

Projektergebnisse zum Projektabschluss

Verwaltungsschale für den Leitungssatz

Berichtszeitraum 1.12.2021 – 30.11.2024

Förderkennzeichen: 13IK005



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



Inhaltsverzeichnis

| | |
|--|----|
| Projektkonsortium | 7 |
| 1 Überblick: Stand des Gesamtprojektes | 9 |
| 1.1 VWS4LS positioniert die Verwaltungsschale in einer zentralen Automotive-Wertkette..... | 9 |
| 1.2 Projektstruktur: 10 Teilprojekte und 5 Anwendungsfälle | 10 |
| 1.3 Projektplanung: Der initiale Projektplan hat im Wesentlichen Bestand | 12 |
| 1.4 Zeitplanung und Arbeitstreffen..... | 16 |
| 1.5 Verortung im Leitbild der Plattform Industrie 4.0 | 18 |
| 2 TP1 - Konzept, Informationsmodelle und Produktbeschreibung | 20 |
| 2.1 AP 1.1 - Zielbeschreibung des Gesamtkonzepts | 20 |
| 2.1.1 Anforderungserhebung | 20 |
| 2.1.2 Klassen vom Materialen | 22 |
| 2.1.3 Prozessübersicht | 22 |
| 2.1.4 PPR-Modell..... | 23 |
| 2.1.5 Fazit | 23 |
| 2.2 AP 1.2 - Entwicklung von Informationsmodellen und Zuweisung der Daten | 24 |
| 2.2.1 VEC und KBL | 25 |
| 2.2.2 ECLASS | 28 |
| 2.2.3 RAMI 4.0..... | 29 |
| 2.2.4 OPC UA | 29 |
| 2.2.5 Fazit | 30 |
| 2.3 AP 1.3 - Erfassen der Geschäftsmodelle für das Informationsmodell | 30 |
| 2.4 AP 1.4 - Beschreibung der Tiefe der Informationsmodelle | 31 |
| 2.4.1 Grundlagen..... | 31 |
| 2.4.2 Prozessliste und Parameter..... | 33 |
| 2.4.3 Produktionsdatensatz | 33 |
| 2.4.4 VEC als Produktdatenmodell..... | 35 |
| 2.4.5 OPC UA Schnittstelle | 37 |
| 2.4.6 Fazit | 44 |
| 2.5 AP 1.5 - Validierung der Standards..... | 44 |
| 2.6 AP 1.6 - Dokumentation der Ergebnisse | 45 |
| 3 TP2 - Entwicklungsprozesse des Leitungssatzes | 47 |
| 3.1 AP 2.1 - Konzept kollaboratives Datenmodell | 47 |
| 3.1.1 Kollaboration in der Leitungssatzentwicklung | 47 |
| 3.1.2 Datenhaltung in der Leitungssatzentwicklung | 48 |
| 3.1.3 Anwendungspotential der VWS | 49 |
| 3.1.4 Fallbeispiel für den Einsatz von VWS bei der Auswahl eines Steckers | 49 |
| 3.1.5 Fazit | 51 |
| 3.2 AP 2.2 - Single-Point-of-Truth-Definition..... | 52 |
| 3.2.1 Herausforderung: Produkt mit mehreren Herstellern | 54 |
| 3.3 AP 2.3 - Prozessbeschreibung LS-Entwicklung | 55 |
| 3.3.1 Analyse des Entwicklungsprozesses | 55 |
| 3.3.2 Entwicklungsprozess mit der Verwaltungsschale | 57 |
| 3.3.3 Analyse und Integration der Einzelkomponenten..... | 58 |
| 3.3.4 Strukturierung der Verbundkomponente | 58 |
| 3.3.5 Anwendungsfall und Voraussetzungen | 59 |
| 3.4 AP 2.4 - Teilmodelle der Verwaltungsschale | 60 |
| 3.4.1 Teilmodelle für die Entwicklung | 60 |
| 3.5 AP 2.5 - Umsetzung Digital Twin LS | 63 |
| 3.5.1 Der Anwendungsfall | 63 |
| 3.5.2 Voraussetzungen für die Umsetzung | 64 |
| 3.5.3 Verwaltungsschalen der Komponenten | 65 |

| | | |
|------------|---|------------|
| 3.5.4 | Erstellung Verbundkomponente | 67 |
| 3.5.5 | Fähigkeitsabgleich der Produktionsmaschinen | 69 |
| 3.5.6 | Abschlussdemonstrator | 70 |
| 3.6 | Fazit | 70 |
| 4 | TP3 - Produktionsprozesse des Leitungssatzes | 71 |
| 4.1 | AP 3.1 - Untersuchung von Anforderungen | 71 |
| 4.1.1 | Vorgehen | 71 |
| 4.1.2 | Ergebnisse | 73 |
| 4.2 | AP 3.2 - Erforschung eines Referenzmodells | 85 |
| 4.2.1 | Datenfluss-Architektur | 85 |
| 4.2.2 | Use Cases | 86 |
| 4.2.3 | CAD2BOP | 89 |
| 4.3 | AP 3.3 - Erforschung von Teilmodellen für Rückmeldedaten | 96 |
| 4.3.1 | Leitungssatzproduktion | 96 |
| 4.3.2 | Produktionsprozesse | 97 |
| 4.3.3 | Prozess VWS | 104 |
| 4.3.4 | Capabilities aus Prozesssicht | 107 |
| 4.3.5 | Prozesssteuerung | 111 |
| 4.4 | AP 3.4 - Proof of Concept | 116 |
| 4.4.1 | Datenvalidierung | 116 |
| 4.4.2 | Datenmapping | 120 |
| 4.5 | AP 3.5 - Validierung der Prinzipien | 123 |
| 4.5.1 | Prozessablaufsteuerung für Produktionsprozesse | 123 |
| 4.5.2 | Datenmapping mit Validierung | 132 |
| 4.5.3 | Micro-Service „Datenvalidierung“ | 132 |
| 4.5.4 | Traceability | 133 |
| 4.5.5 | Transfer in andere Produktserien | 133 |
| 4.6 | Fazit | 141 |
| 4.7 | Anhang: UML-Diagramm | 142 |
| 5 | TP4 - Montage des Leitungssatzes | 147 |
| 5.1 | Definition der Randbedingungen | 148 |
| 5.2 | Einbringen der LS-Tasche in die Karosserie | 150 |
| 5.2.1 | Das Anfahren des Anlieferungspunktes | 150 |
| 5.2.2 | Grobausrichtung der Transporttasche im Innenraum | 152 |
| 5.2.3 | Ablage der Transporttasche innerhalb der Karosserie | 152 |
| 5.3 | Grobauslegung des Leitungssatz | 152 |
| 5.3.1 | Transporttasche Öffnen | 153 |
| 5.3.2 | Leitungssatzstränge im Innenraum auslegen | 153 |
| 5.3.3 | Leitungssatz am Fahrzeugboden montieren | 153 |
| 5.4 | Durchführung von Fädelprozessen | 153 |
| 5.4.1 | Lokalisieren des Einzelstrangs | 154 |
| 5.4.2 | Greifen des Einzelstrangs | 154 |
| 5.4.3 | Leitungsende zum Durchbruch bewegen | 154 |
| 5.4.4 | Leitungsende einfädeln | 155 |
| 5.5 | Dichtungselemente befestigen | 156 |
| 5.5.1 | Lokalisieren des Dichtungselements | 156 |
| 5.5.2 | Grobpositionierung des Dichtungselements | 156 |
| 5.5.3 | Feinpositionierung des Dichtungselements | 156 |
| 5.5.4 | Befestigung des Dichtungselements | 157 |
| 5.5.5 | Senkrechte Einbaurichtung zur Aussparung der Karosserie | 157 |
| 5.6 | Strangbefestigung | 158 |
| 5.6.1 | Clipse und Halterungen befestigen | 158 |
| 5.6.2 | Kabelkanäle und komplexe Halterungen befestigen | 160 |

| | | |
|------------|--|------------|
| 5.6.3 | Einlegen in Kabelkanäle und Schließen des Deckels | 160 |
| 5.7 | Steckverbindung herstellen | 161 |
| 5.7.1 | Konstruktive Gegebenheiten | 162 |
| 5.7.2 | Umsetzung der Steckverbindung im Fahrzeug..... | 162 |
| 5.7.3 | Verriegelungselemente von Gehäusen (z.B. CPA – Connector Position Assurance)..... | 163 |
| 5.8 | Fazit | 166 |
| 6 | TP5 - Integration der Verwaltungsschale..... | 167 |
| 6.1 | AP 5.1 - Anforderungsdefinition..... | 167 |
| 6.1.1 | Allgemeine Anforderungen | 169 |
| 6.1.2 | Verbundkomponenten für den Kabelbaum (AP5.2) | 171 |
| 6.1.3 | Verbundkomponenten Produktionsmittel (AP5.3) | 172 |
| 6.1.4 | Integration der Verwaltungsschalen (AP5.4) | 173 |
| 6.1.5 | Mapping zwischen Produktionsressource und Leitungssatz (AP5.5) | 174 |
| 6.1.6 | Erfassung der Qualitätsdaten (AP5.6) | 175 |
| 6.2 | AP 5.2 - Verbundkomponente Produkt (Leitungssatz)..... | 177 |
| 6.2.1 | Zielsetzung | 177 |
| 6.2.2 | Ermittelte Beziehungen..... | 178 |
| 6.2.3 | Typisierte Beziehungen und Merkmalsfestlegungen | 190 |
| 6.2.4 | Umsetzung der ermittelten Beziehungen | 192 |
| 6.3 | AP 5.3 - Verbundkomponente Ressource (Produktionsmittel) und VIBN | 196 |
| 6.3.1 | Zielsetzung | 196 |
| 6.3.2 | Beispiel-Ressourcen | 196 |
| 6.3.3 | Betrachtete Use Cases | 197 |
| 6.3.4 | Konzept der Verbundkomponente „Ressource“ | 198 |
| 6.3.5 | Virtuelle Inbetriebnahme | 201 |
| 6.3.6 | Ausblick | 205 |
| 6.4 | AP 5.4 - Integration der Verwaltungsschalen | 206 |
| 6.4.1 | Zielsetzung | 206 |
| 6.4.2 | Definition und Umsetzung der Use Cases | 206 |
| 6.5 | AP 5.5 - Mapping zwischen Produktionsressource und Leitungssatz | 210 |
| 6.5.1 | Zielsetzung | 210 |
| 6.5.2 | Modellierung geforderter Fähigkeiten | 210 |
| 6.5.3 | Modellierung angebotener Fähigkeiten..... | 212 |
| 6.5.4 | Algorithmus zum Fähigkeitenabgleich | 215 |
| 6.5.5 | Anbindung von Skills | 217 |
| 6.5.6 | Anhang: Liste von Fähigkeiten und zugehörigen Parametern | 219 |
| 6.6 | AP 5.6 - Erfassung von Qualitätsdaten | 224 |
| 6.6.1 | Zielsetzung | 224 |
| 6.6.2 | Relevante Teilmodelle..... | 224 |
| 6.6.3 | Zusammenspiel der verschiedenen Teilmodelle | 225 |
| 6.6.4 | Prototypische Implementierung | 226 |
| 6.7 | Fazit | 227 |
| 7 | TP6 - Automatisierte Verhandlungsprozesse..... | 228 |
| 7.1 | AP 6.1 - Konzeptentwicklung für die technische und automatisierte Verhandlung | 228 |
| 7.1.1 | Definition von Anwendungsfällen | 228 |
| 7.1.2 | Technische Umsetzung der Verhandlungskonzepte | 233 |
| 7.1.3 | Proof-of-Concept..... | 236 |
| 7.2 | AP 6.2 - Verhandlungsszenarien und -strategien..... | 237 |
| 7.2.1 | Start des Verhandlungsprozesses | 237 |
| 7.2.2 | Ermittlung der möglichen Service Provider..... | 240 |
| 7.2.3 | Übergabe der I4.0-Nachrichten an BPMN-Workflows | 240 |
| 7.3 | AP 6.3 - Capabilities | 243 |
| 7.3.1 | Anforderungen an den Fähigkeitenabgleich in automatisierten Verhandlungsprozessen | 243 |

| | | |
|-------------|---|------------|
| 7.3.2 | Implementierung des erweiterten Fähigkeitenabgleichs | 244 |
| 7.3.3 | Bewertung der Herstellungsprozesse aus Sicht der Produktionsressource..... | 245 |
| 7.4 | AP 6.4 - ECLASS und Industrie 4.0-Sprache..... | 250 |
| 7.5 | Fazit | 252 |
| 8 | TP7 - Data Business Policy, Data Governance und Monetarisierungskonzepte..... | 254 |
| 8.1 | AP 7.1 - Anforderungserhebung..... | 254 |
| 8.1.1 | Datenrichtlinien (Data Business Policy)..... | 254 |
| 8.1.2 | Zugriffskontrolle | 255 |
| 8.1.3 | Rollen und Rechtekonzept | 256 |
| 8.1.4 | Datensicherheit beim Datenaustausch | 256 |
| 8.1.5 | Datensouveränität..... | 258 |
| 8.1.6 | Datenhaltung (Data Storage Policy) im Hinblick auf die gemeinsame Nutzung | 258 |
| 8.1.7 | Datenhaltungsformate und -infrastrukturen | 260 |
| 8.1.8 | Anbindung an Catena-X..... | 261 |
| 8.1.9 | Daten-Monetarisierung..... | 262 |
| 8.2 | AP 7.2 - Fachliche Konzeption der Daten-Policy entlang der Wertkette | 262 |
| 8.2.1 | Integration von Data-Governance-Aspekten in das Metamodell der VWS | 263 |
| 8.2.2 | IDS-Architektur | 268 |
| 8.2.3 | Voraussetzungen für den International Data Space (IDS) | 269 |
| 8.2.4 | Datenmonetarisierung | 271 |
| 8.3 | AP 7.3 - Pilotierung und Erprobung: Erstellung Implementierungsguideline | 275 |
| 8.4 | Fazit | 275 |
| 9 | TP8 - Data Storage Policy, Sicherheit und Anbindung an Catena-X | 277 |
| 9.1 | AP 8.1 - Technische Anforderungsanalyse..... | 277 |
| 9.2 | AP 8.2 - Technische Umsetzungskonzeption und Zielarchitektur..... | 277 |
| 9.3 | AP 8.3 - Vorbereitung der Anbindung an Catena-X | 277 |
| 9.3.1 | Beispielprodukt: Teilleitungssatz | 277 |
| 9.3.2 | Das Anwendungsszenario | 278 |
| 9.3.3 | Ablauf des Szenarios | 278 |
| 9.3.4 | Aufbau der technischen Infrastruktur | 281 |
| 9.3.5 | Umsetzung des Use Cases | 309 |
| 9.3.6 | Whitespots | 332 |
| 9.4 | User Interfaces..... | 336 |
| 9.5 | Ausblick | 338 |
| 9.5.1 | Bewertung Status Quo | 338 |
| 9.5.2 | Blick in die Zukunft | 339 |
| 10 | TP9 - Pilotierung, Erprobung und Demonstrator | 341 |
| 10.1 | AP 9.1: Konzeption und Bauplan des Demonstrators | 341 |
| 10.2 | AP 9.2 - Umsetzung des physischen Demonstrators..... | 341 |
| 10.2.1 | Demonstrator Ausbaustufe 1 (2022) – HMI 2022 | 341 |
| 10.2.2 | Demonstrator Ausbaustufe 2 (2023) - HMI 2023..... | 343 |
| 10.3 | AP 9.3 - Umsetzung des digitalen Demonstrators | 344 |
| 10.3.1 | Demonstrator Ausbaustufe 3 (2024) – HMI 2024 | 344 |
| 10.3.2 | Demonstrator Ausbaustufe 4 (2024) - Ergebnistagung..... | 346 |
| 10.4 | AP 9.4 - Demonstration der Anwendungsfälle | 347 |
| 11 | TP10 – Transfer und Koordination | 349 |
| 11.1 | AP 10.1 - Kontinuierlicher Basistransfer | 349 |
| 11.1.1 | Veranstaltungen | 352 |
| 11.1.2 | Publikationen | 353 |
| 11.2 | AP 10.2 - Allgemeines Implementierungspaket..... | 354 |
| 11.3 | AP 10.3 - Implementierungspaket Leitungssatz..... | 355 |
| 11.4 | AP 10.4 - Ergebnisintegration und Koordination | 355 |
| 12 | Architektur AAS | 356 |

| | |
|---|------------|
| 12.1 Einführung | 356 |
| 12.1.1 Zentrale vs. dezentrale Architektur..... | 357 |
| 12.1.2 Single Point of Truth..... | 358 |
| 12.1.3 Gestaltungsfelder | 358 |
| 12.2 Modularisierung | 360 |
| 12.2.1 Aufteilung des Leitungssatzes in Module..... | 361 |
| 12.2.2 Verwaltungsschalen im Entwicklungsprozess | 362 |
| 12.2.3 Verwaltungsschalen im Produktionsprozess..... | 362 |
| 12.2.4 Inhalte und Verknüpfung der Stakeholder-Verwaltungsschalen | 363 |
| 12.3 Verlinkung | 369 |
| 12.3.1 Typ-Verwaltungsschale des OEM | 369 |
| 12.3.2 Typ-Verwaltungsschale (Konfektionär) | 372 |
| 12.3.3 Instanz-Verwaltungsschale des OEM | 376 |
| 12.3.4 Instanz-Verwaltungsschale des Konfektionärs..... | 377 |
| 12.3.5 Verlinkung des VEC..... | 379 |
| 12.5 Versionierung | 384 |
| 12.5.1 VWS-Aufbau für den Leitungssatz..... | 384 |
| 12.5.2 Versionierungskonzept der Verwaltungsschale | 385 |
| 12.5.3 Zugriff mit Registry und Discovery | 387 |
| 12.5.4 Zugriff ohne Discovery | 388 |
| 12.5.5 Zugriff ohne Registry | 388 |
| 12.6 Synchronisation | 390 |
| 12.6.1 Rahmenbedingungen für den gewählten Lösungsansatz | 390 |
| 12.6.2 Konfiguration der Synchronisationsbeziehung | 391 |
| 12.6.3 Senden von I4.0-Nachrichten zur Synchronisation | 394 |
| 12.6.4 Inhalte der I4.0-Nachrichten zur Synchronisation | 395 |
| 12.6.5 Verarbeiten von I4.0-Nachrichten zur Synchronisation | 396 |
| 12.6.6 Umgang mit Fehlerzuständen | 397 |
| 12.8 Änderungsmanagement | 398 |
| 12.8.1 Ablauf | 398 |
| 12.8.2 Änderungsmanagement in der Verwaltungsschale | 402 |
| 12.10 Rückverfolgbarkeit..... | 405 |
| 12.10.1 Szenario 1: Rückverfolgbarkeit über mehrere Tier-Stufen..... | 405 |
| 12.10.2 Szenario 2: Rückverfolgbarkeit beim Kabelkonfektionär | 406 |
| 12.10.3 Datenvorhaltung | 407 |
| 12.11 Fazit..... | 408 |
| Literaturverzeichnis | 410 |
| Abbildungsverzeichnis | 416 |
| Tabellenverzeichnis | 422 |
| Abkürzungsverzeichnis..... | 423 |
| Autorenverzeichnis..... | 425 |

Projektkonsortium

Das Projekt "Verwaltungsschale für den Leitungssatz" ([VWS4LS](#)) ist hervorgegangen aus der "Innovationsinitiative Leitungssatz" ([IILS](#)) - später umbenannt in "Standardisierungsinitiative Leitungssatz" ([SILS](#)) - als gefördertes Verbundvorhaben im Rahmen des Konjunkturpakets 35c der Bundesregierung "Zukunftsinvestitionen für Fahrzeughersteller und Zuliefererindustrie" ([Kopa 35c](#)).

Die Projektlaufzeit wurde von Dezember 2021 bis Dezember 2024 angesetzt, mit den nachfolgend aufgeführten Partnern:

| | | |
|--|---|--|
|  |  |  |
| Hauptansprechpartner: Michele Laghese Global Director IT & Digital | Hauptansprechpartner: Bernd Jost Managing Director | Hauptansprechpartner: Dr. Alexander Salinas Segura Digital Transformation Specialist |
|  |  |  |
| Hauptansprechpartner: Dr. Matthias Freund Entwicklungsingenieur Digital Engineering | Hauptansprechpartner: Pascal Neuperger Applikationstechniker & Product Support Manager | Hauptansprechpartner: Jürgen Bücker Managing Director |
|  |  |  |
| Hauptansprechpartner: Dr. Michael Buchta Teamleiter Forschungs- und Technologiemanagement | Hauptansprechpartner: Thomas Glockseisen Geschäftsführer | Hauptansprechpartner: Georg Schnauffer Stv. Geschäftsführer |

- [Coroplast Fritz Müller GmbH & Co. KG](#)
- [DiIT GmbH](#)
- [Lisa Dräxlmaier GmbH](#)
- [Festo SE & Co. KG](#)
- [Komax Testing Germany GmbH](#)
- [Kostal Kontaktsysteme GmbH](#)
- [Kromberg & Schubert Automotiv GmbH Co. KG](#)
- [Wezag GmbH & Co. KG](#)

Als assoziierte Partner engagieren sich die Mercedes-Benz AG und die Siemens AG aus der Standardisierungsinitiative Leitungssatz heraus als OEM sowie Hersteller digitaler Tools.

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.

Die [ARENA2036 e.V.](#) fungierte als Verbundkoordinator, während das [VDI-Technologiezentrum \(vditz.de\)](#) den Projektträger repräsentierte.

1 Überblick: Stand des Gesamtprojektes

1.1 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 VWS4LS zur Transformation dieser Wertkette. VWS4LS legt eine Grundlage für eine standardisierte digitale Beschreibung des Leitungssatzes und seiner Komponenten selbst, der Maschinen, mit denen der Leitungssatz produziert wird und die Prozesse, entlang derer die Einzelschritte verkettet werden.

VWS4LS betritt mit dieser Positionierung bisher zumindest aus Sicht der Verwaltungsschale wenig erschlossenes Terrain und das in mehrerer Hinsicht:

- VWS4LS bezieht mit dem Engineering-Prozess die Entstehung des gesamten Lifecycles mit ein. Hier entsteht das Asset, hier muss auch die Verwaltungsschale entstehen.
- VWS4LS adressiert mit dem Leitungssatz eine Automotive-Wertkette, also einen Bereich klassischer Serienfertigung schlechthin. Gleichwohl stellt der Leitungssatz mit seiner nahezu unendlichen Kombinatorik aus Einzelkomponenten eine prädestinierte Industrie 4.0-Komponente in Losgröße 1 dar.
- Mit den Komponenten des Leitungssatzes, für die in VWS4LS Teilmodelle entwickelt werden, aber auch den fünf Anwendungsfällen werden Grundlagen geschaffen, die nicht nur der Wertkette „Leitungssatz“ helfen, sondern weit darüber hinaus nutzbar sind.
- Mit der Perspektive auf der späteren Umsetzung in realen Lieferbeziehungen ergeben sich Ansätze, aber auch Fragestellungen, die bisher wenig Beachtung fanden, aber für die Skalierung der Verwaltungsschale essenziell sind.
- Mit dem Einsatz der Verwaltungsschale legt VWS4LS eine Grundlage für die Catena-X-Readiness in der Wertkette „Leitungssatz“.
- Im Projekt VWS4LS erfolgt eine der ersten Implementierungen des Konzepts der Verbundkomponente und auch des automatisierten Verhandelns nach den Ansätzen der vernetzten Verwaltungsschalen. Darüber hinaus werden z.B. mit der Versionierung Fragen aufgeworfen, die Impulse für die weitere Arbeit in der IDTA liefern.

Aus dieser Positionierung haben sich für das Projekt relevante Chancen, aber auch Herausforderungen ergeben. Die Partner haben in der Planungsphase 2021 viele Entwicklungsaufgaben bereits vorausgesehen und in die Projektplanung mit einfließen lassen. Andere Aufgaben, die sich erst im Projektverlauf als solche erwiesen haben, wurden zusätzlich adressiert. In diesem Bericht werden die erarbeiteten Ergebnisse des Projektes dargestellt um ein Teilprojekt (TP) oder Arbeitspaket (AP) abzuschließen. In dem ersten Kapitel wird die initiale Planung erläutert und welche grundlegenden Abweichungen sich während der Projektlaufzeit ergeben haben. Anschließend wird auf die einzelnen TPs inhaltlich sowie organisatorisch eingegangen.

1.2 Projektstruktur: 10 Teilprojekte und 5 Anwendungsfälle

Die TPs stellten die dominante Strukturierung des Projektes dar. Sie untergliederten sich nach drei wesentlichen Prozessschritten, fünf Querschnittsthemen und zwei übergreifenden Themen: Die Vertikalen-TPs widmeten sich den wesentlichen Wertschöpfungsabschnitten (Entwicklungs-, Produktions- und Montageprozess) des Leitungssatzes. Die fünf Verwaltungsschalen-spezifischen TPs wurden als fachliche Querschnittsprojekte definiert. TP 9 und 10 stellten übergreifende Arbeitsfelder dar, die insbesondere im Hinblick auf eine TP-übergreifende Konsolidierung bei der Demonstration und dem Transfer die Kohärenz sicherstellen.

Für die Bearbeitung der einzelnen Inhalte in den TPs wurden auf mehr als 50 Experten aus den verschiedenen Häusern der Partner zurückgegriffen, siehe Abbildung 1-2. Hinzu kamen weitere Experten, die durch Unterbeauftragungen von [4Soft](#), [Fraunhofer IESE](#), [ifak](#) und [msg systems](#) ihre Expertise in das Projekt im Bereich Verwaltungsschale, dem Vehicle-Electric-Container (VEC) bzw. der Kabelbaumliste (KBL), oder auch aus dem Catena-X Kontext einbringen konnten.

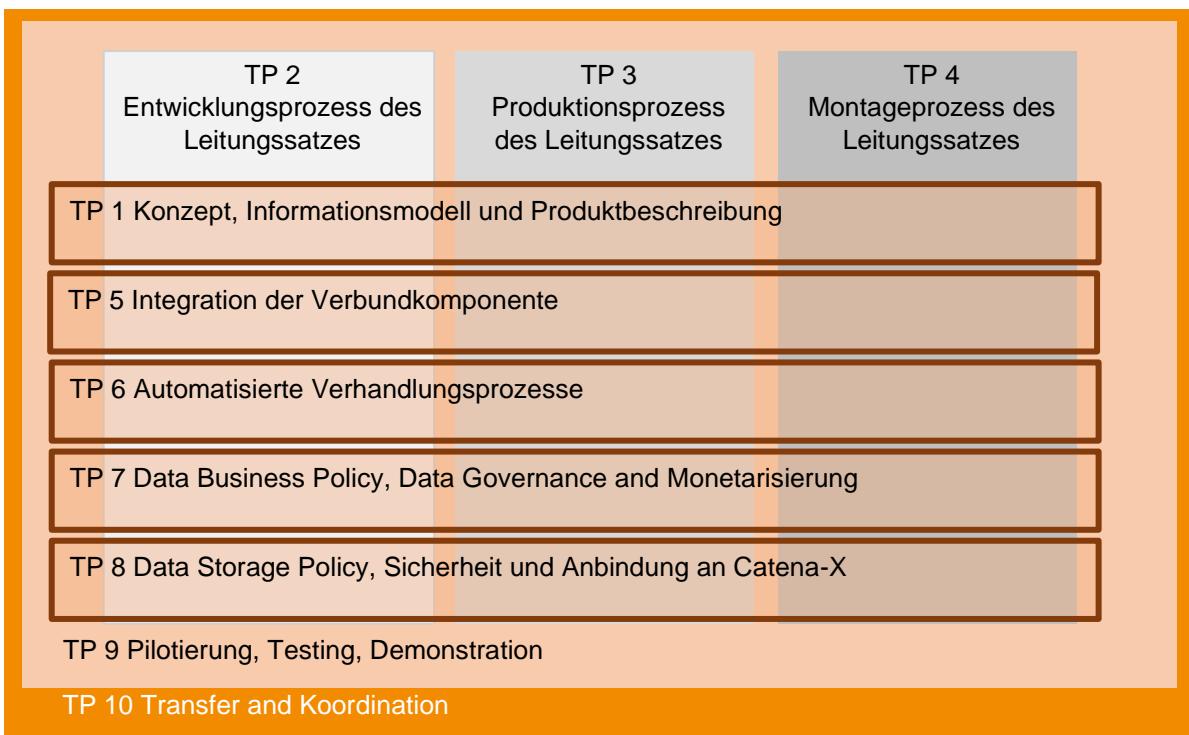


Abbildung 1-1: Matrixstruktur der Teilprojekte in VWS4LS.

| VWS4LS Projekt-Organigramm (Stand 7.10.2024) | | | | | | | | | | | ARENA2036 | |
|--|--------------------------------|--|---|--|---|---|---|---|--|--|------------------------------|--|
| Teilprojekt | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Partner assoz. Partner* | Haupt- Ansprech- partner | Konzepte, Informations- modelle, Produkt- beschreibung | Entwicklungs- prozesse des Leitungssatzes | Produktions- prozesse des Leitungssatzes | Montageprozes- se des Leitungssatzes in der Karosserie | Integration v. Verwaltungsschale/ Verbund- komponenten | Automatisierte Verhandlungs- prozesse | Data Business Policy, Data Governance, Monetarisierung | Data Storage Policy, Sicherheit, Anbindung an Catena-X | Pilotierung, Erprobung, Demonstrator | Transfer und Koordination | |
| Coroplast | x | x | x | x | x | x | x | x | x | x | x | |
| DiiT | x | x | x | x | - | x | x | - | x | x | x | |
| DRÄXLMAIER | x | x | x | x | x | x | x | x | x | x | x | |
| Festo | x | x | x | x | x | x | x | x | x | x | x | |
| Komax Testing | x | x | - | x | - | x | - | - | x | x | x | |
| Kostal | x | x | - | x | - | x | - | - | - | - | - | |
| Kromberg & Schubert | x | x | x | x | x | x | x | x | x | x | x | |
| WEZAG | x | x | x | x | x | x | x | x | x | x | x | |
| ARENA 2036 | x | x | x | x | x | x | x | x | x | x | x | |

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

Das Projekt selbst hat sich auf fünf Use Cases und zehn TPs fokussiert

Die TPs orientierten ihre Arbeiten nicht nur an den Vorhaben des Gesamtprojektes, sondern auch an fünf Anwendungsfällen, die aus Sicht der Leitungssatz-Akteure besondere Herausforderungen, aber auch Potenziale für die Verwaltungsschale darstellen. Diese Anwendungsfälle wurden bereits in der Gesamtvorhabenbeschreibung (GVB) festgelegt:

Tabelle 1-1 Anwendungsfälle bzw. Use Cases von VWS4LS

| | |
|------------|--|
| Use Case 1 | Kollaborative Entwicklung zwischen Original Equipment Manufacturer (OEM), Konfektionär, Komponentenlieferanten |
| Use Case 2 | Berücksichtigung der automatisierten Produktionsfähigkeit im Engineering |
| Use Case 3 | Automatisierung des Änderungsmanagements entlang der gesamten Wertkette |
| Use Case 4 | Automatisierung von flexiblen und modularen Produktionsabläufen |
| Use Case 5 | Rückverfolgung aller Komponenten, Produktions- und Qualitätsdaten |

Kerngedanke der Anwendungsfälle war, dass die inhaltlichen Entwicklungen der TPs alle zur Lösung der Anwendungsfälle beitragen sollen. Während in der GVB bei ca. der Hälfte der TPs noch kein Beitrag zu den einzelnen Anwendungsfällen absehbar war, kann heute festgestellt werden, dass tatsächlich fast alle TP-Ergebnisse einen Beitrag zur Lösung der Anwendungsfälle generieren.

1.3 Projektplanung: Der initiale Projektplan hat im Wesentlichen Bestand

Die ursprüngliche Planung der einzelnen TPs und APs ist in Abbildung 1-3 zu sehen. Die roten Balken beschreiben die initiale Planung auf TP-Ebene, während die grünen Balken die Veränderung und Verschiebungen im Laufe des Projektes widerspiegeln.

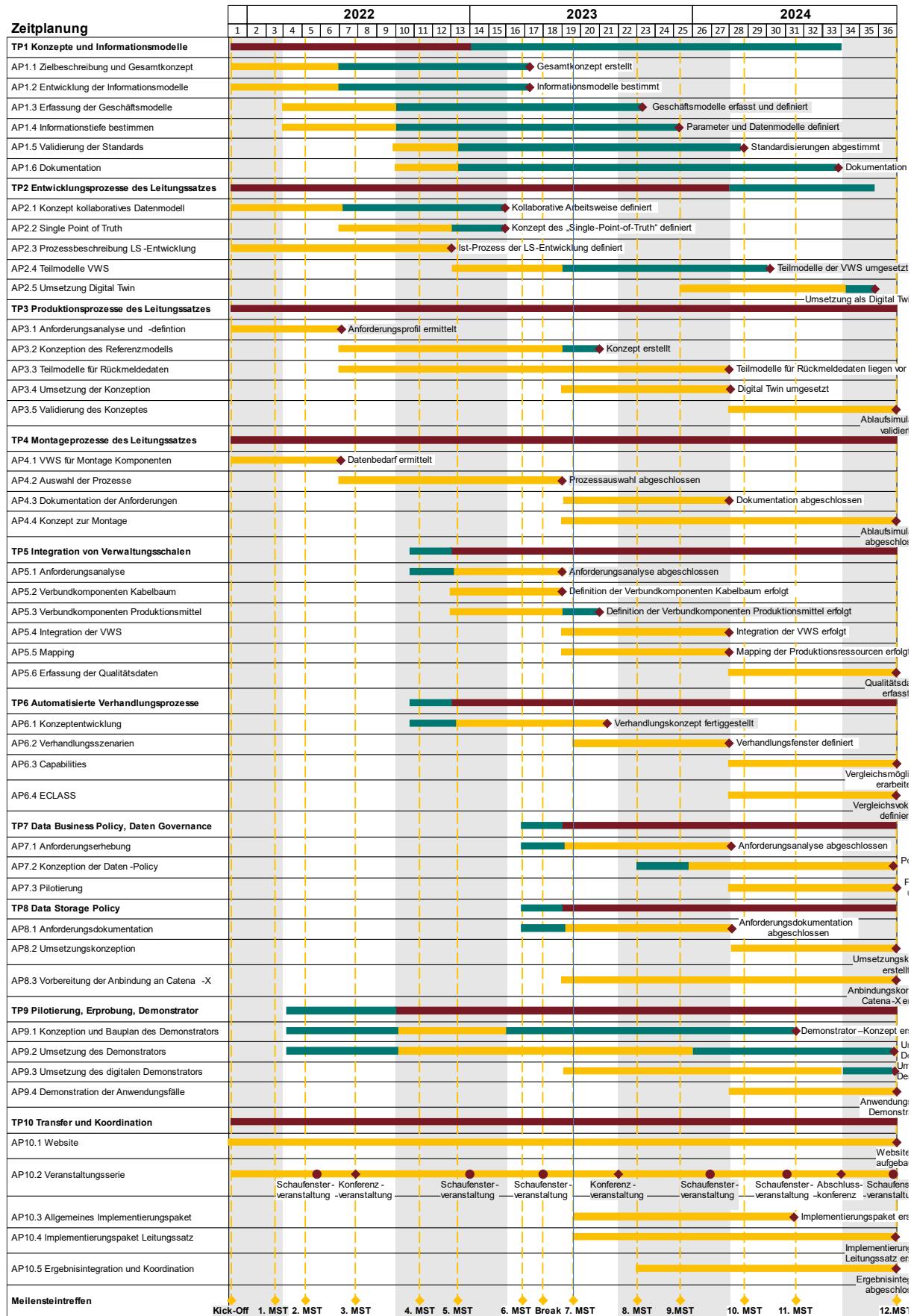


Abbildung 1-3 Initiale (rot) und angepasste Planung (grün) der Teilprojekte und Meilensteine

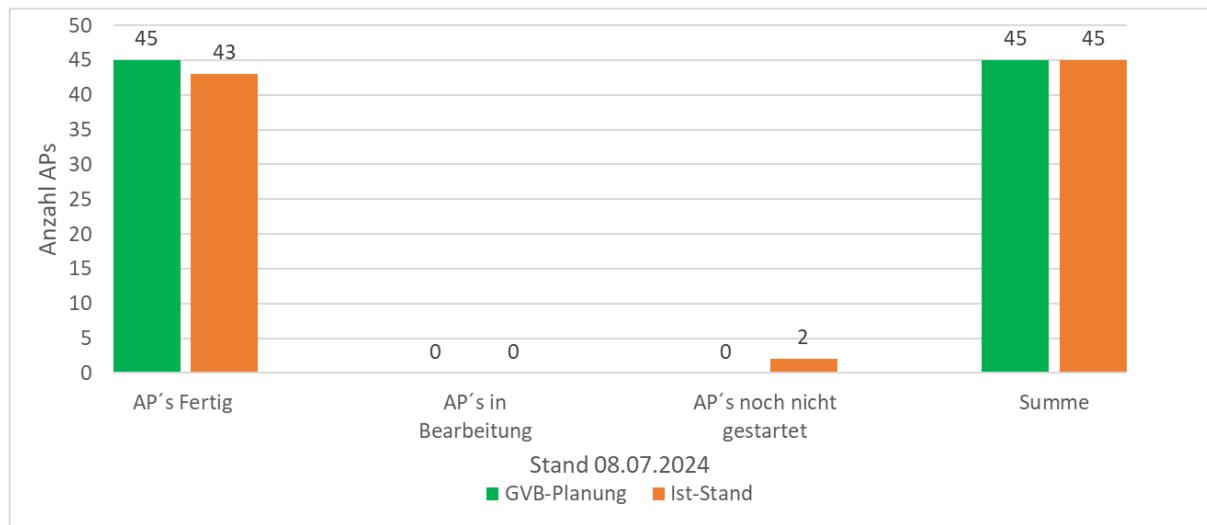
Folgende Anpassungen wurden in Bezug auf die initiale Planung des Projektes vorgenommen:

- Entgegen der ursprünglichen Planung wurde festgestellt, dass sich die in „TP1 - Konzepte, Informationsmodelle und Produktbeschreibung“ geplanten Inhalte nicht innerhalb des geplanten Zeitraums von ca. 12 Monaten abschließend bearbeiten ließen. Vor allem die Definition und Entwicklung der OPC-UA Companion Specification hat deutlich mehr Zeit in Anspruch genommen als ursprünglich geplant, daher wurde der Zeitplan bis zum Projektmonat 33 verlängert.
- Die Erarbeitung der Ergebnisse im TP2-Entwicklungsprozess des Leitungssatzes mussten ebenfalls verlängert werden. Vom Projektmonat 27 auf 35. Das lag vor allem daran, dass das AP2.4- Entwicklung der Teilmodelle deutlich mehr Zeit in Anspruch genommen hat und somit das AP2.5-Umsetzung des Digitalen Zwillings erst später richtig in die Bearbeitung starten konnte.
- Im TP3 wurde lediglich das AP-Konzeption des Referenzmodells von Projektmonat 18 auf 20 verschoben.
- Um den offiziellen Start des TP5 (Integration der Verbundkomponente) Ende 2022 besser vorzubereiten, wurde die Anzahl der Meetings im TP5 schon ab September 2022 erhöht und sich in regelmäßigen Abständen getroffen.
- Um den offiziellen Start des TP6 (Automatisierte Verhandlungsprozesse) Ende 2022 besser vorzubereiten, wurde die Anzahl der Meetings im TP6 schon ab September 2022 erhöht und sich in regelmäßigen Abständen getroffen.
- Um den offiziellen Start des TP7 (Data Business Policy, Data Governance and Monetarisierung) im Mai 2023 besser vorzubereiten, wurde die Anzahl der Meetings im TP7 ab Januar 2023 erhöht und sich in regelmäßigen Abständen getroffen.
- Im TP8, was sich vor allem mit der Anbindung an Catena-X beschäftigt, wurden einige Fragestellungen entdeckt, die ähnlich denen im TP7 waren. Aus diesem Grund wurden die Teilnehmer des TP8 ebenfalls zu den Meetings des TP7 eingeladen, um den gemeinsamen Austausch bereits vor dem offiziellen TP-Start zu suchen.
- Nach der Zusage für das Mitausstellen auf der Hannover Messe 2022 auf dem Stand der Plattform Industrie 4.0, wurde der Start des TP9 (Pilotierung, Erprobung und Demonstration) ebenfalls vorgezogen auf März 2022. Hierfür wurden ein physischer Demonstrator (Ausstellung eines gesamten Cockpit-Leitungssatzes einer Mercedes C-Klasse) und erste Verwaltungsschalen einzelner Komponenten zu einem frühen Stadium im Projekt erstellt.

Um weitere Demonstratoren zu entwickeln, die den Mehrwert der VWS herausstellen sollen, wurden einige APs innerhalb des TP9 verlängert. Die während der Projektbearbeitung entstandenen Erkenntnisse, sollen möglichst vollständig in den Demonstratoren untergebracht werden. Daher wurden bspw. die APs 9.2 und 9.3 bis zum Projektmonat 36 verlängert.

In der Abbildung

| Meilenstein | Zeitpunkt | Ergebnis | Fertigstellungsgrad Midtermevent | Fertigstellungsgrad MST 12 |
|-------------|-----------|---|-------------------------------------|-------------------------------|
| | | E1.1: Gesamtkonzept erstellt | ● | ● |
| | | E1.2: Informationsmodelle bestimmt | ● | ● |
| MS 2 | M 6 | E2.1: Kollaborative Arbeitsweise definiert | ● | ● |
| | | E3.1: Anforderungsprofil ermittelt | ● | ● |
| | | E4.1: Datenbedarf ermittelt | ◐ | ◐ |
| MS 3 | M 9 | E1.3: Geschäftsmodelle erfasst und definiert | ● | ● |
| | | E1.4: Parameter und Datenmodelle definiert | ● | ● |
| | | E1.5: Standardisierungen abgestimmt | ● | ● |
| | | E1.6: Dokumentation der Ergebnisse erfolgt | ◐ | ● |
| MS 4 | M 12 | E2.2: Konzept des „Single-Point-of-Truth“ definiert | ● | ● |
| | | E2.3: Ist-Prozess der LS-Entwicklung definiert | ● | ● |
| | | E10.3: Implementierungspaket erstellt | ◐ | ● |
| MS 5 | M 15 | E9.1: Demonstrator-Konzept erstellt | ● | ● |
| | | E2.4: Teilmodelle der VWS umgesetzt | ● | ● |
| | | E3.2: Konzept erstellt | ● | ● |
| | | E4.2: Prozessauswahl abgeschlossen | ● | ● |
| MS 6 | M 18 | E5.1: Anforderungsanalyse abgeschlossen | ● | ● |
| | | E5.2: Definition der Verbundkomponenten Kabelbaum erfolgt | ● | ● |
| | | E5.3: Definition der Verbundkomponenten Produktionsmittel erfolgt | ● | ● |
| | | E6.1: Verhandlungskonzept fertiggestellt | ● | ● |
| | | E10.4: Implementierungspaket Leitungssatz erstellt | ◐ | ● |
| MS 7 | M 21 | Konferenzveranstaltung durchgeführt | ● | ● |
| MS 8 | M 24 | E9.2: Umsetzung des Demonstrators erfolgt | ● | ● |
| | | E2.5: Umsetzung als Digital Twin erfolgt | ◐ | ● |
| | | E3.3: Teilmodelle für Rückmelddaten liegen vor | ● | ● |
| | | E3.4: Digital Twin umgesetzt | ● | ● |
| | | E4.3: Dokumentation abgeschlossen | ● | ● |
| MS 9 | M 27 | E5.4: Integration der VWS erfolgt | ● | ● |
| | | E5.5: Mapping der Produktionsressourcen erfolgt | ● | ● |
| | | E6.2: Verhandlungsfenster definiert | ◐ | ● |
| | | E7.1: Anforderungsanalyse abgeschlossen | ● | ● |
| | | E8.1: Anforderungsdokumentation abgeschlossen | ● | ● |
| MS 10 | M 30 | Konferenzveranstaltung durchgeführt | ● | ● |
| MS 11 | M 33 | E9.3: Umsetzung des digitalen Demonstrators erfolgt | ● | ● |
| | | E3.5: Validierung der Prinzipien | ◐ | ● |
| | | E4.4: Ablaufsimulationen abgeschlossen | ○ | ○ |
| | | E5.6: Qualitätsdaten erfasst | ● | ● |
| | | E6.3: Vergleichsmöglichkeiten erarbeitet | ◐ | ● |
| | | E6.4: Vergleichsvokabular definiert | ○ | ● |
| | | E7.2: Policy-Konzept liegt vor | ● | ● |
| MS 12 | M 36 | E7.3: Pilotprojekte umgesetzt | ◐ | ● |
| | | E8.2: Umsetzungskonzept erstellt | ◐ | ● |
| | | E8.3: Anbindungskonzept an Catena-X erstellt | ◐ | ● |
| | | E9.4: Anwendungsfälle demonstriert | ● | ● |
| | | E10.1: Webseite aufgebaut | ● | ● |
| | | E10.2: Veranstaltungen durchgeführt | ● | ● |
| | | E10.5: Ergebnisintegration abgeschlossen | ◐ | ● |



1.4 Zeitplanung und Arbeitstreffen

Jahresübergreifend wurden seit Projektbeginn im Dezember 2021 mehr als. 400 TP-Termine wahrgenommen (ca.140 in 2022, ca.170 in 2023 und 120 in 2024 – Stand bis zum 18.9.2024), siehe Abbildung 1-4. Alle TP-Termine werden durch das Project Office koordiniert und vom zentralen VWS4LS-Account dem jeweiligen Verteiler als persönliche Outlook-Einladung zur Verfügung gestellt. Zusätzlich zu diesen Terminen finden innerhalb der TPs, AP-bezogene Termine statt. Diese Termine werden von den jeweiligen Verantwortlichen selbst organisiert.

