-critical joints so that they can be confident that the joint conforms to the safety-standards required by the government or customer.</li> <li>... limit the number of times a bolt or nut can be tightened so that they can prevent damaged or overused parts from being used in safety-critical joints and ensure that customer requirements regarding the use of previously tightened bolts and nuts are met.</li> <li>... know if a tightening operation was performed with the right settings so that they can prove to their customer that the joint met the customer's requirements at the time of delivery.</li> <li>... know which tool was used to perform a tightening operation so that they can verify that only spindles that meet safety standards and are properly calibrated are used for the production.</li> <li>... know the tightening job ID of a tightening operation so that they can link the tightening curve of the operation with the tightening process performed by the station.</li> <li>... know how many times a bolt or nut was tightened so that they can determine how many times a bolt or nut can be re-tightened in the case of rework or repairs.</li> <li>... know the tightening results for every stage of a tightening operation so that they can prove to their customer that all tightening stages were successfully completed within specified conditions.</li> <li>... know at which location a bolt or nut was tightened so that they can verify that every position in an assembly was tightened in the right position, the right order, and no tightening was performed on positions that aren't supposed to be tightened.</li> <li>... control the tightening jobs of a tightening operation so that they can select specific tightening parameters for specific tightening positions or to use different settings when re-tightening a previously tightened position so that the final joint is within customer's expectations.</li> <li>... control the loosening jobs of a tightening operation so that they can select specific loosening parameters for specific tightening positions or to use different settings when loosening a previously tightened position so that the joint is not damaged.</li> </ul>
Reasons	

<b>Fusebox Assembly</b>	
Name	Fusebox Assembly
Objective	Control and verify the correct assembly and installation of components in a fusebox.
Description	<p>The client connected to the server wants to... (KXT)</p> <ul style="list-style-type: none"> <li>... assemble a fusebox so that they can be confident that the fusebox is in good condition before it is delivered to the customer.</li> <li>... control if empty positions should be checked so that they can selectively enable and disable this option depending on whether issues are observed in the field or faster cycle times are needed.</li> <li>... control the threshold at which a component is considered installed so that they can calibrate their process to optimally detect all components in a fusebox.</li> <li>... know if the right component is installed in a position so that they can verify that the fusebox is assembled according to the customer's order.</li> <li>... know which component is installed in a position so that they can know which component was assembled when a wrong component is detected and to take preventive actions to avoid the failure to happen in the future.</li> <li>... know the evaluation score of a component so that they can know how repeatable the evaluation of every position is and to adjust the threshold</li> <li>... know if a component is missing so that they can take preventive actions to avoid this failure to happen in the future.</li> <li>... know if a component is installed in the wrong position so that they can take preventive actions to avoid this failure to happen in the future.</li> <li>... know if a component is installed in the wrong orientation so that they can take preventive actions to avoid this failure to happen in the future.</li> <li>... know if a component is installed in a position that should be empty so that they can take preventive actions to avoid this failure to happen in the future.</li> <li>... know if a label is missing so that they can take preventive actions to avoid this failure to happen in the future.</li> </ul>
Reasons	

<b>Fusebox Height Control</b>	
Name	Fusebox Height Control
Objective	Control and verify the proper insertion and alignment of components in a fusebox.
Description	<p>The client connected to the server wants to... (KXT)</p> <ul style="list-style-type: none"> <li>... check the height of installed components in a fusebox so that they can be confident that the fusebox is in good condition before it is delivered to the customer.</li> <li>... control if empty positions should be checked so that they can selectively enable and disable this option depending on whether issues are observed in the field or faster cycle times are needed.</li> <li>... know if a component is not fully inserted so that they can properly insert the component and take preventive actions to avoid this failure from happening in the future.</li> <li>... know if a component is inserted unevenly so that they can properly insert the component and take preventive actions to avoid this failure from happening in the future.</li> </ul>
Reasons	

<b>Electrical Continuity</b>	
Name	Electrical Continuity
Objective	Control and verify electrical continuity, proper installation, and characteristics of wire harnesses.