Hierbei wurden die Arbeitsgruppentermine, die eigens von den TP-Leitern durchgeführt wurden, nicht berücksichtigt. Zusammenfassend lässt sich aus Projektsicht festhalten, dass intensiv und mit hohem Zeit- und Arbeitseinsatz an der Umsetzung der in der Projektziele gearbeitet wird.

| Teilprojekt | 2022 | | | | | | | | | | | | |
|--|--|---|-------|------|------|-----|------|------|------|-------|------|------|------|
| | Dez. | Jan. | Febr. | März | Apr. | Mai | Juni | Juli | Aug. | Sept. | Okt. | Nov. | Dez. |
| TP 1 Konzept, Informationsmodelle und Produktbeschreibung | 11.1. 3.2. 17.2. 3.3. 17.3. 31.3. 14.4. 28.4. 12.5. 23.5. 9.6. 23.6. 7.7. 21.7. 4.8. | 15.9. 29.9. 13.10. 27.10. 10.11. 21.11. 15.12. | | | | | | | | | | | |
| TP 2 Entwicklung sprozesses des LS | 17.1. 8.2. 2.3. 23.3. | 12.4. 3.5. 24.5. 13.6. 4.7. 26.7. | | | | | | | | | | | |
| TP 3 Produktionsprozess des LS | 12.1. 1.2. 21.2. 18.3. | 8.4. 26.4. 16.5. 7.6. 23.6. 5.7. 29.7. | | | | | | | | | | | |
| TP 4 Montageprozess des LS | 14.1. 4.2. 28.2. 14.3. | 13.4. 29.4. 19.5. 6.6. 24.6. 11.7. 28.7. | | | | | | | | | | | |
| TP 5 Integration von Verwaltungsschalen | 9.2. 23.3. | 2.5. 12.5. 1.1. 15.7. 29.7. | | | | | | | | | | | |
| TP 6 Automatisierte Verhandlungsprozesse | 4.2. 2.2. | 28.3. 15.3. 11.2. 30.3. 11.4. 25.5. 25.5. 8.6. 18.7. 2.9. 9.9. 4.10. 28.10. 21.11. 12.12. | | | | | | | | | | | |
| TP 7 Data Business Policy, Daten Governance, Monetarisierung | | | | | | | | | | | | | |
| TP 8 Data Storage Policy, Sicherheit, Anbindung an Catena-X | | | | | | | | | | | | | |
| TP 9 Pilotierung, Erprobung, Demonstrator | | | | | | | | | | | | | |
| TP 10 Transfer und Koordination | | | | | | | | | | | | | |
| TPL-Telko / TPL-Klausur | 10.1. 17.1. 24.1. 31.1. 17.2. 21.2. 28.2. 2.3. 14.3. 16.3. 30.3. 13.4. 26.4. 27.4. 24.5. 21.9. 27.9. 18.10. | | | | | | | | | | | | |
| Sprint 1 Sprint 2 Sprint 3 Sprint 4 Sprint 5 Sprint 6 Sprint 7 | 1.1. 18.1. 25.1. 1.2. 18.2. 25.2. 1.3. 18.3. 25.3. 1.4. 18.4. 25.4. 1.5. 18.5. 25.5. 1.6. 18.6. 25.6. 1.7. 18.7. 25.7. 1.8. 18.8. 25.8. 1.9. 18.9. 25.9. 1.10. 18.10. 25.10. 1.11. 18.11. 25.11. 1.12. 18.12. 25.12. | | | | | | | | | | | | |
| Sonstige Termine | | | | | | | | | | | | | |
| Meilenstein-Treffen | | | | | | | | | | | | | |
| Kick-off: 6.12. | | | | | | | | | | | | | |
| MST 1: 16.2. | | | | | | | | | | | | | |
| MST 2: 6.4. | | | | | | | | | | | | | |
| MST 3: 28.6. | | | | | | | | | | | | | |
| MST 4: 11.10. | | | | | | | | | | | | | |
| MST 5: 15.12. | | | | | | | | | | | | | |

◆ Vor-Ort Treffen ◆ Virtuelles Treffen

Abbildung 1-4: Teilprojekttreffen 2022

| Teilprojekt | Dez. | 2023 | | | | | | | | | | | |
|--|--------|-------|------|------|----------------------|-------------|-------------|-------------|---------------|------|------|------|----------------|
| | Jan. | Febr. | März | Apr. | Mai | Juni | Juli | Aug. | Sept. | Okt. | Nov. | Dez. | |
| TP 1 Konzept, Informationsmodelle und Produktbeschreibung | | 19.1 | 9.2 | 2.3 | 123.3 | 5.4 | 4.5 | 25.5 | 15.6 | 6.7 | 26.7 | | 7.9 |
| TP 2 Entwicklungprozess des LS | 7.12. | 18.1 | 8.2 | 1.3 | 223.3 | 5.4 | 3.5 | 24.5 | 14.6 | 5.7 | 26.7 | | 6.9 |
| TP 3 Produktionsprozess des LS | 9.12. | 17.1 | 7.2 | 28.2 | 213.3 | 4.4 | 3.5 | 23.5 | 14.6 | 4.7 | 25.7 | | 5.9 |
| TP 4 Montageprozess des LS | 14.12. | 16.1 | 6.2 | 27.2 | 203.3 | 4.4 | 2.5 | 22.5 | 12.6 | 3.7 | 24.7 | | 4.9 |
| TP 5 Integration von Verwaltungsschalen | 13.12. | 20.1 | 13.2 | 6.3 | 27.3 | 17.4 | 8.1 | 30.5 | 19.6 | 10.7 | 31.7 | | 1.9 |
| TP 6 Automatisierte Verhandlungsprozesse | 13.12. | 21.1 | 1.2 | 7.3 | 28.2 | 18.4 | 9.5 | 30.5 | 20.6 | 11.7 | 18 | | 13.9 |
| TP 7 Data Business Policy, Daten Governance, Monetarisierung | 8.12. | 25.1 | 15.2 | 8.3 | 29.3 | 19.4 | 10.5 | 31.5 | 21.6 | 12.7 | 28 | | 4.10 |
| TP 8 Data Storage Policy, Sicherheit, Anbindung an Catena-X | 14.12. | 26.1 | 16.2 | 9.3 | 30.3 | 20.4 | 11.5 | 16 | 22.6 | 13.7 | 3.8 | | 14.9 |
| TP 9 Pilotierung, Erprobung, Demonstrator | 12.12. | 30.1 | 20.2 | | 3.4 | 24.4 | 15.5 | 15.6 | 26.6 | 17.7 | 7.8 | | 18.9 |
| TP 10 Transfer und Koordination | 9.12. | 30.1 | 20.2 | | 3.4 | 24.4 | 15.5 | 15.6 | 26.6 | 17.7 | 7.8 | | 18.9 |
| Architekturenteam | | | | | Big-Picture Kick Off | 13.6 | 26.5 | 30.5 | 16 | 6.6 | 10.6 | 12.6 | |
| Hauptansprechpartner Telko | | | | | Architektur | 2.5 | 22.5 | | 3.7 | | | | Bord. Landshut |
| Sonstige Termine | | | | | HM 2023 | 17.4 – 21.4 | Ludwigsburg | Konferenz 3 | 24.7 | | | | 1.9 |
| Meilenstein-Treffen | | | | | 9.5 – 10.5 | | | | 21.9 | | | | 26.9 |
| | | | | | MST 5: 15.12. | MST 6: 14.3 | MST 7: 13.6 | MST 8: 12.9 | MST 9: 12.12. | | | | |

Abbildung 1-5 Teilprojekttreffen 2023

| Teilprojekt | 2024 | | | | | | | | | | | |
|--|------|-------|------|------|--------------|-------------|-------|-------------|-------|------|-------|----------------------|
| | Jan. | Febr. | März | Apr. | Mai | Juni | Juli | Aug. | Sept. | Okt. | Nov. | Dez. |
| TP 1 Konzept, Informationsmodelle und Produktbeschreibung | 19.1 | 6.2 | 29.2 | 21.3 | 11.4 | 2.5 | 23.5 | 13.6 | 4 | 25.7 | | 5.1 |
| TP 2 Entwicklungprozess des LS | 7.1 | 7 | 26.2 | 20.3 | 10.4 | 29.4 | | | | | 16.10 | |
| TP 3 Produktionsprozess des LS | 17.1 | 7.2 | 28.2 | 20.3 | 10.4 | 29.4 | | | | | 16.10 | |
| TP 4 Montageprozess des LS | 16.1 | 6.2 | 27.2 | 19.3 | 9.4 | 30.4 | 22.5 | | | | 22.10 | |
| TP 5 Integration von Verwaltungsschalen | 23.1 | 13.2 | 5.3 | 26.3 | 16.4 | 7.5 | 28.5 | 18.6 | 9.7 | 30.7 | | 10.9 |
| TP 6 Automatisierte Verhandlungsprozesse | 24.1 | 14.2 | 6.3 | 27.3 | 17.4 | 8.5 | 29.5 | 19.6 | 10.7 | 31.7 | | 11.9 |
| TP 7 Data Business Policy, Daten Governance, Monetarisierung | 25.1 | 15.2 | 7.3 | 25.3 | 18.4 | 8.5 | 28.5 | 19.6 | 11.7 | 18 | | 12.9 |
| TP 8 Data Storage Policy, Sicherheit, Anbindung an Catena-X | 30.1 | 20.2 | 14.3 | 2.4 | 23.4 | 14.5 | 15.6 | 25.6 | 16.7 | 6.8 | | 17.9 |
| TP 9 Pilotierung, Erprobung, Demonstrator | 31.1 | 21.2 | | | | | | 26.6 | 17.7 | 7.8 | | 18.9 |
| TP 10 Transfer und Koordination | 31.1 | 21.2 | | | | | | 26.6 | 17.7 | 7.8 | | 18.9 |
| Hauptansprechpartner Telko | 15.1 | 5.2 | 26.2 | 19.3 | 6.4 | 29.4 | 23.5 | 10.6 | 11 | 22.7 | | 2.9 |
| Sonstige Termine | | | | | HM 2024 | 22.4 – 26.4 | Bord. | Konferenz 5 | 3.5 | | | Bordnetze Digital |
| Meilenstein-Treffen | | | | | 7. & 8.5. | Ludwigsburg | 4.6 | | | | | MST 10: 12.3 |
| | | | | | MST 11: 4.6. | | | | | | | MST 12: 16. & 17.10. |

Abbildung 1-6 Teilprojekttreffen 2024

Nach Schaffung einer gemeinsamen Wissensgrundlage durch Schulungen und Beleuchtung der Gesamtthematik Verwaltungsschale in der Leitungssatzwertkette, musste zunächst ein Modell entwickelt werden, wie die verschiedenen Verwaltungsschalen in einer kollaborativen Wertkette auf Typ- und Instanz-Ebene zusammenhängen.

In den ersten 12 Projektmonaten wurde ein umfassendes Know-How rund um die Verwaltungsschale und deren Einsatzmöglichkeiten aufgebaut. Jedoch wurde auch festgestellt, dass für die tatsächliche Implementierung in reale Industrieprozesse noch einige grundsätzliche Fragestellungen geklärt werden müssen. Grundsätzlich kann für die weiteren 18 Projektmonate an den Meilensteinen in Abbildung 1-3 festgehalten werden. Hinzu kommt, dass der Fokus mehr in Richtung der tatsächlichen Anwendbarkeit gelegt werden soll, um weitere Lücken und Fragestellungen zu entdecken, idealerweise zu klären oder nach Projektende an die treibenden Organisationen weiterzugeben.

Aus VWS4LS-Sicht sollte die reale Anwendung und Implementierung der Verwaltungsschale mehr in den Fokus rücken. Nur dann könnten tatsächliche Fragestellungen oder Themen in Bezug auf die Implementierung entdeckt und anschließend geklärt werden. Die Systemgrenze lediglich innerhalb einer Forschungseinrichtung zu betrachten ist aus Sicht der Industrie für eine Implementierung nicht sinnvoll. Reale Prozesse sind häufig deutlich komplexer als deren Darstellung innerhalb einer Forschungsumgebung.

1.5 Verortung im Leitbild der Plattform Industrie 4.0

Das aktuelle [Leitbild der Plattform Industrie 4.0](#)¹ für ein globales, digitales Ökosystem ergibt sich aus drei Bestandteilen, siehe Abbildung 1-7. Im Folgenden wird erläutert, wie sich die Projektergebnisse aus VWS4LS dort anschließen.

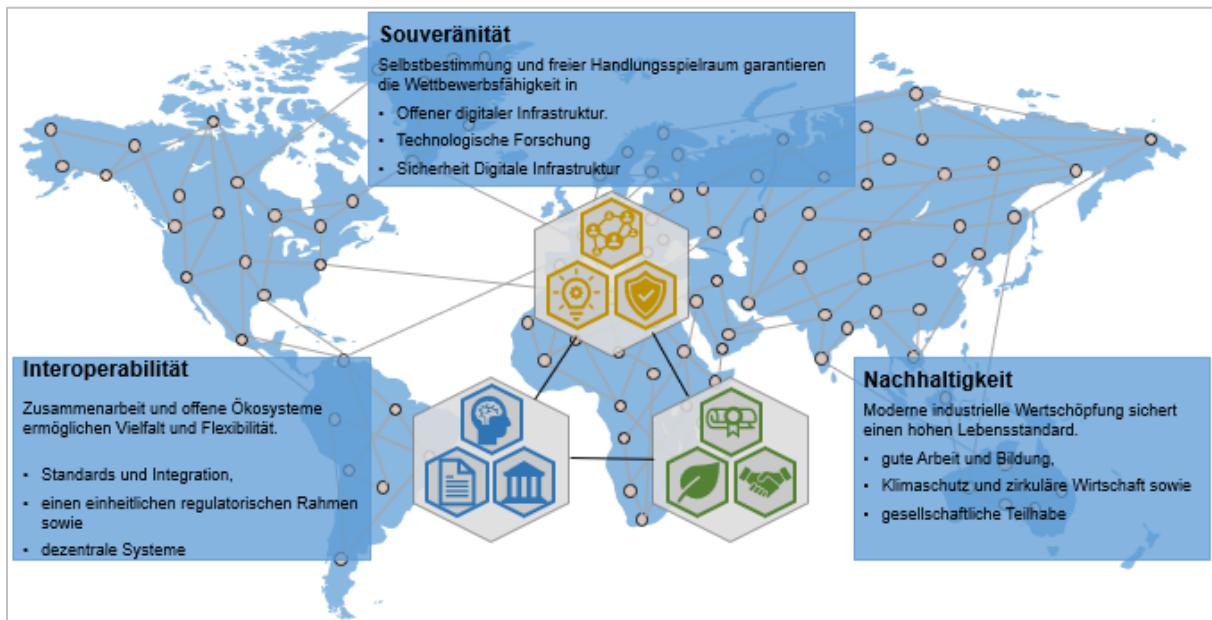


Abbildung 1-7: Leitbild Plattform Industrie 4.0. (Quelle: [Plattform Industrie 4.0, 2019](#))

Souveränität: Souveränität als Leitgedanke betont die Freiheit aller Akteure am Markt (Unternehmen, Mitarbeiter, Wissenschaft, Einzelpersonen), um selbstbestimmte, unabhängige Entscheidungen zu treffen und im fairen Wettbewerb miteinander zu agieren – von der Definition und Gestaltung des individuellen Geschäftsmodells bis zur Kaufentscheidung des Einzelnen innerhalb der I4.0 Ökosysteme [1].

VWS4LS: Im Projektes wird angestrebt, offene und für jeden am Markt zugänglichen Strukturen und Technologien zu verwenden. Diese sollen auf Ihre Eignung geprüft und in die Wertkette des Leitungssatzes integrierte werden. Da sowohl Großunternehmen als auch KMUs innerhalb des Projektes tätig sind, können hier auf verschiedenste Unternehmensstrukturen zurückgegriffen werden und die Anwendbarkeit und Umsetzbarkeit unter verschiedenen Voraussetzungen geprüft werden.

Interoperabilität: Die flexible Vernetzung unterschiedlicher Akteure zu agilen Wertschöpfungsnetzen ist einer der zentralen Kernbausteine digitaler Geschäftsprozesse in der Industrie 4.0. Zur Gestaltung solcher komplexen, dezentral organisierten Strukturen ist die Interoperabilität aller Akteure eine strategische Schlüsselkomponente. Erst ein hohes Maß an Interoperabilität, zu der sich alle Partner eines Ökosystems bekennen und gleichermaßen beitragen, gewährleistet die direkte operative und prozessuale Vernetzung über Unternehmens- und Branchengrenzen hinweg. Umgekehrt ermöglichen interoperable Strukturen und Schnittstellen sowohl Herstellern als auch Kunden die unbeschränkte Teilhabe an digitalen Wertschöpfungsnetzen und damit schließlich die Gestaltung neuer Geschäftsmodelle. Interoperabilität stärkt damit auch Souveränität [1].

VWS4LS: Durch die Implementierung der Verwaltungsschale und des Eclipse DataSpace-Connectors (EDC) soll auf offene Technologien innerhalb des Projektes zurückgegriffen und allen Teilnehmern ermöglicht werden, an die durchgängige und interoperable Wertkette angegeschlossen zu werden. Die jeweiligen Datenbanken aller Teilnehmer sollen dezentral in den Unternehmensgrenzen verbleiben und mittels Rollen- und Rechte-Konzepten für bestimmte Daten an bestimmte Lieferanten zugänglich gemacht werden. Diese Zugriffskontrolle soll hauptsächlich über Attribute-Based-Access-Control (ABAC) umgesetzt werden. Hierfür werden reale Sze-

¹ <https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Leitbild-2030-f%C3%BCr-Industrie-4.0.html>

narien aufgesetzt, wie eine Implementierung von Verwaltungsschale und EDC beispielhaft geprüft werden können. Neben dem grundsätzlichen Konzept, wie Daten auf unternehmensübergreifender Ebene ausgetauscht werden können, müssen die Daten digital bereitgestellt werden. Hierfür werden einzelne Submodelle entwickelt, die je nach Wertschöpfungsstufe angereicht werden können. Diese sollen die Grundlage für den Austausch von Informationen bilden (bspw. Austausch ganzer Submodelle oder Submodell-Elemente).

In den einzelnen Verwaltungsschalen sollen die notwendigen Capabilities (Fähigkeiten) für die Herstellung des jeweiligen Produktes hinterlegt werden, um auch mögliche automatisierte Verhandlungsprozesse möglich zu machen (bspw. frühzeitiger Abgleich der geforderten (Produkt) und bereitgestellten (Ressource) Capabilities).

Nachhaltigkeit: Ökonomische, ökologische und soziale Nachhaltigkeit stellen einen fundamentalen Eckpfeiler der gesellschaftlichen Wertorientierung dar. Diese Aspekte fließen einerseits in Industrie 4.0 ein, andererseits ermöglicht Industrie 4.0 den Nachhaltigkeitsbestrebungen erhebliche Fortschritte. So basieren der Wohlstand und die Lebensqualität jedes Einzelnen zu erheblichen Teilen auf einer zukunfts- und wettbewerbsfähigen Industrie. Das Ökosystem aus Innovation und Umsetzung von Industrie 4.0 liefert damit den Nährboden für Nachhaltigkeit durch Industrie 4.0 genauso wie für eine nachhaltige Industrie 4.0 selbst – und trägt damit schließlich entscheidend zur Erhaltung des Lebensstandards der Gesellschaft bei [1].

VWS4LS: Das Thema rund um die Nachhaltigkeit ist nicht im direkten Scope des Projektes. Es ergeben sich beiläufig Teilergebnisse, die für die Zwecke der Nachhaltigkeit verwendet werden können.

2 TP1 - Konzept, Informationsmodelle und Produktbeschreibung

Im Teilprojekt 1 „Konzept, Informationsmodelle und Produktbeschreibung“ wurden Konzepte rund um das Informationsmodell und die Produktbeschreibung der Leitungssatzherstellung für die Automobilindustrie mittels Verwaltungsschalen (VWS) erarbeitet.

Die VWS 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.

Es 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.

Das Modell soll dabei sowohl in der Entwicklungs- als auch in der Fertigungsphase des Leitungssatzes eingesetzt werden.

In der **Entwicklungsphase** soll das Modell eine hinreichend detaillierte Darstellung des Leitungssatzes ermöglichen, einschließlich aller Komponenten und Eigenschaften. Damit soll die Planung und Optimierung des Designs erleichtert und sichergestellt werden, dass der Leitungssatz alle erforderlichen Spezifikationen und Standards erfüllt.

In der **Fertigungsphase** soll das Modell eine effiziente Kommunikation und Koordination zwischen den an der Produktion des Leitungssatzes beteiligten Maschinen und Systemen ermöglichen. Dies soll dazu beitragen, Fehler zu minimieren, die Produktionszeit zu verkürzen und die Qualität des fertigen Produkts zu verbessern.

Durch die derartige Nutzung des Modells in Entwicklung und Fertigung soll ein hohes Maß an automatisierbarer Kontrolle und Flexibilität gewährleisten werden, um letztendlich einen hochwertigen und effizient produzierten Leitungssatz zu erzielen.

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

2.1 AP 1.1 - Zielbeschreibung des Gesamtkonzepts

Das Hauptziel des AP 1.1 "Zielbeschreibung des Gesamtkonzepts" war die Entwicklung eines Konzepts für die einheitliche Beschreibung der Komponenten des Leitungssatzes. Dies beinhaltet die Definition und Standardisierung der Terminologie und der Spezifikationen, die für die Beschreibung der verschiedenen Komponenten des Leitungssatzes verwendet werden.

Dies soll dazu beitragen, Missverständnisse und Inkonsistenzen zu vermeiden und eine klare und konsistente Kommunikation zwischen den verschiedenen Teams und Stakeholdern zu ermöglichen, die an der Entwicklung und Produktion des Leitungssatzes beteiligt sind.

Darüber hinaus soll dieses Arbeitspaket auch die Grundlage für die kollaborative Entwicklung legen, indem es klare Richtlinien und Prozesse für die Zusammenarbeit und den Informationsaustausch zwischen den verschiedenen Stakeholdern und Teams bereitstellt, um die Effizienz und Effektivität der kollaborativen Entwicklung zu verbessern und letztendlich zu einem hochwertigen und effizient produzierten Leitungssatz zu führen. Von der Vorgehensmethodik her wurde im AP 1.1 zuerst eine Anforderungserhebung durchgeführt, um die dort identifizierten Anwendungsfälle und Parameter als Basis für die Definition einer Prozess- und Ressourcenübersicht für das anschließende Mapping auf das Produkt-Prozess-Ressource-Modell (PPR-Modell) heranziehen zu können.

2.1.1 Anforderungserhebung

Im ersten Schritt wurde eine Liste von Anwendungsfällen zusammen mit Teilnehmern des Teilprojekts TP3 gesammelt, um ein umfassendes Verständnis der verschiedenen Szenarien zu erlangen, in denen der Leitungssatz in der Automobilindustrie eingesetzt wird. Durch die Zusammenarbeit mit den TP3-Teilnehmern konnte eine breite Palette von Anwendungsfällen identifiziert und erfasst werden, die die

Vielfalt und Komplexität der Herausforderungen widerspiegeln, welche bei der Entwicklung und Produktion von Leitungssätzen auftreten können.

Diese Anwendungsfälle (Use Cases) bilden die Grundlage für die Definition und Priorisierung der Anforderungen und Ziele der weiteren Arbeiten im Projekt. Sie ermöglichen es uns auch, die Auswirkungen und den Nutzen unserer Arbeit in einem realen Kontext zu bewerten und zu demonstrieren.

Die Use Cases erfassen verschiedene Sichten auf die Produktion des Leitungssatzes. Als wesentliche Aspekte sind die Produktbeschreibung inklusive der Verarbeitungsspezifikationen für die Komponenten und die Steuerung des Materialflusses durch die Fabrik zu nennen. Die Bedeutung von 3D-Daten (CAD-Daten) des Leitungssatzes und der Ressourcen ist ebenfalls erfasst worden, wie auch die standardisierte Kennzeichnung der physischen Assets, um den Bezug zur Verwaltungsschale herstellen zu können. So ist erkannt, dass die Versionierung von Instanzdaten eine entscheidende Relevanz für die eindeutige Beschreibung und Identifikation eines Leitungssatzes und der Einzelkomponenten besitzt.

Darüber hinaus wurden Themen wie bspw. Grenz- und Sollwerte im MES und die Möglichkeiten zur Anpassung von Prozessparametern erfasst.

Es wurde auch auf die Ausstattung und Konfiguration der Produktionsmaschinen eingegangen, d.h. Aspekte wie die Bereitstellung von Fähigkeiten der Maschinen und Maschinenkomponenten, Übertragung von Rezepten, die Abfrage von Werkzeug-Setups, die Überwachung des Produktionsfortschritts und des Status der Produktionsressourcen sowie die Überwachung des Materialverbrauchs.

| Global-ID | Personen-ID | Titel | Title | Beschreibung |
|-----------|--|---|-------|--|
| GL_23 | BAMA-02 BAMA-03 EmDa_13 | Überwachung des Materialverbrauchs | | Um rechtzeitig Material nachbestellen zu können, und inkorrekt eingestellte Prozessparameter zu erkennen, muss der Materialverbrauch und Ausschuss aller Teile (Kabel, Kontakte, etc.) überwacht werden können. |
| GL_24 | EmDa_16 EmDa_17 EmDa_18 EmDa_19 | Produktbeschreibung inklusive Verarbeitungsspezifikationen der Komponenten für die Durchführung der Produktionsprozesse | | Als Produktionsressource möchte ich alle Spezifikationsdaten (Kabel, Terminals, Seal, Sleeve, ...) auslesen, damit ich Einrichungs- und Prozessschritte planen und durchführen kann. |
| GL_25 | WOAI-01 MAZI-02 MIBR-05 | Eigenschaften von Konnektoren / Kontaktträgern | | Für die Herstellung und richtige Auswahl der Prüftechnik ist es wichtig die Eigenschaften von Konnektoren und die Anforderungen des Leitungssatzes zu haben, um Prüfadapter zu konstruieren/auszuhängen (!!!evtl. ist hier GL_8 mit zu berücksichtigen!!!) |
| GL_26 | MAZI-01 PEBR-02 | Leitungssatzinformationen für die Angebotserstellung | | Für die Angebotserstellung müssen die Leitungssatzinformationen (Zeichnungen, Verbindungspläne, Dimensionen, etc.) bekannt sein. Mit den Informationen aus der Verwaltungsschale Leitungssatz soll eine Auswahl des geeigneten Produktionsequipments durch Abgleich mit den in den Verwaltungsschalen der Ressourcen beschriebenen Fähigkeiten möglich sein. Die Informationen zu den Ressourcen soll eine Mengen/Kapazitätsplanung unterstützen. |

Abbildung 2-1: Arbeitsdokument Anforderungssammlung (Auszug)

Schließlich wurden auch Themen der Qualitätssicherung behandelt, wie die Validierung und Auswahl von Kontaktteilen, die Validierung der Leitungssatzentwicklung, die Abbildung von Varianten und die Rationalisierung der End-Of-Line (EOL) Prüfung. Es wurde auch auf die Bedeutung von Wartungs- und Kalibrierdaten, Wartungsprotokollen, der maximalen Ausstattung und Bestückungsvarianten eingegangen.

Von den ursprünglich 90 gesammelten Anwendungsfällen (Abbildung 2-1) wurden 46 als Grundlage für die Arbeit am Teilprojekt 1 ausgewählt. Relevant für die Auswahl war die Beziehung zur Definition des Datenmodells und den verwendeten Standards. Jeder dieser Anwendungsfälle wurde sorgfältig analysiert und bewertet, um seine Relevanz und seinen Beitrag zur Erreichung der Projektziele zu bestimmen. Die ausgewählten Anwendungsfälle decken ein hinreichend breites Spektrum der Szenarien und Herausforderungen ab, die bei der Herstellung von Leitungssätzen für die Automobilindustrie auftreten

können. Diese Anwendungsfälle bildeten daher die Grundlage für die weiteren Arbeiten im Projekt und um die Anforderungen und Ziele des Projekts effektiv zu definieren und zu priorisieren.

2.1.2 Klassen vom Materialen

In der Herstellung von Leitungssätzen werden verschiedene Materialklassen verwendet, die für die Leistung und Haltbarkeit des Endprodukts entscheidend sind.

- **Wire:** Dies ist das Grundelement eines jeden Leitungssatzes und besteht in der Regel aus Kupfer oder Aluminium. Diese Metalle werden aufgrund ihrer hervorragenden elektrischen Leitfähigkeit und Flexibilität ausgewählt. Die Drähte sind üblicherweise mit Isolationsmaterialien wie PVC, Polyethylen oder Polypropylen ummantelt, um elektrische Störungen zu verhindern.

- **Terminal:** Kontaktteile sind Metallteile, die verwendet werden, um zwei oder mehrere Drähte miteinander zu verbinden. Sie werden um den Draht gecrimpt, um eine starke, leitfähige Verbindung zu schaffen. Crimps müssen präzise ausgeführt werden, um eine zuverlässige Verbindung zu gewährleisten.

- **Seal:** Dichtungen werden eingesetzt, um die Verbindungspunkte vor Feuchtigkeit, Staub und anderen Umwelteinflüssen zu schützen. Sie bestehen häufig aus Gummi oder Silikon und bieten eine zusätzliche Isolationsschicht, die die Langlebigkeit des Leitungssatzes verbessert.

- **Sleeve:** Schutzhüllen dienen dazu, die Drähte vor physischen Schäden und Umwelteinflüssen wie Hitze oder Abrieb zu schützen. Materialien wie Nylon oder Polyester werden oft für diese Schutzhüllen verwendet, da sie eine hohe Widerstandsfähigkeit bieten.

- **Housing:** Gehäuse sind die äußeren Komponenten, die die elektrischen Verbindungen eines Leitungssatzes umschließen. Sie sind in der Regel aus Kunststoff oder Metall gefertigt und dienen dazu, die inneren Komponenten zu schützen und zu organisieren.

2.1.3 Prozessübersicht

Die Leitungssatzindustrie zeichnet sich durch eine Reihe von spezialisierten und komplexen Prozessen aus, die die Produktion von hochwertigen und funktionalen Leitungssätzen ermöglichen. Diese Prozesse sind integraler Bestandteil des Fertigungsablaufs und sind sorgfältig darauf ausgelegt, präzise Designanforderungen zu erfüllen.

In einem integrierten Produktionssystem existieren dabei mehrere Arten von Fertigungskompetenzen:

- **Drahtschneiden:** Hier werden die einzelnen Drahtstränge, die zur Bildung des Endprodukts verwendet werden, geschnitten, gecrimpt und versiegelt.
- **Montage:** Hier werden die Drahtstränge zusammen auf einer Montageplatte platziert, in die richtigen Gehäuse platziert sowie verklebt.
- **Qualitätsprüfung:** Hier wird der Leitungssatz auf elektrische, mechanische und andere Eigenschaften geprüft, bevor er an den OEM ausgeliefert wird.

Dabei werden die folgenden gängigen Arten von Produktionsmaschinen zur Herstellung von Leitungssätzen eingesetzt:

- **Schneidemaschinen:** Diese werden verwendet, um Drähte auf bestimmte Längen zu schneiden.
- **Abisoliermaschinen:** Diese Maschinen entfernen die Isolierung von den Enden der Drähte.
- **Crimpmaschinen:** Werden verwendet, um Anschlüsse oder Stecker an den Enden der Drähte anzubringen.
- **Löt-/Verzinnungsstationen:** Diese werden zum Löten oder Verzinnen von Drahtenden verwendet.
- **Drahtverdrillmaschinen:** Werden verwendet, um mehrere Drähte miteinander zu verdrillen.
- **Drahtmarkierungsmaschinen:** Sie markieren Drähte zu Identifikationszwecken.
- **Bündelmaschinen:** Damit werden Drähte zu einem Leitungssatz gebündelt.

- **Kabelbinder- und Schrumpfschlauchmaschinen:** Werden zum Anbringen von Kabelbindern und zum Schrumpfen von Schläuchen für zusätzlichen Drahtschutz verwendet.
- **Prüfmaschinen:** Zur Durchführung von elektrischen Sicherheitstests und zur Gewährleistung der Funktionalität.

Hierbei gibt es Maschinen, die mehrere dieser Fertigungsschritte kombinieren können, und es gibt einzelne Fertigungsschritte, die manuell durchgeführt werden.

2.1.4 PPR-Modell

Ein etablierter methodischer Ansatz für die Fertigungsdatenmodellierung ist das PPR-Modell [2] [3]. Es ermöglicht eine ganzheitliche Sicht auf ein Fertigungsprojekt aus unterschiedlichen Sichtweisen und einen Datenaustausch über Fachbereichsgrenzen hinweg. Dieser Ansatz erlaubt damit eine effiziente Gestaltung und Rekonfiguration von Fertigungsressourcen, um damit die Reduzierung der Entwicklungszeit und der Rüstkosten zu erreichen.

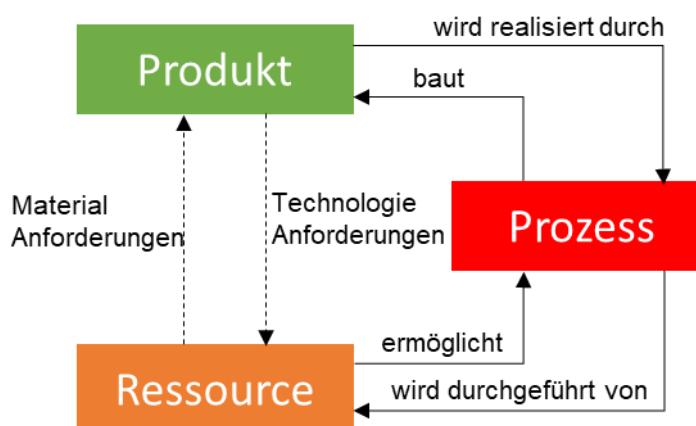


Abbildung 2-2: PPR-Modell

Die PPR-Modellierung [4] teilt den Produktionsvorgang in folgende Aspekte auf (Abbildung 2-2):

- **Produkt:** Bezeichnet den End- oder Zwischenprodukttyp. Es besteht ein Unterschied zwischen einem Produkttyp und dem 'realisierten' Werkstück zur Laufzeit
- **Prozess:** Stellt Veränderungen an einem Produkt dar, die während der Produktion auftreten. Prozesse können aus verschiedenen Bereichen stammen, wie z.B. Fertigung, Transport, Montage, usw.
- **Ressource:** Sind die Systemkomponenten (Hardware, Software) die zur Durchführung der Prozesse benötigt werden.

Eine Herausforderung besteht darin, häufige Produktänderungen innerhalb kurzer Zeiträume zu berücksichtigen. Dies ist aufgrund der Limitierungen aktueller Ansätze zum Entwurf, Aufbau und der Rekonfiguration von Automatisierungssystemen schwierig. Ebenso führen die steigende Menge und Vielfalt an Informationen aus der Produktion zu Engpässen, welche aber auf effiziente Art und Weise ausgetauscht und wiederverwendet werden sollen.

Ein effektiver Ansatz, um die Anforderungen der Produkte an die Montageautomatisierung zu erfüllen, ist die Integration von Datenmodellen zur Kopplung der PPR-Domänenmodelle. Damit können Daten aus verschiedenen Ingenieurdomänen und Ingenieurwerkzeugen effektiv verknüpft werden.

2.1.5 Fazit

Im Arbeitspaket 1.1 wurden die grundlegenden Anforderungen an die Leitungssatzproduktion erfasst und analysiert. Dies beinhaltete eine Zusammenstellung der typischen Materialklassen, Produktionsprozesse und Produktionsequipment in der Leitungssatzindustrie. Mit den gesammelten Informationen

konnte ein umfassendes Verständnis der spezifischen Anforderungen und Herausforderungen dieser Branche erlangt werden. Anschliessend wurde das PPR-Modell zur Erstellung von detaillierten Datenmodellen ausgewählt. So ist eine hinreichend genaue Datenmodellierung der realen Produktionsprozesse in der Leitungssatzindustrie als Grundlage für die Umsetzung von Automatisierungslösungen in diesem Umfeld möglich.

2.2 AP 1.2 - Entwicklung von Informationsmodellen und Zuweisung der Daten

Im AP 1.2 „Entwicklung von Informationsmodellen und Zuweisung der Daten“ wurde basierend auf den folgenden vorhandenen Standards die Datenmodellierung untersucht.

KBL [5] wurde als XML-basiertes Datenaustauschformat entwickelt, nachdem sich die bis dahin übliche Verwendung von Konstruktionszeichnungen als nicht mehr hinreichend erwiesen hatte. Die Zielsetzung der KBL ist die digitale Beschreibung der Produktspezifikation eines Kabelbaums als Grundlage für den Austausch zwischen OEM und Konfektionär. Dabei wird oft das Containerdatenformat HCV („Harness Container for Viewing“) verwendet, um bspw. KBL und SVG-Daten (skalierbare Vektorgrafik) zu kombinieren.

VEC [6] ist ebenfalls ein XML-basiertes Datenaustauschformat und als moderner Nachfolger der KBL zu sehen, entwickelt für den Austausch von Daten im Entwicklungsprozess des physikalischen Bordnetzes mit Übergang zur Produktion in der Automobilindustrie. Er bietet ein integriertes Modell mit umfassenden Möglichkeiten, unter anderem zur Beschreibung von Kabelbaumkomponenten, verschiedenen elektrologischen Sichten, sowie der Produktspezifikation des Kabelbaums mit dessen geometrischer Form (3D & 2D).

Open Platform Communications Unified Architecture (OPC-UA) [7] ist ein maschinenorientierter Kommunikationsstandard, der auch semantische Modellierung unterstützt und für die industrielle Automatisierung entwickelt wurde. Er ermöglicht eine sichere und zuverlässige Kommunikation zwischen verschiedenen Geräten und Softwareplattformen.

ECLASS [8] ist ein etabliertes System für die semantische Klassifizierung und Beschreibung von Produkten und Dienstleistungen, die im Kontext der Industrie 4.0 verwendet werden. Es soll festgestellt werden, ob ECLASS eine nützliche Ressource für die Klassifizierung von Stammdaten in Datenmodellen der Leitungssatzindustrie sein kann.

Reference Architecture Model for Industry 4.0 (RAMI 4.0) [9] ist ein Referenzarchitekturmodell, und könnte als Rahmen für die Modellierung der Konzepte innerhalb des Projekts dienen, denn durch die Nutzung internationaler Standards ermöglicht RAMI 4.0 eine nahtlose Kommunikation und Zusammenarbeit zwischen unterschiedlichen Systemen und Akteuren.

Das Ergebnis der Analyse zielte darauf ab, die besten Ansätze für die Nutzung dieser Standards innerhalb des Projekts zu identifizieren.

2.2.1 VEC und KBL

Im nächsten Schritt des Projekts wurde eine Analyse der KBL- und VEC-Datenmodelle durchgeführt, um das optimale Modell für das Projekt auszuwählen. KBL und VEC sind standardisierte Datenformate, die entwickelt wurden, um den Austausch, die Zusammenarbeit und die Archivierung von Informationen über das physische Bordnetz im Automobilsektor zu erleichtern. Die Modelle wurden unter dem Dach der prostep ivip Association [10] in Kooperation mit dem Verband der Automobilindustrie (VDA) [11] entwickelt.

Als Basis für eine Bewertung ist ein grundlegendes Verständnis der Ausrichtung von KBL & VEC (Abbildung 2-3), sowie deren Historie und Zusammenhänge notwendig [12].



Abbildung 2-3: KBL & VEC

KBL: Die KBL entstand im Zeitraum 1999 – 2005 (erste Veröffentlichung als VDA Recommendation) mit verschiedenen Erweiterungen in den folgenden Jahren. Die Zielsetzung der KBL ist die digitale Beschreibung der Produktspezifikation eines Kabelbaums als Grundlage für den Austausch zwischen OEM und Konfektionär (vgl. Abbildung 2-4).

VEC: In der Anwendung der KBL zeigte sich aber auch, dass dieser spezifisches Anwendungsfall nur ein kleiner, wenngleich auch ein sehr wichtiger, Teil der in der Bordnetzentwicklung existierenden Schnittstellen und Datenaustauschszenarien ist. Da die Abgrenzung der einzelnen Schnittstellen zum Teil unternehmens- und prozess-spezifisch ist, aber die ausgetauschten Inhalte und Strukturen an vielen Stellen überlappen, wurde der VEC als einheitliche digitale Sprache in der Wertschöpfungskette „physisches Bordnetz“ entwickelt (siehe „erweiterter Scope“ in Abbildung 2-4).

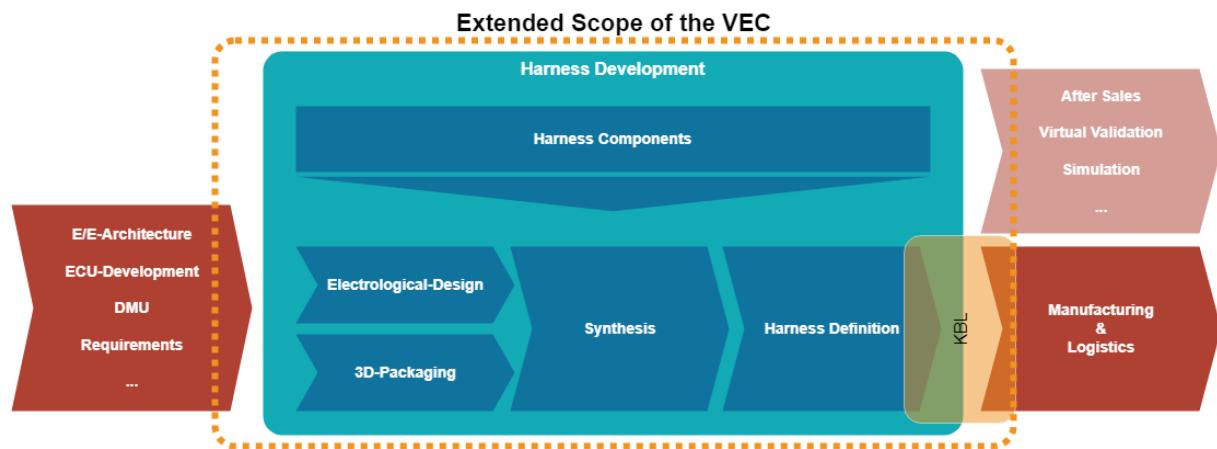


Abbildung 2-4 Vergleich Scope VEC & KBL [12]

2.2.1.1 Ausdrucksfähigkeit der Modelle

Unter dem Gesichtspunkt der Ausdrucksfähigkeit der beiden Modelle ist die KBL eine echte Untermenge des VEC, d.h. alle Sachverhalte, die sich in der KBL abbilden lassen, können auch im VEC ausgedrückt werden, umgekehrt ist dies nicht der Fall. Hinweis: Diese Aussage betrifft die Ausdrucksfähigkeit der beiden Modelle. Die dafür verwenden Modellierungskonzept und Struktur, also die Syntax, sind andere. Das heißt, eine KBL-XML-Dateien ist nicht gleichzeitig auch ein gültiges VEC-XML. Lediglich die Inhalte der Datei würden sich verlustfrei auch als VEC-XML ausdrücken lassen.

Dieser Unterschied ergibt sich im Wesentlichen durch zwei Dimensionen:

- (1) Der VEC hat einen deutlich größeren Scope. Insbesondere bietet der VEC die Möglichkeit, die Struktur eines Artikels zu definieren. Beispielsweise ist mit dem VEC eine detaillierte Abbildung von Komponentendaten möglich, zum Beispiel Aufbau und Eigenschaften von Leitungen, Informationen zu Kontakteilen, Ermittlung von Kammerpositionen in Steckern usw. Die KBL beschränkt sich hier im Wesentlichen auf die Sachnummer der Komponente und ein paar einfache Eigenschaften, die auf einer klassischen Leitungssatzzeichnung zu erkennen sind, zum Beispiel Farbe und Querschnitt von Leitungen oder Farbe und Kodierung von Steckern. Insbesondere wurde beim Design des VECs Wert darauf gelegt immer eine Nachverfolgbarkeit zwischen den Informationen der verschiedenen Disziplinen der Bordnetzentwicklung (z.B. Geometrie, Elektrologik, E/E-Architektur) zu erreichen.
- (2) Die KBL wird konzeptionell nicht mehr weiterentwickelt. Auch im Kern des Scopes der KBL haben sich einerseits die Anforderungen an die Detailtiefe einer Leitungssatzbeschreibung geändert, andererseits sind aber auch neue Komponentenarten und Technologien hinzugekommen, bzw. haben Bedeutung gewonnen, zum Beispiel modulare Stecker, Stecker mit mehreren Bündelanschlusspunkten, Hochvolt-Anwendungen, komplexere Leitungen, mehrteilige Kontaktierungssysteme. Diese Anforderungen sind bei der Entwicklung des VECs berücksichtigt worden, aber aus verschiedenen Gründen nicht mehr in die KBL eingeflossen (siehe Abschnitt 2.2.1.2).

2.2.1.2 Lebenszyklus und industrieller Einsatz

Die KBL befindet sich derzeit im „Maintenance Modus“, d.h. Themen, die den derzeitigen industriellen Einsatz gefährden, werden behandelt, es findet allerdings keine aktive Umsetzung neuer Anforderungen und keine Erweiterungen des Scopes statt.

Der Grund hierfür ist, dass eine Erweiterung des Scopes der KBL bzw. auch die Umsetzung der neuen Anforderungen im Scope der KBL nicht möglich gewesen wäre, bzw. sehr starke Änderungen an verschiedenen Modellierungskonzepten erfordert hätte. Hierzu muss man sich auch vor Augen führen, dass der Kern der KBL-Modellierung inzwischen 25 Jahre alt ist und auf dem damaligen Wissen und den entsprechenden Anforderungen und Modellierungskonzepten basiert. Das Ergebnis dieses „Weiterentwicklungsprozesses“ wäre eine „neue“ KBL gewesen, die aus Gesichtspunkten der Rückwärtskompatibilität, Wiederverwendbarkeit von Schnittstellen-Implementierungen und Daten keinen Mehrwert gegenüber einem direkten Umstieg auf den VEC geboten hätte.

Aus diesem Grund haben die zuständigen Gremien entschieden, die verfügbaren Ressourcen auf eine Weiterentwicklung des VECs zu konzentrieren. Der VEC wird aktuell aktiv weiterentwickelt und neue Anforderungen umgesetzt.

Die KBL ist Stand heute das verwendete Standardformat an der Schnittstelle OEM / Konfektionär für die Spezifikation einzelner Leitungssätze. Zudem wird die KBL in verschiedenen Prozessen zum Austausch von Leitungssatz-Geometriendaten und in seltenen Fällen für Elektrologik verwendet. Bei diesen Anwendungen handelt es sich aber um Anwendungen, die nicht im Scope der KBL liegen und ein hohes Maß an proprietärer Interpretation erfordern.

Der VEC wird bei verschiedenen OEMs für die Bereitstellung von Kabelbaum-Komponentendaten verwendet. Außerdem gibt es im Einsatz befindliche Implementierungen für den Austausch von Geometrie und Systemschaltplandaten. Außerdem wird er von verschiedenen Unternehmen für die Bereitstellung eines „digitalen Zwilling“ des physischen Bordnetzes an Folgeprozess (z.B. Kundendienst) genutzt. Neue Umsetzungsprojekte kommen hinzu.

2.2.1.3 Abdeckung Anforderungen im Kontext VWS4LS

Im Kontext des Projekts VWS4LS gibt es verschiedene Anwendungsfälle bei denen VEC bzw. teilweise auch KBL in Frage kommen. Im Folgenden drei Beispiele:

1. Produktspezifikation Leitungssatz: Dieser Anwendungsfall ist grundsätzlich sowohl mit KBL als auch VEC abbildbar. Fraglich ist dabei aber, ob die KBL für Anwendungsszenarien die notwendige Detailtiefe beinhaltet, zum Beispiel Kontaktierungssituationen an HV-Leitungen.
2. Bereitstellung von Komponentendaten: Zur Parametrisierung von Fertigungsprozessen sind detaillierte Informationen über Kabelbaumkomponenten notwendig, bspw. Terminalmaße oder Kammerkoordinaten in Steckern. Diese Information können nur mit dem VEC bereitgestellt werden.
3. Definition von Halbprodukten: Für eine Abbildung des Herstellungsprozesses ist die Definition von Halb-/Zwischenprodukten notwendig, beispielsweise eine geschnittene & abisolierte Leitung. Durch die flexiblere Modellierung können solche Produkte mit dem VEC abgebildet werden. Mit der KBL ist dies nicht immer möglich, oder nur durch sehr „individuelle“ Interpretation der Modellelemente.

2.2.1.4 Bewertung

Zusammenfassend lässt sich sagen, dass die KBL derzeit für die Beschreibung des Produkts „Leitungssatz“ aufgrund der längeren Historie und der Stabilität etablierter Prozesse die breitere Verwendung hat. Allerdings lassen sich gewisse Anwendungsfälle im Projektkontext mit der KBL gar nicht abbilden.

Der VEC hingegen ist aufgrund der Umsetzung neuer Anforderungen, sowohl fachlich wie auch technologisch (z.B. Unterstützung von Ontologien), der kontinuierlichen Weiterentwicklung und der zunehmenden Anzahl an Umsetzungen die Wahl der Zukunft.

Insgesamt bietet VEC eine fortschrittlichere Lösung für die Leitungssatzfertigung, während KBL aufgrund seiner weit verbreiteten Nutzung weiterhin relevant bleibt.

Beide Modelle haben ihre Vorteile: KBL ist weit verbreitet und hat eine etablierte Position in der Branche, während VEC eine fortschrittlichere Architektur bietet. Nach eingehender Analyse im Projekt wurde daher entschieden, sowohl das KBL- als auch das VEC-Datenmodell zu unterstützen.

2.2.2 ECLASS

ECLASS [8] ist ein weit verbreitetes Klassifikationssystem für Produkte und Dienstleistungen, welches in verschiedenen Branchen eingesetzt wird und eindeutige semantische Referenzen bereitstellt. Die Beschreibung von Leitungssätzen erfordert detaillierte Informationen über Steckverbinder, Leitungen, Verbindungen, Isolierungen und vieles mehr. ECLASS bietet zwar eine allgemeine Struktur und einige Definitionen mit Projektbezug (Auszüge siehe Abbildung 2-5 und Abbildung 2-6), aber es fehlen spezifische Attribute und Klassen, die für die detaillierte Beschreibung von Leitungssätzen geeignet sind. Lediglich die in ECLASS vorhandenen allgemeinen Einheits-Klassifizierungen kamen im TP3 "Produktionsprozesse des Leitungssatzes" zur Anwendung.

ECLASS BASIC 14.0 (de)

- 44 Fahrzeugtechnik, Fahrzeugkomponente
 - 44-04 Elektrik, Elektronik (KFZ)
 - 44-04-01 Bordnetz (KFZ, low voltage)
 - 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)
 - 44-04-01-04 Ladegerät (KFZ, low voltage)
 - 44-04-01-05 Masseband, Batterieleitung (KFZ, low voltage)
 - 44-04-01-90 Bordnetz (KFZ, low voltage, nicht spezifiziert)
 - 44-04-01-91 Bordnetz (KFZ, low voltage, Teile)
 - 44-04-01-92 Bordnetz (KFZ, low voltage, Zubehör)

Abbildung 2-5: ECLASS-Klassifizierungen für KFZ-Bordnetze [8]

ECLASS BASIC 14.0 (de)

- 17 Maschine, Apparat (für besondere Anwendungsbereiche)
 - 17-13 Maschine für Elektrohandwerk
 - 17-13-01 Maschine zur Kabelbe- und -verarbeitung ⓘ
 - 17-13-01-01 Crimp- und Abisolierautomat ⓘ
 - 17-13-01-02 Abisolierautomat (Elektro) ⓘ
 - 17-13-01-03 Ablängautomat (Elektro) ⓘ
 - 17-13-01-90 Maschine zur Kabelbe- und -verarbeitung (nicht spezifiziert)
 - 17-13-01-91 Maschine zur Kabelbe- und -verarbeitung (Teile)
 - 17-13-01-92 Maschine zur Kabelbe- und -verarbeitung (Zubehör)

Abbildung 2-6: ECLASS-Klassifizierungen für Produktionsmaschinen [8]

Eine weitere Prüfung fand TP6 "Automatisierte Verhandlungsprozesse" statt. Dort wurde nach semantischen Referenzen für ein I4.0-Vokabular recherchiert, damit es sowohl von Menschen als auch Resourcen interpretiert werden kann. Es wurde keine derzeit ausreichende Überdeckung mit dem im Projekt benötigten Vokabular festgestellt. Um unternehmens- bzw. industrieübergreifende-Interoperabilität im Projektkontext zu erreichen, wurden daher die bereits vorhandenen semantischen Definitionen der anderen Standards als ausreichend betrachtet (OPC UA und VEC). Als weitere Quelle für semantische Referenzen wäre prinzipiell auch die [DIN 72036](#) [13] geeignet, jedoch müsste hierfür noch eine eindeutige ID-Bildung definiert werden, z.B. in Form von IRIs (Internationalized Resource Identifier nach [RFC 3305](#)).

Erwähnenswert in diesem Zusammenhang ist auch das „Common Data Dictionary“ der IEC [14], worin ebenfalls tendenziell für den Leitungssatz verwendbare Klassifizierungen definiert sind, z.B. „[0112/2//61360 4#AAA548 \(car plug\)](#)“² und „[0112/2//61360 4#AAA611 \(connector tool\)](#)“³.

² https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TU0/0112-2--61360_4%23AAA548

³ https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TU0/0112-2--61360_4%23AAA611

2.2.3 RAMI 4.0

Das Referenzarchitekturmodell Industrie 4.0 (kurz RAMI 4.0) ordnet verschiedene Aspekte der Produktion in einem einzigen Modell, bestehend aus einer mehrdimensionalen Struktur, die verschiedene Ebenen, Lebenszyklusphasen und Hierarchieebenen umfasst, um die Komplexität der Industrie 4.0 übersichtlich abzubilden. Es umfasst eine Kommunikationsebene, die Lebenszyklen von Anlagen und Produkten sowie Automatisierungs- und IT-Ebenen. RAMI 4.0 ist ein kubisches Schichtenmodell, das in der [DIN SPEC 91345](#) [15] definiert ist. Die Dimensionen des Würfels beschreiben die Architektur von Assets, ihren Lebenszyklus und ihre Zuordnung zu Hierarchieebenen [9].

RAMI 4.0 ist gut geeignet als übergeordneter Referenzrahmen für die Klassifikation der zu erstellenden Lösungsbestandteile im VWS4LS-Projekt. Es wurde jedoch entschieden, es nicht im Projekt zu berücksichtigen, da damit personelle Ressourcen gebunden, aber kein direkter anwendungsbezogener Mehrwert entstehen würde.

2.2.4 OPC UA

OPC UA (Open Platform Communications Unified Architecture) ist ein Standard für eine Kommunikationsarchitektur, die für den Datenaustausch als plattformunabhängige, service-orientierte Architektur (SOA) entwickelt wurde und nicht nur den Transport von Maschinendaten, wie Regelgrößen, Messwerte, Parameter etc. ermöglicht, sondern diese auch maschinenlesbar semantisch zu beschreiben. Dabei werden nicht nur Betriebsdaten berücksichtigt, sondern auch Spezifikationen und Fähigkeiten der beteiligten Maschinen. Das ermöglicht die Kommunikation zwischen Produktionssteuerungssystemen wie MES (Manufacturing Execution Systems) und den Produktionsanlagen [16]. Die OPC UA Technologie bietet [grundlegende Bausteine](#) [7] eines klassischen Kommunikationsframeworks, die eine einheitliche Basis für die Kommunikation und den Datenaustausch bereitstellen und eine einheitliche Beschreibung und Kommunikation zwischen verschiedenen Systemen ermöglicht. Einer dieser Bausteine ist bspw. das Informationsmodell des Typenschildes ([Machinery Nameplate](#)) [17], um Informationen über die Maschine, ihre Funktionen und Eigenschaften bereitzustellen zu können.

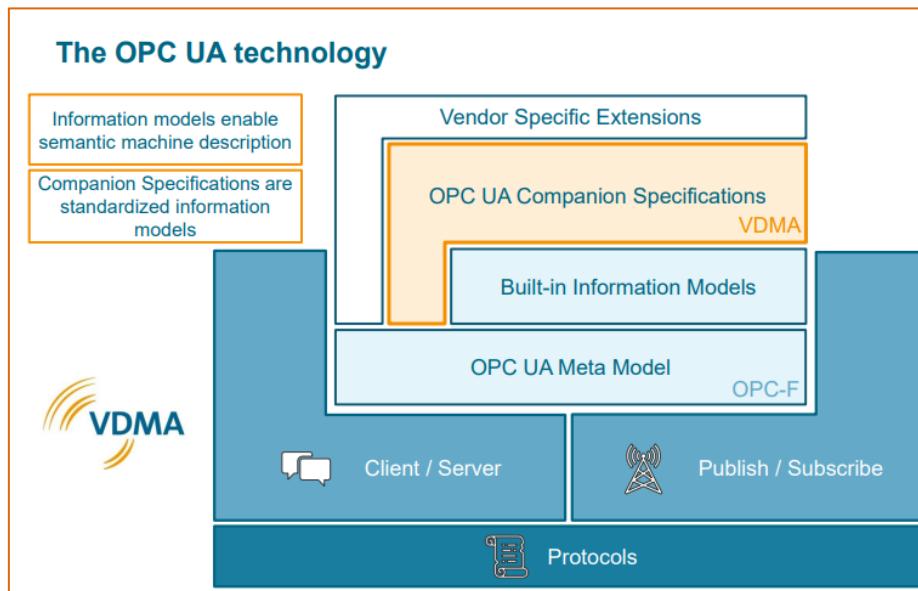


Abbildung 2-7: OPC UA Technologie (Quelle: VDMA)

Im Bereich der Leitungssatzfertigung gab es bislang keine spezifische Standardisierung für OPC UA. Da die Anforderungen an die Beschreibung von Leitungssätzen komplex und spezifisch sind, würden [die verfügbaren allgemeinen Companion Specifications](#) [7] nur bedingt ausreichen. Deshalb wurde im Projekt die Entscheidung getroffen, eine Companion Specification speziell für die Leitungssatzfertigung zu entwickeln.

Diese neue Companion Specification «OPC 40570: OPC UA for the Wire Harness Manufacturing Industry» [18] wird die spezifischen Anforderungen der Branche berücksichtigen und eine einheitliche Kommunikation zwischen den beteiligten Akteuren ermöglichen.

Insgesamt ist OPC UA ein zentrales Element der Digitalisierung in der Fertigungsindustrie, und die Entwicklung einer spezifischen Companion-Spezifikation für die Leitungssatzfertigung wird die Interoperabilität und Effizienz weiter verbessern.

2.2.5 Fazit

Im Projekt wurden wesentliche Schlüsselentscheidungen getroffen, um die Interoperabilität und Effizienz innerhalb der Leitungssatzfertigung zu verbessern:

Unterstützung von VEC und KBL: VEC bietet im Vergleich zu KBL eine fortschrittlichere Architektur und ermöglicht eine umfassendere Produktbeschreibung, einschließlich Stücklisten und Verbindungsinformationen. Trotzdem wurde beschlossen, neben dem VEC- auch das KBL-Datenmodell zu unterstützen, da es bei mehreren OEMs etabliert ist und von bestehenden Systemen unterstützt wird.

OPC UA als Kommunikationsstandard: Während VEC und KBL als Modell für die statische Produktbeschreibung dient, soll OPC UA als Kommunikationsstandard die dynamische Interaktion der verschiedenen Systeme in der Leitungssatzfertigung ermöglichen, einschließlich MES (Manufacturing Execution Systems) und Produktionsanlagen. Außerdem soll OPC UA die Grundlage für die Prozess- und Resourcendatenmodelle bilden.

Entwicklung einer Companion Specification: Da es keine spezifische Standardisierung für die Leitungssatzfertigung in OPC UA gab, wurde beschlossen, eine Companion-Specification in Zusammenarbeit mit dem VDMA zu entwickeln. Diese Spezifikation wird die spezifischen Anforderungen der LS-Branche berücksichtigen und eine einheitliche Kommunikation ermöglichen.

Fokus auf den Schneidraum: Eine weitere Entscheidung bestand darin, den Fokus auf die Produktionsprozesse im Schneidraum zu legen. Hier werden die Kabelbäume zugeschnitten und vorbereitet, bevor sie in die Montage gehen.

VWS-Datenmodell für den Leitungssatz: VEC und KBL-Dateien sowie OPC UA Nodeset-Dateien sollen unverändert in die VWS eingehängt bzw. verlinkt werden.

Insgesamt zielen diese Entscheidungen darauf ab, die Leitungssatzfertigung effizienter und flexibler zu gestalten, um den Anforderungen der modernen Automobilindustrie gerecht zu werden.

2.3 AP 1.3 - Erfassen der Geschäftsmodelle für das Informationsmodell

Im Arbeitspaket 1.3 „Erfassen der Geschäftsmodelle für das Informationsmodell“ sollen die bestehenden Geschäftsmodelle und ihre wirtschaftlichen Auswirkungen untersucht werden, um festzustellen, wie sie effektiv in das Informationsmodell integriert werden können. Dabei werden sowohl kurzfristige als auch langfristige wirtschaftliche Aspekte berücksichtigt, um sicherzustellen, dass das Modell nachhaltig und zukunftsorientiert ist.

→ Es wurde die Entscheidung getroffen, AP 1.3 in das TP 7 zu überführen, um eine fokussierte und spezialisierte Behandlung des Themas Geschäftsmodelle zu gewährleisten, in enger Abstimmung mit den Anforderungen des Informationsmodells.

2.4 AP 1.4 - Beschreibung der Tiefe der Informationsmodelle

Das Arbeitspaket 1.4 „Beschreibung der Tiefe der Informationsmodelle“ beschäftigte sich mit der detaillierten Beschreibung von Informations- und Datenmodellen für die Leitungssatzfertigung.

Ein modellbasiertes Engineering erfasst das Fachwissen durch Designregeln, um die Qualität des Produktes zu verbessern, die Effizienz der Produktion zu steigern, sowie den automatisierten Datenaustausch zu unterstützen. Das digitale Leitungssatz-Modell soll daher als Grundlage dienen, um automatisiert detaillierte Arbeitsanweisungen zu generieren und eine präzise und effiziente Produktion in der Leitungssatzfertigung zu ermöglichen.

2.4.1 Grundlagen

Die Entwicklung des Leitungssatzes erfolgt durch den OEM, typischerweise unter Verwendung der etablierten Modellierungsstandards KBL [5] und VEC [6].

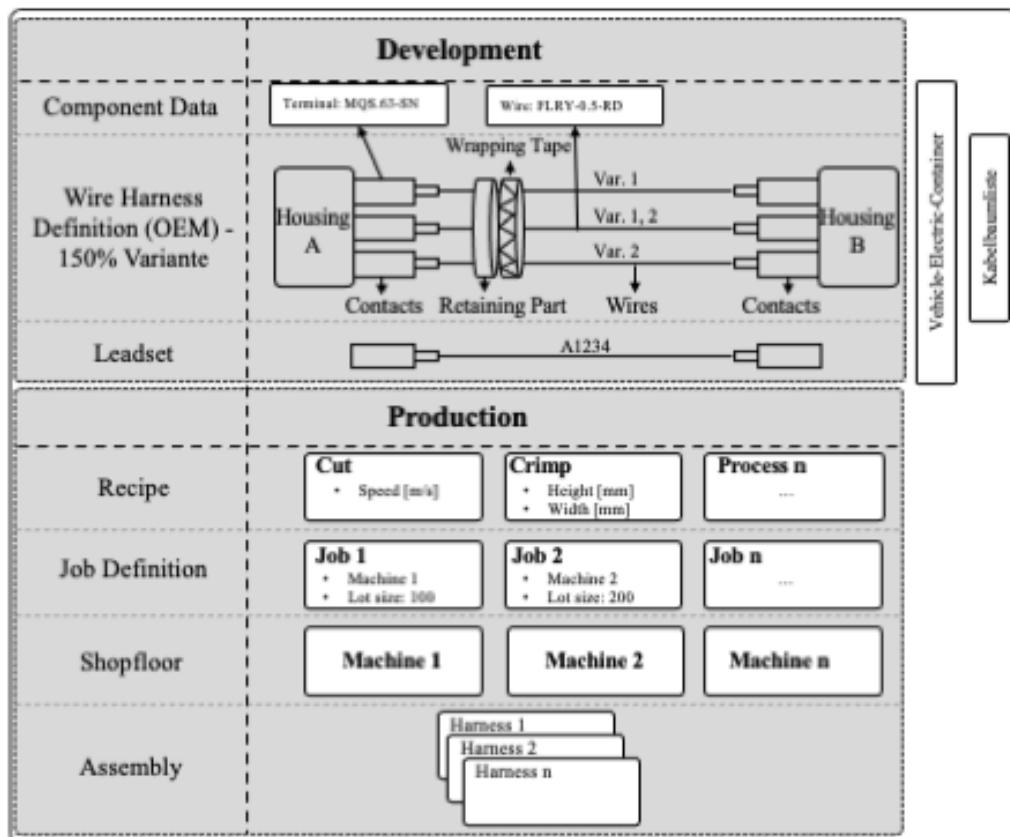


Abbildung 2-8: Leitungssatzarchitektur

Der vom OEM auf dieser Basis bereitgestellte sog. „150%-Leitungssatz“ umfasst alle möglichen Varianten für den Leitungssatz einzelner Fahrzeugmodelle. Berücksichtigt wird dabei nicht nur die Standardkonfiguration, sondern auch zusätzliche Optionen und Ausstattungsvarianten. Diese umfassen unterschiedliche Motorisierungen, Sonderausstattungen, Länderanforderungen und individuelle Kundenwünsche. Das Ergebnis ist ein umfassendes Modell, das mehr Variationen umfasst, als jemals in einem einzigen real gefertigten Leitungssatz (in Abgrenzung dazu auch als sog. „100% Leitungssatz“ bezeichnet) enthalten sein können.

Diese Vielfalt führt zu der Notwendigkeit, für reale Modellkonstellationen dedizierte Produktionsaufträge ausleiten zu können. Jedes Fahrzeugmodell erfordert spezifische Leitungssatzvarianten, die auf individuellen Konfigurationen basieren. Diese müssen präzise hergestellt werden, um sicherzustellen, dass alle Komponenten korrekt verdrahtet sind.

Der Produktionsprozess besteht aus folgenden wesentlichen Bestandteilen:

1. Leitungssatzentwurf durch den OEM:

- Der OEM erstellt das Leitungssatzdesign mithilfe von CAD-Systemen und CAE-Tools.

2. Extraktion der Produktionsinformationen:

- Der Leitungssatzhersteller extrahiert aus dem OEM-Design die Informationen, die für die Herstellung jedes Leitungssatzes erforderlich sind. Dies umfasst Details zu Steckern, Kabeln, Längen, Verbindungen und Positionen.
- Die Daten werden in einem strukturierten Format gespeichert, um die Produktion effizient zu steuern.

3. Verteilung der Produktionsaufträge:

- Der Hersteller verteilt die Produktionsaufträge auf die jeweiligen Maschinen für die verschiedenen Fahrzeugmodelle.
- Jeder Leitungssatz wird gemäß den spezifischen Anforderungen für das jeweilige Fahrzeugmodell hergestellt.

4. Senden eines Produktionsauftrags an eine Maschine:

- Die Maschinen erhalten die erforderlichen Informationen, z. B. Kabeltyp, Länge, Verbindungen und Positionen.
- Übermittlung der Produktionsaufträge an die Maschinen mittels OPC UA.

5. Datenerfassung über die produzierte Charge:

- Während der Produktion werden Daten wie Produktionszeit, Materialverbrauch und Qualitätsprüfungen erfasst.
- Der digitale Zwilling des Produkts (VWS) wird mit diesen Daten angereichert, zentral abgelegt gespeichert und kann für Rückverfolgbarkeit und Qualitätskontrolle genutzt werden.

6. Montage und Endprüfung des Leitungssatzes:

- Nach der Produktion der Einzelkomponenten werden die Leitungssätze montiert und auf ihre korrekte Verdrahtung überprüft, um die Qualität sicherzustellen und mögliche Fehler zu identifizieren.
- Endprüfung mit elektrischen Tests, Widerstandsmessungen und Funktionsüberprüfungen.
- Anreicherung des digitalen Zwillings des gefertigten Endprodukts um diese Qualitätssicherungsdaten.

2.4.2 Prozessliste und Parameter

Im TP 3 wurden verschiedene Analyseaktivitäten durchgeführt, um eine Liste der Produktionsprozesse und ihrer Parameter zu erstellen. Schließlich wurden spezifische Kategorien erstellt, um diese Parameter zu gruppieren und den Produktionsprozess effizient zu gestalten. Es wurde ein detailliertes Prozess- und Parametermodell definiert, um die verschiedenen Produktionsprozesse und Parameter effizient und einheitlich verwälten zu können. Hierfür wurde zunächst jedem identifizierten Produktionsprozess eine eindeutige Bezeichnung sowie eine Beschreibung hinterlegt. Anschließend wurde jedem Prozess ein verantwortlicher Konsortialpartner zugewiesen, dessen Aufgabe darin bestand, die relevanten Prozessparameter nach Folgendem Schema zu definieren:

- **Name:** Ein eindeutiger und leicht verständlicher Name für den Parameter.
- **Einheit:** Die Maßeinheit des Parameters (Meter, Kilogramm, Sekunden, etc.).
- **Anforderungsgrad:** Definiert, inwiefern ein Parameter für den jeweiligen Prozess erforderlich (mandatorisch) oder optional ist.
- **Datentyp:** Der Datentyp des Parameters (Zahl, Text, Boolean, etc.)
- **Beschreibung:** Eine kurze Beschreibung des Parameters, die dessen Bedeutung und Verwendungszweck im Kontext des Produktionsprozesses erklärt.

Die so angereicherten Parameterdefinitionen wurden in die Prozessparameterliste übertragen und von allen Konsortialpartnern überprüft und validiert, um sicherzustellen, dass alle Definitionen korrekt und konsistent sind.

Nach der Validierung und Finalisierung wurden alle erfassten Prozesse und Prozessparameter an den VDMA zur Integration in die Companion Specification «OPC UA for the Wire Harness Manufacturing Industry» (VDMA 40570) [18] weitergegeben.

2.4.3 Produktionsdatensatz

Ein Datensatz für die Produktion besteht aus 5 Prozessdatenbausteinen (siehe Abbildung 2-9). Prozesse können von der Montage über das Crimpen bis hin zu Tests und Qualitätskontrollen reichen. Jeder dieser Prozesse muss klar definiert und dokumentiert sein, um eine effiziente und fehlerfreie Produktion zu gewährleisten. Nach Abschluss der Produktion wird für jeden Prozess ein Satz von Resultaten zurückgeliefert, welche die Qualität und Genauigkeit jedes einzelnen Schrittes im Produktionsprozess widerspiegeln.

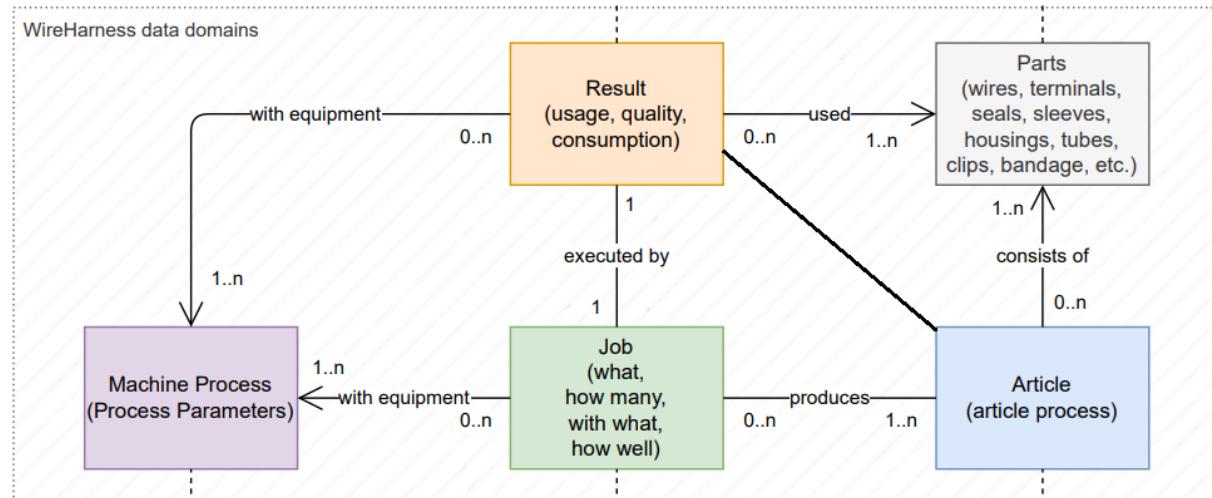


Abbildung 2-9: WireHarness Data Domains

Um eine Arbeitsauftragsanforderung (Job Order Request) für einen Fertigungsauftrag zusammenzustellen (Abbildung 2-10), muss als Voraussetzung eine Artikelspezifikation gemäß dem VEC-Standard vorliegen, welcher eine standardisierte Beschreibung von Fahrzeugelektronikkomponenten (*Part Info*) bereitstellt und somit die Kommunikation und den Datenaustausch zwischen verschiedenen Systemen und Partnern in der Automobilindustrie ermöglicht.

Zusätzlich zur Artikeldefinition muss die Spezifikation eine Liste von Teilen (*Part*) enthalten, die für die Produktion benötigt werden. Dazu gehören beispielsweise Anschlüsse, Kabel, Dichtungen und weitere relevante Komponenten. Das VEC-Datenmodell wird als Grundlage für die Beschreibung von *Article* und *Part* genutzt und in den OPC UA Datentypen überführt.

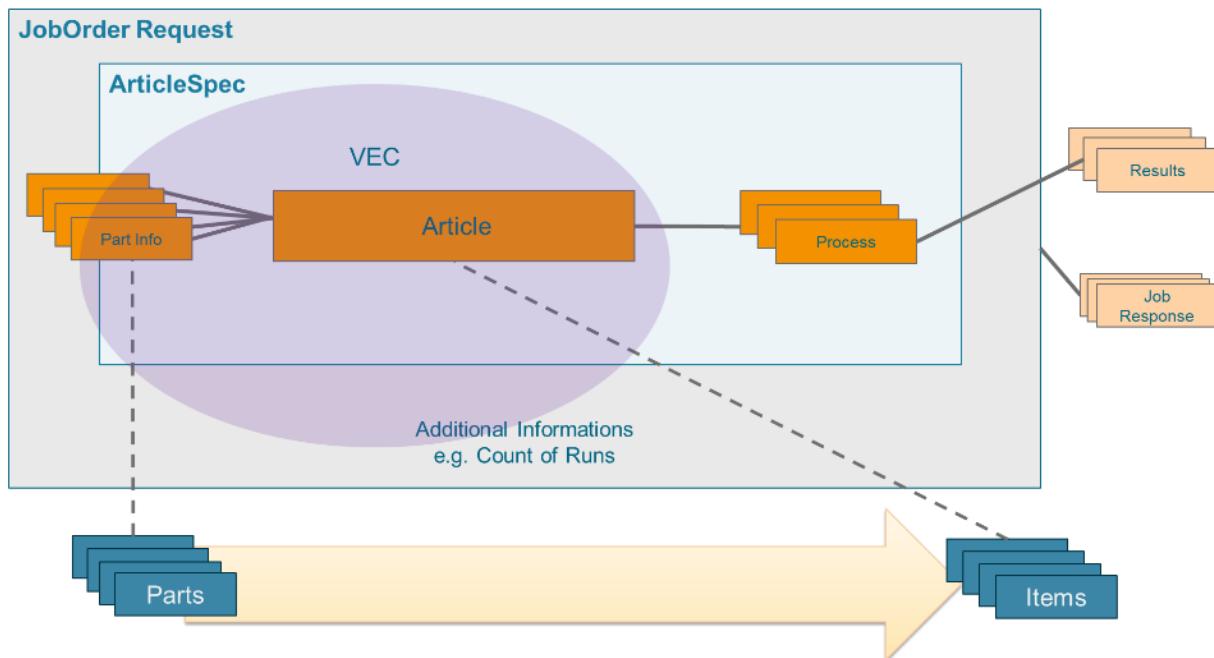


Abbildung 2-10: Zusammenhänge im Produktionsdatensatz [18]

Neben der Teileliste ist auch eine Liste der durchzuführenden Prozesse erforderlich. Ein solcher Datensatz für die Produktion besteht aus den 5 Prozessdatenbausteinen in Abbildung 2-9. Die Prozesse können von der Montage über das Crimpen bis hin zu Tests und Qualitätskontrollen reichen. Jeder dieser Prozesse muss klar definiert und dokumentiert sein, um eine effiziente und fehlerfreie Produktion zu gewährleisten. Nach Abschluss eines Produktionsvorgangs wird für jeden Prozess ein Satz von Resultaten zurückgeliefert. Diese Ergebnisse sind von entscheidender Bedeutung, da sie die Qualität und Genauigkeit jedes einzelnen Schrittes im Produktionsprozess widerspiegeln.

Schließlich wird eine Arbeitsantwort (*JobResponse*) erstellt, die eine Zusammenfassung der Produktionsergebnisse enthält. Diese Zusammenfassung bietet einen Überblick über die durchgeführten Arbeiten und die erreichten Ergebnisse. Sie dient als Nachweis (sog. „Audit Trail“) für die Einhaltung der Spezifikationen und als Qualitätsdokumentation. Die Arbeitsantwort ist somit ein wesentlicher Bestandteil des gesamten Produktionsprozesses und ermöglicht eine kontinuierliche Verbesserung und Optimierung der Arbeitsabläufe.

Im Produktionsprozess entstehen *Items* als konkrete Instanzen eines *Article*, d.h. können eine Seriennummer haben. Diese werden im OPC UA Adressraum als *Items* (Knoten) repräsentiert, welche die eigentlichen Datenpunkte (Attribute) enthalten und somit die Objekte im OPC UA Server abbilden.

2.4.4 VEC als Produktdatenmodell

Abbildung 2-11 skizziert die Nutzung des VEC-Modells im Kontext des Projekts. Jede Komponente, Terminal, Wire, Connector usw. kann mit ihren technischen Attributen im VEC-Modell beschrieben werden. Unter Verwendung dieser Komponenten wird dann eine Artikelbeschreibung aufgebaut, wobei dabei jede einzelne Verwendung greifbar sein muss, da diese einerseits unterschiedliche Eigenschaften erhalten können, vergleiche bspw. die *StrippingLength* in der Grafik, oder die Länge der Leitungen. Anderseits unterscheiden die Verwendung sich im Beziehungsgeflecht, das diese mit anderen Komponenten eingehen. Vergleiche bspw. Terminal „X“ an roter bzw. grüner Leitung.

Diese Artikelbeschreibung stellt dann das Grundgerüste für weitere Informationsinhalte dar. Diese können die Elemente der Artikelbeschreibung aus als Bezugspunkte nutzen. Beispielsweise kann die Definition von Fertigungsprozessen darüber eindeutig definieren für welches Element ein bestimmter Prozess gilt, also welche Leitung geschnitten werden soll, oder welche Leitungsende in einem Schritt abgesetzt werden müssen.

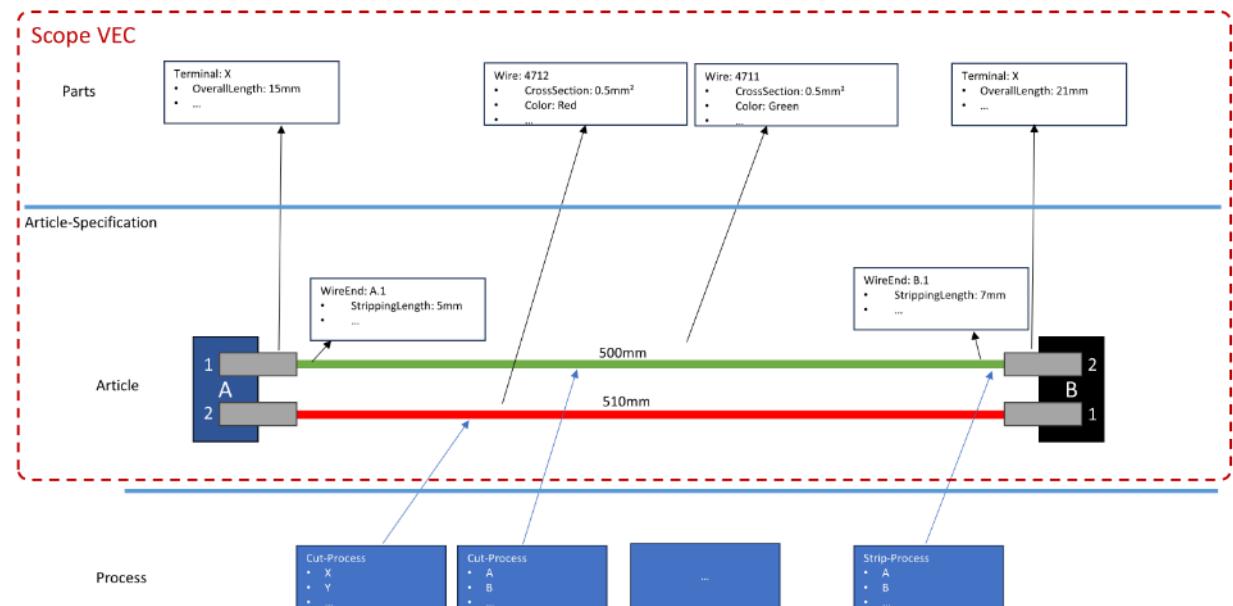
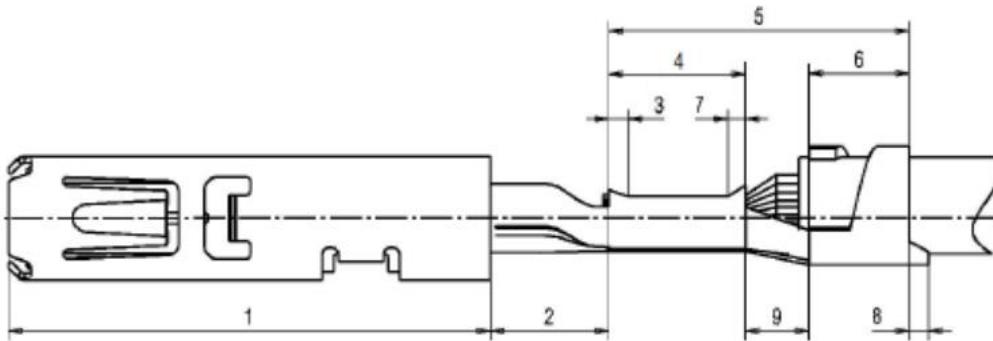


Abbildung 2-11: Nutzung des VEC-Modells

Da der VEC den kompletten Entwicklungsprozess des physischen Bordnetzes unterstützt ist das Modell sehr umfangreich. Für die im Projekt relevanten Anwendungsfälle werden lediglich die Bereiche „Komponentenbeschreibung“ und „Leitungssatzdefinition“ benötigt.

Wie in den vorhergehenden Abschnitten beschrieben besteht ein Datensatz für die Produktion aus 5 Prozessdatenbausteinen (siehe Abbildung 2-9). Die Bausteine „Article“ und „Parts“ liegen dabei im Kern des Scopes des VEC (siehe Abbildung 2-12 mit dem Beispiel eines Kontaktteils).



| TerminalReceptionSpecification (VEC.electrical_parts) | WireReceptionSpecification (VEC.electrical_parts) |
|---|--|
| <p>«Deprecated» + cavityDesign : String [0..1] +platingMaterial : Material [0..*] +primaryLockingType : PrimaryLockingType [0..1] +pullOutForce : NumericalValue [0..1] +testingPullOutForce : NumericalValue [0..1] +contactRangeLength : NumericalValue [0..1]</p> | <p>+coreCrossSectionArea : ValueRange [0..1] +insulationDisplacementLength : NumericalValue [0..1] +multiContact : Boolean [0..1] +wireReceptionType : WireReceptionType [0..1] +wireElementOutsideDiameter : ValueRange [0..1] +platingMaterial : Material [0..*] +sealable : Boolean [0..1] +frontBellMouthLength : NumericalValue [0..1] +conductorCrimpLength : NumericalValue [0..1] +conductorCrimpShape : CrimpShape [0..1] +conductorCrimpLegHeight : NumericalValue [0..1] +rearBellMouthLength : NumericalValue [0..1] +crimpConnectionLength : NumericalValue [0..1] +insulationCrimpLength : NumericalValue [0..1] +insulationCrimpShape : CrimpShape [0..1] +insulationCrimpLegHeight : NumericalValue [0..1] +cutOffTabLength : NumericalValue [0..1] +connectionBLength : NumericalValue [0..1] +sheetThickness : NumericalValue [0..1] +wireTipProtrusion : ValueRange [0..1]</p> |
| TerminalSpecification (VEC.electrical_parts) | |
| <p>+voltageRange : ValueRange [0..1] +sealingType : TerminalSealingType [0..1] +connectionALength : NumericalValue [0..1] +overallLength : NumericalValue [0..1]</p> | |

Abbildung 2-12: Beispiel VEC Terminal Datamodell (Quelle: VEC Recommendation V2.1.)

Eine erste Bewertung der benötigten Parameter (siehe „2.4.2 Prozessliste und Parameter“) ergab eine hohe Abdeckung der benötigten Informationen durch den VEC, und zwar sowohl für den Bereich „Parts“ als auch für den Bereich „Article“.

Eine Artikeldefinition ist mit der KBL ebenfalls grundsätzlich möglich, aber es fehlen verschiedenste Attribute, die im Kontext der Fertigung relevant sind (z.B. Abisolierlängen an Leitungsenden). Da der VEC derartige Attribute bereits enthält, wurde im Weiteren der Fokus auf den VEC gelegt, insbesondere für die Demonstratoren und die OPC UA Integration.

Im Rahmen des Projekts wurden für Parameter, die im VEC-Modell als fehlend identifiziert wurden, eine Reihe von Feature-Requests an den VEC gestellt (siehe Abbildung 2-13 und dort zum heutigen Zeitpunkt auch bereits weitestgehend in der aktuellen Version 2.1.0 umgesetzt [19]):

- Identifikation von Werkzeugen für Komponenten.
- Attribute für Schläuche die in der Verwendung geschlitzt werden.
- Verschiedenste Parameter für die Beurteilung von Crimps (z.B.: Crimp-Höhe & Breite am Kontaktteil parametrisiert, Abzugskräfte, Crimp-Höhe und Breite in einer konkrete Verwendung)
- Parameter zur Behandlung von Leitungsenden.
- Parameter zur automatischen Bestückung von Gehäusen (z.B. Kammerkoordinaten, Einstiektiefe & anwendbare Kräfte).

| Ty... ↑ : | Key | Status | Created ↓ | Updated | Summary |
|-----------|-------------|------------|--------------|--------------|--|
| + | KBLFRM-1244 | OUTLINED ▾ | Jul 26, 2024 | Jul 26, 2024 | CrimpBarrelType an der WireReceptionSpecification |
| + | KBLFRM-1243 | OUTLINED ▾ | Jul 26, 2024 | Jul 26, 2024 | Zusätzliche Parameter für Schlitten und partiellen Abzug |
| + | KBLFRM-1237 | OUTLINED ▾ | Jun 5, 2024 | Jul 26, 2024 | Seal Position auf der Leitung |
| + | KBLFRM-1236 | OUTLINED ▾ | May 29, 2024 | Jul 26, 2024 | Partieller Abzug bei der Abisolierung von Leitungsenden |
| + | KBLFRM-1210 | RESOLVED ▾ | Aug 31, 2023 | Jan 8, 2024 | Added MaximumInsertionForce to PluggableTerminalSpecification |
| + | KBLFRM-1209 | RESOLVED ▾ | Aug 30, 2023 | Jan 8, 2024 | Added TestingPullOutForce to TerminalReceptionSpecification |
| + | KBLFRM-1208 | RESOLVED ▾ | Aug 30, 2023 | Nov 16, 2023 | Defined Coordinate System for Terminals and added WireReception.Rotation |
| ■ | KBLFRM-1205 | RESOLVED ▾ | Jul 26, 2023 | Sep 11, 2023 | Fixed direction of X-Axis in Illustration "Geometric Properties of Connector Housings (Definitions)" |
| + | KBLFRM-1202 | RESOLVED ▾ | Jul 13, 2023 | Jan 8, 2024 | Added multiple detail parameters for automatic crimp processes |
| + | KBLFRM-1201 | DEFINED ▾ | Jul 13, 2023 | May 21, 2024 | Verzinnen von Leitungsenden |
| + | KBLFRM-1200 | RESOLVED ▾ | Jul 13, 2023 | Jan 8, 2024 | Added CrimpSizes to WireMountingDetail |
| + | KBLFRM-1199 | RESOLVED ▾ | Jul 13, 2023 | Jan 8, 2024 | Refined Definition of WireEnd.PositionOnWire, added Stripping- & CutBackLength |
| + | KBLFRM-1194 | RESOLVED ▾ | Jun 1, 2023 | Jul 27, 2023 | Added "ProcessingInstruction" to OpenEnumeration "DocumentType" |
| + | KBLFRM-1177 | RESOLVED ▾ | Nov 8, 2022 | Jan 8, 2024 | Definition of Crimp Width & Height |
| + | KBLFRM-1176 | RESOLVED ▾ | Nov 8, 2022 | Sep 28, 2023 | Added pullOffForce for Crimp-Definitions |
| + | KBLFRM-1175 | RESOLVED ▾ | Nov 8, 2022 | Dec 20, 2023 | Added TubeRoles with Attributes for Slits |
| + | KBLFRM-1084 | RESOLVED ▾ | May 11, 2021 | Sep 28, 2023 | Added Concept of ResourceVersion for Relations to Tools & Machines |

Abbildung 2-13: Feature Requests an den VEC

2.4.5 OPC UA Schnittstelle

OPC UA hat für die unterschiedlichsten Anwendungsfälle eine Reihe von sog. Companion Specifications veröffentlicht [7], von denen einige möglicherweise eine vielversprechende Basis für die im Leitungssatz-Kontext zu lösenden Herausforderungen darstellen (Abbildung 2-14).

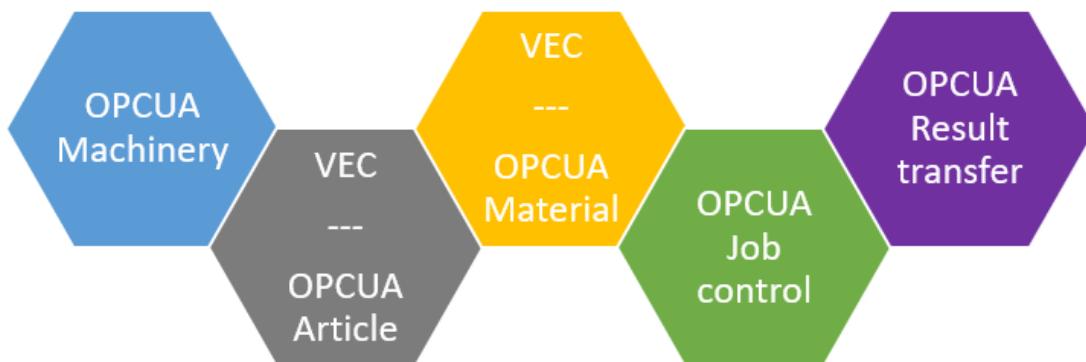


Abbildung 2-14: Relevante OPC UA Spezifikationen

Die folgenden OPC UA Companion Specifications wurden als geeignete Basis für die VWS4LS-Anwendungsfälle identifiziert:

«OPC 40001-1 Machinery Basic Building Blocks» [17]

«OPC 40001-3: Machinery Job Mgmt» [20]

«OPC 40001-101: Machinery Result Transfer» [21]

«OPC 10031-4: ISA-95-4 Job Control» [22]

Für die in Abbildung 2-14 aufgeführten fiktiven «OPC UA Article» und «OPC UA Material» wurden in der Folge dedizierte Lösungen erarbeitet.

Für eine effektive Kommunikation mit der Maschinerie über eine OPC UA-Schnittstelle müssen bestimmte Datencontainer spezifiziert werden:

Identification: Enthält Informationen zur Identifizierung der Leitungssatzmaschine.

MachineryItemState: Zeigt den aktuellen Zustand der Leitungssatzmaschine an.

MachineryOperationMode: Gibt die Art der von der Maschine durchgeföhrten Aufgaben an.

JobManagement: Beinhaltet Informationen und Methoden, die sich auf die Aufträge beziehen. Dies umfasst die Zustände der aktuellen Aufträge und Methoden zum Einfügen, Entfernen, Freigeben und Unterbrechen eines Auftrags.

ResultManagement: Ist der Einstiegspunkt für das Ergebnismanagement, d.h. die Behandlung der Arbeitsantworten (JobResults).

MaterialManagement: Enthält Informationen und Methoden, die sich auf das Material beziehen (z.B. materialbezogene Prozesswerte). Dies umfasst die Zustände des aktuellen Materials, das von der Maschine bekannt ist, und Methoden zum Einfügen und Entfernen von Material.

ArticleManagement: Beinhaltet Informationen und Methoden, die sich auf den Artikel beziehen (z.B. materialbezogene Prozesswerte). Dies umfasst die Zustände des aktuellen Materials, das von der Maschine bekannt ist, und Methoden zum Einfügen und Entfernen eines Artikels.