Description	<p>The client connected to the server wants to... (KXT)</p> <p>... test the electrical continuity and additional characteristics of a wire harness so that they can be confident that the harness is in good condition before it is delivered to the customer.</p> <p>... control if a short-circuit test should be performed so that they can be confident that the harness is in good condition before it is delivered to the customer. In some cases the short-circuit test is not needed or not wanted.</p> <p>... control the threshold at which a wire is considered short-circuited so that they can be confident that the resistivity between wires is high enough to prevent any electrical side-effects.</p> <p>... know if a component is installed in the wire harness so that they can be confident that the harness is completely assembled according to the customer's order.</p> <p>... know if a component has a particular characteristic so that they can be confident that the correct components were assembled in a harness.</p> <p>... know if a wire is missing so that they can prevent the harness from being delivered, repair the harness, and take preventive actions for future harnesses.</p> <p>... know if a wire is installed in the wrong terminal so that they can prevent the harness from being delivered, repair the harness, and take preventive actions for future harnesses.</p> <p>... know if two wires are short-circuited so that they can prevent the harness from being delivered, repair the harness, and take preventive actions for future harnesses.</p> <p>... know if two wires are swapped so that they can prevent the harness from being delivered, repair the harness, and take preventive actions for future harnesses.</p> <p>... know the reason for a specific failure so that they can detect systematic errors in production and take preventive actions for future harnesses.</p>
Reasons	

## Autorenverzeichnis

An der Projektverlängerung vom 01.12.2024 bis zum 30.6.2025 haben mitgewirkt:

	Name	Christian Kosel
	Organisation	ARENA2036 e.V.
	Funktion	Forschungskordinator
	<p>Christian Kosel ist studierter Maschinenbauer mit dem Fokus auf der virtuellen Produktentwicklung, der Automatisierung von Entwicklungs- und Produktionsprozessen sowie vernetzten Produktentstehungsprozessen. Seit 2020 ist er in der ARENA2036 in verschiedenen Rollen in den Bereich der Leitungssatzaktivitäten der ARENA2036 tätig und koordiniert das Forschungsprojekt VWS4LS von Beginn an. Darüber hinaus hat er die Leitung von TP4 und TP9 innerhalb des Projekts übernommen.</p>	
	Name	Georg Schnauffer
	Organisation	ARENA2036 e.V.
	Funktion	Stellvertretender Geschäftsführer und Forschungskordinator Industrie 4.0
	<p>Georg Schnauffer ist seit 2019 Forschungskordinator Industrie 4.0 der ARENA2036 und seit 2021 stellvertretender Geschäftsführer. Nach dem Studium der technisch orientierten BWL an der Uni Stuttgart arbeitete er am Fraunhofer Institut für Fabrikbetrieb und Fabrikautomatisierung IFF als Mitarbeiter und später Abteilungsleiter für Wissens- und Innovationsmanagement. 2005 wechselte er in die Konzernzentrale von thyssenkrupp und übernahm dort die Verantwortung für das strategische Wissensmanagement. 2014 bis 2018 war er Mitglied des Vorstands der Gesellschaft für Wissensmanagement. 2015 übernahm er in der Zentrale der Fraunhofer-Gesellschaft die Verantwortung für das Corporate Business Development im Bereich Industrie 4.0 und wechselte 2017 in die Geschäftsstelle der Plattform Industrie 4.0 u.a. als Leiter des Transfer-Netzwerk Industrie 4.0 des BMWi und BMBF.</p>	
	Name	Bernd Kärcher
	Organisation	ARENA2036 e.V.
	Funktion	Projektoffice
	<p>Bernd Kärcher unterstützt im Projektoffice und kümmert sich um das Ergebniscontrolling.</p> <p>Davor war er über viele Jahre im Bereich Forschung/Vorentwicklung für Festo tätig.</p> <p>Er ist Mitglied in der AG1 der Plattform Industrie 4.0 seit Gründung.</p>	

	Name	Markus Rentschler
	Organisation	ARENA2036 e.V.
	Funktion	Forschungskordinator Digitale Interoperabilität
	<p>Markus Rentschler war für das Teilprojekt 12 „Produktkatalog“ zuständig und hat Teile des Teilprojektes 13 „Standards und Middleware“ erarbeitet. Sein beruflicher Werdegang kann auf <a href="#">LinkedIn</a> eingesehen werden.</p>	
	Name	Pascal Neuperger
	Organisation	Komax Testing Germany GmbH
	Funktion	Produktentwickler & Support Manager
	<p>Pascal Neuperger ist seit 2005 bei Komax Testing Germany GmbH in Porta Westfalica tätig und arbeitet dort als Produktenwickler und Support Manager.</p> <p>Im Projekt VWS4LS trug er im Wesentlichen bei zu den Teilprojekten 1, 3, 5, 12 und 13. Er leitete die Architekturgruppen Verlinkung, Rückverfolgbarkeit und das Teilprojekt 13.</p>	
	Name	Miguel Rodriguez
	Organisation	Komax TSK
	Funktion	Projekt Leiter Digital Marketplace
	<p>Leiter Teilprojekt 1</p>	
	Name	Torsten Hansen
	Organisation	KOSTAL Kontakt Systeme GmbH & Co. KG
	Funktion	Gruppenleiter der Entwicklung mit Schwerpunkt Tools & Prozesse
	<p>Herr Hansen ist mitunter verantwortlich für die Umsetzung der Digitalisierung der entwicklungsbezogenen Produktdaten im Hause KOSTAL Connectors. Hierzu gehört die Administration und Pflege des Online Kataloges.</p> <p>In der Projektverlängerung des Projektes VWS4LS hat er das Teilprojekt TP 12, Produktkatalog begleitet.</p>	