2.4.5.1 Identifikation

Die OPC UA Companion Specification «OPC 40001-1 Machinery Basic Building Blocks» [17] dient zur Kommunikation zwischen Maschinen (M2M) und mit übergeordneten Systemen der Industrie 4.0 (z.B. MES) und unterstützt die folgenden Anwendungsfälle:

Maschinenidentifikation und Typenschild: Jede Maschine hat eine eindeutige Identifikation und ein Typenschild, die in einem OPC UA Server repräsentiert werden. Diese Informationen enthalten wichtige Details wie den Maschinentyp, das Modell, den Hersteller, das Herstellungsdatum und andere spezifische Informationen, die eine genaue Identifizierung und Klassifizierung der Maschine ermöglichen.

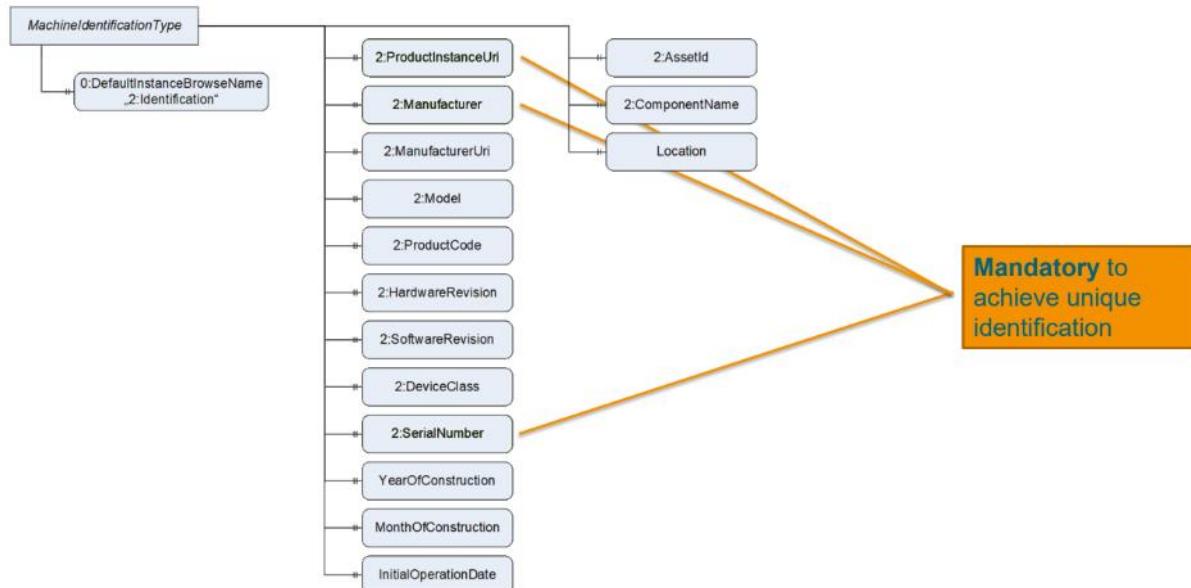


Abbildung 2-15: `MachineIdentificationType`. (Quelle: OPC UA for Machinery [17])

Finden aller Maschinen in einem Server: OPC UA ermöglicht es, alle Maschinen in einem Server zu finden. Dies ist besonders nützlich in großen industriellen Umgebungen, wo Hunderte oder sogar Tausende von Maschinen vorhanden sein können. Durch die Verwendung von OPC UA können Benutzer schnell und effizient alle Maschinen in einem Server lokalisieren und identifizieren.

Komponentenidentifikation und Typenschild: Ähnlich wie bei der Maschinenidentifikation hat auch jede Komponente einer Maschine eine eindeutige Identifikation und ein Typenschild. Diese enthalten Informationen wie den Komponententyp, das Modell, den Hersteller und andere spezifische Details. Dies ermöglichen eine genaue Identifizierung und Klassifizierung der Komponenten.

Finden aller Komponenten einer Maschine: Mit OPC UA können Benutzer alle Komponenten einer bestimmten Maschine finden. Dies ist besonders nützlich für die Wartung und Fehlerbehebung, da es ermöglicht, schnell die spezifischen Komponenten zu identifizieren, die gewartet oder ersetzt werden müssen.

Maschinenüberwachung: Durch die kontinuierliche Überwachung der Maschinenleistung und des Betriebszustands mittels OPC UA können Benutzer potenzielle Probleme frühzeitig erkennen und proaktive Massnahmen ergreifen, um Ausfallzeiten zu vermeiden und die Effizienz zu verbessern.

| Applicability: OPC UA Server running | |
|--------------------------------------|---|
| Not available | The unit is not available and does not perform any activity*. (e.g. Switched Off, in Energy Saving Mode) |
| Out of Service | The unit is not functional and does not perform any activity*. (e.g. Error, Blocked) |
| Not Executing | The unit is available & functional and does not perform any activity*. It waits for an action from outside to start or restart an activity*. |
| Executing | The unit is available & functional and is actively performing an activity* (pursues a purpose) |

* activity = part of the production, preparation or maintenance process

Abbildung 2-16: MachineryItemState. (Quelle: OPC UA for Machinery [17])

2.4.5.2 Job Management

Die OPC UA Companion Specifications «**OPC 10031-4: ISA-95-4 Job Control**» [22] und «**OPC 40001-3: Machinery Job Mgmt**» [20] ermöglichen eine standardisierte Verwaltung von Arbeitsaufträgen innerhalb von Maschinen und Anlagen (vgl. Abbildung 2-10). Im Folgenden sind einige Kernelemente und deren Zusammenhänge beschrieben (Abbildung 2-17).

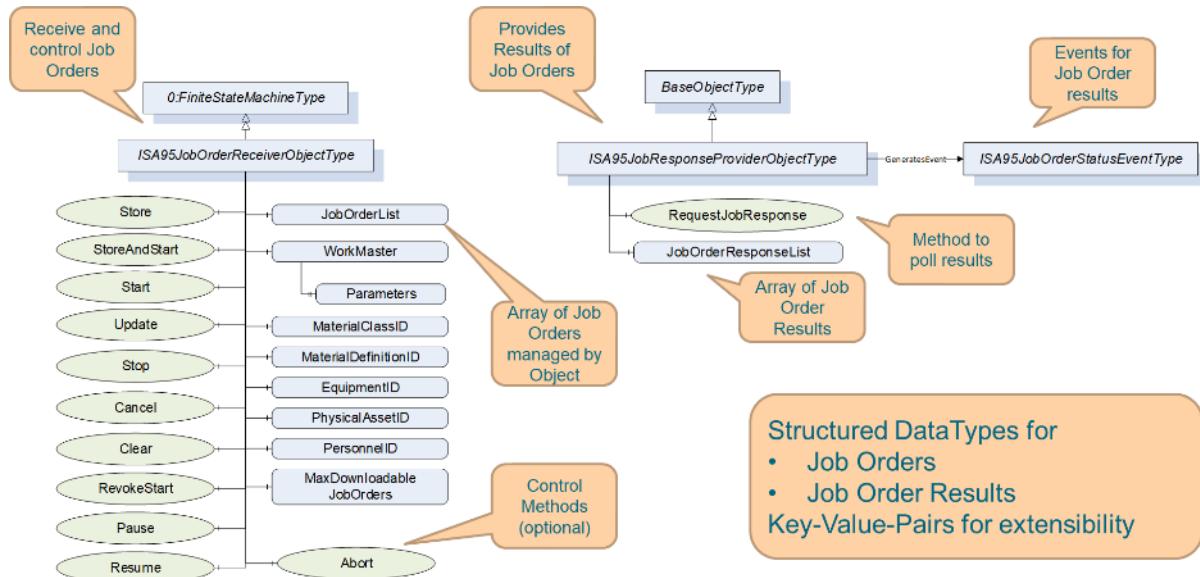


Abbildung 2-17: Relevante OPC UA Informationsmodelle [17] [20] [21] [22]

2.4.5.2.1 ISA95JobOrderReceiverObjectType

Als Kernelement enthält *ISA95JobOrderReceiverObjectType* die Methoden zur Entgegennahme von Arbeitsaufträgen und optionale Definitionen zulässiger Arbeitsauftragsinformationen.

Dieser Objekttyp definiert die Metadaten der Zustandsmaschine für die verwalteten Arbeitsaufträge. Arbeitsaufträge werden dem System hinzugefügt, indem die Methoden *Store* oder *StoreAndStart* aufgerufen werden, woraufhin der Arbeitsauftrag in den Zustand *NotAllowedToStart* oder *AllowedToStart* versetzt wird. In beiden Zuständen wird der Arbeitsauftrag noch nicht ausgeführt. Durch Aufrufen der *Update*-Methode kann der Arbeitsauftrag noch geändert werden. Der Start eines Arbeitsauftrags wird durch die *Start*-Methode ausgelöst, die den Zustand von *NotAllowedToStart* zu *AllowedToStart* wechselt.

2.4.5.2.1.1 JobOrderList

Die *JobOrderList* definiert eine schreibgeschützte Liste von Jobauftragsinformationen, die vom Server verfügbar sind. Die zurückgegebenen Informationen werden vom Server definiert und können eine vollständige Kopie eines empfangenen Jobauftrags sein oder lediglich die IDs und den Status der Jobaufträge enthalten. Sollten keine Jobaufträge vom Server verfügbar sein, kann dieser Wert auch null sein.

2.4.5.2.1.2 WorkMaster

Der *WorkMaster* definiert einen schreibgeschützten Satz von *WorkMaster*-IDs, die in einem Jobauftrag angegeben werden können, sowie den schreibgeschützten Satz von Parametern, die für einen bestimmten *WorkMaster* festgelegt werden können. Falls es keine Einschränkungen gibt, welche *WorkMaster*-IDs mit einem Jobauftrag gesendet werden können, kann dieser Wert ebenfalls *NULL* sein.

Rezepte, die mit dem Jobauftrag verbunden sind, werden unter Verwendung der *WorkMasterID* identifiziert. Die ID im *ISA95WorkMasterDataType* ist eine Identifikation des Work Masters und kann interne Identifikationen, OPC UA Node-IDs, Web-Links oder ein zwischen Client und Server vereinbartes Format sein. Diese Strukturen ermöglichen eine flexible und effiziente Verwaltung von Arbeitsaufträgen und Ressourcen in industriellen Automatisierungssystemen und unterstützen die Integration von Produktionsprozessen mit der Unternehmens-IT.

2.4.5.2.2 ISA95JobResponseProviderObjectType

Ein weiterer wichtiger Bestandteil ist der *ISA95JobResponseProviderObjectType*, welcher eine Methode zur Entgegennahme von unaufgeforderten Arbeitsantwortanfragen beinhaltet. Dies ermöglicht es, auf eine Arbeitsauftragsanfrage zu reagieren und eine entsprechende Antwort zu generieren. Die Kombination des *ISA95JobOrderReceiverObjectType* mit der *ReceiveJobOrder*-Methode und des *ISA95JobResponseProviderObjectType* mit der *RequestJobResponse*-Methode bildet eine Transaktion, die es ermöglicht, einen Arbeitsauftrag zu senden und nach einer Arbeitsantwort zu fragen.

Die Spezifikation definiert auch den *ISA95JobOrderStateMachineType*, der die Untertypen der Zustände *NotAllowedToStart*, *AllowedToStart*, *Ended* und *Interrupted* definiert. Diese Zustände und Übergänge sind entscheidend für das Verständnis, wie Arbeitsaufträge im System hinzugefügt, ausgeführt und entfernt werden. Die Zustandsmaschine bietet eine klare Struktur, die es ermöglicht, den Lebenszyklus eines Arbeitsauftrags von der Erstellung bis zur Fertigstellung zu verfolgen.

2.4.5.3 Material und Artikel Management

Teile und Artikel innerhalb dieser Begleitspezifikation setzen sich aus Strukturen von zwei verschiedenen Quellen zusammen: dem ISA-95 Materialmodell, welches für die allgemeine Verwendung vorgesehen ist, und dem domänenspezifischen VEC (Vehicle Electric Container) Modell (vgl. 2.4.4). Diese wurden in die Begleitspezifikation integriert. Folglich bestehen sowohl Teil als auch Artikel immer aus zwei Segmenten: der generischen Komponente, die vom ISA-95-Modell abgeleitet ist, und dem Leitungssatzspezifischen Element aus dem VEC-Modell.

Das ISA-95 Materialmodell bietet eine umfassende Struktur zur Darstellung von Materialien und deren Eigenschaften, die in verschiedenen Industrien und Anwendungen genutzt werden können. Es ermöglicht eine einheitliche Beschreibung von Materialien, unabhängig von der spezifischen Anwendung oder dem Sektor. Dieses Modell ist besonders nützlich, um die Kompatibilität und Interoperabilität zwischen verschiedenen Systemen und Organisationen zu gewährleisten.

Das VEC-Modell hingegen ist speziell auf die Anforderungen der Fahrzeugelektrik und des Leitungssatzes zugeschnitten. Es berücksichtigt die einzigartigen Eigenschaften und Anforderungen, die mit der elektrischen Verkabelung von Fahrzeugen verbunden sind. Durch die Integration dieses Modells in die Begleitspezifikation wird sichergestellt, dass die spezifischen Bedürfnisse dieses Bereichs angemessen adressiert werden.

Durch die Zweiteilung der Struktur in ein generisches und ein spezifisches Segment können Anwender der Begleitspezifikation die Vorteile beider Modelle nutzen.

Um die Informationsbausteine „Parts“ und „Article“ aus dem VEC-Modell für die OPC UA Schnittstelle zu verwenden, muss keine neue Modellierung für diese Aspekte entwickelt, sondern kann die Modellierung des VECs wiederverwendet werden. Die OPC UA definiert allerdings eine eigene Beschreibungssprache für Daten, das sog. „OPC UA Nodeset“ [23]. Um hier eine gute Integration zu erreichen und Entwickler von OPC UA Schnittstellen davon zu entlasten zwei unterschiedliche Datenformate verstehen zu müssen, erfolgte die Integration des VEC-Models nicht mittels der veröffentlichten XML Schema Repräsentation. Stattdessen wurde eine Transformation in Form eines XSLT-Skripts entwickelt [24], die das als XML vorliegende VEC-Model gemäß definierter Regeln in ein OPC UA Nodeset umwandelt. Gleichzeitig wird dabei das Modell auf die für den Scope der OPC UA Schnittstelle relevanten Klassen heruntergefiltert.

2.4.5.4 ResultManagement

Im speziellen Kontext von «OPC 10031-4: ISA-95-4 Job Control» [22] werden Ergebnisse von Produktionsprozessen erfasst, überwacht und analysiert. Dies umfasst Aspekte wie Qualitätsprüfungen, Materialdaten, Produktionsstatistiken und vieles mehr. Dazu gibt es zwei Arten von Prozessparametern

- Die sog. **Validierung** überprüft stichprobenhaft (z.B. einmal am Anfang einer zu fertigenden Charge), ob ein Produkt den Anforderungen entspricht, da manche Validierungsmethoden zerstörerisch sind, so dass die geprüften Teile nicht weiterverwendet werden können. Ein Beispiel hierfür ist die Crimpzugkraft, die sicherstellt, dass eine Verbindung dauerhaft hält.
- Die sog. **Überwachung** hingegen ist eine fortlaufende Qualitätssicherung, die für jedes produzierte Exemplar sicherstellt, dass die Produktionsstandards eingehalten werden. Die Crimpkraftanalyse („Crimp Force Analysis“, CFA) ist ein Beispiel für die fortwährende Überwachung einer Prozesseigenschaft bei der Herstellung von Crimpverbindungen.

Beide Prozesse sind entscheidend, um die Sicherheit und Zuverlässigkeit der Produkte zu gewährleisten.

Ein typischer Variablenbedarf wird wie folgt exemplarisch ermittelt: Vor der Produktion einer Charge von Kabeln wird ein Validierungsprozess durchgeführt, um sicherzustellen, dass die Spezifikationen eingehalten werden. Dieser Prozess beinhaltet die Überprüfung der Crimphöhe gemäss den festgelegten Variablen in der Artikelbeschreibung

- In der Artikelspezifikation wird eine Variable *CheckCrimpHeight* festgelegt, die bestimmt, dass die Crimphöhe überprüft wird.
- Die Spezifikation der Teile enthält die Maße und Toleranzen für die Produktion, z.B. die *NominalCrimpHeight*.
- Eine dritte Variable mit dem Namen *ActualCrimpHeight* wird verwendet, um die an einem bestimmten Kontaktpunkt gemessenen Werte zurückzugeben.

Nach der Validierung werden die tatsächlichen Werte für die Crimphöhen durch die Variable *ActualCrimpHeight* erfasst und zur weiteren Analyse zurückgegeben.

«OPC 40001-101 Machinery Result Transfer» [21] befasst sich mit dieser Übertragung von Ergebnissen und bietet Mechanismen zur Verwaltung und Übertragung von Produktionsdaten.

2.4.5.5 Wire Harness Manufacturing

Dieser Abschnitt beschreibt die Entstehung der OPC UA Companion Specification "OPC 40570: OPC UA for Wire Harness Manufacturing" [18].

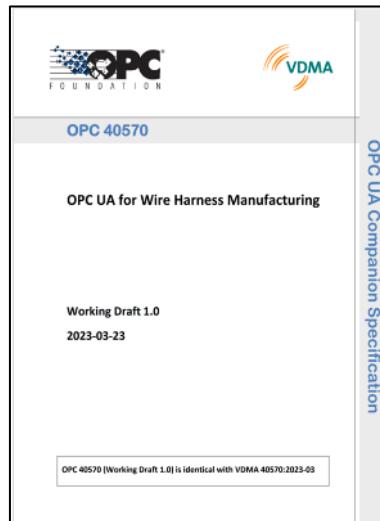


Abbildung 2-18: OPC40570: Wire Harness Manufacturing

Als in TP 1 erkannt wurde, dass eine Verallgemeinerung und Standardisierung der Maschinenschnittstellen von entscheidender Bedeutung sind, wurde mit Unterstützung des VDMA (Verband Deutscher Maschinen- und Anlagenbauer) die OPC UA Arbeitsgruppe 'Wire Harness Manufacturing' ins Leben gerufen.

In ersten Schritten wurde wie oben bereits beschrieben bestehende OPC UA-Standards betrachtet, die auch in der Leitungssatz-Branche Anwendung finden könnten. Dies sollte eine erneute Standardisierung von bereits vorhandenen Standards verhindern. Folgende existierende Standards wurden daher als Grundlage einbezogen:

- «OPC 40001-1 Machinery Basic Building Blocks» [17]
- «OPC 40001-3: Machinery Job Mgmt» [20]
- «OPC 40001-101: Machinery Result Transfer» [21]

Basierend auf diesen bestehenden Standards wurde die Integration der [Prozessliste](#) analysiert und die relevanten Prozesse für die Version 1.0 diskutiert und festgelegt. Dies war notwendig, da der bestehende Detailgrad der Prozessliste zu umfangreich für eine erste Version war. In der Version 1.0 der Companion Specification sind daher folgende Prozesse zu finden:

- Cut
- Strip
- Slit
- Crimp
- Seal
- Sleeve

Nach der Festlegung der Prozesse war zudem eine Synchronisation zwischen der OPC UA-Arbeitsgruppe und TP3 erforderlich, da aus Sicht der Maschinen diverse Prozesse mit zusätzlichen Daten angereichert werden mussten. Ebenso erfolgte eine Aufgliederung der Prozesse auf Maschinenseite in die Bereiche "Spezifikation", "Validierung" und "Überwachung".

2.4.6 Fazit

Im Rahmen von AP 1.4 erfolgte eine intensive Auseinandersetzung mit der Tiefe der Informationsmodelle, um sicherzustellen, dass alle relevanten Daten und Informationen erfasst und in der erforderlichen Detailtiefe definiert sind. Ein Hauptaugenmerk lag dabei auf den bestehenden Standards VEC und OPC UA und die Granularität ihrer jeweiligen Datenstrukturen.

Während dieser Analyse haben sich einige Anpassungen und Erweiterungen an den bestehenden Standards herauskristallisiert, um die spezifischen Anforderungen der Kabelbaumindustrie zu erfüllen.

Änderungsvorschläge am VEC, um die vorhandenen Strukturen zu optimieren und die Datenrepräsentation zu verfeinern, wurden beim zuständigen Gremium eingereicht und dort zum heutigen Zeitpunkt auch bereits zu großen Teilen umgesetzt [19].

Eine Reihe von existierenden OPC UA Companion Specifications wurde als geeignete Basis für die Anforderungen der Leitungssatz-Industrie identifiziert und darauf aufbauend in Kooperation mit dem VDMA die OPC UA Companion Specification «OPC 40570: OPC UA for the Wire Harness Manufacturing Industry» [18] definiert.

Das Ziel dieser Spezifikation ist es, alle relevanten Aspekte und Anforderungen der Kabelbaumindustrie abzudecken und somit eine Grundlage für die Implementierung und Nutzung von OPC UA zu schaffen. Sie dient zudem als Bindeglied für die Anwendung der Verwaltungsschale und dem interoperablen Datenverkehr zwischen Ressourcen und Steuerungssystemen.

Durch die enge Zusammenarbeit mit dem VDMA und die Integration der Erkenntnisse in die OPC UA Spezifikation wurde ein nachhaltiger Beitrag geleistet, die Industrie 4.0-Vision in der Kabelbaumindustrie voranzutreiben und eine effiziente, standardisierte Kommunikation und Datenverarbeitung in diesem Bereich zu ermöglichen.

2.5 AP 1.5 - Validierung der Standards

Arbeitspaket 1.5 „Validierung der Standards“ sollte sich mit der Analyse der Informationsmodelle der bestehenden Standards beschäftigen, um sicherzustellen, dass alle relevanten Daten und Informationen erfasst und in der erforderlichen Detailtiefe definiert sind. Die detaillierten Ergebnisse der Analyse wurden zur besseren Verständlichkeit des Dokuments bereits im Kapitel 3 beschrieben, hier nochmals eine Zusammenfassung der Resultate. Ein Hauptaugenmerk lag auf KBL, VEC und OPC UA und die Granularität ihrer jeweiligen Datenstrukturen. Insbesondere wurden in diesem AP die folgenden Beiträge erarbeitet:

KBL und VEC: Die bestehenden Datenmodelle wurde analysiert und für den VEC entsprechende Anpassungen und Erweiterungen vorgenommen [19], um es besser auf die Bedürfnisse der Leitungssatzindustrie abzustimmen. Diese Anpassungen umfassen sowohl strukturelle als auch funktionale Verbesserungen, um die Genauigkeit der Datenmodellierung zu erhöhen (Abbildung 2-13). Dadurch ist das VEC-Datenmodell in der aktuellen Version 2.1.0 [19] noch leistungsfähiger und anwendungsfreundlicher geworden. Ergänzend wurde beschlossen, neben dem VEC- auch das KBL-Datenmodell zu unterstützen, da es bei mehreren OEMs etabliert ist und von bestehenden Systemen unterstützt wird.

OPC UA: Bestehende Bausteine wie «OPC 10031-4: ISA-95-4 Job Control» [22], «OPC 40001-3 Machinery Job Mgmt» [20] und «OPC 40001-101 Machinery Result Transfer» [16] wurden analysiert und als solide Grundlage identifiziert für die Entwicklung der neuen Companion Specification «OPC 40570: OPC UA for Wire Harness Manufacturing» [18]. Der Standard «OPC 40001-1 Machinery Basic Building Blocks» [17] bietet grundlegende Bausteine für die Maschinenidentifikation, die Überwachung von Maschinenzuständen und die präventive Wartung. Diese Bausteine ermöglichen eine einheitliche und strukturierte Darstellung von Maschineninformationen, was die Interoperabilität und Integration verschiedener Maschinen und Systeme erleichtert. «OPC 40001-3 Machinery Job Mgmt» [20] definiert die Bausteine für die Ausführung und Steuerung von Arbeitsaufträgen. Es umfasst die Verwaltung von Jobaufträgen, die Definition von Jobparametern und die Rückmeldung von Jobergebnissen, was eine effiziente Planung und Durchführung von Produktionsprozessen ermöglicht. Der Standard «OPC 40001-101 Machinery Result Transfer» [21] befasst sich mit der Übertragung von Ergebnissen und bietet Mechanismen zur Verwaltung und Übertragung von Produktionsdaten. Gemeinsam mit dem VDMA wurde im TP1 neue Companion Specification «OPC 40570: OPC UA for the Wire Harness Manufacturing Industry» [18] erarbeitet, um einen einheitlichen Standard für die Kommunikation und Interaktion zwischen

verschiedenen Systemen in der Leitungssatzindustrie zu etablieren, was wiederum zu einer Umsetzung von Industrie 4.0 in Richtung einer vernetzten und intelligenten Fertigung in diesem Bereich beiträgt.

Verwaltungsschale: Für die VWS wurde ein Konzept definiert, welches den Austausch von Informationen zwischen Produktionsunternehmen in der Leitungssatzindustrie ermöglicht. Die VWS nutzt dabei die in TP1 erarbeiteten und verbesserten Datenmodelle von VEC, OPC UA und ECLASS, um eine effiziente, modulare und standardisierte Kommunikation und Datenverarbeitung zu gewährleisten. Daraus wurden Initiativen für neue Submodelle abgeleitet und in den entsprechenden Teilprojekten erarbeitet: „IDTA 02056-1-0 Data Retention Policies“ [25] und „IDTA 02031-1-0 Bill of Process“ [26] im TP3. Im TP5 wurde der Anwendungsfall der virtuellen Inbetriebnahme (VIBN) unter Einsatz der VWS untersucht. Im TP6 wurden I4.0-Nachrichten definiert, um Synchronisation von Werten und Verhandlungsprozesse zwischen Leitungssatz-VWS zu ermöglichen.

ECLASS: Es wurden Definitionen und Klassifikationen recherchiert, die bereits in ECLASS vorhanden sind und potenziell für VWS4LS-Artefakte verwendet werden können. Die Beschreibung von Leitungssätzen erfordert detaillierte Informationen über Steckverbinder, Leitungen, Verbindungen, Isolierungen und vieles mehr. ECLASS bietet zwar eine allgemeine Struktur, aber es fehlen spezifische Attribute und Klassen, die für die genaue Beschreibung von Leitungssätzen geeignet sind. Deshalb wurden die bereits vorhandenen semantischen Definitionen der anderen Standards (VEC und OPC UA) als ausreichend für den Projektkontext betrachtet.

RAMI 4.0 wurde, wie schon weiter oben unter 2.2.3 beschrieben, als gut geeigneter übergeordneter Referenzrahmen für die Klassifikation der zu erstellenden Lösungbestandteile im VWS4LS-Projekt identifiziert. Es wurde jedoch entschieden, im Projekt keine Aufwände in diese Richtung zu investieren, da einerseits personelle Ressourcen gebunden, jedoch kein direkter anwendungsbezogener Mehrwert entstehen würde.

2.6 AP 1.6 - Dokumentation der Ergebnisse

Im Rahmen von VWS4LS wurde eine Vielzahl von Arbeitsdokumenten erzeugt, in denen die Bearbeitung und Ergebnisse der einzelnen Teilprojekte und Arbeitspakete detailliert dokumentiert wurde. Zu diesen Dokumenten gehören unter anderem auch die Anforderungserhebung, die Prozess- und Parameterliste über alle Produktionsprozesse der Leitungssatzindustrie, die Betrachtungen zur Systemarchitektur sowie der Interaktionen und Schnittstellen zwischen den verschiedenen Komponenten des Systems. Dabei entstand auch ein VWS4LS-Datenmodellkonzept (Abbildung 2-19).

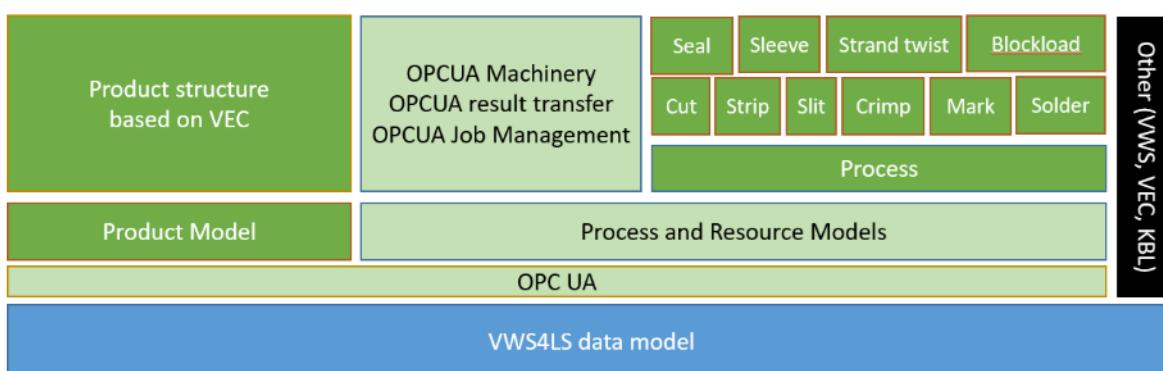


Abbildung 2-19: VWS4LS Datenmodell

Zu den im Verlauf des Teilprojekt 1 getroffenen wesentlichen Grundsatzentscheidungen für das Gesamtprojekt VWS4LS sowie die Beiträge zur Verbesserung und Weiterentwicklung der Standards und Technologien für die Kabelbaumindustrie gehören insbesondere:

Verbesserung des VEC-Datenmodells: Das bestehende VEC-Datenmodell wurde analysiert und entsprechende Anpassungen und Erweiterungen vorgenommen, um es besser auf die Bedürfnisse der Leitungssatzindustrie abzustimmen. Durch diese Arbeit ist das VEC-Datenmodell in der aktuellen Version 2.1.0 [19] noch leistungsfähiger und anwendungsfreundlicher geworden, wodurch es sich besser für die Nutzung in unterschiedlichen Anwendungsfällen eignet.

Weiterentwicklung der VWS: Die VWS vereinfacht den Austausch von Informationen zwischen Produktionsunternehmen in der Kabelbaumindustrie. Teilprojekt 1 hat mittels der erarbeiteten und verbesserten Datenmodelle von VEC [6] und OPC UA [18] sowie der neuen Submodelle „IDTA 02056-1-0 Data Retention Policies“ [25] und „IDTA 02031-1-0 Bill of Process“ [26] maßgeblich zur Weiterentwicklung der Verwaltungsschale (VWS) beigetragen. Dabei wurden nicht nur theoretische Grundlagen, sondern auch praktische Lösungen geschaffen [24], um die Implementierung und Akzeptanz der VWS in der Industrie zu fördern. So wurde bspw. eine Testimplementierung erstellt, die auf der Hannover Messe im April 2024 präsentiert wurde. Hervorzuheben ist, dass diese Demonstration vom VDMA als eine der ersten umfassenden Integrationen zwischen der VWS und OPC UA anerkannt wurde. Die Testimplementierung umfasste verschiedene Anwendungsfälle, die die Flexibilität und Vielseitigkeit der VWS verdeutlichten. Die Präsentation auf der Hannover Messe ermöglichte es, die Vorteile und Möglichkeiten der VWS einem breiten Publikum aus Industrie und Forschung vorzustellen und wertvolles Feedback zu sammeln. Die Anerkennung durch den VDMA unterstreicht die Bedeutung dieser Arbeit und zeigt, dass die Integration von VWS und OPC UA einen wichtigen Schritt in Richtung einer vernetzten und intelligenten Fertigung darstellt.

Entwicklung einer OPC UA Companion Specification: In einer engen Zusammenarbeit mit dem VDMA wurde die Entwicklung der neuen Companion Specification «OPC 40570: OPC UA for Wire Harness Manufacturing» [18] umgesetzt. Dies trägt dazu bei, einen Standard für die Kommunikation und Interaktion zwischen verschiedenen Systemen in der Kabelbaumindustrie zu etablieren, was wiederum zu einer effizienteren und zukunftssicheren Umsetzung von Industrie 4.0 in diesem Bereich beiträgt.

Durch die Bemühungen in diesen Bereichen hat VWS4LS maßgeblich zur Weiterentwicklung und Standardisierung in der Kabelbaumindustrie beigetragen und damit die Grundlage für eine erfolgreichere und effizientere Implementierung von Industrie 4.0 in diesem Sektor geschaffen (Abbildung 2-20).



Abbildung 2-20: Im Rahmen von VWS4LS erstellte neue Standards

3 TP2 - Entwicklungsprozesse des Leitungssatzes

Im Teilprojekt 2 "Entwicklungsprozesse des Leitungssatzes" wurde die Entstehung des Digitalen Zwilling entlang der Anwendungsfälle analysiert, um die Abläufe auf die Verwaltungsschale und die Erzeugung interoperabler Daten und Formate für den gesamten weiteren Lebenszyklus auszurichten. Das Teilprojekt wurde in folgende Arbeitspakete aufgeteilt, deren Resultate hier zusammenfassend erläutert werden:

- AP 2.1 – Konzept kollaboratives Datenmodell
- AP 2.2 – Single-Point-of-Truth-Definition
- AP 2.3 – Prozessbeschreibung LS-Entwicklung
- AP 2.4 – Teilmodele der Verwaltungsschale
- AP 2.5 – Umsetzung Digital Twin LS

3.1 AP 2.1 - Konzept kollaboratives Datenmodell

Im AP 2.1 "Konzept kollaboratives Datenmodell" wurde als wesentlicher Inhalt die Datenhaltung im kollaborativen Entwicklungsprozess betrachtet und im Ergebnis ein dezentraler Ansatz als das geeigneter Konzept angesehen.

Denn bei der Entwicklung und Herstellung eines Leitungssatzes sind verschiedene Akteure in einem komplexen Prozess involviert. Der OEM als Fahrzeugentwickler, der Konfektionär als Koordinator der Leitungssatzentwicklung sowie die unterschiedlichen Lieferanten-Abstufungen, die sog. „Tier“. Als Tier-1 werden typischerweise Systemlieferanten wie Hersteller von Steuergeräten oder eben auch Hersteller von Leitungssätzen bezeichnet. Tier-2 bezeichnet Lieferanten von einzelne Komponenten für das Gesamtsystem. Weitere Zulieferer von indirekten Gütern oder Ressourcen werden als Tier-3 bezeichnet. Zudem muss vor einem Einsatz von Komponenten deren Freigabe vom eigentlichen Auftraggeber (hier dem OEM) eingeholt und entsprechend dokumentiert werden. Diese Akteure verteilen sich dabei auf mehrere Unternehmen, die oft auch in verschiedenen Ländern ansässig sind. Darüber hinaus sind die Informationsflüsse zwischen den Akteuren situativ geprägt und lassen sich nicht allgemeingültig definieren, was Richtung und Reihenfolge anbelangt.

Die dahinterliegenden Datensätze müssen für die Aktualisierung und Änderung den beteiligten Akteure sichtbar und zugänglich gemacht werden. Mit Einsatz der VWS könnte diese Datenhaltung entweder in einem zentralen oder dezentralen Ansatz realisiert werden. Gegen den zentralen Ansatz sprachen vor allem Akzeptanzvorbehalte seitens der Akteure bezüglich der Datensicherheit, da sie ihre als sensibel angesehenen Daten in einem zentralen Repository ablegen müssten, dessen Zugänge sich aber nicht unter Ihrer alleinigen Kontrolle befinden. Deshalb werden von diesen Akteuren selbst gehostete Repositories favorisiert, wo sie die Hoheit darüber besitzen, welche Daten nach außen für wen zugänglich gemacht werden.

3.1.1 Kollaboration in der Leitungssatzentwicklung

Bei der Entwicklung und Herstellung eines Leitungssatzes sind verschiedene Parteien involviert, die vom OEM über den Tier-1, bis zum Tier-2 reichen. Der der Auftraggeber ist in diesem Szenario der OEM und stellt Produktanforderungen in Form von Funktionsmodulen. Als Tier-1 werden Systemlieferanten wie Hersteller von Steuergeräten und eben auch Hersteller von Leitungssätzen bezeichnet. Der Tier-1 entwickelt den Leitungssatz anhand der vom OEM zur Verfügung gestellten Funktionen. Dabei wird im ersten Schritt eine Umschlüsselung von Funktions- in Produktionsmodule durchgeführt. Der Grund hierfür ist, dass ein Funktionsmodul, lediglich eine Funktion beschreibt aber in den seltensten Fällen auf einer Ressource produziert werden kann. Demgegenüber kann das Produktionsmodul so heruntergebrochen werden, um eine modulare Produktion zu ermöglichen. Lieferanten, die einzelne Komponenten (Stecker, Leitungen oder Terminals) für die Gesamtsystemkomponente herstellen, bezeichnet man als Tier-2. Weitere Zulieferer von indirekten Gütern oder Ressourcen werden als Tier-3 bezeichnet.

In der nachfolgenden Abbildung 3-1 wird eine Vernetzung der Wertkettenteilnehmer während der Leitungssatzentwicklung aus Sicht des Konfektionärs dargestellt. Auf eine vollständige Darstellung aller Informationsflüsse wurde dabei auf Grund der Komplexität und der Übersichtlichkeit verzichtet.

Es soll gezeigt werden, dass es eine Vielzahl an Mitwirkenden und keinen allgemeingültigen Informationsfluss gibt, der sich durch eine vorgegebene Richtung sowie Reihenfolge immer gleich anwenden lässt. Zu sehen ist, dass es zwischen Tier-1 und Tier-2 einen Austausch geben kann und diese Informationen direkt zum OEM geleitet werden aber auch über einen anderen Tier zum OEM kommuniziert werden können. Des Weiteren kann es auch den Fall geben, dass Informationen eines Tier-3 dem OEM größtenteils nicht vorgelegt werden, wenn dies nicht zwingend erforderlich.

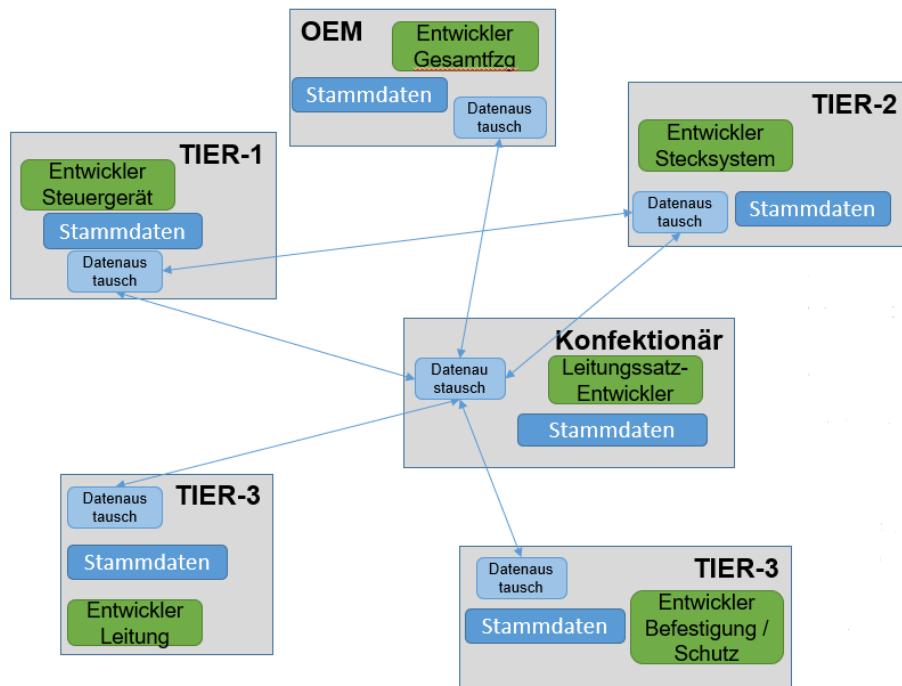


Abbildung 3-1: Kollaborationsbeziehungen in der Leitungssatz-Entwicklung

Größtenteils findet die Leitungssatzentwicklung durch den Konfektionär in der IT-Systemumgebung des OEMs statt. In der Leitungssatzentwicklung ist der Tier-1 die Schnittstelle zum OEM und zu den verschiedenen Komponentenherstellern (Tier-2). Seitens des OEMs wird der Tier-1 unter anderem mit CAD-Daten von der Karosserie und einer ersten Schaltplan- bzw. Elektrologik versorgt. Diese Datenpakete beinhalten auch Informationen wie bspw. welche Komponenten ggf. bereits festgelegt sind, für welche Funktion oder Spezifikation eine bereits vorhandene Komponente verwendet werden soll oder ob ggf. eine neue Komponente entwickelt werden muss.

3.1.2 Datenhaltung in der Leitungssatzentwicklung

Die meisten OEMs pflegen ihre eigenen Stammdaten, wie beispielsweise Teilesachnummern. Zwischen den verschiedenen Tiers und OEMs erfolgt der Austausch der Komponenten über die verschiedensten Schnittstellen und mit unterschiedlichen Datenformaten (siehe Abbildung 3-2). Darüber hinaus gibt es keine „Rückkopplung“ bei Änderungen o.ä., die für eine Synchronisation der Datensätze sorgen kann. Nach der Freigabe durch den OEM eines definierten Entwicklungsstandes wird ein filebasierter Datenstand aus den Systemlandschaften des OEMs exportiert und an den Tier-1 „übergeben“. Beim Tier-1 werden diese Daten verwendet und um Produktionsinformationen angereichert. Diese beiden Datenstände, der vom OEM und der vom Tier-1, existieren parallel und haben keine automatisierte Verlinkung oder Rückkopplung zueinander. Es entstehen somit bereits auf Stammdatenebene Datenbrüche und Inkonsistenzen bei den Datensätzen und die Anwendung der Verwaltungsschale bietet hier viel Verbesserungspotential.

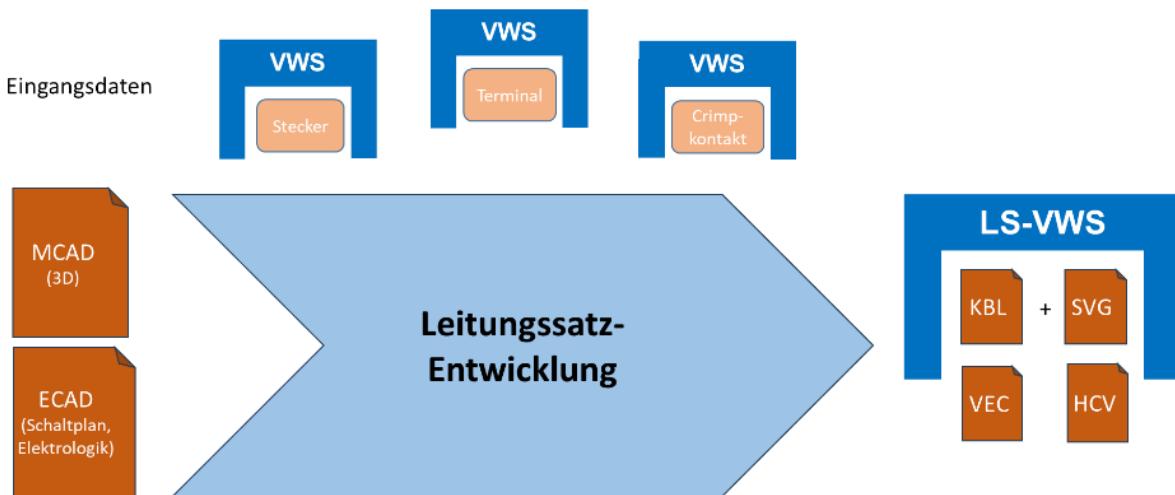


Abbildung 3-2: Eingangs- und Ausgangsdaten im Entwicklungsprozess im Überblick

3.1.3 Anwendungspotential der VWS

Der Einsatz der VWS würde vor allem bei der Übertragung von Produktinformationen zwischen den Wertkettenteilnehmern einen großen Mehrwert bieten, da im aktuellen Prozess zwischen den Wertkettenteilnehmern proprietäre Schnittstelle und unterschiedliche Datenformate bedient werden müssen. Die „Neutralität“ der auf XML-basierten VWS könnte zu einer grundsätzlichen Reduktion der proprietären Schnittstellen führen. Dabei würden Stammdatenattribute, CAD-Daten und Schaltpläne, die die elektrologischen Verbindungen und weitere Informationen des Leitungssatzes enthalten, in der VWS hinterlegt. Für die Produktstammdaten kann hierzu auf etablierte Datenstandards wie KBL und VEC zurückgegriffen werden. Bezuglich der Ausgangsdaten aus den verschiedenen Engineering-Tools würde die VWS also eine Art Container darstellen, in der diese Daten aus dem Prozess Leitungssatzentwicklung als Ergebnis „abgelegt“ sind. Es spielt dabei keine Rolle ob Dateien in der VWS abgelegt bzw. verlinkt werden oder ob die Leitungssatzdaten darin „aufgelöst“ abgelegt sind.

Der Einsatz von VWS hätte dabei nur bedingt Auswirkung auf die Engineering-Tools, die bei der Leitungssatzentwicklung verwendet werden. Für die nahtlose Anwendung müssten diese Softwaretools lediglich um eine Schnittstelle zum Import- bzw. Export der VWS erweitert werden. Am wichtigsten für die Umsetzung des interoperablen Digitalen Zwillings ist die Umsetzung der durchgängigen Verlinkung.

3.1.4 Fallbeispiel für den Einsatz von VWS bei der Auswahl eines Steckers

In diesem Abschnitt soll ein potenzieller Ablauf einer Kollaboration zwischen dem Konfektionär und mehreren Tier-2 dargestellt werden. Ziel ist es, aufgrund einer Anforderung, die richtige Auswahl eines Steckgehäuses durchzuführen.

Annahme dabei ist, dass es im Datenökosystem (z.B. [Catena-X](#)) ein gehostetes zentrales Repository gibt, in dem die Teilnehmer „Angebote“ hinterlegen sowie „Anfragen“ für „Dienstleistungen“ stellen können (siehe Abbildung 3-3).

Der Konfektionär kann dort eine Anfrage starten, die grundlegende Attribute enthält, die ein bestimmtes Produkt (z.B. Steckgehäuse) haben soll. Das Repository agiert hier also als zentrales Register für „Angebote“ und „Anfragen“.

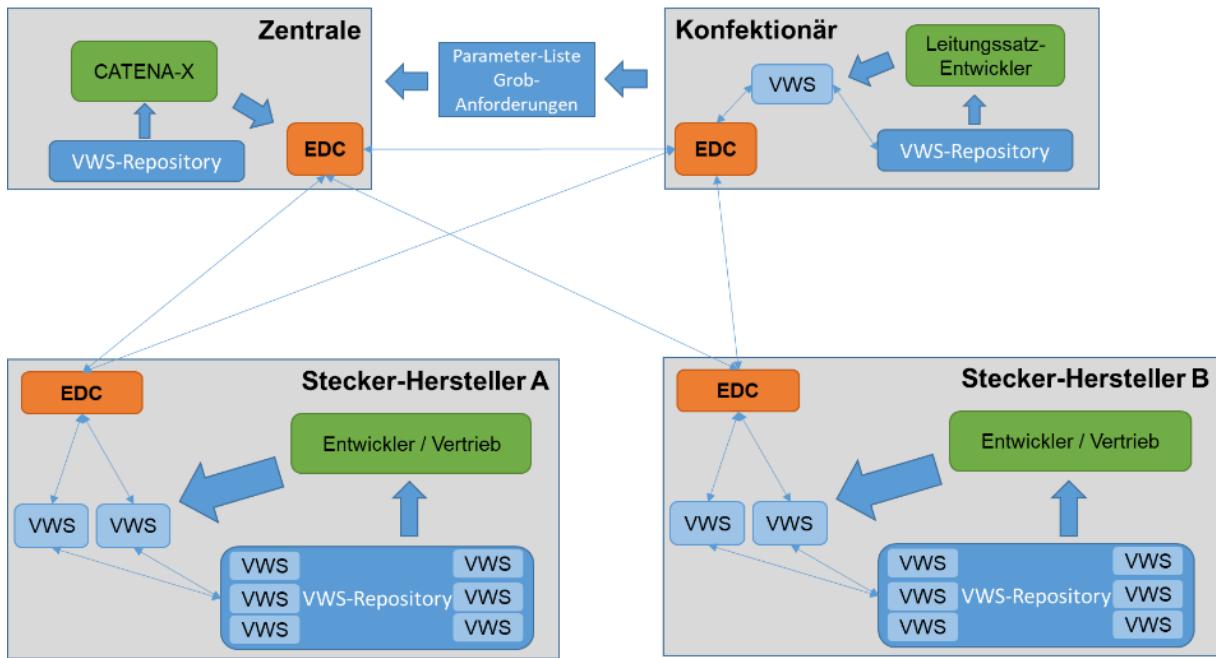


Abbildung 3-3: Anfrage beim Komponentenhersteller stellen

Im Repository wird dann nachgesehen, ob es passende Kandidaten für diese Anfrage gibt. Als Ergebnis erhält der anfragende Konfektionär eine Liste möglicher Kandidaten, hier Stecker-Hersteller A (oder auch A und B), die eine Lösung für die Anfrage bieten können (Abbildung 3-4).

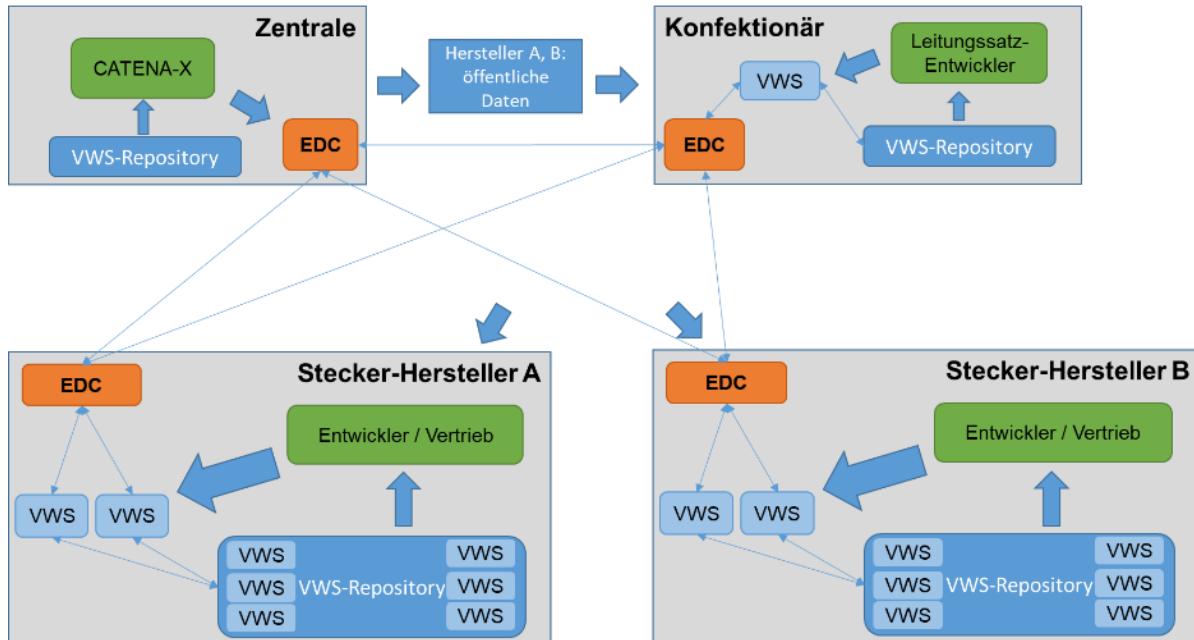


Abbildung 3-4: Auskunft auf Anfrage

Im Anschluss, wenn der Konfektionär die Daten der Anbieter für seine Anfrage aus dem Repository zurück erhält, kann er mit den Anbietern in einen detaillierten Austausch gehen. Hierbei können dann noch weitere Bedingungen (z.B. kommerzieller Natur) abgeklärt werden, die weiter ins Detail gehen und evtl. eine Geheimhaltungsvereinbarung voraussetzen (Abbildung 3-5).

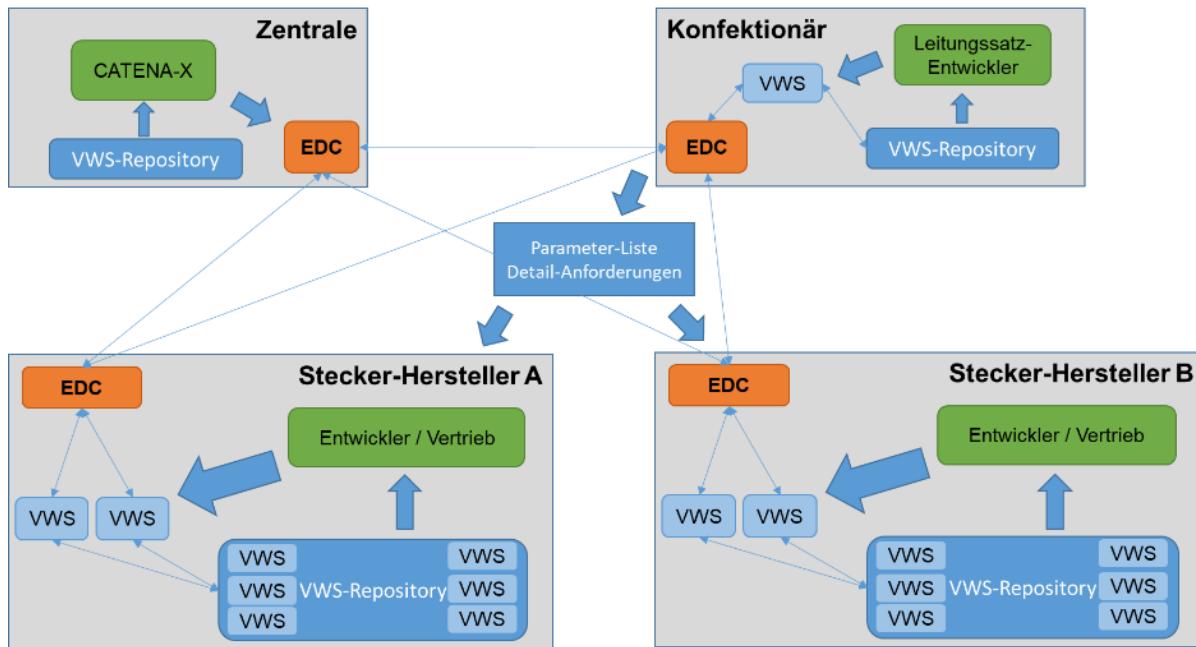


Abbildung 3-5: 2. Anfrage mit Detail-Anforderungen

Damit wäre ein repräsentatives Anwendungsszenario auf Basis einer dezentralen Datenhaltung mit Verwaltungsschalen konzeptionell grob beschrieben. Neben öffentlich einsehbaren Informationen können in einem weiteren Schritt dann detaillierte und nichtöffentliche Informationen ausgetauscht werden, zudem bleibt die Zuständigkeit für die Aktualität der Daten in der VWS weiterhin beim jeweiligen Hersteller.

3.1.5 Fazit

Ein kollaboratives Datenmodell in der Produktentwicklung besteht aus einer zentralen, strukturierten Datenbasis, die von verschiedenen Teams und Abteilungen innerhalb eines Unternehmens gemeinsam genutzt und bearbeitet wird. Es ermöglicht eine Echtzeit-Zusammenarbeit, bei der alle Beteiligten, wie Entwickler, Designer oder Marketingexperten, auf dieselben Informationen zugreifen und diese aktualisieren können. Dieses Modell unterstützt die interdisziplinäre Zusammenarbeit, sorgt für konsistente und aktuelle Daten und erhöht die Effizienz des Entwicklungsprozesses. Durch Rollen- und Rechtemanagement wird sichergestellt, dass die Datenintegrität gewahrt bleibt. Zudem ermöglicht es die Versionierung und Rückverfolgbarkeit von Änderungen, was die Transparenz der Produktentwicklung steigert. Für die Übertragung eines kollaborativen Datenmodell-Ansatzes auf eine Unternehmensübergreifende Ebene, müssen einige Grundbedingungen berücksichtigt werden:

- Jeder Hersteller erstellt für seine Produkte eigene VWSen, die in einem eigenem Repository gehostet und verwaltet werden.
- Zugriffsrechte regeln die Sichtbarkeiten der Inhalte bei den Entwicklungspartnern. Diese Zugriffsrechte müssen vom Eigentümer der Datensätze auf VWS-Ebene oder Teilmödell-Ebene erteilt werden, damit die nachfolgende Instanz mit diesen Informationen weiterarbeiten kann.
- Die beteiligten Stakeholdern im Entwicklungsprozess können die bereits heute etablierten Tools wie [L Dorado⁴](#), [Siemens Harness Designer⁵](#), [Vector PREVision⁶](#) etc. weiterverwenden. Voraussetzung wäre, dass die Tools entsprechend ergänzt werden, um die Ergebnisse VWS-konform zu transferieren.
- Zwischenergebnisse können mit entsprechenden Methodiken zur Versionierung abgegrenzt und an andere Teilnehmer weitergegeben werden.

⁴ <https://trias-mikro.de/produktloesungen/bordnetz-design-mit-kbl-standard/>

⁵ <https://plm.sw.siemens.com/de-DE/capital/products/capital-wiring-harness-designer/>

⁶ <https://www.vector.com/de/de/produkte/produkte-a-z/software/preevision/>

- Am Ende kann die finale Produkt-VWS zusammengestellt und freigegeben werden, um mit dem physischen Produkt an den Kunden geliefert zu werden.

3.2 AP 2.2 - Single-Point-of-Truth-Definition

Im AP 2.2 „Single-Point-of-Truth-Definition“ wurden Lösungsansätze für das Datenmanagement betrachtet. Wie in AP2.1 beschrieben, ist die Entwicklung eines Leitungssatzes ein stark kollaborativer Prozess mit mehreren Akteuren. Durch die kollaborative Arbeit und die Anforderung einer dezentralen Datenhaltung in verteilten Repositories entsteht die Gefahr von Inkonsistenzen in der Datenhaltung. Damit ist es notwendig, einen Referenzpunkt festzulegen, den sog. „*Single-Point-of-Truth*“ oder **SPoT** für alle Daten und Parameterwerte eines Leitungssatzes entlang seiner Wertschöpfung, um eine inkonsistente Datenhaltung in den nebeneinander existierenden Datensilos zu vermeiden.

Der SPoT bezeichnet damit die Datenquelle oder Datenquellen, aus der sämtliche angeschlossenen Systeme für eine bestimmte Prozessphase bedient werden. Im Entwicklungsprozess liegt dieser Referenzpunkt oftmals in der Datenbank des Kunden (OEM). Durch den intensiven Austausch zwischen Konfektionär, OEM und/oder Tier-1/2/3 aber liegen die aktuellen Daten bzw. Parameter nicht mehr unbedingt in der Datenbank des Kunden. Es ist dann Aufgabe des SPoT, diese Datenbeziehungen zu verwalten.

Mit Einsatz der VWS bietet sich die Möglichkeit, der SPoT in einer VWS auf einem VWS-Repository oder -Speicherort zu definieren und damit die Aktualisierung und Änderung sämtlichen Dateien für beteiligte Akteure sichtbar und zugänglich zu machen. Die Implementierung der VWS könnte entweder in einem zentralen oder dezentralen Ansatz realisiert werden. Im zentralen Ansatz (siehe Abbildung 3-6) wird die VWS auf einem zentralen Repository gespeichert. Die jeweilige Akteure haben je nach Rechte-Management Leserecht auf bestimmte Teilmodelle innerhalb der VWS oder können auch Schreibrecht auf relevante Sub-Modelle besitzen, um die Daten oder Parameter zu pflegen. Diese VWS wird anschließend als de-facto SPoT eingesetzt. Die Vorteile von diesem Ansatz sind u.a. zentrale Freigaben sowie aktuelle Dateien und Parameter auf einem zentralen (festen) Speicherort. Dieser Ansatz fordert eine gute Berechtigungssteuerung, einen höheren Aufwand für den Speicherort sowie höhere Anforderungen an die Server-Performance. Diese Anforderung zeigen sich als die Nachteile von einem zentralen Ansatz.

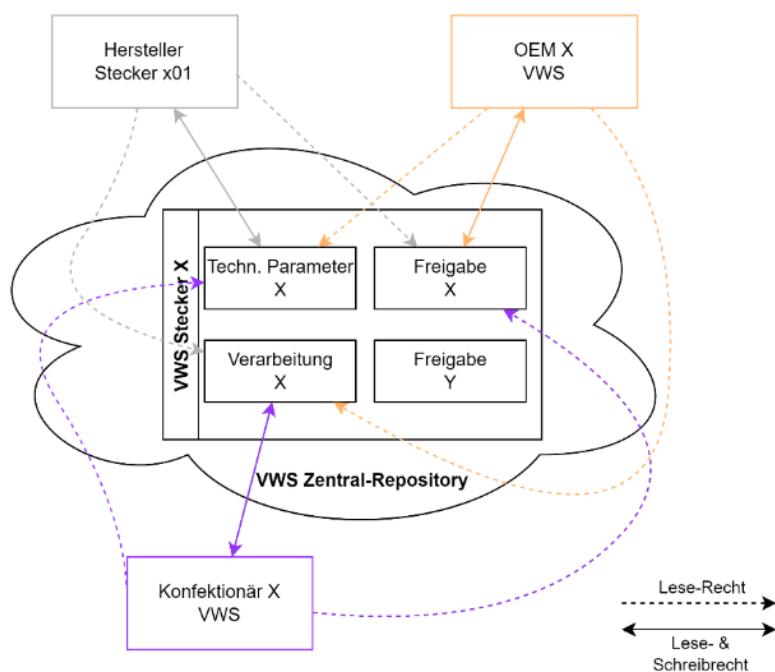


Abbildung 3-6: Beispiel für zentralen Ansatz

Ein dezentraler Ansatz (siehe Abbildung 3-7) bildet die andere Möglichkeit, um einen SPoT zu definieren. Dabei besitzt jeder Akteur seine eigene VWS der Komponente, die lokal innerhalb der eigenen IT-Systemlandschaft gespeichert wird. Die Anforderung an die IT-Infrastruktur für diese Art von dezentralem Repository ist nicht so hoch wie beim zentralen Ansatz. Jeder Akteur kann die VWS mit relevanten Teilmodellen für den internen, als auch relevanten Teilmödelle von externen Entwicklungspartnern aufbauen. Die Teilmödelle anderer Entwicklungspartner werden in der VWS eines Akteures entweder als eine Kopie oder eine Verlinkung implementiert. Die schwierigste Anforderung dieses Ansatzes ist die Synchronisierung der individuellen VWSen oder individuellen Teilmödellen von anderen Entwicklungspartnern. Jeder Data-Owner könnte sein eigenes Teilmödell auf seiner lokalen VWS ändern. Findet eine Änderung statt, dann sollte der Data-Owner diese Änderung (automatisch) mitteilen und eine (automatische) Synchronisierung für andere Entwicklungspartnern ermöglichen. Auf Grund der verteilten VWS ist die Definition des SPoT im Vergleich mit dem zentralen Ansatz nicht trivial.

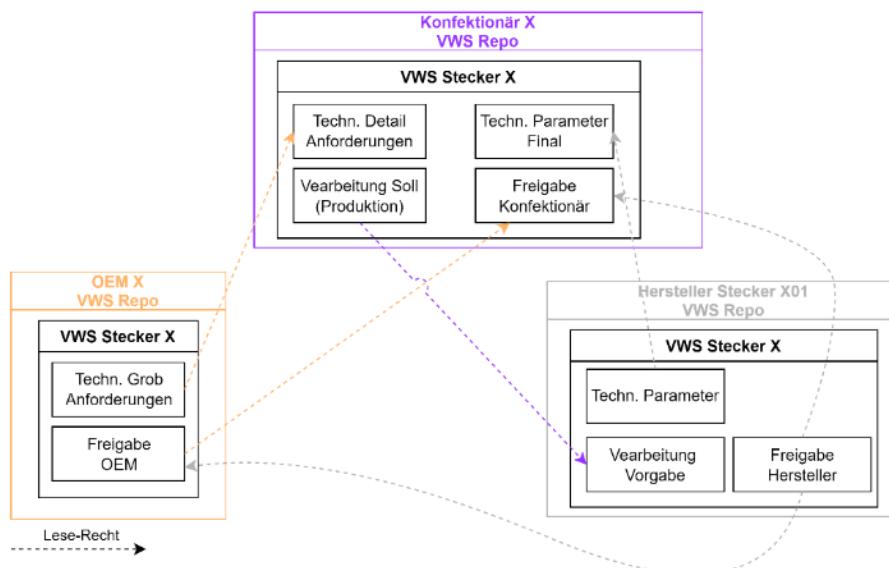


Abbildung 3-7: Freigaben und Änderungs-Management beim dezentralen Ansatz

Deshalb wurde im Rahmen dieses APs diskutiert, wie ein SPoT bei einem dezentralen Entwicklungsprozess genau definiert werden kann. Es wurde dabei festgestellt, dass der Ort des SPoT abhängig von der Phase des Lebenszyklus ist. Während der Entwicklungsphase befindet sich der SPoT auf der VWS des Kunden (OEM). In der Produktionsphase wird die VWS auf Instanz-Ebene erweitert. Während des Produktionsprozesses beim Konfektionär werden sämtliche Ist-Werte von Fertigung erfasst und in der VWS integriert. Daher befindet sich der SPoT bei dieser Lebenszyklusphase des Leitungssatzes beim Konfektionär, siehe Abbildung 3-8.



Abbildung 3-8: SPoT-Bezeichnung beim dezentralen Einsatz

3.2.1 Herausforderung: Produkt mit mehreren Herstellern

Eine im Leitungssatz verwendete Komponente wie z.B. ein Stecker muss nicht unbedingt nur in einem Land hergestellt werden, da auch Zulieferer mehrere Werke betreiben. In Abbildung 3-9 sind zwei mögliche Ursprungsländer für einen Stecker aufgelistet.

Bei zwei möglichen Stecker-Herstellern können folgende Szenarien eintreten:

- Eine OEM-Teile-Nr., zwei verschiedene Hersteller mit je eigener Teile-Nr.
- Pro Teile-Nr. müssen die jeweiligen Parameter & Spezifikationen verwaltet werden, d.h. die VWS von Stecker ax und Stecker ay.

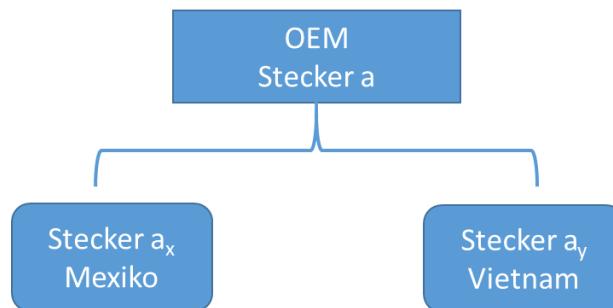


Abbildung 3-9: Beispiel für einen Stecker, der an zwei verschiedenen Standorten verwendet wird

Diese Unterschiede bilden sich in unterschiedlichen Sachnummern für ein funktional identisches Bauteil ab, was in entsprechenden Tabellen niedergelegt wird und bei Änderungen entsprechend gepflegt werden muss. Unabdingbar ist dabei, dass beide Sachnummern mit einer Freigabe belegt sind, siehe Abbildung 3-10. Hier sind die Freigaben seitens des OEM, des Herstellers und des Konfektionärs notwendig, um das entsprechende Bauteil überhaupt verwenden zu können.

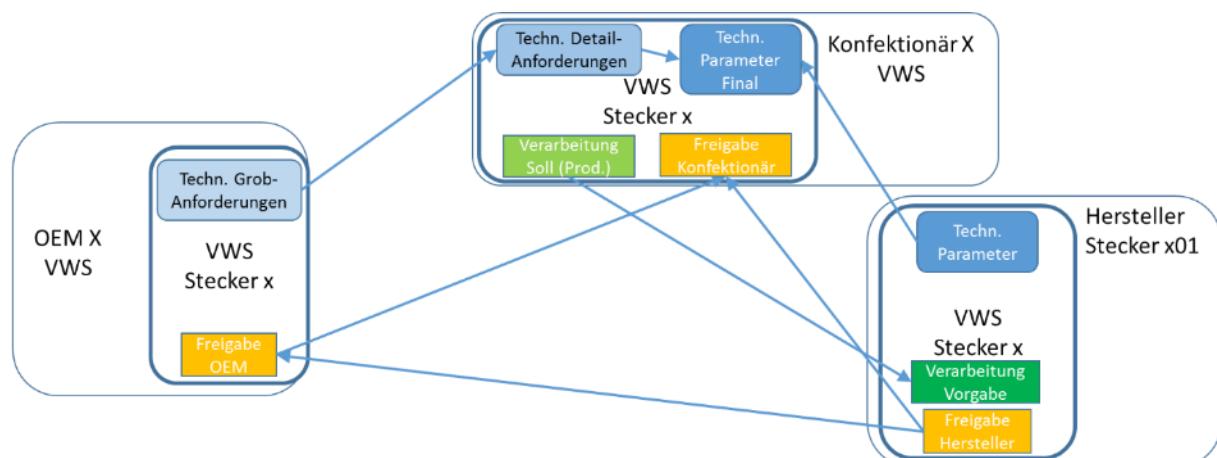


Abbildung 3-10: Freigaben und Änderungs-Management

Durch dieses Mapping von Zuordnungen können die verschiedenen Teile-Bezeichnungen und Nummerierungen weitergeführt und je nach Anforderungen angepasst werden. Zudem wird eine Rückverfolgbarkeit gewährleistet und es kann für spätere Betrachtungen, wie z.B. Lieferketten-Analysen oder CO2-Footprint, eine genauere Analyse durchgeführt werden.

3.3 AP 2.3 - Prozessbeschreibung LS-Entwicklung

Im AP 2.3 wurde eine generischer Leitungssatz-Entwicklungsprozess erarbeitet, d.h. eine vereinheitlichte Beschreibung aller wesentlichen Prozessschritte und der dazugehörigen Bedarfe (Input, Output). Dazu musste der Ist-Prozess der Leitungssatz-Herstellung analysiert werden, mit den folgenden Zielsetzungen:

- a.) Systembedarf abklären,
- b.) Gesamtprozess in Teilprozesse/Prozessschritte unterteilen,
- c.) Informationsbedarfe der Teilprozesse ableiten,
- d.) Kosten-Aspekte berücksichtigen.

Im Anschluss wird die Umsetzung in die Verwaltungsschale beschrieben und erste Tests der Konzepte durchgeführt.

Die resultierenden Analysedaten finden sich mit Input als Anforderung und Output als Zwischenergebnis gesammelt in „[VWS4LS_Submodell_TP2_Entwicklungsprozess_2023_05_22.xlsx](#)“⁷.

3.3.1 Analyse des Entwicklungsprozesses

Ein Überblick des generischen Entwicklungsprozess ist in Abbildung 3-11 grafisch dargestellt.

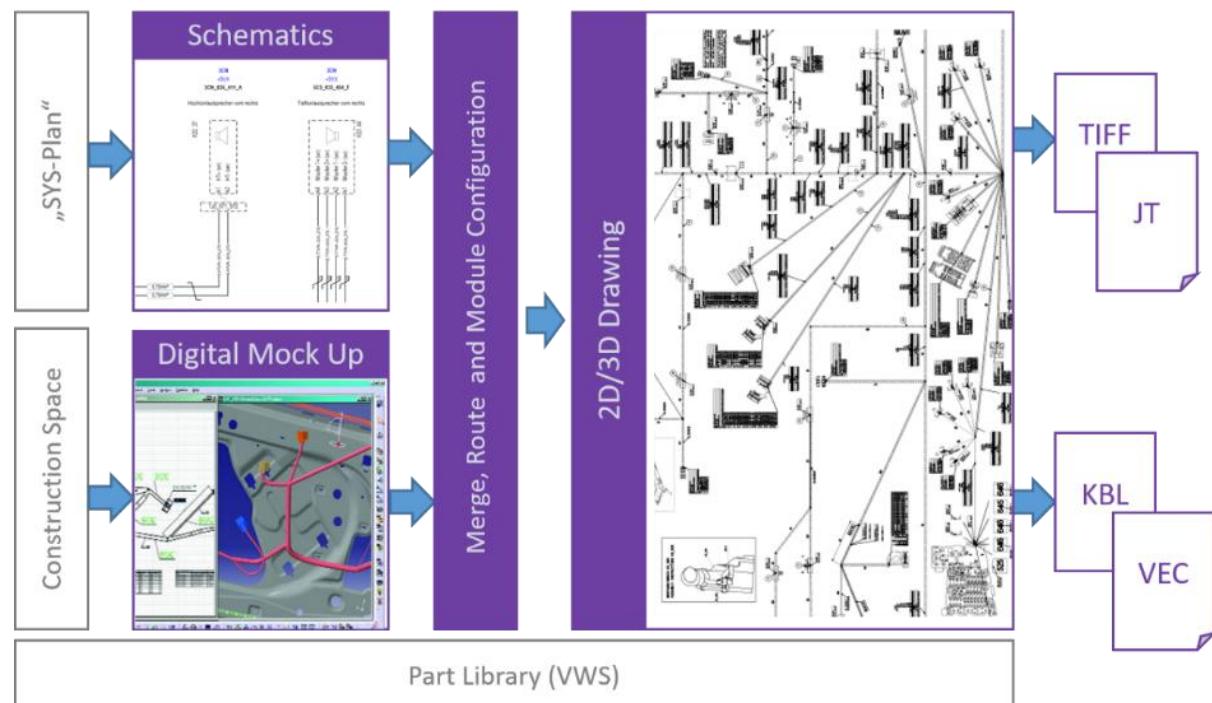


Abbildung 3-11: Überblick Teilprozesse der Leitungssatzentwicklung

Die detaillierte Ansicht der Prozessschritte ist in Abbildung 3-12 dargestellt. Als Ausgangspunkt werden ein 3D-Modell des Bauraums sowie Systemschaltpläne benötigt. Der Bauraum wird meist partitioniert, d.h. abschnittsweise, zur Verfügung gestellt, um die Datenmenge zu begrenzen. Hierfür werden einschlägige CAD-Datenformate verwendet. Die Systemschaltpläne stellen die elektrischen Daten dar, angefangen von der Verbindungsliste bis hin zum Strombedarf der einzelnen Sensoren und Aktoren. Die Entwicklung wird dann parallel auf elektrischer und mechanischer Domäne durchgeführt. In der mechanischen Domäne wird der Leitungssatz grafisch als ein „Rohr“ dargestellt, als Platzhalter für die spätere Verlegung. Der Kunde bzw. OEM und der Komponentenentwickler stellen das 3D-Teilmodell zur Verfügung. Der Leitungssatzentwickler fügt diese 3D-Teilmodelle zusammen und daraus entsteht ein 3D-

⁷ https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP02/Beispieldaten/VWS4LS_Submodell_TP2_Entwicklungsprozess_2023_05_22.xlsx

Modell für den Leitungssatz. Aus der 3D-Teil-Bibliothek kann der LS-Entwickler den passenden Gegenstecker auswählen und am Ende des „Rohres“ platzieren. Neben den mechanischen Eigenschaften bietet dieser Prozessschritt auch die Möglichkeit eine sogenannte „Temperatur-Landkarte“ zu pflegen, da die Umgebungstemperatur eine große Rolle bei Dimensionierung einer Leitung spielt. Daher ist es sehr hilfreich diese Information im 3D-Modell zu integrieren. Als Zwischenergebnis aus diesem Prozess wird das 3D-Modell als *Digital Mock Up* (DMU) erzeugt. In der elektrischen Domäne wird die Information über die Bordnetz-Architektur bzw. der Systemplan vom OEM in eine spezifische Bordnetztopologie umgewandelt und als Zwischenergebnis der Kabel-Schaltplan erzeugt. Dieser Kabel-Schaltplan beinhaltet unter anderem den Typ, Querschnitt, Verortung⁸ des sog. „Splice“⁹ einer Leitung sowie die Anforderungen von Kontaktobерflächen.

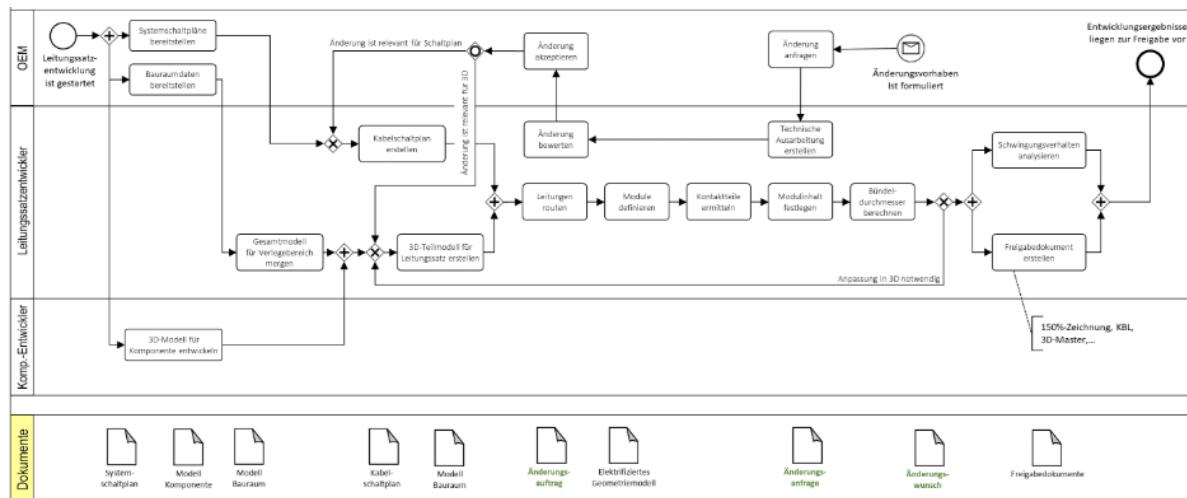


Abbildung 3-12: Referenzprozess der Leitungssatz-Entwicklung

Nach Vorbereitung der 3D-Modelle sowie des elektrischen Schaltplans können mit der Information aus dem Schaltplan die Leitungsgruppe oder -Bündel, sowie deren Dimensionen abgeleitet werden.

Nach Start der Leitungssatz-Entwicklung mit den notwendigen Dokumenten bzw. Dateien spaltet sich der Prozess auf in den Kabelschaltplan und 3D-Teilmodell erstellen. Nach dem Mergen (Zusammenführen) aller 3D-Teilmodelle werden die elektrischen und 3D-Daten zusammengeführt und das Routen der Leitungen beginnt. Im Anschluß werden die Module definiert, die zusammengehörige Leitungen in ein Modul bündelt, die konkreten Kontaktteile für alle Anschlüsse ermittelt und abschließend der Bündeldurchmesser an allen relevanten Stellen des Kabelsatzes berechnet. Sollten hier Anpassungen z.B. einer Karosserie-Durchführung notwendig werden, wird wieder zurück zur Erstellung der 3D-Teilmodelle gegangen und der Prozess nochmals durchgeführt. Sobald nach dem Bündeldurchmesser berechnen keine Anpassung mehr durchzuführen ist, wird je nach Kundenanforderung das Schwingungsverhalten (meist bei Tür-Übergängen) analysiert und abschließend die Freigabe-Dokumente erstellt. Diese umfassen die entsprechenden Zeichnungen und die dazu gehörigen elektrischen und sonstigen Informationen wie Teilelisten etc.

Bei einer Anfrage auf Änderung wird eine technische Ausarbeitung erstellt, die alle Aspekte und Daten der Änderung umfasst. Das können rein mechanische Änderungen seitens Lage der Anschluß-Buchse bei einem Steuergerät sein oder erhöhter Strombedarf auf einem Anschluss-Pin. Diese Änderung wird von allen betroffenen Zulieferern, dem Konfektionär und dem OEM bewertet und final akzeptiert. Danach muss die Änderung entsprechend in den vorhandenen Leitungssatz eingepflegt und die Umsetzung in die Produktion geplant werden.

Der bisher erzeugte Schaltplan gilt für einen sogenannten „150%-Leitungssatz“, welcher die Informationen von allen möglichen Verbindungen aus verschiedene Ausstattung eines Fahrzeugtyps enthält.

⁸ Hier ist nicht die geometrischen Verortung im Fahrzeug gemeint, sondern die Verzweigung eines Kabels nach einer bestimmten Länge auf mehrere Kabelsegmente.

⁹ <https://ecad-wiki.prostep.org/specifications/vec/guidelines/component-types/splices/>

Diese Verbindungen werden nie komplett in einem Fahrzeug gebaut, da oftmals eine Ausstattungsoption die andere ausschließt. Deshalb wird nun der Prozessschritt *Modularisierung* durchgeführt und dabei der Leitungssatz in Basis-Leitungssatz und weitere optionale Module unterteilt. Am Ende des Modularisierungsprozesses wird der Durchmesser der Leitungsbündel erneut berechnet. Wenn der neue Durchmesser die entsprechende Anforderung nicht erfüllt, dann wird ein iterativer Prozess ausgelöst und der Entwicklungsprozess wird von der Erstellung des 3D-Teilmodells bis zur Berechnung der Bündeldurchmesser wiederholt. Als Ergebnisse des LS-Entwicklungsprozess werden verschiedene Freigabedokumente erstellt, unter anderem die „150%-Zeichnung“, KBL bzw. VEC und 3D-Muster der LS-Verlegung.

3.3.2 Entwicklungsprozess mit der Verwaltungsschale

Wenn der eingangs betrachtete Entwicklungsprozess auf Basis der Verwaltungsschale umgesetzt werden soll, beginnt der Vorgang mit der Bereitstellung einer VWS durch den OEM, welche die zentrale Informationen und Datenstrukturen für die Leitungssatzentwicklung enthält. In dieser VWS sind Freigabelisten für Komponenten, CAD-Zeichnungen sowie relevante Normen und Anforderungen enthalten. Diese umfassende Datensammlung dient als eine zentrale und konsistente Datenquelle, die als Grundlage für alle weiteren Entwicklungsaktivitäten dient.

Im weiteren Verlauf des Entwicklungsprozesses entsteht dann beim Tier-1 eine Struktur zusammenhängender Verwaltungsschalen, die die Anforderungen und Daten der jeweiligen Entwicklungsstufen integriert und konsistent weiterführt. Hierzu wurde untersucht, wie dezentrale Verbundkomponenten im Kontext der Verwaltungsschale umgesetzt werden können, um eine flexible und skalierbare Integration sicherzustellen.

Diese Prozessbeschreibung zeigt, wie die Verwaltungsschale als zentrales Element in der Leitungssatzentwicklung genutzt werden kann, um die Integration von Einzelkomponenten zu erleichtern, die Kommunikation zwischen OEM, Tier-1 und Tier-2 zu optimieren und die Effizienz des gesamten Entwicklungsprozesses zu steigern. Durch die gezielte Bereitstellung spezifischer Produktinformationen wird die Zusammenarbeit entlang der Lieferkette effizient und sicher gestaltet.

In der nachfolgenden Abbildung 3-13 soll der beschriebene Workflow nochmals zusammengefasst auf einer abstrakten Ebene visualisiert werden.

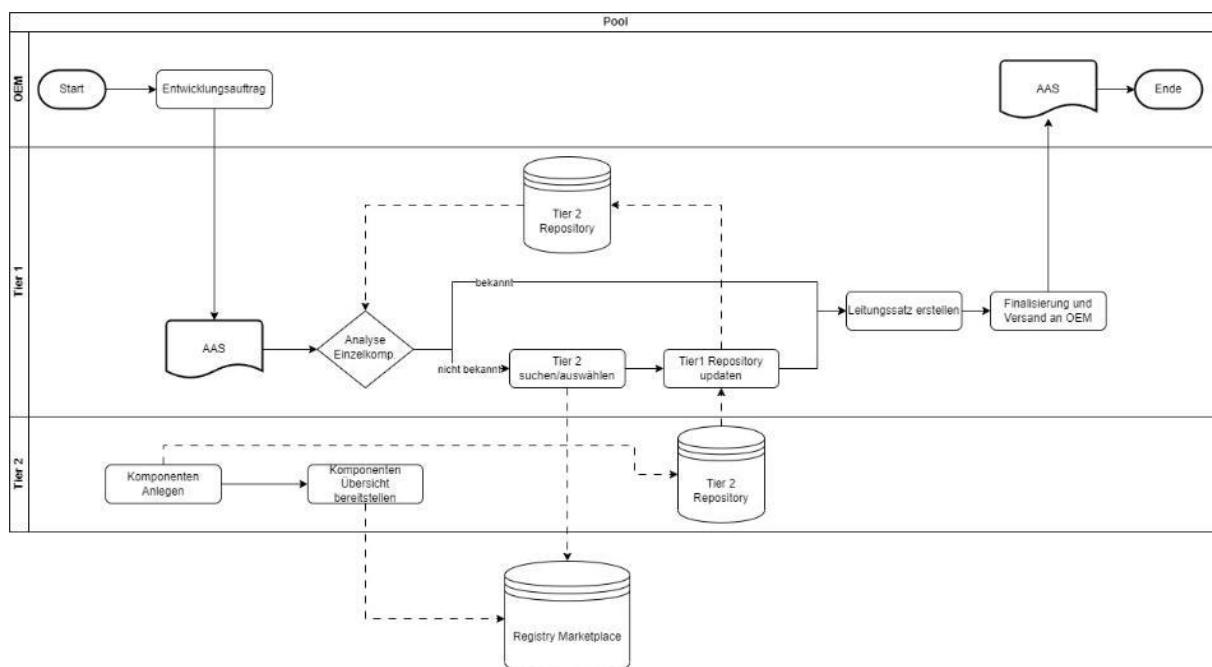


Abbildung 3-13: Workflow Entwicklung Leitungssatz

3.3.3 Analyse und Integration der Einzelkomponenten

Der Tier-1 beginnt den Prozess unter anderem, indem er die im Entwicklungsauftrag enthaltenen Einzelkomponenten analysiert und in zwei Kategorien unterteilt:

1. **Bekannte Einzelkomponenten:** Diese Komponenten sind bereits in der Tier-1-Datenbank vorhanden und können direkt mit der entsprechenden Produktsichtweise weiterverwendet werden. Die Verwaltungsschale des Tier-1 ermöglicht es, diese Komponenten effizient in den Entwicklungsprozess zu integrieren.
2. **Unbekannte Einzelkomponenten:** Für Komponenten, die nicht in der Tier-1-Datenbank vorhanden sind, gibt es zwei Möglichkeiten:
 - o **Generell bekannte und freigegebene Komponenten:** Die Komponente ist bereits freigegeben beim OEM, aber beim Tier-1 noch nicht in der Datenbank vorhanden. Der Tier 1 kontaktiert den jeweiligen Komponentenhersteller (Tier-2) und fordert den Zugriff auf die Produkt-VWS an, um die fehlenden Informationen zu integrieren.
 - o **Suche nach Komponentenentwicklungspartner:** Falls keine direkte Integration der unbekannten Komponenten möglich ist, kann ein Szenario sein, dass der Tier-1 einen Entwicklungspartner (Tier-2) für die entsprechende Komponente ausfindig machen muss, der die Anforderungen des Entwicklungsauftrags erfüllen kann. Anschließend sind Freigabeprozesse beim OEM zu durchlaufen und die Integration der Daten in die entsprechenden Datenbanken.

Sobald die unbekannten Komponenten erfolgreich angelegt und die VWS-Produktsichtweisen erstellt sind, kann die Aggregation der Verbundkomponente beginnen.

3.3.4 Strukturierung der Verbundkomponente

Nachdem der Tier-1-Lieferant den Entwicklungsauftrag in Form einer Verwaltungsschale (VWS) vom OEM erhalten hat, beginnt die eigentliche Leitungssatzentwicklung. Zunächst analysiert der Tier-1 die übergebenen Daten und identifiziert die notwendigen Schritte zur Integration und Erstellung der Verbundkomponente der Verwaltungsschale.

1. **Identifikation und Erzeugung der notwendigen Teilmodelle:** Basierend auf den vom OEM bereitgestellten VWS-Informationen muss der Tier-1 die spezifischen Teilmodelle für die Leitungssatzentwicklung definieren. Diese Teilmodelle umfassen unter anderem:
 - o **Produktteilmodelle:** Enthalten spezifische Produktinformationen wie technische Spezifikationen, CAD-Daten und freigegebene Komponenten.
 - o **Prozessteilmodelle:** Diese Modelle beinhalten Informationen zu Fertigungsprozessen, Montagereihenfolgen und Prüfanforderungen.
 - o **Ressourceneilmodelle:** Beinhaltet alle notwendigen Ressourceninformationen, z.B. Werkzeuge, Maschinen und Materialien.
2. **Verknüpfung der Teilmodelle:** Um eine konsistente und funktionale Verwaltungsschale zu erstellen, müssen die einzelnen Teilmodelle korrekt verknüpft werden. Dies stellt sicher, dass alle relevanten Informationen in einem zentralen Kontext zusammengeführt und genutzt werden können. Die Verknüpfung erfolgt durch die Definition eindeutiger Beziehungen zwischen den Modellen, die den gesamten Lebenszyklus des Leitungssatzes abdecken.
3. **Integration der Tier-2 VWS:** Für die vollständige Aggregation der Verbundkomponente müssen die VWS der Tier-2-Lieferanten in die 150%-Variante des Leitungssatzes integriert werden. Wichtig ist, dass der Tier-1 nicht die vollständigen VWS der Tier-2-Lieferanten erhält, sondern nur spezifische Sichten, die die relevanten Produktinformationen enthalten. Diese selektiven Sichten müssen vom jeweiligen Tier-2 bereitgestellt und vom Tier-1 in die eigene VWS integriert werden.

Um den Prozess effizienter zu gestalten, kommen Automatisierungstools zum Einsatz, wie bspw. das [VWS4LS-AASPE-Plugin](https://github.com/VWS4LS/vws4ls-aaspe-plugin)¹⁰. Solche Tools unterstützen die automatisierte Erstellung und Verknüpfung

¹⁰ <https://github.com/VWS4LS/vws4ls-aaspe-plugin>

der Teilmodelle, indem sie Standardprozesse abbilden und somit den manuellen Aufwand reduzieren, da durch den Einsatz solcher Tools viele Schritte automatisiert durchgeführt werden können, was die Konsistenz und Effizienz des gesamten Entwicklungsprozesses erheblich steigert.

3.3.5 Anwendungsfall und Voraussetzungen

Der OEM erteilt einem Tier-1-Lieferanten einen Entwicklungsauftrag und übergibt dazu eine Verwaltungsschale (VWS). Der Dateninhalt der VWS umfasst alle relevanten Informationen zum Produktmodell wie beispielsweise Komponenten, CAD-Daten, Elektrologik (ELOG) und elektrische Schaltpläne, die auf den Freigabeprozessen aus AP5.2 basieren. Der Tier-1-Lieferant analysiert diesen Entwicklungsauftrag, integriert die bereitgestellten Informationen in seine eigene Datenbank und entwickelt darauf aufbauend den Leitungssatz.

Die Leitungssatzentwicklung ist ein komplexer und iterativer Prozess, der aus mehreren Prozessschritten aufgebaut wird. Diese Prozessschritte erzeugen Zwischenartefakte, die als Input bzw. Anforderung für den nächsten Schritt verwendet werden können oder Auslöser für einen iterativen Prozess sind.

Eine strukturierte Herangehensweise gewährleistet die effiziente Koordination der komplexen Interaktionen zwischen den verschiedenen Projektbeteiligten. Die Verwaltungsschale unterstützt dabei das Änderungsmanagement und fördert die frühzeitige Einbindung von Produktions- und Montagespezifika, wodurch die Gesamteffizienz und Qualität der Leitungssatzentwicklung verbessert werden.

Die beschriebenen Entwicklungsprozesse werden nun durchgeführt, unter Verwendung der bereits verfügbaren Standard-Tools, um z.B. Leitungslängen zu ermitteln. Die dabei anfallenden Ausgangs-Daten werden dann in einem Übergangs-Szenario ermittelt und anschließend in der VWS abgelegt werden. Zu einem späteren Zeitpunkt sollte dies von den genutzten Software-Tools direkt bewerkstelligt werden. Durch den Einsatz von Automatisierungstools für die schnelle und einfache Übernahme der Daten in die VWS kann der Tier-1 erhebliche Zeit- und Kostenvorteile erzielen. Die Tools ermöglichen:

- Automatisierte Anlage von Teilmodellen: Die Teilmodelle werden auf Basis der vom OEM bereitgestellten VWS automatisch generiert und mit den relevanten Daten gefüllt.
- Effiziente Verknüpfung der Modelle: Die automatisierte Verknüpfung der Teilmodelle minimiert Fehler und stellt sicher, dass die Modelle in einem einheitlichen Kontext stehen.
- Schnelle Integration von Tier-2 VWS-Sichten: Die spezifischen Sichten der Tier-2 VWS werden effizient in die Verbundkomponente integriert, ohne dass interne Daten der Tier-2-Lieferanten offengelegt werden müssen.

Diese Schritte sind entscheidend, um die Verwaltungsschale als zentrales Element in der Leitungssatzentwicklung erfolgreich zu nutzen. Die integrierten und verknüpften Teilmodelle bilden eine konsistente Datenbasis, die für die gesamte Entwicklung und spätere Produktion genutzt werden kann.