	Name	Mario Angos
	Organisation	Coroplast Group
	Funktion	Project Manager Digital Transformation
	<p>Mario Angos studierte Maschinenbau an der Universität Stuttgart mit den Schwerpunkten Produktentwicklung und Konstruktionstechnik. Derzeit arbeitet er als Projektmanager in der Abteilung Digitale Transformation bei der Coroplast Group und verfügt über mehr als zwei Jahre Erfahrung in diesem Bereich. Darüber hinaus hat Mario Angos umfangreiche Expertise in der Forschung zu digitalen Zwillingen erworben, die er im Rahmen seiner Promotion am Institut für Technologien und Management der digitalen Transformation (TMDT) an der Universität Wuppertal vertieft. Im Projekt VWS4LS leitete er das Teilprojekt 7 (Data Business Policy, Data Governance und Monetarisierungskonzepte) und wirkte zudem am Teilprojekt 8 mit.</p>	
	Name	Johannes Becker
	Organisation	4Soft GmbH
	Funktion	Managing Consultant
	<p>Johannes Becker hat Informatik an der Technischen Universität München studiert und arbeitet seit 2006 für die 4Soft GmbH. Dort leitet er den Geschäftsbereich Bordnetze und ist seit dem Jahr 2010 als Dienstleister für die Projektgruppen VES-WF / ECAD-IF des prostep ivip tätig und prägt die Standards KBL und VEC maßgeblich. Im Projekt VWS4LS war er Teammitglied des TP1 und in den Architekturgruppen Verlinkung sowie Synchronisierung brachte er querschnittlich das Know-How zu den Produktmodellstandards KBL &amp; VEC ins Projekt ein.</p>	
	Name	Rene-Pascal Fischer
	Organisation	Fraunhofer IESE
	Funktion	Scientist
	<p>Rene Fischer ist studierter Informatiker und arbeitet seit 2021 am Fraunhofer IESE an der Entwicklung und dem Transfer von Industrie 4.0 Konzepten in verschiedene Domänen. Des Weiteren promoviert er mit dem Ziel eine automatische Zertifizierung und Validierung von Digitalen Zwillingen zu ermöglichen.</p> <p>Im Rahmen des Projektes VWS4LS trug er wesentlich zur Prozesssteuerung als Teil der Verwaltungsschale, dem Architekturteam sowie der Umsetzung der entwickelten Konzepte bei.</p>	

	Name	Jannis Jung
	Organisation	Fraunhofer IESE
	Funktion	Engineer
	<p>Jannis Jung ist studierter Informatiker und arbeitet seit 2021 am Fraunhofer IESE an der Entwicklung von Industry 4.0 Konzepten und Co-Simulationen.</p> <p>Im Rahmen des Projektes VWS4LS trug er wesentlich zum Architekturteam sowie der Umsetzung der entwickelten Konzepte bei.</p>	
	Name	Melanie Stolze
	Organisation	Institut für Automation und Kommunikation e.V.
	Funktion	Wissenschaftliche Mitarbeiterin
	<p>Nach ihrem Studium in der Elektro- und Informationstechnik arbeitet Melanie Stolze seit 2022 im ifak e.V. an den Forschungsschwerpunkten der Standardisierung von Technologien der Industrie 4.0 und der Weiterentwicklung digitaler Produktionssysteme mit. Zudem promoviert sie mit dem Ziel, mit Hilfe der Verwaltungsschale den täglichen Umgang von Mitarbeitern mit komplexen Maschinen sicher und effizient zu gestalten.</p> <p>Im Rahmen von VWS4LS hat das ifak beratende Tätigkeiten im Bereich der Verwaltungsschale in TP2 und TP6 übernommen. Zudem wurde zusammen mit den Projektpartnern ein Demonstrator in TP6 entwickelt, der das automatisierte Ausschreibungsverfahren mit Hilfe proaktiver Verwaltungsschalen und Business Process Models ermöglicht.</p>	