3.4 AP 2.4 - Teilmodelle der Verwaltungsschale

Im AP 2.4 „Teilmodelle der Verwaltungsschale“ wurden Anforderungsanalysen für die VWS-Teilmodelle der in AP 2.3 definierten LS-Prozesse durchgeführt, unter Einbeziehung jeder Komponente und jedes Prozessschrittes.

Der Entwicklungsprozess ist die „Wiege“ des Digitalen Zwillings. Entlang der Anwendungsfälle werden die Abläufe auf die Verwaltungsschale und die Erzeugung interoperabler Daten und Formate für den gesamten weiteren Lebenszyklus ausgerichtet. Die VWS soll dabei nicht die etablierten Daten-Formate ersetzen, wie z.B. CAD-Dateiformate, KBL, VEC etc., sondern ergänzen und die Daten vor allem automatisiert und maschinenlesbar bereitstellen. Deshalb ist es neben der Auswahl passender existierender bzw. neu zu erstellender Teilmodelle auch wichtig, anhand von Anwendungsfällen das Zusammenspiel zu betrachten. Hierbei liegt der Fokus zuerst nur auf dem Entwicklungsprozess des Leitungssatzes. Der Übergang zu Produktion und Montage (TP3 und TP4) soll im nachfolgenden AP 2.5 betrachtet werden.

3.4.1 Teilmodelle für die Entwicklung

Die zu lösende Fragestellung ist, welche Daten zu welchem Zeitpunkt in der Verwaltungsschale (VWS) zur Verfügung stehen bzw. stehen müssen und in wie diese strukturiert sein müssen.

Hierfür wurden zu Beginn des Arbeitspaketes 2.4 für den Referenzprozess alle existierenden relevanten Daten aus anderen Teilprojekten zusammengetragen. Aus diesen Daten wurde abgeleitet, welche Entwicklungspartner und die ihnen zugeordneten Komponenten mit deren Daten in der Leitungssatzentwicklung eine Rolle spielen. Danach fand eine Betrachtung statt, welcher Entwicklungspartner eine VWS für die Komponenten im Leitungssatz und für den Leitungssatz selbst besitzt. Das Ergebnis ist in Abbildung 3-14 zu sehen. Da im Entwicklungsprozess mit Typen von Leitungssätzen gearbeitet wird, wurde hier derselbige Begriff verwendet. Eine Instanz wird in diesem Fall erst erzeugt, wenn ein Leitungssatz-Typ in die Fertigung gegeben wird und dort physisch produziert wird.

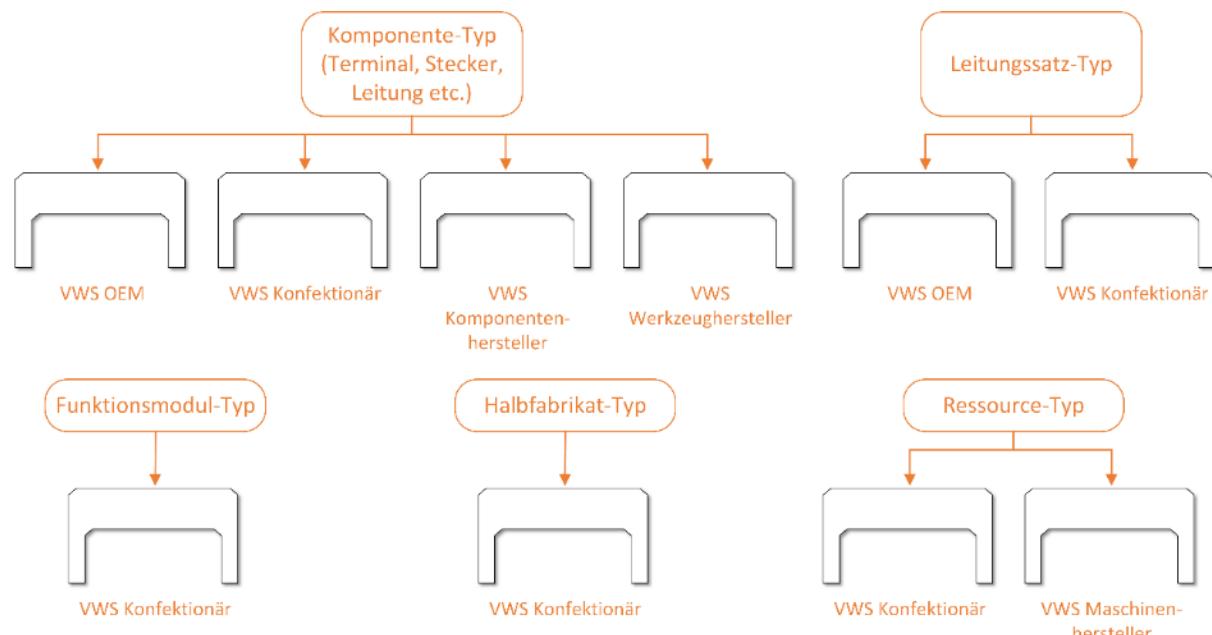


Abbildung 3-14: Assets und ihre VWS im Entwicklungsprozess des Leitungssatzes

Es wurde ein Grad an Modularität gewählt, sodass durch die Verknüpfung von VWS, wie z. B. der Leitungssatz-Typ-VWS mit den Halbfabrikat- und Komponenten-Typ-VWS, eine Wiederverwendung ermöglicht wird. Beispielsweise kann der Fall eintreten, dass ein bestimmter Halbfabrikat-Typ für mehrere Leitungssatz-Typen Anwendung findet. Neben den Komponenten-Typen für den Leitungssatz-Typ, wurde auch dem Ressourcen-Typ eine VWS seitens des Maschinenherstellers und des Konfektionärs zugewiesen, sodass der Konfektionär noch zusätzliche ressourcenrelevante Daten speichern kann, die später bei der Wahl einer für die Produktion geeigneten Maschinen-Instanz beachtet werden müssen.

Mit der Festlegung der Verwaltungsschalen, die in der Leitungssatzentwicklung benötigt werden, konnte die inhaltliche Zuordnung der Entwicklungsdaten zu den von der „Industrial Digital Twin Association e.V.“ (IDTA) spezifizierten und bereits veröffentlichten Teilmodell-Templates [27] erfolgen. Dabei trat der Fall ein, dass nicht alle zu dem Zeitpunkt veröffentlichten Teilmodelle die Abbildung der vorliegenden Inhalte ermöglichen. Es wurde daher eine Anforderungsanalyse durchgeführt, die

- alle im Prozess beteiligten Assets mit deren VWS bei den jeweiligen Entwicklungspartnern einbezieht.
- die Teilmodelle für die jeweiligen VWS mit Details zu Inhalten einzelner Teilmodelle bezogen auf die Lebenszyklusphase des Leitungssatzes (Entwicklung, Produktion und Montage) zeigt.
- die Notwendigkeit vom Teilen und Synchronisieren von Teilmodellen und VWS mit anderen Entwicklungspartnern festlegt.

Ein kleiner Ausschnitt aus der Analyse für einen Komponenten-Typen (z. B. Gehäuse, Leitung, Terminal) ist in Abbildung 3-15 zu sehen. Aus diesem Beispiel ist ersichtlich, dass sowohl OEM, Konfektionär, Komponentenhersteller und in manchen Fällen auch der Werkzeughersteller eine VWS für den Komponenten-Typen besitzt. Dabei ist in diesem Fall der Komponentenhersteller der SPoT für die Komponentendaten. Alle anderen Entwicklungspartner können ebenfalls eigene VWS zu dem Komponenten-Typ erstellen, verlinken aber immer auf die originale Komponenten-Typ-VWS des Komponentenherstellers, wenn zu dieser eine Verbindung existiert. Dies wurde in der Tabelle mit „[...]“ VWS der [...]“ in der Spalte der Teilmodelle vermerkt. Teilinhalte der originalen VWS werden seitens des Komponentenherstellers den anderen Entwicklungspartnern über Zugangsberechtigungen zugänglich gemacht.

| Asset | Stakeholder | teilmodelle | genauerer teilmodellinhalt | Sollen die Daten mit externen Partnern geteilt werden? (Bei Mehrfachauflösung, sind optimale Partner dabei) | Wie werden Daten geteilt? | Synchronisierung mit geteilten "Kopien" erforderlich? |
|---|--|--|---|---|--|---|
| Komponente (Typ) -> Terminal, Leitung, Gehäuse usw. | OEM | Dokumente | Verarbeitungsnormen | ja mit Konfektionär | | ja |
| | | | Freigabezeichnung | ja mit Komponentenhersteller | | ja |
| | | Technische Daten originale VWS der Komponentenhersteller | Anforderungen KDL Ansprechpartner für Nutzer der VWS | ja mit Konfektionär | | ja |
| | | | | ja mit Konfektionär | | ja |
| | | | | ja mit Konfektionär | | ja |
| | Konfektionär | Komponenten VWS des OEM mit Link des Komponentenherstellers | | nein | | |
| | | Dokumente | Verarbeitungsspezifikation Vorlage Freigabezeichnung 2D-Zeichnung | ja mit Konfektionär, OEM | Online-Katalog | ja |
| | | | | ja mit Konfektionär, OEM, Werkzeughersteller | Zu beachten ist, dass die Informationen in den Systemen der Kunden eingepflegt werden müssen | ja |
| | | Technische Daten Kontaktinformationen Digitales Typenschild 3D CAD Modell required Capability CarbonFootprint | ja mit Konfektionär, OEM, Werkzeughersteller | ja mit Konfektionär, OEM, Werkzeughersteller | ja | ja |
| | | | ja mit Konfektionär, OEM, Werkzeughersteller | ja mit Konfektionär, OEM, Werkzeughersteller | ja | ja |
| | | | ja mit Konfektionär, OEM, Werkzeughersteller | ja mit Konfektionär, OEM, Werkzeughersteller | eingepflegt werden müssen aber nicht verändert werden (nur lesernder Zugriff) | ja |
| | Komponentenhersteller | *Produktionsdaten* | Verarbeitungsanforderungen interne Daten | nein auf Typenbasis | Online-Katalog | - |
| | | | | | Zu beachten ist, dass die Informationen in den Systemen der Kunden eingepflegt werden müssen aber nicht verändert werden (nur lesernder Zugriff) | ja |
| | Maschinenhersteller (Werkzeughersteller) | originale VWS des Komponentenherstellers | | nein | | |

Abbildung 3-15: Ausschnitt aus der Anforderungsanalyse für VWS und deren Teilmodelle

Im Folgenden wird anhand des Beispiels des Komponenten-Typen näher auf die in der VWS enthaltenen Teilmodelle und ihren Zweck eingegangen. Zu den ersten oft verwendeten Teilmodellen gehören das digitale Typenschild [28], die Kontaktinformationen und die technischen Daten der Komponente. Weiterführend werden in die VWS auch die Dokumente, wie z.B. die Verarbeitungsspezifikationen, die Freigabezeichnung und die 2D-Zeichnung sowie das 3D CAD-Modell abgelegt. Zusätzlich wird das Teilmodell „Capability Description“ [29] integriert, um die von dem Komponenten-Typen geforderten Fähigkeiten aus den Verarbeitungsanforderungen in die VWS aufnehmen zu können. Da das Teilmodell sich zu diesem Zeitpunkt noch in Entwicklung befindet, wurde der aktuell verfügbare Arbeitsstand verwendet. Mit den aufgezählten Teilmodellen konnten nicht alle Daten abgebildet werden. Dazu gehören das Produktmodell, welches als KBL [5] oder VEC [6] vorliegt, sowie die Vorlage für die Produktionsdaten, die erst mit der Produktion der Komponente beschrieben werden. Beide sind mit * umklammert. Für das Produktmodell-Teilmodell wird per XPath-Referenzen der VEC bzw. die KBL hinterlegt, um die Daten aus dem Modell in der VWS verfügbar zu machen. Dieses Teilmodell spielt vor allem in den VWS des Leitungssatz-Typs und seiner Modul- und Halbfabrikat-Typen eine Rolle.

Von der Komponenten-Typ-VWS weg hin zu den VWS des Leitungssatz-Typs und der Halbfabrikat-Typen wird das Teilmodell „IDTA 02011: Hierarchical Structures Enabling Bill of Material“ [30] mehrfach verwendet:

- als **Stückliste** über die im Leitungssatz-Typ oder dem Halbfabrikat-Typ enthaltenen Assets, mit Referenzen zu dem Produktmodell-Teilmodell.
- als **Konfigurationsliste** der möglichen Varianten, aus den verschiedenen Komponenten- und Halbfabrikat-Typen.
- als **Produktionsliste** mit den bestellten Varianten, die dann an den Produktionsprozess übergeben werden.

Das Konzept zur Struktur der VWS und der Zugriffsberechtigungen in der Wertschöpfungskette sind dem Abschnitt „Architektur AAS“ zu entnehmen.

Mit den aus der Anforderungsanalyse gewonnenen Inhalten konnten die Schnittstellen und auszutauschenden Inhalte sowohl zwischen den einzelnen Entwicklungspartnern als auch zwischen den Lebensphasen des Leitungssatzes herausgearbeitet werden. So ist beispielsweise in Abbildung 3-16 der Austausch der Leitungssatz-Typ-VWS zwischen dem OEM und dem Konfektionär zu sehen. Die orange hinterlegten VWS sind durch ihre Teilmodelle gekennzeichnet, wobei deren Zugriff für externe entweder durch grüne (Zugriff gewehrt) oder rote (Zugriff verboten) Balken gekennzeichnet ist. Bei dem 3D CAD-Modell gibt es z. B. eine Mischung dieser beiden Zugriffsrechte, da die Inhalte des Teilmodells für den Konfektionär manchmal nur direkt bei dem OEM zugriffsbereit liegen. Ansonsten kann der Konfektionär sich die VWS mit allen für ihn freigegebenen Teilmodellen vom OEM herunterladen und in seinem System speichern.

Die Leitungssatz-Typ-VWS des OEM bleibt von dem Konfektionär unberührt und wird mit Hilfe der *specificAssetId* der VWS von der Leitungssatz-Typ-VWS des Konfektionärs referenziert. Gründe dafür liegen in der einfacheren Synchronisation von Änderungen seitens des Datenbereitstellers, wie hier dem OEM. Genaue Informationen dazu und zu der Verlinkung sind im Abschnitt „Architektur AAS“ zu finden.

Basierend auf der vom Leitungssatz-Typ-VWS des OEM abgeleiteten VWS zu seinem eigenen Leitungssatz-Typ, kann der Konfektionär mit der Erstellung und Beschreibung der VWS für die Funktionsmodul-, Halbfabrikat- und Komponenten-Typen beginnen. Die weiteren Austauschprozesse mit den dazugehörigen VWS sind bereits in dem Abschnitt „Architektur AAS“ detailliert beschrieben.

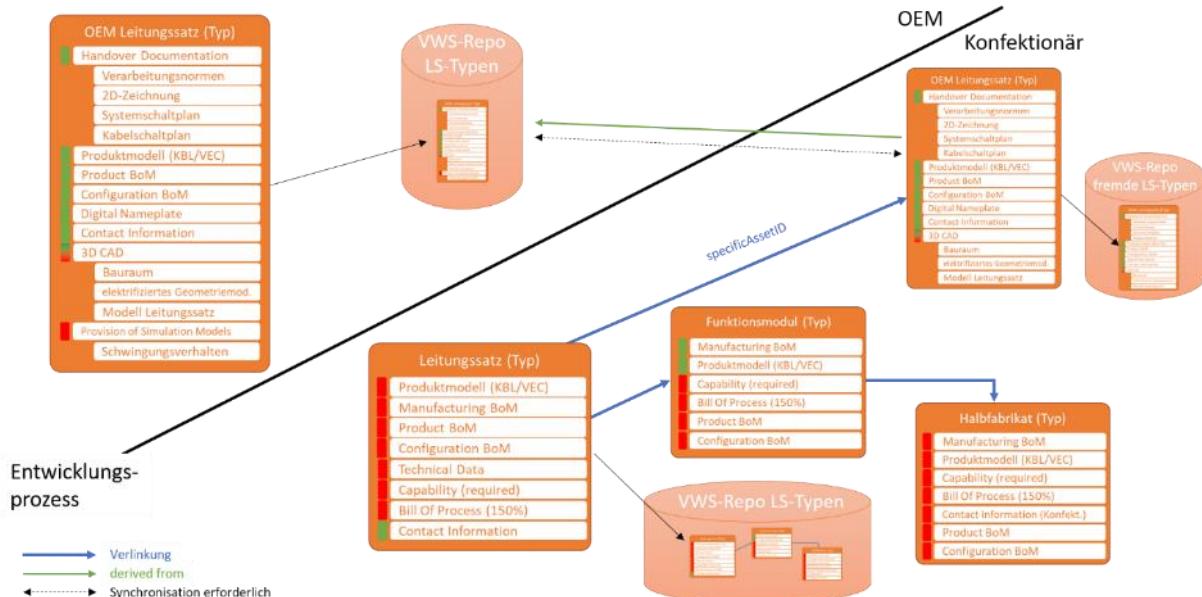


Abbildung 3-16: VWS-Schnittstellen im Entwicklungsprozess zwischen OEM und Konfektionär

3.5 AP 2.5 - Umsetzung Digital Twin LS

Im Rahmen des Arbeitspakets 2.5 „Umsetzung Digital Twin LS“ wurde das im vorherigen Arbeitspaket 2.4 definierte Vorgehen zur Erstellung der Typ-Verwaltungsschale für den Leitungssatz beim Konfektionär sowie die definierte Aufteilung der Informationen in unterschiedliche Teilmodelle auf ihre Anwendbarkeit in der Leitungssatzentwicklung erarbeitet. Im Vordergrund stand dabei die Rolle der VWS im Gesamtentwicklungsprozess, die Integration der Komponenten-Verwaltungsschalen sowie der Informationsaustausch über Unternehmensgrenzen hinweg, mit dem Ziel, den gesamten Entwicklungsablauf effizienter, transparenter und besser strukturiert zu gestalten.

3.5.1 Der Anwendungsfall

Um den Workflow zu verproben, wurde der Beispieleitungssatz aus dem PPR-Workshop verwendet (siehe Abbildung 3-17). Da dieser Beispieleitungssatz bisher nur als schematische Illustration vorhanden war, wurden im Vorfeld für alle notwendigen Komponenten die entsprechenden Teile inklusive der Teilenummern definiert. Um möglichst detaillierte Information zu allen Komponenten zur Verfügung zu haben, wurden vorzugsweise Teile verwendet, die von den VWS4LS-Projektteilnehmern hergestellt werden. Darauf aufbauend wurden alle notwendigen Leitungssatzdaten (Schaltplan, 3D-Modell, VEC usw.) erzeugt.

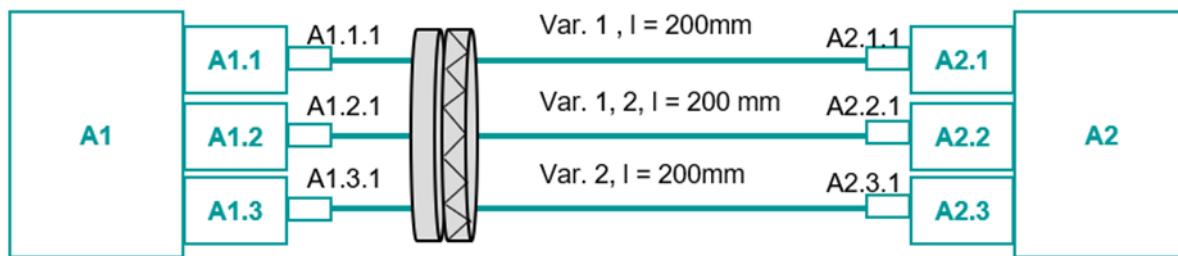


Abbildung 3-17: Beispieleitungssatz aus dem PPR-Workshop

Um die zuvor definierten Teilmodelle für den Entwicklungsprozess mit Leitungssatzdaten zu verproben und mit entsprechenden Daten zu befüllen, musste ein Leitungssatz im VEC-Format erstellt werden (siehe [DRX00012_0002.02_AO_2023_09_08.vec](#)¹¹).

Mit den festgelegten Teilen wurde der Entwicklungsprozess entsprechend durchgeführt.

Die zur Verfügung gestellten abstrahierten 3D-Modelle der Komponenten wurden für die Erstellung des DMU-Modells des Leitungssatzes verwendet. Da hier das Konzept eines 3D-Master Prozesses angewendet wurde, sind alle relevanten Daten des Leitungssatzes, bis auf die elektrologische Verschaltung, im DMU-Modell einzuarbeiten.

Zusätzlich wurden der entsprechende Schaltplan für den Leitungssatz erstellt, um die elektrologische Verbindung zu erhalten, die wie zuvor erwähnt nicht im 3D entsteht.

Im anschließenden Schritt werden die Daten aus DMU und Schaltplan zusammengeführt, um eine vollumfängliche Beschreibung des Leitungssatzes zu erhalten. Im letzten Schritt wird dann noch die sogenannte Modularisierung gemacht. Hierbei wird der 150%-Umfang des Leitungssatzes in sogenannte Funktionsmodule partitioniert.

Als Output des Entwicklungsprozesses wird nach heutigen Stand eine 2D-Zeichnung (SVG) und eine KBL-Datei erzeugt (siehe Inhalt von [DRX00012_0002.02_AO_2023_09_08.hcv](#)). Abbildung 3-18 zeigt den Leitungssatz dargestellt in der SVG-Zeichnung. In der KBL sind alle Daten zum Leitungssatz in einer standardisierten Form abgelegt und dient als Austauschformat zwischen OEM und Konfektionär.

¹¹https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP02/Beispieldaten/DRX00012_0002.02_AO_2023_09_08.vec

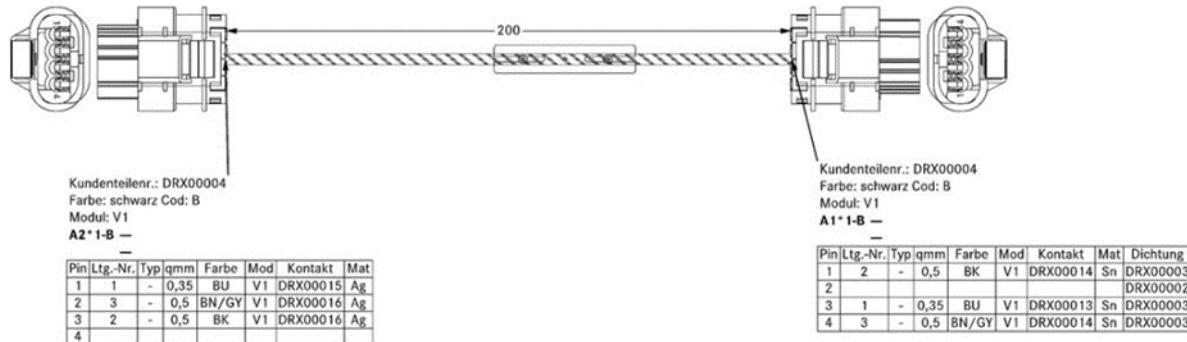


Abbildung 3-18: SVG-Darstellung des Leitungssatzes

Die nachfolgende Abbildung 3-19 zeigt die Verbindungsliste (WIRE) und die Materialliste (BOM) für den Leitungssatz, aufgeteilt nach den jeweiligen Varianten/Modulen. Je nach Kundenanforderung müssen die WIRE und die BOM auch auf der SVG-Zeichnung dargestellt werden.

| V1 V1 | | V2 V2 | | | | | | | | | | V3 V3 | | | | | | | | | |
|----------|---------|-------|---------|---------|----------|----------|----------|-----------|-------------|--------------|--------------|---------------|---------------------|------------------|-----------------------------|----------------|---------|-------|--|--|--|
| Ltg.-Nr. | Typ/qmm | Farbe | Von REF | Von Pin | Von Feld | Nach REF | Nach Pin | Nach Feld | Von Kontakt | Von Dichtung | Nach Kontakt | Nach Dichtung | Länge ohne Zuschlag | Kundenteilenr. | Benennung | Teilezeichnung | Einheit | Menge | | | |
| 1 | - | 0,35 | BU | A1*1-B | 3 | 4A | A2*1-B | 2A | DRX00013 | DRX00003 | DRX00015 | 200 | DRX00001 | Fixiere, 9_MM_BK | DRX00001 | Sek. | 2 | | | | |
| 2 | - | 0,5 | BK | A2*1-B | 2 | 2A | A1*1-B | 1 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00002 | Blindstöpsel orange 3,4 | DRX00002 | Sek. | 1 | | | |
| 3 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00003 | ELA gelb 1,1-2,1 | DRX00003 | Sek. | 3 | | | |
| 4 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00004 | STECKHÜLSENGEHÄUUSE | DRX00004 | Sek. | 2 | | | |
| 5 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00005 | Leitungsbefestiger | DRX00005 | Sek. | 1 | | | |
| 6 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00006 | DRX00006 | DRX00006 | Sek. | 1 | | | |
| 7 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00007 | BUchse MLK 1,2 Sn 0,22-0,35 | DRX00007 | Sek. | 1 | | | |
| 8 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00008 | BUchse MLK 1,2 Sn 0,5-0,7 | DRX00008 | Sek. | 1 | | | |
| 9 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00009 | BUchse MLK 1,2 Ag 0,22-0,35 | DRX00009 | Sek. | 2 | | | |
| 10 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00010 | BUchse MLK 1,2 Ag 0,5-0,5 | DRX00010 | Sek. | 1 | | | |
| 11 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00011 | DRX00011 | DRX00011 | mm | 200 | | | |
| 12 | - | 0,5 | BN/GY | A2*1-B | 2 | 2A | A1*1-B | 4 | 4A | DRX00016 | DRX00014 | DRX00003 | 200 | DRX00012 | DRX00012 | DRX00012 | mm | 200 | | | |

Abbildung 3-19: WIRE und BOM mit dargestellter Modularisierung (V1, V2 und V3)

Zu einem frühen Zeitpunkt des Projektes wurde sich darauf verständigt, das VEC-Format zur Beschreibung des Leitungssatzes zu verwenden. Hintergrund dieser Entscheidung ist, dass der VEC mehr Informationsfelder zur Beschreibung des Leitungssatzes zur Verfügung stellt.

Allerdings sind viele heutige gebräuchliche Leitungssatzentwicklungstools noch nicht befähigt, eine Produktbeschreibung des Leitungssatzes im VEC-Format auszugeben.

Daher musste die erzeugte KBL-Datei mittels eines Konverters in das VEC-Schema überführt werden, mit dem Nachteil, dass der Inhalt der VEC zur Beschreibung des Produktes dem der KBL entspricht und somit die Vorteile des VEC-Formats brachliegen.

3.5.2 Voraussetzungen für die Umsetzung

Um diesen Anwendungsfall erfolgreich umsetzen zu können, sind generell folgende Voraussetzungen notwendig:

- Vorhandensein von VWS für alle relevanten Komponenten:** Es wird vorausgesetzt, dass alle notwendigen Komponenten des PPR-Beispiels (Abbildung 3-17) in Form von Verwaltungsschalen bereits vorhanden sind und nicht erst erzeugt werden müssen.
- Zugriff auf spezifische VWS-Sichten der Tier-2-Lieferanten:** Der Tier-1 muss in der Lage sein, die spezifischen Produktinformationen der VWSen der Tier-2-Lieferanten zu erhalten, um diese in sein Repository zu integrieren und anschließend in die Entwicklungsdaten einzubinden. Dies erfordert eine koordinierte Bereitstellung der entsprechenden Informationen durch die Tier-2-Lieferanten.
- Infrastruktur für dezentrale VWS-Verwaltung:** Eine dezentrale Infrastruktur, die die Integration und Verwaltung der Verwaltungsschalen unterstützt, ist notwendig, um die Flexibilität und Skalierbarkeit der Leitungssatzentwicklung zu gewährleisten.

- **Etablierung eines kollaborativen Datenmodells:** Ein kollaboratives Datenmodell, das als Single-Point-of-Truth fungiert, muss etabliert sein, um eine nahtlose Integration aller relevanten Datenquellen und Prozesse sicherzustellen.

3.5.3 Verwaltungsschalen der Komponenten

Beim Stecksystem wurde der MLK 1,2 ausgewählt (Abbildung 3-20), da dieser nicht im vollen Umfang der Geheimhaltungspflicht von KOSTAL Kontakt Systeme GmbH & Co.KG unterliegt, sondern von weiteren Mitbewerbern aus dem Arbeitskreis genutzt werden kann. Die zugehörige VWS-Datei ist die „[VWS4LS_Terminal_MLK_1-2_0.5-0.75mm_2024_01.aasx](#)“¹² (Abbildung 3-21).

Es wurde ein 4-poliges Gehäuse ausgewählt, das für beide Seiten eingesetzt wird. Allerdings ist die Seite A2 ungedichtet und das Gehäuse der Seite A1 gedichtet. Das heißt die Leitungen auf der Seite A1 werden mit Einzelleitungsdichtungen (ELA) verarbeitet und die nichtbelegten Kammern mit sogenannten Blindstopfen bestückt.

Für dieses Beispiel sind Terminals mit vier verschiedenen Teilenummern notwendig. Dies ergibt sich durch die eingesetzten Querschnitte der Leitungen von 0,35 und 0,5 mm². Dafür sind zwei unterschiedliche Terminals notwendig, da der Crimpbereich einmal für 0,35mm² verwendbar ist und ein weiteres für 0,5 - 0,75mm². Nochmals zwei weitere Varianten kommen hinzu, da der Einsatz der ELA-Terminals einen anderen Isolationscrimpbereich erfordern.

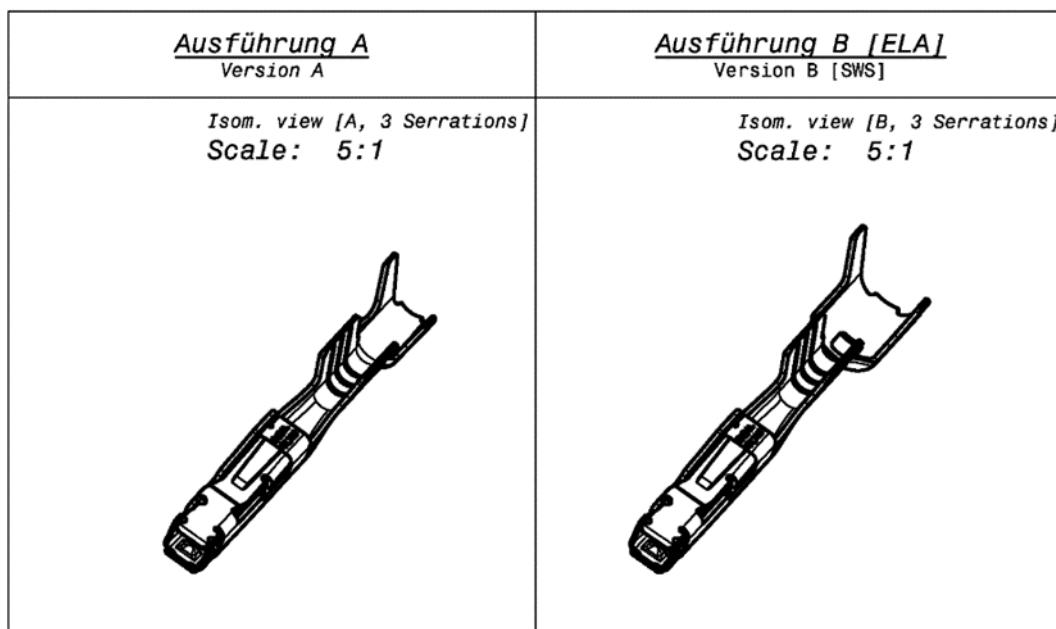


Abbildung 3-20: MLK 1,2 Terminal mit/ohne ELA (KOSTAL Kontakt Systeme GmbH & Co. KG)

Um die einzelnen Typ-Verwaltungsschalten aus Tier-2 Sicht zu erstellen, wurden als Input die Verarbeitungsspezifikationen, das Datenblatt und die dazugehörigen neutralen Kundenzeichnungen verwendet.

¹² https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP02/Beispieldaten/VWS4LS_Terminal_MLK_1-2_0.5-0.75mm_2024_01.aasx

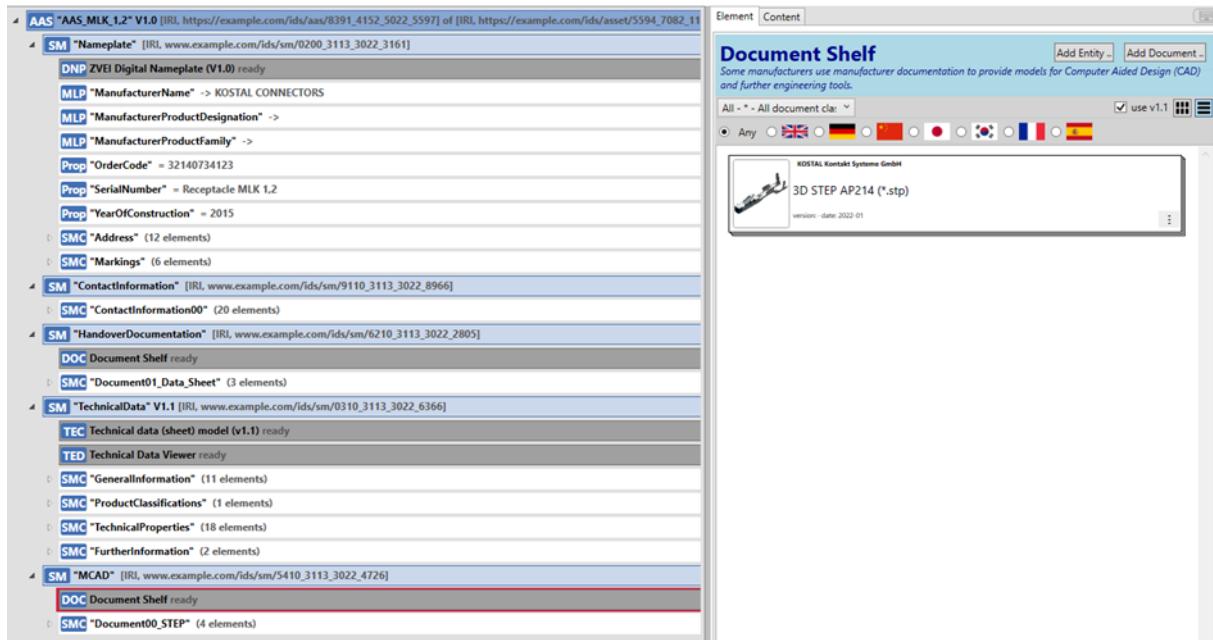


Abbildung 3-21: Auszug aus einer Typ-Verwaltungsschale Terminal

Um das PPR-Beispiel zu vervollständigen, wurden auch für das Halteteil und die Bandagierung jeweils eine entsprechende Teilenummer festgelegt.

Typischerweise haben unterschiedliche Partner der Wertschöpfungskette unterschiedliche Sichtweisen auf eine Komponente. Dies fängt bei unterschiedlichen (internen) Teile- bzw. Sachnummern an und geht bis hin zu unterschiedlichen Grenzwerten für den Einsatz einer Komponente. Um dies abzubilden, wurden für jede Komponente nicht nur eine sondern jeweils drei Verwaltungsschalen inkl. zugehöriger Assets erstellt, um die unterschiedlichen Sichtweisen (OEM, Tier1, Tier2) abbilden zu können. Dabei wurde der im Rahmen von TP5.2 entwickelte Mechanismus zur Äquivalenz-Beschreibung zwischen den verschiedenen Assets genutzt. Abbildung 3-22 zeigt dies Anhand der Tier1-Verwaltungsschale für eine Komponente, welche per „*specificAssetID*“ auf die äquivalenten Assets sowohl bei einem OEM als auch bei einem Tier2 verweist. Dies erlaubt das automatische Sachnummern-Mapping, d.h. das automatische Auffinden eines äquivalenten Assets/Verwaltungsschale eines anderen Partners der Wertschöpfungskette.

| Element | Content |
|--|---|
| AssetAdministrationShell (according IEC63278) | |
| Referable: | |
| idShort: | Draexlmaier_200018565 |
| HasExtension: | |
| Identifiable: | |
| id: | www.draexlmaier.com/ids/aas/9093_3151_2132_7387 |
| id (Base64): | d3d3LmRyYWV4bG1haWVyLmNvbS9pZHMvYWFzLzkwOTNfMzE1MV8yMTMyXczODc= |
| HasDataSpecification (Reference): | |
| AssetInformation | |
| Kind (of AssetInformation): | |
| kind: | Type |
| globalAssetId: | |
| globalAssetId: | www.draexlmaier.com/ids/asset/3330_0010_1010_5147 |
| assetType: | |
| specificAssetId: | |
| Element 1: partNumber | |
| semanticId: | (GlobalReference) 0173-1#02-AAO676#003 |
| name: | partNumber |
| value: | 200018565 |
| externalSubjectId: | (GlobalReference) www.draexlmaier.com |
| Element 2: partNumber | |
| semanticId: | (GlobalReference) 0173-1#02-AAO676#003 |
| name: | partNumber |
| value: | DRX00017 |
| externalSubjectId: | (GlobalReference) www.oem.com |
| Element 3: partNumber | |
| semanticId: | (GlobalReference) 0173-1#02-AAO676#003 |
| name: | partNumber |
| value: | YA00356565 |
| externalSubjectId: | (GlobalReference) www.coficab.com |

Abbildung 3-22: Äquivalenzbeziehungen per „specificAssetId“

3.5.4 Erstellung Verbundkomponente

Auf Basis der erstellten Daten für den Gesamtleitungssatz wurde anschließend die zugehörige Typ-Verwaltungsschale aus Sicht des OEM mit entsprechenden Teilmodellen (vgl. Abbildung 3-16) erstellt, die sog. „Verbundkomponente“ (VBK). Abbildung 3-23 zeigt die Verwaltungsschale „[Leitungssatz und Komponenten Typ OEM.aasx](#)¹³ sowie exemplarisch die dort enthaltene 2D-Zeichnung (vgl. Abbildung 3-18) als Teil des „HandoverDocumentation“-Teilmodells.

¹³ https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP02/Beispieldaten/Leitungssatz_und_Komponenten_Typ_OEM.aasx

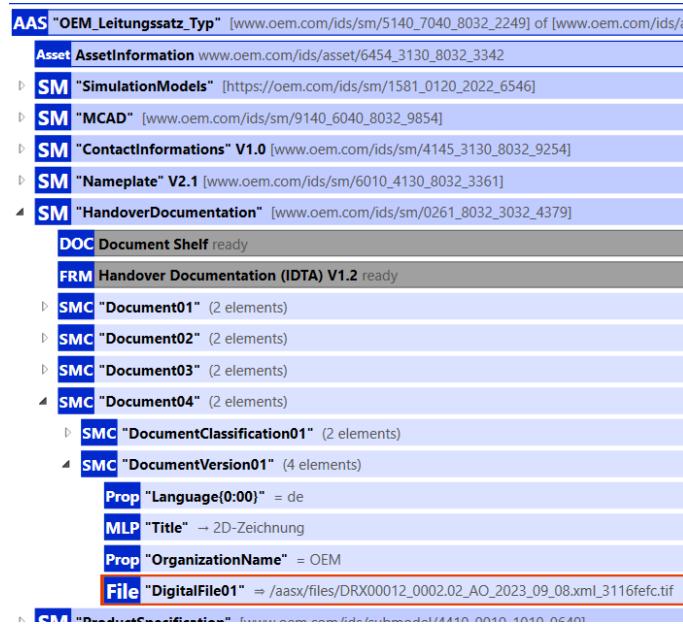


Abbildung 3-23: OEM-Verwaltungsschale für den Gesamtleitungssatz

Durch das im Rahmen von TP5.2 sowie des Architekturteams entwickelte [VWS4LS-AASPE-Plugin¹⁴](#) für den AASX Explorer konnte aus der importierten Produktspezifikation in Form des VEC-Modells automatisch die Product-BOM erstellt und mit den Verwaltungsschalen der Einzelkomponenten verknüpft werden. Abbildung 3-24 zeigt dies anhand der Komponente (bzw. der Komponenten-Verwendung) „A1*1-B“, welches über die sog. *GlobalAssetId* mit dem entsprechenden Komponenten-Typ verknüpft ist.

| | |
|---|---|
| SM "LS_Product_BOM_01" [www.oem.com/ids/s] | Supplemental Semantic IDs: |
| BOM Bill of Material - Graph display ready | Qualifiable: |
| Prop "ArcheType" = Full | HasDataSpecification (Reference): |
| Ent "EntryNode" | ConceptDescription cannot be looked up within the AAS environment! |
| Ent "A1*1-B" | Entity |
| Rel "HasPart_A1*1-B" | Statements: Please add statements via editing of sub-ordinate entities |
| Ent "NOT-STABLE-d2e1182" | entityType: SelfManagedEntity |
| Rel "HasPart_NOT-STABLE-d2e1182" | GlobalAssetId (in case of self managed asset, preferred) |
| Ent "NOT-STABLE-d2e1189" | globalAssetId: www.oem.com/ids/asset/9100_0010_1010_4810 |

Abbildung 3-24: Verknüpfung Komponenten-Verwendung mit Komponenten-Typ per GlobalAssetId

Ebenfalls unter Zuhilfenahme des [VWS4LS-AASPE-Plugin](#) konnte anschließend die Ableitung der Leitungssatz-Verwaltungsschale für den Konfektionär durchgeführt werden (vgl. Abbildung 3-16). Hierbei wurden die im Rahmen des Architekturteams entwickelten Prinzipien zur Verlinkung der unterschiedlichen Sichten angewendet. So verweisen z.B. Elemente in der Produkt-BOM in der Verwaltungsschale des Konfektionärs auf die äquivalenten Elemente in der Produkt-BOM des OEMs. Diese Duplizierung mit Verlinkung ermöglicht einerseits die Nachvollziehbarkeit des Ursprungs von Komponenten, andererseits aber auch die Anpassung der Produkt-BOM durch den Konfektionär z.B. durch Ergänzen von Elementen, die vom OEM nicht betrachtet werden (Klebebänder oder andere für Produktion/Transport notwendige Komponenten).

Auf Basis der OEM-Verwaltungsschale für den Leitungssatz-Typ konnten anschließend Schritt für Schritt benötigte Halbfabrikate abgeleitet werden. Abbildung 3-25 zeigt ein solches Halbfabrikat: eine geschnittene Leitung. Auch hier wurden wieder die im Rahmen des Architekturteams entwickelten Strukturen bzgl. der Verlinkung von Elementen angewendet. Abbildung 3-25 zeigt dies durch die Manufacturing-BOM des Halbfabrikats: Für dessen Herstellung wird lediglich eine einzige Komponente benötigt

¹⁴ <https://github.com/VWS4LS/vws4ls-aaspe-plugin>

– die Leitung als Ausgangsmaterial. Diese Leitung ist jedoch über eine entsprechende *SameAs*-Beziehung aus der Manufacturing-BOM des Gesamt-Leitungssatzes referenziert, wodurch eine durchgängige Erkundbarkeit der Produktstruktur vom Gesamtleitungssatz bis hin zu den Halbfabrikaten auf den untersten Ebenen gewährleistet wird.

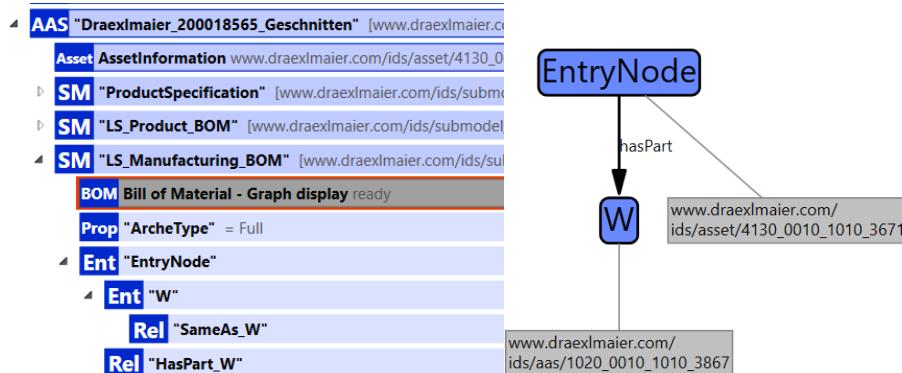


Abbildung 3-25: Verwaltungsschale für ein Halbfabrikat "Geschnittene Leitung"

Die Integration der verschiedenen Teilmodelle von unterschiedlichen Zulieferern in die Verwaltungsschale wurde erfolgreich in Verbindung mit der [BaSyx](#)¹⁵ Middleware umgesetzt, die als zentrale Plattform für die Verwaltung und Integration der Verwaltungsschalen diente.

Die BaSyx Middleware ermöglichte eine flexible und skalierbare Integration der Teilmodelle der verschiedenen Zulieferer. Im Rahmen dessen wurden spezifische Sichten der Verwaltungsschalen von Tier-2 Lieferanten eingebunden, sodass lediglich die relevanten Produktinformationen, jedoch ohne interne Produktionsdaten, in die zentrale Verwaltungsschale des Tier-1 integriert wurden. Dies sicherte den Schutz sensibler Daten und gewährleistete gleichzeitig die Konsistenz und Verfügbarkeit für die Leitungsschatzentwicklung notwendigen Informationen.

3.5.5 Fähigkeitsabgleich der Produktionsmaschinen

Ein weiterer Aspekt bei der Integration war die Abstimmung zwischen den produktseitigen Anforderungen und den verfügbaren Fähigkeiten der Produktionsmaschinen. In einem ersten Schritt wurde eine Analyse durchgeführt, um die erforderlichen Fähigkeiten des Produkts für die Produktion zu ermitteln. Darauf aufbauend wurde untersucht, welche Fähigkeiten auf Seiten der Produktionsmaschinen vorhanden sind. In dem Implementierungsschritt bzw. dem Demonstrator wurden zwei verschiedene Produktionsmaschinen über die Verwaltungsschale angebunden und deren Fähigkeiten modelliert.

Durch den Abgleich der Produktanforderungen mit den Maschinenfähigkeiten konnten die Produktionsressourcen optimal genutzt und die Produktionseffizienz gesteigert werden. Die Verwaltungsschalen der Maschinen enthielten detaillierte Informationen zu ihren Fähigkeiten, die über die BaSyx Middleware zentral verwaltet und den spezifischen Anforderungen des Leitungssatzes gegenübergestellt wurden. Dieser Abgleich ermöglichte eine zielgerichtete Planung und Steuerung der Produktionsprozesse.

¹⁵ <https://wiki.basyx.org>

3.5.6 Abschlussdemonstrator

Das im Rahmen von AP 2.5 modellierte System wurde auf den Abschlussdemonstrator übertragen, um dessen Interaktion mit den anderen Teilprojekten zu analysieren. Der Demonstrator diente dabei als Testumgebung, in der die Integration der VWSen sowie die Interaktion der Teilmodelle in einer realistischen Umgebung nach dem folgenden Schema evaluiert wurden:

1. **Projektion des Modells auf den Demonstrator:** Dies beinhaltete die Implementierung der Verwaltungsschale und der Teilmodelle in der Testumgebung, um die reale Anwendung der entwickelten Struktur zu überprüfen.
2. **Testen der Integration:** Im Demonstrator wurden alle relevanten Schnittstellen und Interaktionen zwischen den verschiedenen Teilprojekten getestet. Hierbei wurden insbesondere die Verknüpfungen zwischen den Teilmodellen und deren Integration in die BaSx Middleware überprüft.
3. **Erfassung der Resultate:** Die durchgeführte Tests ermöglichte wertvolle Einblicke in die Funktionalität der Verwaltungsschale und der Teilmodelle im Zusammenspiel. Die Ergebnisse zeigten, wie gut die Teilmodelle integriert werden konnten, und identifizierten mögliche Schwachstellen oder Optimierungspotenziale.
4. **Optimierungsmaßnahmen:** Auf Basis der Testergebnisse wurden Anpassungen und Verbesserungen am System vorgenommen. Diese Maßnahmen dienten dazu, die Effizienz der Integration zu erhöhen und sicherzustellen, dass alle Komponenten nahtlos zusammenarbeiten.

3.6 Fazit

Das übergeordnete Ziel des gesamten Vorhabens war es, eine dezentrale Datenhaltung zu ermöglichen, während gleichzeitig eine Aggregation der Komponentenverwaltungsschalen erreicht wird. Durch die erfolgreiche Umsetzung konnte dieses Ziel weitgehend realisiert werden. Die wesentlichen Ergebnisse umfassen:

1. **Integration der Teilmodelle:** Die Verwaltungsschale des Tier-1 konnte erfolgreich mit spezifischen Sichten der Zulieferer-Verwaltungsschalen angereichert werden, ohne sensible interne Informationen offenzulegen. Die dezentrale Datenhaltung wurde beibehalten, während die Aggregation der relevanten Daten durch die zentrale Verwaltungsschale ermöglicht wurde.
2. **Fähigkeitsabgleich** zur Optimierung der Produktion. Die Verwaltungsschalen der Maschinen ermöglichen eine präzise Zuordnung der Produktionsressourcen.
3. **Ergebnisse des Abschlussdemonstrators** bestätigten die erfolgreiche Integration und Interaktion der Teilmodelle. Gezielte Optimierungsmaßnahmen bestätigten die Funktionalität der entwickelten Struktur.
4. **Einsatz der BaSx Middleware** als zentrale Plattform für die Kommunikation und Datenverarbeitung zwischen den verschiedenen Akteuren.
5. **Einsatz des AASX-Explorers** und des in VWS4LS entwickelten [VWS4LS-AASPE-Plugin](#) unterstützte die schnelle und fehlerfreie Erstellung der VWS, was sehr hilfreich war, um effizient LS-VWSen erzeugen zu können. Allerdings ist der Package Explorer als Prototyping-Tool und nicht für Produktivanhwendungen gedacht, daher wäre eine Integration der Funktionalität zur Generierung von LS-VWSen in die Engineering-Applikations-Toolkette wünschenswert. Denkbar wäre bspw. eine BaSx-Backend-Funktionalität analog zu dem AASPE-Plugin, mit der anhand einer vorliegenden KBL- oder VEC-Datei eine VWS mit den notwendigen Submodellen erzeugt und möglichst weitgehend inhaltlich vorbefüllt werden kann.

Insgesamt zeigte das TP2, dass die Verwaltungsschale als zentrales Werkzeug zur Steuerung und Optimierung der Leitungssatzentwicklung und der zugehörigen Produktionsprozesse eingesetzt werden kann. Die Kombination aus Verwaltungsschalen, Middleware und Fähigkeitsabgleich kann wesentlich zur Effizienzsteigerung und Flexibilität in der Produktionsplanung beitragen.

Die erfolgreich durchgeführten Tests im Abschlussdemonstrator bestätigten das Ziel der erfolgreichen Integration einer dezentralen Datenhaltung mit zentraler Aggregation der Verwaltungsschalen. Die in dem Zusammenspiel aller Komponenten gewonnenen Erkenntnisse wurden final dokumentiert, um auch über der Laufzeit dieses Projektes hinaus zur Verfügung zu stehen.

4 TP3 - Produktionsprozesse des Leitungssatzes

Im Teilprojekt 3 "Produktionsprozesse des Leitungssatzes" wurden Konzepte und Verfahren entwickelt, welche die Informationen der Verwaltungsschalen der Produkte, Ressourcen und Prozessbeschreibungen sowie den aktuellen Anlagenzuständen kombinieren, aufbereiten und verarbeiten, um automatisiert die Produktionsmaschinen bzw. deren Montage-Skills zu konfigurieren und zu parametrieren.

Kernthema war daher die Erforschung eines Konzeptes bzw. Modelles, wie unter diesen Bedingungen der Digitale Zwilling des gesamten Leitungssatzes angewendet werden kann.

Das Teilprojekt wurde in folgende Arbeitspakete aufgeteilt, deren Resultate in diesem Dokument zusammenfassend erläutert werden:

- AP 3.1 - Untersuchung von Anforderungen
- AP 3.2 - Erforschung eines Referenzmodells
- AP 3.3 - Erforschung von Teilmodellen für Rückmelddaten
- AP 3.4 - Proof of Concept
- AP 3.5 - Validierung der Prinzipien

4.1 AP 3.1 - Untersuchung von Anforderungen

Im Arbeitspaket 3.1 „Untersuchung von Anforderungen“ lag der primäre Fokus auf der Erfassung und Analyse aller Anforderungen für die Anwendung des digitalen Zwillings eines Leitungssatzes unter Nutzung der Verwaltungsschale. Dies erfolgte durch explizite Partnerzusammenarbeit innerhalb des Projektes, um eine Anforderungsbasis zu erstellen. Auf diese Basis wurde im weiteren Projektverlauf zurückgegriffen, um die Berücksichtigung bzw. Implementierung der einzelnen Anforderungen zu prüfen.

4.1.1 Vorgehen

Für die Anforderungsspezifikation wurde ein fünfstufiges Phasenmodell verfolgt. Abbildung 4-1 zeigt die einzelnen Phase und welche Teilprojekte bzw. Partner für die jeweilige Phase verantwortlich waren. In den folgenden Unterkapiteln wird jede einzelne Phase genauer beschrieben.

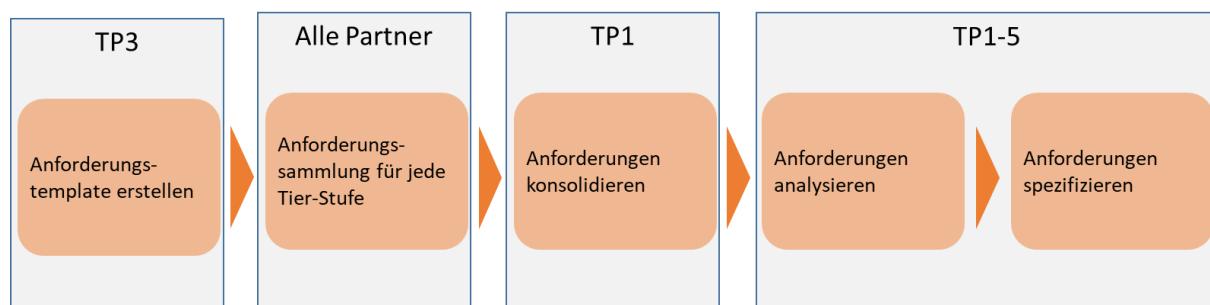


Abbildung 4-1: Phasenmodell für die Anforderungsspezifikation

4.1.1.1 Phase 1 - Anforderungstemplate erstellen

Die Anforderungserhebung wurde für mehrere Teilprojekte geplant, um möglichst zum Projektstart eine solide Anforderungsbasis zu schaffen. In TP3 wurde hierfür ein einheitliches Anforderungstemplate erarbeitet und allen TPs zur Verfügung gestellt. Das Anforderungstemplate wurde auch als Grundlage für die Anforderungsdokumentation entwickelt, damit für die später folgende Anforderungskonsolidierung die Zusammenführung aller Anforderungen in einem Dokument effizient erfolgen konnte.

Das Template wurde in einer Excel-Datei aufgebaut und beinhaltet folgende Informationen für jede Anforderung:

- **ID**
Ein eindeutiger Identifier, bestehend aus den beiden ersten Buchstaben des Vor- und Nachnamens und einer laufenden Nummer. Beispiel: Max Mustermann erstellt seine erste Anforderung mit ID „MaMu-01“, die Zweite mit „MaMu-02“, und so weiter.
- **Titel**
Die Kurzbezeichnung der Anforderung.
- **Beschreibung**
Eine ausführliche Beschreibung der Anforderung mit Bezug zu den jeweiligen Anwendungsfällen und wichtigen Kriterien.
- **PPR-Kategorie**
Eine Zuordnung der Anforderung in die drei Kategorien Produkt, Prozess und Ressource, wobei Prozess in die Unterkategorien Entwicklungsprozess, Produktionsprozess und Montageprozess unterteilt wurde.
- **Wertschöpfungsstufe**
Eine Zuordnung der Anforderung zu einer oder mehreren Wertschöpfungsstufen, als Unterkategorisierung der PPR-Kategorien (Entwicklung, Produktion, Montage, etc.).
- **Datenbedarf**
Daten, die für die Umsetzung der Anforderung zwingend notwendig sind.
- **Priorisierung**
Einordnung der Anforderung in die Kategorien: Muss (Gesetzlich), Soll oder Wunsch.
- **Link zu User Story**
Wenn zu einer Anforderung noch eine User Story beschrieben wurde, konnte diese hier verlinkt werden.
- **Unternehmen, Autor, E-Mail**
Informationen zum Anforderungsersteller.

Zusätzlich wurden Felder für die Teilprojekte 1-8 und den Use Cases 1-5 hinzugefügt, um die Anforderung relevanten Teilprojekten bzw. Use Cases zuordnen zu können.

4.1.1.2 Phase 2 - Anforderungssammlung

In Phase 2 wurde das Anforderungstemplate allen Konsortialpartnern für die Anforderungssammlung zur Verfügung gestellt. Der Zeitraum für die Anforderungssammlung betrug 4 Wochen. In diesem Zeitraum wurden unternehmensinterne Anforderungen aus den verschiedenen Fachbereichen vom jeweiligen Konsortialpartner gesammelt und in das Anforderungstemplate eingetragen. Zum Ende dieser Phase wurden alle Anforderungsdokumente zentral abgelegt.

4.1.1.3 Phase 3 - Anforderungskonsolidierung

Im Rahmen der Anforderungskonsolidierung wurden die einzelnen Anforderungsdokumente der Konsortialpartner strukturell bereinigt und zu einem partnerübergreifenden Dokument zusammengefasst. Das vorgegebene Template ermöglichte hierbei ein unkompliziertes Zusammenführen aller Einzeldokumente in einem Gesamtdokument. Nach Beendigung der Anforderungskonsolidierung konnten insgesamt 166 Anforderungen in einer zusammengefassten Anforderungsliste integriert werden.

4.1.1.4 Phase 4 - Anforderungsanalyse

Für die Anforderungsanalyse wurde auf Grundlage der einheitlichen Anforderungsliste eine Arbeitsgruppe eingesetzt. Diese hatte die Aufgabe alle Anforderungen auf Vollständigkeit und Verständlichkeit zu prüfen. Unklare oder unvollständige Anforderungen wurden identifiziert und an den Anforderungsersteller zur Nachbesserung weitergeleitet und ergänzt.

4.1.1.5 Phase 5 - Anforderungsspezifikation

In einer weiteren Arbeitsgruppe wurden schließlich alle Anforderungen auf Redundanz geprüft. Dies beinhaltete die Zusammenfassung inhaltstechnisch gleicher und überlappender Anforderungen, damit ein einheitliches Verständnis zu jeder Anforderung erzielt werden konnte. In diesem Zuge wurden zusätzliche globale Identifier integriert, auch um den Bezug zu den ursprünglichen Anforderungen zu erhalten. In dieser Phase konnten insgesamt 89 Anforderungen spezifiziert werden für die weitere Nutzung und Bearbeitung im Projekt.

4.1.2 Ergebnisse

Im Rahmen des AP 3.1 wurde eine Vorlage für die Anforderungserhebung entwickelt, TP übergreifend von allen Konsortialpartnern angewendet, partnerübergreifend konsolidiert und bereinigt. Hieraus ist eine Liste mit konkreten Anforderungen an den digitalen Zwilling eines Leitungssatzes und dessen Umsetzung durch Nutzung von Verwaltungsschalen entstanden.

Die Anforderungsliste wurde durch TP1 ergänzt, um die Anforderungen gegen projektrelevante Themen wie OPC UA, KBL, VEC und MIKO abzugleichen. Das „[Manufacturing Interface Komax“ \(MIKO\)](#)¹⁶ ist eine Schnittstelle für den Datenaustausch an kabelverarbeitende Automaten der Fa. Komax, basierend auf den Protokollen [HTTP](#) und [MQTT](#).

Die gesammelten Ergebnisse finden sich in der Anforderungstabelle „[VWS4LS_AP3.1_Requirementssammlung_2022_05_06.xlsx](#)¹⁷.

¹⁶<https://www.komaxgroup.com/de-de/products/software-and-networking/miko>

¹⁷https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP03/Beispieldaten/VWS4LS_AP3.1_Requirementssammlung_2022_05_06.xlsx

| Global-ID | Personen-ID | Titel | Title | Beschreibung | Description |
|-----------|--------------------|---------------------------------------|---|--|---|
| GL_1 | ALKI-05 PANE-02 | Grenz- und Soll-werte | Limits and nominal values | <p>Für die Auslegung des Prüfequipments und Bewertung von Messwerten müssen die Grenz- bzw. Sollwerte bekannt sein.</p> <p>Diese Anforderung richtet sich speziell an Werte die in der Produktion überprüft und bewertet werden müssen und somit ausschlaggebend für die Bewertung eines OK-Teils sind.</p> <p>Die Grenzwerte entsprechen hierbei den Bewertungskriterien für einen Messwert. Sie sind zudem ausschlaggebend für die Wahl eines geeigneten Messsystems.</p> <p>Es ist nicht notwendig für jeden Messwert eine absolute Ober- und Untergrenze oder eine unabhängige Toleranz zu definieren, so lange sich die Grenzwerte aus den vorhandenen Angaben ableiten lassen.</p> <p>Beispiele:</p> <ul style="list-style-type: none"> • Widerstand 1-0,05 Ohm • Drehmoment 15 Nm ±10% • Länge 100 mm (Allgemeintoleranz ISO 2768-mK) • Isolationswiderstand >4.5 MΩ <p>Dabei ist zu beachten, dass die Prüftoleranz von den Zeichnungsaufgaben abweichen können. Beispielsweise kann eine Verschraubung mit 15 Nm ±10% in der Zeichnung definiert sein und tatsächlich mit 15 Nm ±5% bewertet werden.</p> | The limits and nominal values must be defined, so that the test equipment can be designed and measured value can be evaluated. |
| GL_2 | EmDa_21 | | Data access for machine manufacturers - Monitoring (rw) | Als Maschinenhersteller will ich in Abhängigkeit der aktivierten Überwachungssysteme (bei der aktuellen Produktspezifikation) die zugehörigen Überwachungstools ermitteln können, um damit die Messungen durchführen zu können. | As a machine manufacturer I want to read the state of all enabled monitoring processes so that I can choose the appropriate measurement equipment/resource to perform the task depending on the product specification |
| GL_3 | ROMI-01 HORÖ-03 | Istanzdatenversion | Crimp/Seal data version | Als Manager möchte ich in der Lage sein, eine Version der Instanzdaten zu erstellen, damit ich ihre Parameter ändern kann. Historische Instanzdaten müssen zugänglich sein, um z.B. Fehlerursachen ermitteln zu können. | As a manager, I want to be able to create a version of the instance data so I can change its parameters. |
| GL_4 | ROMI-02 | Versionierung von Instanzdaten | Versioning of Crimp/Seal data versions | Als Manager möchte ich die verschiedenen Versionen von Instanzdaten visualisieren, damit ich nachvollziehen kann, wer Änderungen vorgenommen hat und warum | As a manager, I want to visualize the different versions of instance data so I can understand who made changes and why |
| GL_5 | ROMI-03 | Zulassung Prozess-daten (Crimpda-ten) | Approval crimp data | Als Manager möchte ich „freigegebene“ neue Sätze von Prozessdaten (bspw. Crimp- und Werkzeugdaten) aus meiner Datenbank abrufen, damit sie in den Produktionsmaschinen verwendet werden können. | As a Manager I want to fetch "Released" new sets of crimp and tool data from my database so that it can be used in production machines. |

| | | | | | |
|------|---|---|--|--|--|
| GL_6 | TOKR05 ROMI-07 | Möglichkeiten der Parameter-Anpassung der Komponenten Bestehende Produktionsparameter ändern | Possibilities of parameter adjustment of the components Modify existing production parameters | Möglichkeiten der Parameter-Anpassung in der laufenden Maschine. D.h. bei Produktionsänderungen soll es möglich sein, prozessrelevante Parameter einer Komponente anzupassen. Als Kunde möchte ich die neuesten Versionen von Prozessdaten (Crimp- und Werkzeugdaten) aus einer Datenbank abrufen, damit ich neue Werkzeuge verwenden oder die Parameter der bereits Verwendeten ändern kann. | Possibilities of parameter adjustment in the running machine. This means that it should be possible to adjust process-relevant parameters of a component in case of production changes. As a Customer I want to fetch the latest versions of crimp and tool data from a database so that I can use new tools or modify the parameters of the ones I am already using. |
| GL_7 | HORÖ-02 | Ausstattung und Konfiguration der Maschine | Equipment and configuration of the machine | Um die Fehlerursache zu ermitteln, müssen die Ausstattungs- und Konfigurationdaten der Maschine zum Zeitpunkt der Prüfung bekannt sein. | In order to determine the cause of a defect, the equipment and configuration data of the machine at the time of the test must be known. |
| GL_8 | FRST-01 ALKI-01 RoMi_04 RoMi_05 JOZI-07 JOZI-01 MIBR-08 RoMi_07 BeJo_03 | Eindeutige Beschreibung und Identifikation des Leitungssatzes und aller Einzelkomponenten im Leitungssatz | | <p>Für alle im Leitungssatz vorhandenen Teile (auch Zusammenbauteile, Kaufteile o.ä.) gibt es eine Beschreibung/Auflösung aller Einzelkomponenten (in der KBL/VEC/...). Diese können durch eine Art "hierarchische Klammerung" als Zusammenbauteil gekennzeichnet sein</p> <p>Für die eindeutige Zuweisung von Steckern und Leitungssatzkomponenten muss jede Komponente eindeutig identifiziert sein (Steckerbezeichnung, Steckernummer, XCode, etc.).</p> <p>Für die automatische Zuordnung von Prüfaufnahmen und Leitungssatzkomponenten muss die eindeutige (global) Typidentifikation aller Komponenten bekannt sein.</p> <p>Für die Auslegung des Prüfequipments müssen die Merkmale für alle Komponenten des Leitungssatzes bekannt sein (Steckerbezeichnung, Ausstattung, Kontaktbezeichnungen, etc.).</p> <p>Als API-Benutzer möchte ich auf die Daten jedes Kabels zugreifen, damit ich dessen Länge, Crimp, Dichtung, Aufdruck und andere Informationen lesen kann</p> <p>Jedes Produkt (ein Kabel oder ein Bündel) erhält eine eigene ID zur Identifikation. Weiter müssen alle Teilprodukte (entnehme zwei Leitungen aus dem Bündel und werden verdrillt) müssen ebenfalls eine eigene ID erhalten.</p> | |
| GL_9 | JuNe-02 FRST-07 | OEM-Input - Last- und Temperaturprofile von Verbrauchern und Bauraum zur Leitungsauslegung | | Zur Ermittlung des zu verwendenden Leitungsquerschnitt und Isolationsmaterial sind Umfeldparameter wie die Bauraumtemperatur und die Lastprofile (Dauerstrom, dynamische Last usw.) der zu versorgenden Verbraucher notwendig. Wenn diese Parameter bekannt sind, lässt sich eine weitestgehend automatisch Leitungsdimensionierung und Sicherungsauslegung erreichen. Sowie eine robust Auslegung hinsichtlich Spannungsfälle ohne der Gefahr einer Über- oder Unterdimensionierung. | |

| | | | | | |
|-------|---|---|---|---|--|
| GL_10 | DIN-N-05 PEBR-04 | Geometrische In-formationen von Tüllen und Schächten, sowie des Verlegeweges innerhalb | Geometrical representa-tion of grommets and cable ducts and routing inside them | Um eine eindeutige digitale Beschreibung des Leitungssatzes zu bekommen ist die geometrische Ausgestaltung innerhalb von Tüllen und Schächten erforderlich, sowie die geometrische Ausgestaltung der Tüllen und Kabelschäfte selbst. Die geomtrische Ausgestaltung wird verwendet für die Konstruktion von Werkzeugen, Kabelformbrettern und Kabelformbrettaufnahmen. | |
| GL_11 | DIZA-01 | Maschinenidentifi-kation | Machine identification | Um die Fehlerursache an der richtigen Maschine zu ermitteln, muss die Maschinenidentifikation bekannt sein. | In order to determine the cause of a fault on the correct machine, the machine identification must be known. |
| GL_12 | BAMA-04 BeJo-01 EmDa_11 GENE-02 GENE-04 CaWe_009 HORÖ-01 ArBe-02 GENE-09 MaFe-08 CaWe_012 CaWe_013 EmDa_20 BeJo_06 | Horizontale Rück-verfolgbarkeit | Horizontal Traceability | Für die Herstellung einer horizontalen Rückverfolgbarkeit müssen Rückmelddaten aus dem Prozess/ von der Maschine abgefragt und in der Verwaltungsschale des hergestellten Produktes gespei-chert werden. Dies gilt insbesondere für Rückmeldungen zu in dem Prozess integrierten Qualitätsüberwachungen. Bietet die Maschine eine aktive Verwaltungsschale ist es sinnvoll, wenn die Rückmelddaten darüber interaktiv abgefragt werden können, um diese Daten z.B. zur Entscheidungsfindung in Verhandlungsprozessen zwischen Verwaltungsschalen zu nutzen. Auch das für einen Prozess verwendete Rezept soll nachvollziehbar sein. | |
| GL_13 | FRST-05 DIN-N-09 DIN-N-04 JOZI-06 PEBR-01 JOZI-02 JOZI-03 | Detailierte geom-e-trische Informatio-nen zu Kontaktträ-gern und weiteren Anbauteilen (Tüll-en, Clipse, Kabel-führungen, Kabel-binder usw.) | | Geometrische Informationen zu Außenkonturen (Länge, Höhe, Breite) und Innenkonturen von Anbauteilen und Kontaktträger, sowie zusätzliche konstruktive Informationen beispielsweise bei Kontaktträgern Kammerkoordinaten, Stecktiefen usw. Diese Informationen sind u.a. auch notwendig für die Konstruktion von Prüfaufnahmen Für eine (teil-)automatisierte Ableitung des Tischlayouts muss die genaue Lage aller Stecker inklusive Einlagerichtung bekannt sein. Für die Bereitstellung von Hilfswerkzeugen und Produktionshilfs-mitteln müssen die Anbauteile (z.B. Kabelbinder) bekannt sein. | |
| GL_14 | SEAI-01 SEAI-02 DIN-N-01 DIN-N-02 DIN-N-07 FRST-04 | Verarbeitungs- und Verfahrensinfor-mation für diverse Produkttypen (Splice, UTP-ver-drillte Leitung, Wicklung usw.) | | Alle notwendigen Verarbeitungspezifikationen und dazugehörige Parameter müssen den entsprechenden Produkttypen datentechnisch zugewiesen und/oder eindeutig identifizierbar sein. Beispiele: <i>UTP:</i> Schlaglänge, Entdrilllänge, mit Wicklung gesicherte Enden usw. <i>Splice:</i> Verbinderart (End- oder Durchgangsverbinder), Verbindungs-technik (U-Schall, Crimp, ...), Gedichtet/Ungedichtet usw. <i>Wicklung:</i> Überlappungsgrad und -winkel von Wickelstrecken muss | |

| | | | | | |
|-------|--|---|--|--|--|
| | | | | im Datenmodell hinterlegt werden. Zur (automatisierten) Verarbeitung wird sowohl der Überlappungsgrad als auch der Überlappungswinkel benötigt. | |
| GL_15 | SEAI-03 FRST-03 FRST-06 WOAI-03 | Vererbung von Klassifikation, Sicherheitsrelevanz, Verbauinformation, Anforderungen usw. von Brodnetzkomponenten auf den Leitungsstrang bzw. dessen Komponenten | | Sicherheitsanforderungen die beispielsweise aus FuSi (ISO 26262), ASPICE, Bauraum o.ä. kommen muss diese Information an die angeschlossenen Komponenten übergeben/vererbt werden, um die Komponenten richtig auszuwählen (Festlegung und Verwendung von Standard-Attributen für functional safety (FuSi) und ISO26262 relevante Umfänge). Diese Information muss auch systemübergreifend transferiert werden können. Ebenso brauchen Komponenten entsprechende Properties ob diese Anforderungen erfüllt werden. Beispiel: Steuergeräte mit entsprechenden Anforderungen müssen diese an den Leitungsstrang und dessen Komponenten vererben. FuSi-relevante Verbindungen, die über Trennstellen geführt werden, müssen diese Information an die Trennstelle übergeben. Für die Herstellung der Prüftechnik ist es wichtig die kritischen Abgriffe (z.B. Airbag) für den Test und das Layout zu haben | |
| GL_16 | FRST-05 WOAI-04 | Detaillierte Informationen zum Kontaktträger bezüglich der Kompatibilität zu Terminals, EADs | | Kompatibilitäten von Kontaktträgern zu Terminalsystemen inklusive der verwendbaren Einzeladerdichtungen, maximalen Leitungsdurchmesser usw. u.a. auch für die Prüftechnik relevant zur Konstruktion von Prüfadaptern | |
| GL_17 | FRST-05 | Detaillierte Informationen zum Kontaktträger bezüglich der Verarbeitungsspezifikation | | Für die automatisierte Fertigung werden die Ausrichtung des Terminals zur Kammer des Kontaktträgers benötigt sowie Parameter für Stecktiefe, Steckkräfte usw. | |
| GL_18 | MIBR-01 | Position der Kontakte im Steckergehäuse | | Die Position der Kontakte im Steckergehäuse muss bekannt sein, damit im Fehlerfall die genaue Fehlerposition visualisiert werden kann. | |
| GL_19 | FRST-02 DIN-N-08 | Einheitliche Deklarierung / Klassifizierung von Komponenten (Stammdaten) | | Damit die Kundendaten ohne weitere "Manipulation/Anpassung" bearbeitet und aufbereitet werden können, müssen die Komponenten/Stammdaten einer einheitlichen Klassifikation und Mindestanforderung an Attributierung unterliegen. Auf Seitens OEM sowie auch der Lieferanten z.B: Tüllen / Schrumpfschlauch / Glattschläuche / Wellschläuche / Geflechtsschläuche / Rollschläuche Kabelkanäle / nichtelektrische Leitungen / usw. | |
| GL_20 | DIN-N-03 | Verzinnen von Litzen nach verbinden Terminal mit Leitung | | Die Kenntlichmachung, dass Litzen nachträglich verzinnt werden müssen, ist notwendig; heute wird dies teilweise über Text based instructions gemacht, dies ist jedoch nicht einheitlich umgesetzt. Zum einen kann dies eine Anforderung vom OEM sein, so dass dies über KBL/VEC kenntlich gemacht sein muss. Andererseits kann dies | |

| | | | | | |
|-------|--|---|--|---|--|
| | | | | auch eine Verarbeitungsrichtliche für das Terminal sein, was dann über ein Attribut des Terminals kenntlich gemacht werden muss. | |
| GL_21 | DIN-N-10 | Detaillierte Informationen zu Kontaktteilen / Kabelschuhen bezüglich Verbindungsverfahren und Geometriedaten | | Für Kontaktteile muss eine vollständige geometrische Beschreibung vorliegen im unverarbeiteten und verarbeiteten Zustand (Daten wie z.B. Kontaktteilüberstand usw). Ebenso soll das Verbindungsverfahren des Kontaktierungsbereichs als Merkmal mit übermittelt werden. | |
| GL_22 | DIN-N-13 | Befestigungstechnik für Clipse und Kabelführungen | | Mit welcher Prozesstechnik Clipse und Kabelführungen am Leitungsstrang befestigt werden (z.B. Wickelband, Kabelbinder usw.) soll mit übertragen werden. Die Typanweisung zur Befestigung des Clips / Kabelführung muss definiert sein und die Information des einzelnen Clips / Kabelführung (je Position) muss ebenso definiert und übermittelt werden. | |
| GL_23 | BAMA-02 BAMA-03 EmDa_13 | Überwachung des Materialverbrauchs | | Um rechtzeitig Material nachbestellen zu können, und inkorrekt eingestellte Prozessparameter zu erkennen, muss der Materialverbrauch und Ausschuss aller Teile (Kabel, Kontakte, etc.) überwacht werden können. | |
| GL_24 | EmDa_16 EmDa_17 EmDa_18 EmDa_19 | Produktbeschreibung inklusive Verarbeitungsspezifikationen der Komponenten für die Durchführung der Produktionsprozesse | | Als Produktionsressource möchte ich alle Spezifikationsdaten (Kabel, Terminals, Seal, Sleeve, ...) auslesen, damit ich Einrichts- und Prozessschritte planen und durchführen kann. | |
| GL_25 | WOAI-01 MAZI-02 MIBR-05 | Eigenschaften von Konnektoren / Kontaktträgern | | Für die Herstellung und richtige Auswahl der Prüftechnik ist es wichtig die Eigenschaften von Konnektoren und die Anforderungen des Leitungssatzes zu haben, um Prüfadapter zu konstruieren/auszuwählen (!!evtl. ist hier GL_8 mit zu berücksichtigen!!!) | |
| GL_26 | MAZI-01 PEBR-02 | Leitungssatzinformationen für die Angebotserstellung | | Für die Angebotserstellung müssen die Leitungssatzinformationen (Zeichnungen, Verbindungspläne, Dimensionen, etc.) bekannt sein. Mit den Informationen aus der Verwaltungsschale Leitungssatz soll eine Auswahl des geeigneten Produktionsequipments durch Abgleich mit den in den Verwaltungsschalen der Ressourcen beschriebenen Fähigkeiten möglich sein. Die Informationen zu den Ressourcen soll eine Mengen/Kapazitätsplanung unterstützen. | |
| GL_27 | PEBR-03 PEBR-05 JOSC-04 | Steuerung des Materialflusses durch die Fabrik | | Mit den Informationen aus der Verwaltungsschale der Leitungssatz-Instanz soll der Weg des Produktes durch die Fertigung geplant und gesteuert werden. Die einzelnen Anlagen sollen mit Produktdaten und Prozess Vorgabeparametern versorgt werden. | |
| GL_28 | Anforderung wurde gelöscht und muss nicht weiter berücksichtigt werden. Um den Mehraufwand der Zeileanpassung nicht vornehmen zu müssen bleibt die Zeile leer. | | | | |

| | | | | | |
|-------|--|--|--------------------------|---|---|
| GL_29 | TOKR03 GENE-10 | Detektierbare Komponente/Resource im Maschinennetzwerk | | Auffinden der Komponente in der Maschine zum Zwecke der autom. Verhandlungen (Discovery). Dabei wird die VWS der Komponente im Maschinennetzwerk instanziert. Analoges gilt für das Auffinden von Produktionsressourcen: Als Produktionsresource möchte ich über einen automatisierten Verhandlungsprozess durch Abgleich meiner Capabilities mit Anforderungen entscheiden, welchem Dienst/Steuerungssystem (MES) ich mich anschließen kann, um so ein Plug&Produce Szenario zu realisieren. | |
| GL_30 | GENE-11 | Heartbeat & Wiederverbinden | | Als Produktionsresource möchte ich den Status der Verbindung zum Steuerungssystem abfragen, so dass ich das Wiederverbinden oder den Wechsel auf einen Fallback-/Offlinemodus einleiten kann. | |
| GL_31 | CaWe_010 GENE-07 MiRo-03 CaWe_001 | Übertragen von Rezepten | | Als Steuerungssystem möchte ich ein Rezept an meine Produktionsressource senden, so dass ich die Durchführung meines Produktionsplanes vorbereiten kann oder Rezepte oder Einstellungen von einer Maschine auf eine andere übertragen kann. | |
| GL_32 | CaWe_014 GENE-13 GENE-14 CaWe_013 | Abfragen von Werkzeug-Setup | Read die-set and tool ID | Als Steuerungssystem (MES) oder Produktionsresource möchte ich die eingerichteten Werkzeuge (Artikelnummer, Seriennummer, Position) abfragen, so dass ich über die Produktionsfreigabe entscheiden kann und Daten für die Rückverfolgbarkeit sammeln kann. | |
| GL_33 | BAMA-01 MiRo-01 MiRo-03 | Überwachung des Produktionsfortschritts und Status der Produktionsressourcen | | Für eine optimale Planung der Maschinennutzung muss der aktuelle Produktionsfortschritt überwacht werden können. Der Status, Werte der Prozessüberwachung und ggf. Fehlerzustände sollen von den Produktionsressourcen abgefragt werden können. | |
| GL_34 | MaBa-01 TOKR04 MaFe-10 | Bereitstellung von Fähigkeiten der Maschinen und Maschinenkomponenten | | Zur Prozessbeschreibung werden Fähigkeiten der Komponenten (Maschine/Anlage) / Ressourcen benötigt. Mit diesen Fähigkeiten können (z.B. KI gestützte) Automatismen und automatisierte Verhandlungsprozesse durchgeführt werden. | |
| GL_35 | MiRo-04 | Ausführen von Aktionen auf Produktionsressource | | Als Kontrollsysteem will ich Aktionen/Befehler auf der Produktionsressource ausführen. | |
| GL_36 | MIBR-07 | Leitungs-Routen | Wire routing information | Für die Bewertung von kritischen Signalpfaden, müssen die Leitungs-Routen bekannt sein. | For the evaluation of critical signal paths, the wire routes must be known. |
| GL_37 | JuNe-01 TOKR01 | Digitaler Materialkatalog | | Auf Basis von Designparametern können Materialvorschläge (z.B. passende Stecker) aus den verfügbaren Verwaltungsschalen der Komponenten ausgegeben werden, durch Abgleich der Anforderungen mit den in der VWS hinterlegten Merkmalen der Komponenten. Die Daten je Kategorie der Leitungssatzkomponente sind noch zu beschreiben. Die Attribute werden neben dem Entwicklungsprozess auch zur Steuerung bzw. Validierung der Produktionsprozesse benötigt. In der VWS soll sowohl die Herstellteilenummer als auch die OEM-Teilenummer zu finden sein. Nicht Funktionale Anforderung an OPCUA Companion spec | |

| | | | | | |
|-------|--------------------|--|--|---|---|
| GL_38 | WOAI-02 MaFe-03 | 3D-Daten (CAD-Daten) vom Leitungssatz | | <p>Es werden die CAD-Daten des Leitungssatzes benötigt um darauf aufbauend weiterführende Themen zu prüfen, designen, simulieren usw.</p> <p>Beispiele:</p> <ul style="list-style-type: none"> Prüftisch Gestaltung Leitungsstrang autom. in Karosserie zu verlegen usw. | |
| GL_39 | MaFe-03 | 3D-Daten (CAD-Daten) von Ressourcen | | <p>Es werden die CAD-Daten der Ressourcen z.B. Greifer, Schrauber, Roboter etc.</p> <p>Als Datenformat wird ein Open Standard Format empfohlen, z.B. glTF, Collada etc, jedoch nicht proprietäre Formate wie JT oder 3DXML.</p> | |
| GL_40 | ArBe-01 | Standardisierte Kennzeichnung der physikalischen Assets um den Bezug zur Verwaltungsschale herstellen zu können. | | <p>Einheitliche Beschreibung und Spezifikation von Anforderungen für die Codierung(=Kennzeichnung) von Komponenten und Systemen zur Traceability zwischen den Unternehmen und auch innerhalb des Unternehmens. Aufgrund der sehr unterschiedlichen Funktionen und Geometrien der Leitungssatzkomponenten wird es notwendig sein, verschiedene Arten der physischen Kennzeichnung am Artikel zu nutzen. Die daraus resultierenden Anforderungen an die Hardware (z.B. Barcodescanner) sind zu berücksichtigen, damit erforderliche Hardware nicht dem durchgängigen Einsatz der VWS im Wege steht.</p> | |
| GL_41 | ArBe-04 | Validierungsdaten für Kontaktteile für Kontakteilauswahl und Validierung der Leitungssatzentwicklung. | | <p>Um zu überprüfen, ob die Verwendung eines Steckersystems mit dem ausgewählten Leitungsmaterial möglich ist, soll die VWS des Kontaktteils vereinheitlichte Informationen zu den freigegebenen/Validierten Kombinationen enthalten (z.B. Leitermaterial, Leiteraufbau, Leitungshersteller, Querschnittsbereich).</p> | Freigabeinformationen des Kontakteilhersteller Freigabeinformationen des OEM Erlaubter Querschnittsbereich Erlaubte Leitungsart (Leitermaterial+Litzenaufbau) Erlaubte Leitungshersteller |
| GL_42 | SeOl-01 | Variantenabbildung | | <p>Das Datenmodell muss in der Lage sein, die Variantenvielfalt eines Leitungssatzes abzubilden.</p> <p>D.h. eine KBL / VEC muss maximal ausgeprägt sein, jedoch ein Kennzeichen für Varianten für Komponenten mitgeben können.</p> | |
| GL_43 | HeWi-01 | Projekt IILS: TP5 - Rationalisierung EoL-Prüfung | | <p>Erweiterte und eindeutige Produktionsdaten müssen einem Leitungssatz zugeordnet werden können, um mithilfe dieser Daten eine EoL-Prüfung zu rationalisieren</p> | |
| GL_44 | FRST-08 | Konfigurationsinformationen an Komponenten und Leitungen | | <p>Wenn alle Komponenten und Leitungen eine entsprechend auswertbare Ausstattungs-/Konfigurationslogik erhalten, können Bestellungen auf Konsistenz geprüft werden und Prüfprogramme gezielter auf den spezifischen KSK erstellt werden.</p> | |
| GL_45 | DIN-N-06 | Detaillierte Beschreibung der Außenkonturen vor allem bei Kreuzungen und Ausbindungen | | <p>Bei den Ausbindungen und Kreuzungen sollen die Bündeldurchmesser berücksichtigt werden und datentechnisch auswertbar sein</p> | |

| | | | | | |
|-------|----------|--|---|--|---|
| GL_46 | DIN-N-11 | Verarbeitungs- und Typinformation für Tüllen | | Informationen zum Tüllentyp sollen als Merkmal mit übergeben werden, handelt es sich um eine einteilige Tülle mit Fädelprozess oder um eine mehrteilige Tülle die nachträglich angebracht werden kann. Ebenso soll es eine Definition dafür geben, ob es sich um eine Schaumtülle handelt. | |
| GL_47 | DIN-N-12 | Information zur Ausführungsart von Schläuchen | | Mit den Schläuchen soll ein Merkmal übermittelt werden, ob diese in geschlitzter oder geschlossener Ausführung verwendet werden sollen. | |
| GL_48 | TOKR02 | Teilmodell Simulation | Simulation submodel | Für die Simulation der Prozesse werden Simulationsmodelle der beteiligten Komponenten benötigt. | For the simulation of the processes, simulation models of the involved components are required. |
| GL_49 | MIBR-02 | Verbindungsinformationen | Connection information | Für die automatische Erstellung von elektrischen Prüfprogrammen müssen die Verbindungsinformationen vorhanden sein. | For the automatic creation of electrical test programs, connection information must be available. |
| GL_50 | MIBR-03 | Bilder von der Kontaktseite der Stecker | Picture of the contact side of the connectors | Für die Fehlervisualisierung muss ein Bild von der Kontaktseite des Steckers vorhanden sein. | A picture of the contact side of a connector must exist, so that it can be displayed in case of a failure. |
| GL_51 | PANE-01 | Seriennummer des Leitungssatzes | Serial number of the harness | Für die lückenlose Rückverfolgbarkeit aller Prüfergebnisse, muss die Serialnummer des Leitungssatzes bekannt sein. | For the complete traceability of all test results, the serial number of the harness is required. |
| GL_52 | MIBR-04 | Artikelnummer | Part Number | Für die automatische Konfiguration des Prüfprogramms muss die Artikelnummer des Leitungssatzes bekannt sein. | For the automatic configuration of the test program, the article number of the harness must be known. |
| GL_53 | MIBR-06 | Eigenstabile Gehäuse | Intrinsically stable housings | Für die prozesssichere Aufnahme von großen Gehäusen (z.B. Sicherungsboxen) muss sichergestellt werden, dass die Gehäuse eigenstabil sind. | For the reliable intake of large housings (e.g. fuse boxes), it must be ensured that the housings are intrinsically stable. |
| GL_54 | MIBR-09 | Prüfaufgaben | Inspection tasks | Für die automatische Programmgenerierung müssen die Prüfaufgaben bekannt sein. | For automatic program generation, the inspection tasks must be known. |
| GL_55 | JOSC-01 | Prüfanforderungen und Vorschriften | Test requirements and regulations | Für die Auslegung des Prüfequipments müssen die Prüfanforderungen und Vorschriften (standortbezogen) bekannt sein. | For the design of the test equipment, the test requirements and regulations (plant based) must be known. |
| GL_56 | JOSC-02 | Hallenlayout | Production layout | Für die Auslegung der Prüfmaschine muss das Hallenlayout inklusive Anschlussmöglichkeiten und Arbeitsrichtung bekannt sein. | For the design of the testing machine, the production layout including supply sockets and working direction must be known. |
| GL_57 | JOZI-03 | Bereitstellungsart | Supply mode | Für die Bereitstellung geeigneter Schnittstellen und Peripheriegeräten muss die Bereitstellungsart (z.B. Trolley) bekannt sein. | For the provision of suitable interfaces and peripheral devices, the mode of supply (i.e. trolley) must be known. |
| GL_58 | JOZI-04 | Produktions-Stückzahlen | Production quantities | Für die Wahl eines geeigneten Werkstoffs für die Steckeraufnahmen müssen die Produktionsstückzahlen bekannt sein. | In order to choose a suitable material for the connector intakes, the production quantities must be known. |
| GL_59 | JOZI-05 | Kennzeichnungsart | Labeling type | Für die Kennzeichnung von Leitungssätzen nach bestandener Prüfung muss die erforderliche Kennzeichnungsart (Körperpunkt, Label, etc.) bekannt sein. | For the marking of harnesses after the test has passed, the desired type of marking (dot marking, label, etc.) must be known. |
| GL_60 | JOSC-05 | Bewertungskriterien für Prüfergebnisse | Evaluation criteria for test results | Für die endgültige Bewertung eines Prüfergebnisses (reparabel, Ausschuss, etc.) müssen die Bewertungskriterien bekannt sein. | For the final evaluation of a test result (repairable, scrap, etc.), the evaluation criteria must be known. |
| GL_61 | ROMI-04 | Prozessdaten (Crimpdaten) zur Validierung einreichen | Submit crimp data to validation | Als Manager möchte ich überprüfte Änderungen an Prozessdaten (Crimp- und Werkzeugdaten) aus meiner Validierungsressource an meine Datenbank „übermitteln“, damit sie in der gesamten Organisation verwendet werden können | As a Manager I want to "Submit" reviewed changes of crimp and tool data from my validation resource to my database so that it can be used across the whole organization |

| | | | | | |
|-------|---------|--|---|--|--|
| GL_62 | ROMI-09 | Synchronisierung Prozessparameter über Organisationen hinweg | Synchronization process parameters across organizations | Als Kunde möchte ich Prozessparameter über mehrere Datenbanken hinweg synchronisieren, damit ich die neuesten genehmigten Parameter verwenden kann | As a Customer I want to synchronize process parameters across multiple databases so that I can use the latest parameters approved |
| GL_63 | HORÖ-06 | Wartungs- und Kalibrierdaten | Maintenance and calibration data | Um eine regelmäßige Wartung und Kalibrierung der Maschine durchzuführen, müssen die Wartungs- und Kalibrierdaten bekannt sein. | In order to carry out regular maintenance and calibration of the machine, the maintenance and calibration data must be known. |
| GL_64 | HORÖ-07 | Wartungsprotokolle | Maintenance logs | Um die Fehlerursache zu ermitteln, müssen die Wartungsprotokolle (Reinigung, Wartung, etc.) bekannt sein. | In order to determine the cause of the error, the maintenance protocols (cleaning, maintenance, etc.) must be known. |
| GL_65 | ALKI-03 | Maximalausstattung | Maximal configuration | Um die Prüfmaschine für die Prüfung aller möglichen Leitungssatzvarianten auszustatten, muss die Maximalausstattung des Leitungssatzes bekannt sein. | In order to equip the testing machine for testing all possible harness variants, the maximum configuration of the harness must be known. |
| GL_66 | ALKI-04 | Bestückungsvarianten | Assembly variants | Für die Auslegung des Prüfequipments müssen alle Bestückungsvarianten bekannt sein. | For the design of the test equipment, all assembly variants must be known. |
| GL_67 | ALKI-06 | Projektbezeichnung bzw. Fahrzeugidentifikation | Project name or vehicle identification | Für eine eindeutige Zuordnung des Projektes muss die Projektbezeichnung bzw. Fahzeugidentifikation bekannt sein. | For a clear assignment of the project, the project name or vehicle identification must be known. |
| GL_68 | ArBe-03 | Traceability For- and Backwards | | Stand aktuell - Rheels (Spulen) der Kontakte sind mit Traceabilitydaten codiert - wie kann diese Information nach der Vereinzelung abgesichert werden | |
| GL_69 | ArBe-05 | Spezifikationsanforderungen Produkt | | Anforderungen und Spezifikationen des 1st Tier und OEM durchgängig auf Produktebene in digitaler Form | |
| GL_70 | GENE-01 | Auftragsfreigabe | | Als Produktionsressource möchte ich eine Freigabe für den (Wieder-)Start der Produktion eines Auftrags vom Steuerungssystem erhalten. | |
| GL_71 | GENE-03 | Qualitätsprüfungen anstoßen | | Als Steuerungssystem (MES) möchte ich Qualitätsprüfungen anstoßen, die implementiert durch meine Ressource sind, so dass ich den Produktionsstart freigeben oder verweigern kann und Rückmelddaten sammeln kann. | |
| GL_72 | GENE-05 | Schicken des Auftrages an die Maschine | | Als Steuerungssystem möchte ich einen Auftrag an meine Produktionsressource mit Produktspezifikation und Menge senden, so dass ich meinen Produktionsplan durchführen kann. | |
| GL_73 | GENE-06 | Schicken von Konfigurationsänderungen an die Maschine | | Als Steuerungssystem möchte ich meine Produktionsressource konfigurieren (Grundeinstellungen, nicht Rezept, z.B. automatischer Werkzeugwechsel), so dass ich die Durchführung meines Produktionsplanes vorbereiten kann. | |
| GL_74 | GENE-08 | Abrufen von Rückmelddaten für die Rückverfolgbarkeit | | Als Steuerungssystem möchte ich die gemessenen Prozessparameter (Rückmelddaten) anfragen, so dass ich Daten für die Rückverfolgbarkeit sammeln kann. | |
| GL_75 | GENE-12 | Abfragen von Material-Setup | | Als Steuerungssystem (MES) möchte ich die eingerichteten Materialien (Artikelnummer, Chargennummer) abfragen, so dass ich über die Produktionsfreigabe entscheiden und Daten für die Rückverfolgbarkeit sammeln kann. | |

| | | | | | |
|-------|----------|---|---|--|--|
| GL_76 | GENE-15 | Abfragen von Material-Setup | | Als Produktionsressource möchte ich Informationen über das ein-gerichtete Material (Artikelnummer, Seriennummer, Verarbei-tungsspezifikation) sammeln, so dass ich Daten für die Rückverfolg-barkeit sammeln kann und diese an andere VWS weiterleiten kann. | |
| GL_77 | MaFe-04 | Zugriff auf Verwaltungsschalen | | Für die Produktion bzw. Montage des Kabelbaums wird ein stan-dardisierter Zugriff auf die Verwaltungsschale benötigt, um Daten zu lesen und zu schreiben. Hierfür muss eine Kommunikationstech-nologie (z.B. REST über HTTP) zur Verfügung gestellt werden | |
| GL_78 | MaFe-05 | Repository für Ver-waltungsschalen | | Die in der Anlage zur Verfügung gestellten verwaltungsschalen müssen über eine Repository standardisiert zur Verfügung gestellt werden | |
| GL_79 | MaFe-06 | Physikalische Da-ten von Produkten und Ressourcen | | Es werden physikalische Daten von Produkten benötigt, insb. Masse und Trägheit | |
| GL_80 | MaFe-07 | Kinematikbeschrei-bung von Produk-teu und Ressour-cen | | Es werden die Kinematik der Ressourcen benötigt, insb. des Greif-ers | |
| GL_81 | CaWe_006 | | Hardwired safe modes | | As a crimp machine manufacturer, I want to have the op-eration modes "safe shutdown" and "local operation" to be implemented by an electrical signal connection (e.g.: "Emergency shutdown loop") so that I can implement the safety functions by hardware elements. |
| GL_82 | CaWe_007 | | Locking and unlocking machine for manual opera-tion | | As an OEM, I want to have means to lock and unlock opera-tion of a crimp machine, which is part of a partially auto-mated system, so that I can make sure that it will only be operated in relation to a specific manufacturing job. |
| GL_83 | CaWe_015 | | Verify die-set and tool | | As a crimp machine manufacturer, I want to have the ma-chine verifying that the currently loaded program can be safely executed with the installed die-set and tool so that I can prevent damage of the machine. |
| GL_84 | EmDa_11 | | Data access for OEMs - gather quality infor-mation | | As an OEM I want to be able to identify each part of a pro-duced harness instance (according to BOM) so that I can visual-ize the product |
| GL_85 | RoMi_01 | | Hierarchical product data model | | As an API user I want to query my product data model so that I can group the details in a hierarchical structure |
| GL_86 | RoMi_02 | | Harness as root element | | As an API user I want to access the Product data model over a Harness root element so that I have always the same entry point |
| GL_87 | RoMi_03 | | Harness metadata | | As an API user I want to access metadata for a Harness so that I can read the model identifier, Unique harness ID, and other data |
| GL_88 | MiRo-02 | | Receive events from re-source | | As a Control System I want to receive events of my produc-tion resources so that I can react to changes in my produc-tion line |

| | | | | | |
|-------|---------|--|-----------------|--|--|
| GL_89 | MaFr-01 | | Lock a resource | | As a Control System I want to be able to reserve a resource for production exclusively |
|-------|---------|--|-----------------|--|--|

4.2 AP 3.2 - Erforschung eines Referenzmodells

Im Arbeitspaket 3.2 wurde auf Grundlage der Ergebnisse des AP3.1 ein Referenzmodell erforscht, mit dem es in Zukunft möglich ist, einen automatisierten Austausch eines Digitalen Zwillinges eines Leitungssatzes mittels der Verwaltungsschale durchzuführen.

Wertschöpfungskettenübergreifend gilt es die verschiedenen Systemlandschaften der einzelnen vor- und nachgelagerten Wertschöpfungsstufen (Bsp. OEM, Tier-n) miteinander zu verbinden, um eine Daten durchgängigkeit über die VWS zu erreichen. Dies betrifft Systeme wie bspw. PLM, MES sowie Leitungssatzspezifische Entwicklungs- und Produktionssysteme. Dies wird anhand der nachfolgend beschriebenen Use Cases veranschaulicht.

Die wesentlichen Elemente dieses Modells beziehen sich auf die VWS für Produktionsmittel, Produktionsprozesse und der Integration der Daten-/Informationsmodelle aus TP1. Für letzteres wurde in AP3.2 noch ein Konzept namens CAD2BOP definiert, welches CAD-Daten und Parameter erfasst, die einen starken Einfluss auf eine spätere Automatisierung der Produktions- und Montageprozesse in der Wertschöpfungskette für Leitungssätze haben.

4.2.1 Datenfluss-Architektur

Für einen unternehmensübergreifenden Datenaustausch über einen IDS wie bspw. [Catena-X](#) sind Konzepte für den unternehmensinternen Datenzugriff, -aggregation und -konvertierung notwendig, um alle relevanten Daten bereitzustellen zu können. Hierzu wurde eine initiale High-Level Architektur entworfen, um alle Elemente für die Datengenerierung und -haltung in einem Modell zusammenzufassen (siehe dazu die grün hinterlegten Bestandteile in Abbildung 4-2).

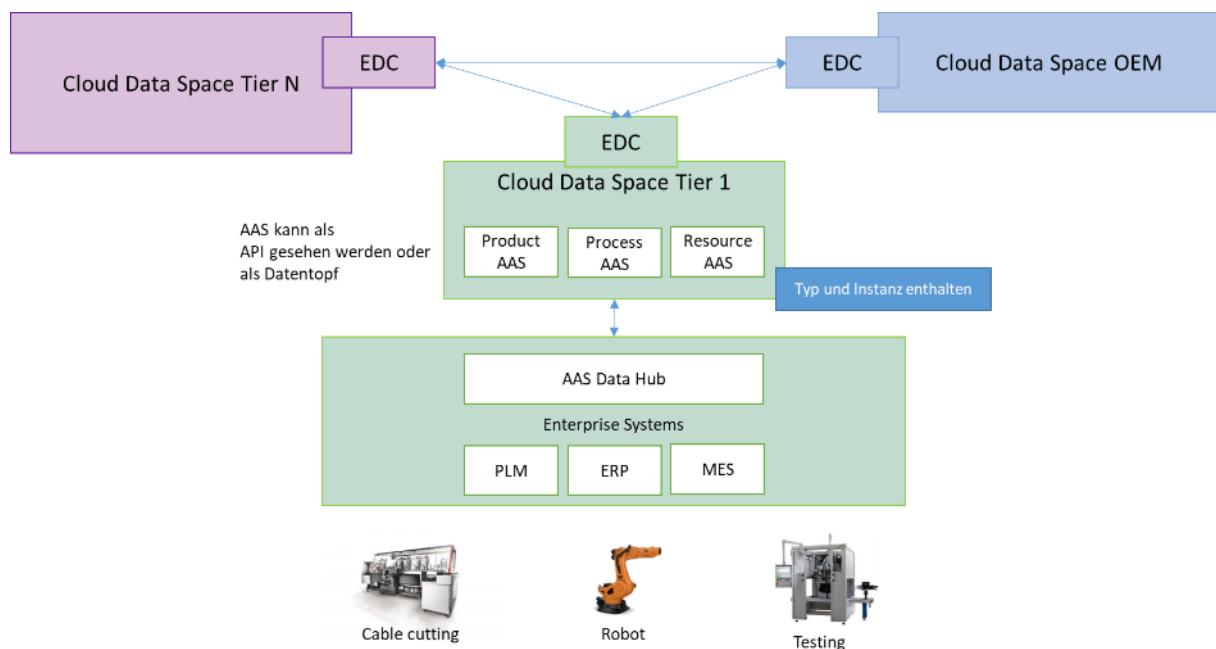


Abbildung 4-2: High-Level Architektur

Die unterste Ebene beschreibt den Shopfloor mit allen dazugehörigen Produktions- und Testanlagen. Diese Ebene kommuniziert mit Unternehmensanwendungen in einer darüber liegenden Schicht wie bspw. einem Manufacturing Execution System (MES) zur Produktionssteuerung oder einem Enterprise Resource Planning (ERP) System, um Produktionsressourcen sowie Materialverbräuche zu managen. Für eine Kommunikation mit einer Verwaltungsschale ist meist eine ETL-Schicht (ETL: Extract, Transform, Load) [31] notwendig, da die Daten in den Unternehmensanwendungen in den seltensten Fällen bereits passend für eine Integration in die Verwaltungsschale vorliegen. Diese Transformation erfolgt im sog. „AAS Data Hub“. In der vorgeschlagenen Architektur besteht die Annahme, dass die Verwal-

tungsschalen in einem sog. „Cloud Data Space“ gehalten werden, in dem für jedes Mitglied in der Wertschöpfungskette sowohl die Typ- als auch Instanz-VWS gespeichert sind und über APIs angesprochen werden können. Für den unternehmensübergreifenden Datenaustausch wird angenommen, dass dieser über den in Catena-X entwickelten [Eclipse Dataspace Connector](#)¹⁸ (EDC) erfolgen wird.

4.2.2 Use Cases

Aufbauend auf der High-Level Architektur wurden drei verschiedene Use Cases definiert, um die Datenflüsse im Kontext einer konkreten Anwendung darzustellen. Diese Use Cases werden in den folgenden Unterkapiteln vorgestellt.

4.2.2.1 UC1: Predictive Maintenance

In diesem Use Case werden Sensordaten von der Maschine dem Maschinenhersteller zur Verfügung gestellt für Predictive Maintenance. Die Sensordaten werden durch Produktinformationen angereichert.

Die Architektur für diesen Use Case ist in *Abbildung 4-3* dargestellt, wobei eine Produktionsanlage (hier der Maschinentyp „Crimpwerkzeug“) eine AAS kompatible Interfacetechnologie besitzt und direkt mit einer Verwaltungsschale kommunizieren kann. Der andere Maschinentyp (Prüfgerät) besitzt keine AAS kompatible Interfacetechnologie und kann somit nicht direkt mit einer Verwaltungsschale kommunizieren. Die für den Predictive Maintenance Service notwendigen Daten sind in der Verwaltungsschale „[EHC10-PLK14_5_50mm2_instance1.aasx](#)“ hinterlegt. Die einzelnen Datenpunkte sind auf der rechten Seite in der Abbildung gelistet.

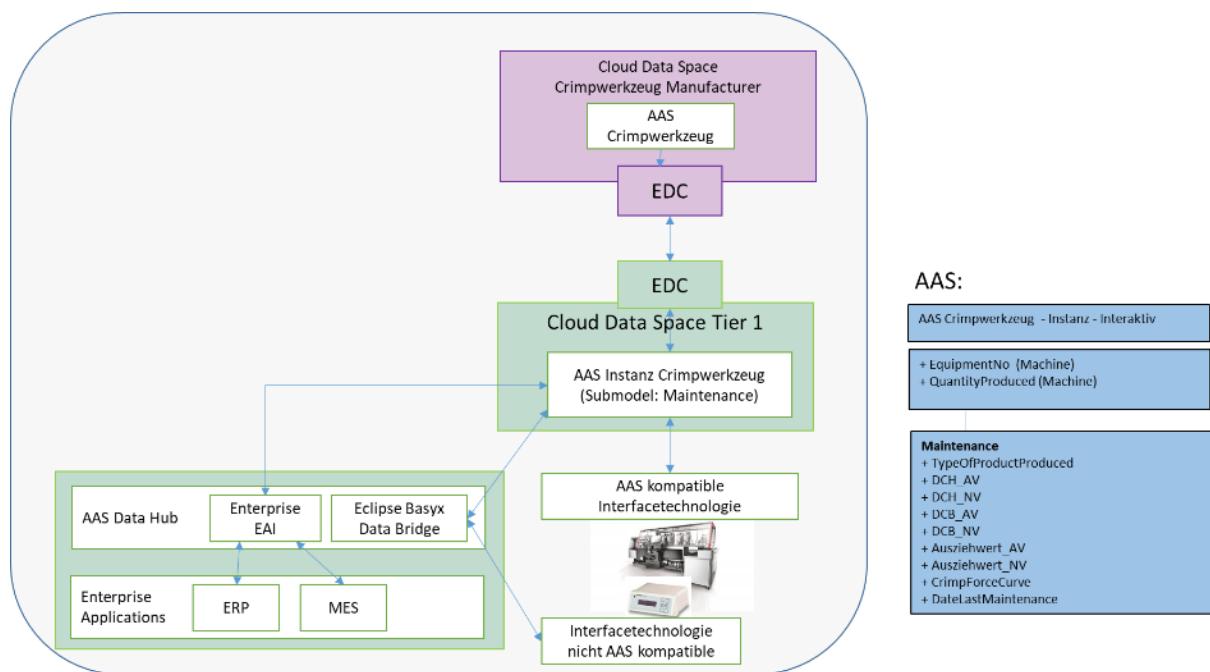


Abbildung 4-3: Use Case Predictive Maintenance

Für den zweiten Fall verläuft der Datenfluss über den zuvor erwähnten AAS Data Hub. Dieser beinhaltet in diesem Szenario ein EAI-System (Enterprise Application Integration) für eine prozessorientierte Integration von Anwendungssystemen in heterogenen IT-Anwendungsarchitekturen. Hierüber ist die Kommunikation zu einem ERP oder MES möglich. Daneben wurde die [Eclipse BaSyx DataBridge](#)¹⁹ (nachfolgend DataBridge), integriert für eine unkomplizierte Datenintegration. Sie unterstützt verschiedenste Protokolle und erlaubt eine einfache Transformation von Daten. D.h. das Prüfgerät in diesem Szenario kommuniziert über die DataBridge mit der Verwaltungsschale und integriert dort alle Produktionsdaten, die in einem weiteren Schritt als Rückmeldedaten über das EAI ans MES übergeben werden.

¹⁸ <https://github.com/eclipse-edc/>

¹⁹ <https://github.com/eclipse-basyx/basyx-databridge>

4.2.2.2 UC2: Pay-per-Use für Anlagenbetreiber

Eine Ressource soll mittels Pay-Per-Use Modell betrieben werden. Dabei ist es nötig, bestimmte Daten direkt in der Maschine zu halten, damit sie vor Manipulation geschützt sind. Dazu wird die Ressourcen-Verwaltungsschale direkt in die Maschine integriert, damit der Anlagenhersteller sichergehen kann, dass die Nutzdaten der Maschine unverfälscht sind.

In Abbildung 4 werden wie im Use Case zuvor 2 Szenarien beschrieben:

In Szenario 1 kommunizieren die Anlagen, beispielhaft als ‚Cable cutting‘ und ‚Testing‘ dargestellt, über Unternehmensanwendungen und dem AAS Data Hub mit den einzelnen Verwaltungsschalen. Szenario 2 wird anhand eines Industrieroboters dargestellt. Hier besteht eine direkte Kommunikation zwischen der Produktionsressource und ihrer zugehörigen Verwaltungsschale.

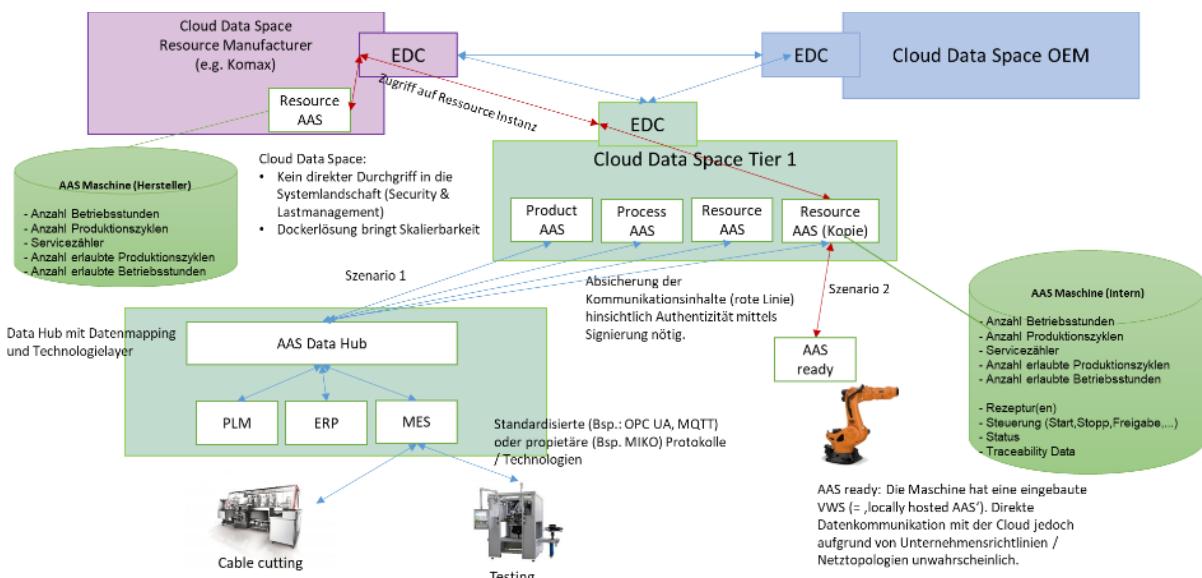


Abbildung 4-4: Pay-per-Use: Maschinenhersteller greift auf Ressource AAS bei Kunde zu

Die Produktion von Leitungssätzen heutzutage folgt Szenario 1, in dem die Produktionsressourcen über ein MES und dort verwalteten Produktionsaufträgen gesteuert werden. Die Kommunikation erfolgt über standardisierte Protokolle wie bspw. OPC UA oder Message Queuing Telemetry Transport (MQTT) oder proprietäre Protokolle wie bspw. MIKO. Die Rückmelddaten dieser Produktionsressourcen werden im MES hinterlegt und sind über den AAS Data Hub für die Verwaltungsschalen im Cloud Data Space Tier 1 zugänglich.

Für den Anwendungsfall „Pay-Per-Use“ erscheint eine solche Anbindung jedoch nicht praktikabel, ist es doch erforderlich, dass hierbei zwischen der Ressource und ihrem Hersteller (im Bild ‚Ressource Manufacturer‘) manipulationssicher kommuniziert werden kann. Entsprechende Möglichkeiten zur Durchleitung signierter Daten durch die Unternehmensanwendungen müssten erst noch geschaffen werden.

Für die Pay-Per-Use Ressource wird daher Szenario 2 angewendet (siehe Abbildung 4-4): Sie kommuniziert mittels ihrer eingebauten AAS direkt mit dem unternehmensinternen Cloud Data Space. Dort liegt aus Datenhaltungs- und Performancegründen gegebenenfalls noch eine Kopie der Ressourcen-Verwaltungsschale „Ressource AAS (Kopie)“. Sowohl die ressourceninterne Verwaltungsschale „AAS Maschine (intern)“ als auch ihre Kopie umfassen sowohl Daten, die der Hersteller für das Abrechnungsmodell Pay-per-Use benötigt, als auch weitere Daten, die nicht mit dem Hersteller geteilt werden sollen, zum Beispiel Rezepturen und Traceability-Daten.

Die Kommunikation mit dem Hersteller erfolgt von EDC (Tier 1) zu EDC (Resource Manufacturer), woüber die relevanten Daten für das Pay-per-Use vom Tier 1 an diesen weitergegeben werden. Die interne Kommunikation mit den Unternehmensanwendungen erfolgt über den AAS Data Hub direkt mit „Ressource AAS (Kopie)“.

4.2.2.3 UC3: OEE-Berechnung

Die OEE-Berechnung ist von zentralem Interesse, wenn ein Betreiber von Anlagen und Maschinen eine Erhöhung der Effizienz anstrebt. Auch für Anlagen- und Maschinenhersteller sind diese Daten von hoher Wert. Dieser KPI findet oft Verwendung bei FAT's (Factory Acceptance Test) und kann auch im späteren Verlauf dem Hersteller wertvolle Informationen zur Weiterentwicklung seiner Maschinen liefern.

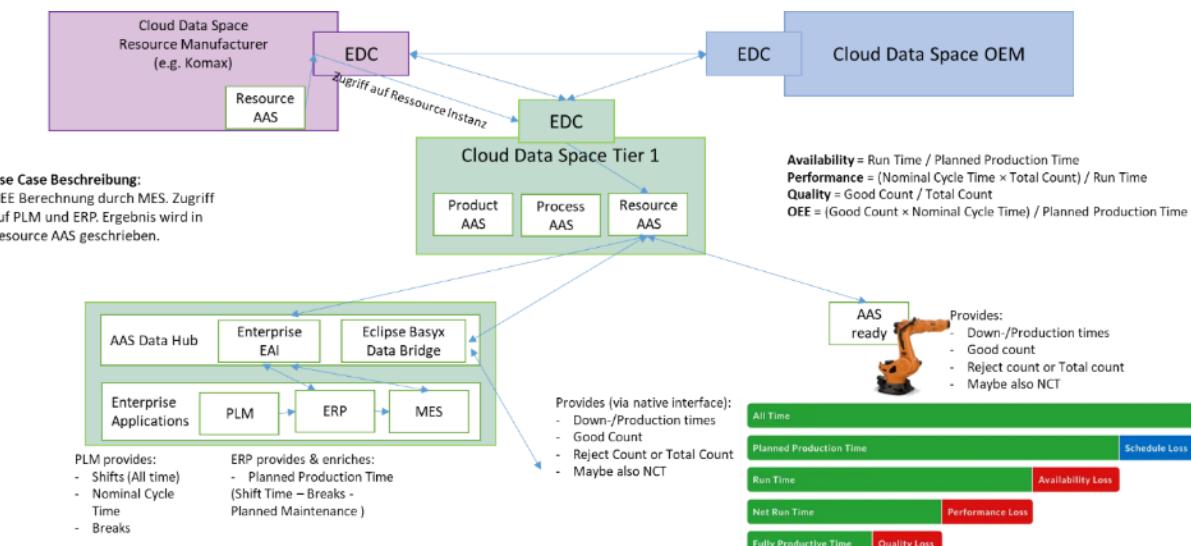


Abbildung 4-5: OEE-Berechnung: MES greift auf Ressource AAS zu

In diesem Use Case kommt folgende Annahme zum Einsatz. Die "Eclipse BaSyx Data Bridge" ist im Einsatz und ermöglicht somit die Anbindung von Maschinen mit "hur" proprietären Schnittstellen. Des Weiteren wird unterstellt, dass die AAS der Maschine mit der AAS-Kopie der Ressource des Tier 1 synchronisiert ist, bzw. wird (s. 5). Ein MES muss daher nicht mehr zwingend mit der Ressource direkt kommunizieren, sondern kann über verfügbare EAI (Enterprise Application Integration) Zugang zu den benötigten Daten erhalten. Bei der Anreicherung der AAS der Ressource mit OEE-Daten ist darauf zu achten, dass auch die Ermittlung der einzelnen OEE-Bestandteile (Availability, Perfomance, Quality) transparent dargestellt wird. Sowohl in Bezug auf die Datenbasis als auch die Berechnungen.

4.2.3 CAD2BOP

Hinter „Computer-Aided-Design to Bill-of-Process“ (CAD2BOP) steht die Erstellung eines Konzepts zur Definition von CAD-Daten und weiteren Parametern, die einen starken Einfluss auf eine spätere Automatisierung der Produktions- und Montageprozesse in der Wertschöpfungskette für Leitungssätze haben. Im Speziellen werden hier Gehäuse und Kontakte betrachtet, um den Prozess der Steckerbestückung (das sog. „Blockload“) mit Automatisierungsparametern zu unterstützen.

4.2.3.1 Ist-Zustandsbeschreibung

Für die Bestückung von Gehäusen werden teilautomatisierte Maschinen verwendet (z.B. [Komax Omega Serie](#)), die dazu in der Lage sind, nach der Fertigung der Kabel und der Kontaktierung diese entsprechend in ein Gehäuse zu platzieren.

Hierzu werden unbestückte Gehäuse auf einer Palette mit speziell angefertigten Halteteilen platziert (siehe Abbildung 4-6). Für jede Kammer muss der Bediener anschließend die Position, Stecktiefe, Ausrichtung der Verriegelung sowie die Steck- und Rückzugskraft anlernen. Das ist notwendig, um einen sicheren Sitz der Kontaktteile nach der Herstellung zu gewährleisten. Um einen wiederholbaren Prozess zu gewährleisten, muss der Bediener zudem jede weitere Palette nach demselben Muster bestücken.

Ebenso ist das Prüfen solcher Gehäuse mit einem hohen Maß an Handarbeit verbunden. Hier werden oft die Gegenstellen durch manuelle Vermessung der Originale hergestellt, statt sie aus bestehenden Konstruktionen abzuleiten.

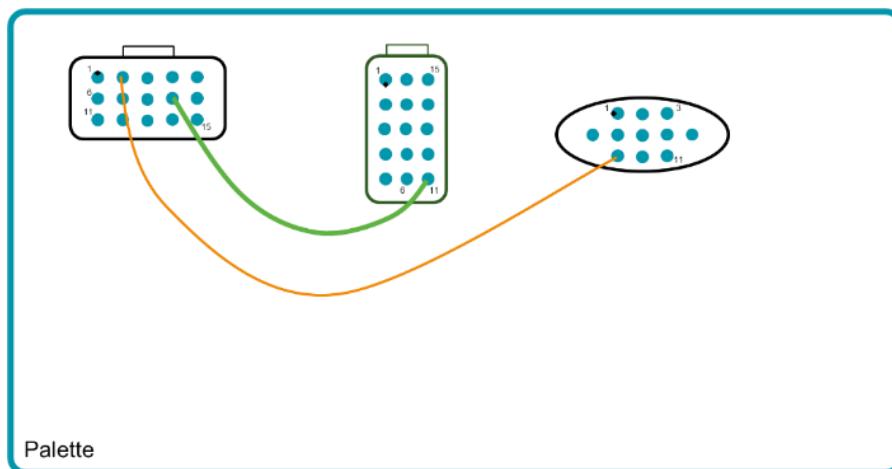


Abbildung 4-6: Palette mit zu bestückenden Gehäusen

4.2.3.2 Anforderungen

4.2.3.2.1 Allgemein

- Die nachfolgenden geometrischen Beschreiben, vor allem für die elektrischen Steckverbindungen (Gehäuse), bezüglich des Nullpunktes und der Achssysteme haben den „Leitfaden zur Erstellung von Elektrik-Bauteilen mit CATIA V5“ des CES-Arbeitskreises (CES = Car Electrical System) als Grundlage.
- Das Informationsmodell des VEC [6] bildet die Datenbasis.
- Daten, die nicht aus dem VEC extrahiert werden können, müssen über die VWS bereitgestellt werden.
- CAD-Daten (z.B. im STEP-format) liefern auch bei vereinfachten Modellen aussagekräftige Informationen (z.B. über Außenmaße).
- Eine einheitliche Ausrichtung der Konstruktionen und Referenzpunkte muss definiert sein.

4.2.3.2.2 Gehäuse (siehe Abbildung 4-7)

- Der Mittelpunkt der Kammer 1 in der Ebene der elektrischen Kontaktfläche des Gehäuses ist der Nullpunkt Gehäuses und der Ursprung des Koordinatensystems.
- Die Z-Achse entspricht bei Stiftgehäusen der Steckrichtung, somit zeigt diese nach außen. Bei Buchsengehäusen zeigt die Z-Achse in das Gehäuse hinein und verläuft gegen die Steckrichtung des Gehäuses. Siehe dazu auch Abschnitt 4.2.3.3.1.
- Die X-Achse zeigt auf weitere Reihen (sofern vorhanden). Diese wird auch als Spurmaß bezeichnet.
- Die Y-Achse zeigt auf die nächste Kammer derselben Reihe (sofern vorhanden). Diese wird auch als Rastermaß bezeichnet.
- Orientierung/Drehwinkel der Primärverriegelung im Winkelmaß gegenüber der Y-Achse im Uhrzeigersinn um die Z-Achse (siehe Beschreibung von VEC-Gehäusen).
- Alle weiteren Koordinatensysteme referenzieren auf das Ursprungskoordinatensystem.

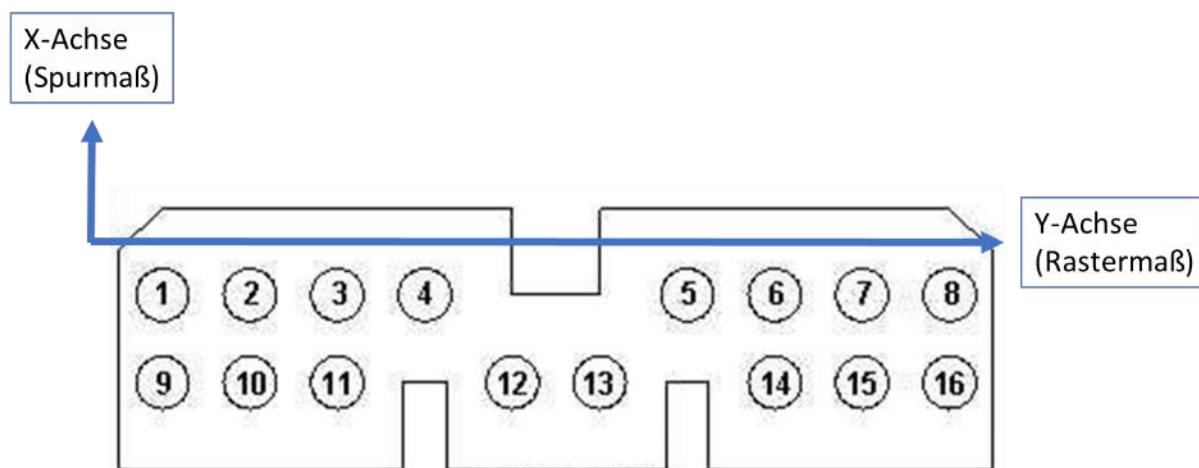


Abbildung 4-7: Gehäuse

4.2.3.2.3 Kontakt (siehe Abbildung 4-8, 4 und 5)

- Der Nullpunkt des Terminals liegt im Kreuzungspunkt der Mittelachse (Z-Achse) und der Endlage des Terminals in der Kammer (Endlage des Terminals in der Kammer, siehe Abbildung 4-8).
- Die Z-Achse des Terminals orientiert sich an der Richtung der Z-Achse des Gehäuses. Beim Buchsenterminal zeigt die Z-Achse in Richtung des Crimps und beim Stiftterminal entgegen der Richtung des Crimps.
- Die Y-Achse zeigt in Richtung der Primärverriegelung (0°).
- Die X-Achse ergibt sich aus der Rechte-Hand-Regel
- Um eine Orientierung im Raum zu gewährleisten, ist ein weiterer Parameter notwendig (Parameter Name: CrimpOrientation). Dazu wird der Drehwinkel der Crimpöffnung (bei F-Crimp) im Verhältnis zur Y-Achse angegeben. Ausschließlich positive Werte und analog zur Bestimmung des Drehwinkels wie beim Gehäuse (im Uhrzeigersinn). Beispiele dazu zeigen die Abbildung 4-8.



Abbildung 4-8: Kontaktbeispiel 1 (Y-Achse zeigt in Richtung Primärverriegelung)

Beim Ein- / Zusammenführen des Terminals in die Gehäusekammer ist zu beachten, dass die Nullpunkte bzw. die Achsen der Nullpunkte von Gehäuse und Terminal in der Endposition nicht übereinander liegen. Die Beschreibung der Nullpunkte für Terminal und Gehäuse wurde bereits zuvor gemacht und in der Abbildung 4-9 nochmal visuell verdeutlicht. Die Distanz zwischen den beiden Nullpunkten kann mit einem Offsetwert angegeben werden und daraus die Endposition des Terminals in der Kammer bestimmt werden.

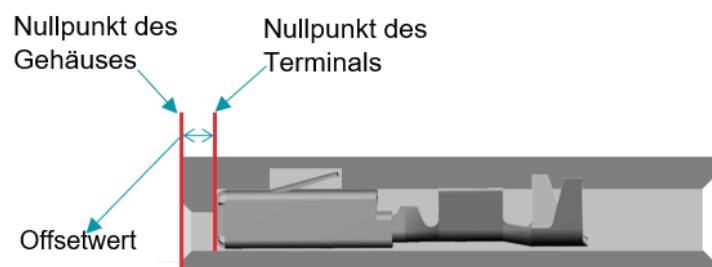


Abbildung 4-9: Definition des Nullpunkts für das Terminal und gegenüber dem Gehäuse

Zu beachten ist beim Stiftkontakt, dass sich der Nullpunkt nicht an der Spitze des Terminals befindet, sondern analog zum Buchsengehäuse/-terminal, wie in Abbildung 4-10 gezeigt, dort wo sich das Kammerende befindet.

Zudem ist beim Stiftkontakt eine zusätzliche Längenangabe des Stiftes (vom Nullpunkt bis zur Kontaktspitze) notwendig. Dies ist erforderlich um eine „Hindernisumfahrung“ zu ermöglichen, damit das Terminal nicht kollidiert wenn es zu der erforderlichen Kammer geführt und anschließend gesteckt wird.

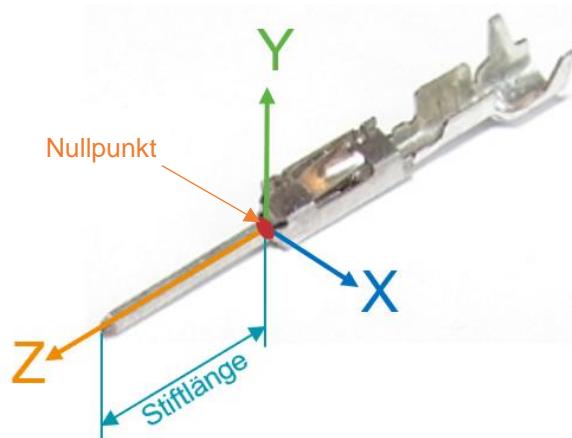


Abbildung 4-10: Stiftkontakt

Besonderheit bei Terminals mit mehreren Primärverriegelungen

Bei Terminals mit mehreren Primärverriegelungen (Beispiel siehe Abbildung 4-11) obliegt es dem Konstrukteur/Hersteller des Terminals zu welcher Primärverriegelung die Y-Achse ausgerichtet ist, dies gilt nur wenn die Terminals in mehreren Positionen steckkompatibel in der Kammer sind.

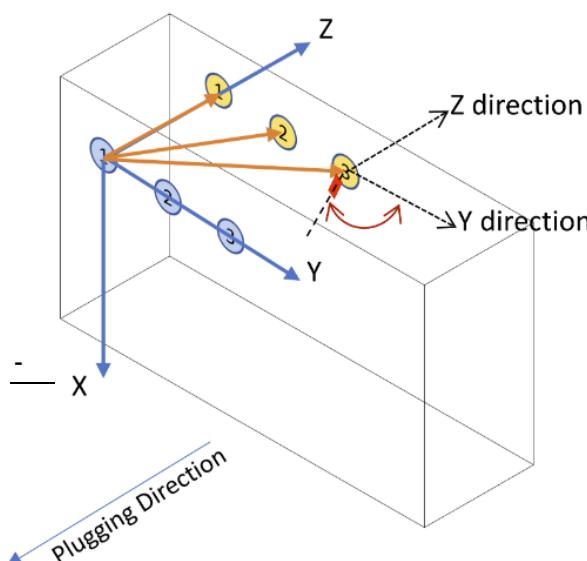
Sollte bei mehreren Primärverriegelungen keine Steckkompatibilität in mehreren Positionen vorhanden sein, ist die Ausrichtung der Achsensysteme, von Terminal und Gehäuse, analog wie bei Terminals mit nur einer Primärverriegelung auszuführen.



Abbildung 4-11: Kontaktbeispiele mit zwei Verriegelungen

4.2.3.3 VEC als führender Standard

Dieses Konzept basiert auf diesem Auszug aus den Spezifikationen des VEC (Vehicle Electrical Container, Abbildung 4-12):



- ① Cavity opening within the electrical contact surface
- ② Cavity opening within the entrance surface of the connector during the insertion process of the terminal.
- Position vector of the cavity (center entrance surface)
- Postion of the primary locking
- ↷ Rotation angle of the primary locking against the Y-axis, around Z-Axis (right hand rule)

Abbildung 4-12: VEC-Definition für Steckrichtung und Achsensystem (Source Prostep)

Der folgende Abschnitt enthält alle relevanten Definitionen, die für die entsprechenden Modellelemente im VEC erforderlich sind.

Die geometrischen Eigenschaften von Steckergehäusen stehen in engem Zusammenhang mit der LocalGeometrySpecification, die im Abschnitt "Koordinatensysteme der Komponenten" beschrieben wird.

Alle Koordinaten und Werte sind im gleichen Koordinatensystem definiert. Alle geometrischen Eigenschaften, die nicht ausschließlich für Verbindergehäuse gelten, werden dort definiert (z.B. BoundingBox, PlacementPoints / SegmentConnectionPoints).

Das 3D-Modell und das Koordinatensystem eines Steckergehäuses sind wie folgt definiert:

- Der Ursprung des Steckers liegt in der Ebene der elektrischen Kontaktfläche in der Mitte der Öffnung der/des Kammer/Stifts mit der kleinsten alphanumerischen Kennung (z. B. Hohlraum/Stift 1 oder A1).
- Die Y-Achse liegt in der Ebene der elektrischen Kontaktfläche und zeigt in Richtung der Mitte der/des Kammer/Stifts mit der nächsthöheren alphanumerischen Kennung (z. B. Kammer/Stift 2 oder A2). Bei einpoligen Steckergehäusen in Richtung der Kodierung, sofern vorhanden.
- Die Z-Achse steht senkrecht zur Ebene der elektrischen Kontaktfläche und zeigt bei Buchsengehäusen in das Gehäuse und bei Stiftgehäusen nach außen (der Stecker in der Abbildung oben ist eine Buchse). Bei weiblichen Gehäusen zeigt die Z-Achse also in die entgegengesetzte Richtung der Steckrichtung. Bei männlichen Gehäusen zeigt sie in die gleiche Richtung wie die Steckrichtung.
- Die X-Achse ist das Vektorprodukt von Y und Z.

Für Kammern gibt es zwei relevante Positionen im Stecker:

- Die Position in der elektrischen Kontaktfläche (blau hervorgehoben), und
- Die Position in der Eintrittsfläche während des Einführens (gelb hervorgehoben).

Die Lage der Kammer innerhalb des Verbinders wird als der Winkel zwischen der Y-Achse und der primären Verriegelung in der Kammer definiert.

<https://ecad-wiki.prostep.org/specifications/vec/v202/component-characteristics/geometric-properties-of-connector-housings-definitions/>

In den folgenden Unterkapiteln werden die erforderlichen Attribute für die Automatisierung genannt und beschrieben. Es wird dabei zwischen Attributen des Gehäuses und des Kontakts unterschieden.

4.2.3.3.1 Attribute des Gehäuses

In diesem Abschnitt wird nur das Bezugssystem des Steckverbinders beschrieben. Für die Automatisierung an einer Maschine muss das Bezugssystem entsprechend den entstehenden Anforderungen erweitert werden. Diese Möglichkeit bietet z.B. die OPC UA Companion Specification "Relative Spatial Location".

4.2.3.3.1.1 HousingCoordinates

- Der Ursprungsnnullpunkt muss auf der Mitte der Kammeröffnung mit der kleinsten alphanumerischen Zahl (z. B. 1, A1 usw.) innerhalb der elektrischen Kontaktfläche liegen. Er liegt immer in der Ebene der X- und Y-Achsen
- Die Ausrichtung der Achsen entspricht der VEC-Spezifikation

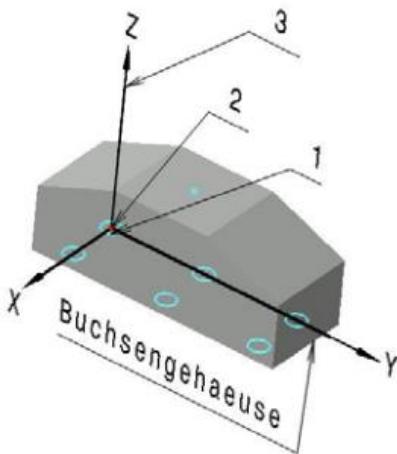


Abbildung 4-14: Buchsengehäuse mit Pinbild

1=Pin 1, 2=BauteilUrsprung, 3=Achsensystem

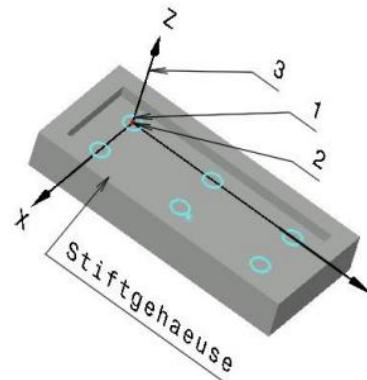


Abbildung 4-13: Stiftgehäuse mit Pinbild

1=Pin 1, 2=BauteilUrsprung, 3=Achsensystem

4.2.3.3.1.2 CavityNumber

- Jede Kammer eines Gehäuses hat seine eigene eindeutige Nummer (z. B. "A5") zur Identifizierung

4.2.3.3.1.3 CavityId

- Definiert den Typ der Kammer (z. B. NanoMQS)

4.2.3.3.1.4 CavityRotation

- Winkel zwischen Y-Achse und Primärverriegelung (um die Z-Achse im Uhrzeigersinn)
- Der Winkel ist immer positiv und beginnt bei 0° auf der Y-Achse und dreht sich im Uhrzeigersinn.

4.2.3.3.1.5 CavityCoordinate

- Die Koordinaten des Mittelpunkts der Kammeröffnung innerhalb der elektrischen Kontaktfläche in Bezug auf den Ursprungsnulnpunkt (CoordinatesHousing). Sie liegt immer in der Ebene der X- und Y-Achsen

4.2.3.3.1.6 OffsetInsertDirectionXCavityCenter

- Verschiebung des Einführpunktes auf der X-Achse (notwendig, wenn Nullpunkt der Kammer und des Kontakts nicht übereinstimmen)

4.2.3.3.1.7 OffsetInsertDirectionYCavityCenter

- Verschiebung des Einführpunktes auf der Y-Achse (notwendig, wenn Nullpunkt der Kammer und des Kontakts nicht übereinstimmen)

4.2.3.3.1.8 InsertionDistance

- Einstecktiefe, die der Kontakt erreichen muss (gemessen von der Kammeröffnung innerhalb der Eintrittsfläche). Die Bewegung erfolgt ausschließlich entlang der Z-Achse (Richtung variiert je nach männlich oder weiblich)

4.2.3.3.1.9 InsertionTrajectory

- Einfügepfad (Richtung, Drehung, Geschwindigkeit) ist ein optionales Attribut

4.2.3.3.2 Attribute des Kontakts

Die Konstruktion des Kontakts ist das Gegenstück zur VEC-Spezifikation des Gehäuses. Das bedeutet, dass die Steckrichtung des Kontakts ebenfalls über die Z-Achse dargestellt werden muss. Der Ursprungsnnullpunkt des Koordinatensystems des Kontakts muss in der Mittellinie des Kontakts an der Endposition der Kammer liegen.

TerminalOrientation: Der Winkel der Primärverriegelung um die Z-Achse. Immer positiv, beginnend mit 0° auf der Y-Achse und im Uhrzeigersinn

TerminalCoordinates: Der Ursprungsnnullpunkt des Kontaktes muss auf der Endlage in der Kammer an der Mittelline liegen. Die Richtung der Z-Achse zeigt in Richtung Crimp der Z-Achse des Gehäuses und ist die Bewegungsachse der Steckrichtung

MaximumInsertionForce: Die maximale Kraft, die beim Einführen bis zur primären Verriegelung des Kontakts aufgebracht werden darf.

MaximumTerminalPullOutForce: Die maximale Auszugskraft, die aufgebracht werden darf, bis die Primärverriegelung zerstört ist.

CrimpOrientation: Bezugspunkt der Öffnung des Crimp in Relation zur Primärverrastung (Y-Achse) des Terminals, wird in Grad (ausschließlich positive Werte) angegeben, Drehrichtung im Uhrzeigersinn.

MaximumTerminalPullOutLength: Maximale Auszugslänge.

Overlength: Überlänge (Kabelspitze bis Terminal Ende).

4.2.3.4 Umsetzungskonzept

Viele benötigte Parameter liegen bereits im VEC vor [32] und müssen Anwendung in der Konstruktion der entsprechenden Bauteile finden. Vereinfachte CAD-Modelle (z.B. im STEP-Format) müssen eine klare Auskunft über die Außengeometrie von Gehäusen sowie notwendigen Informationen über die einzelnen Kammern liefern. Diese Daten sind essenziell für eine automatisierte Bestückung, da sie ein in sich geschlossenes Bezugssystem haben.

Dieses Bezugssystem muss nun in einer Maschine in Relation gesetzt werden. Hierfür ist folgendes Szenario entstanden:

Die Gehäuse werden mittels Universalhalterungen frei auf einer Palette (früher Lochplatte) fixiert. Ein Vermessungssystem ermittelt die relative Positionierung zum Nullpunkt der Palette. Die Palette selbst hat zum Zeitpunkt der Bestückung eine definierte Position relativ zum Nullpunkt der Maschine (z.B. durch Nullstellung des Greifarms). Aus der relativen Positionierung der unterschiedlichen Komponenten und der Beschreibung des Gehäuses lässt sich somit vollautomatisch die Kammerposition ermitteln und der Steckvorgang durchführen. Für die Ausführung der relativen Positionsermittlungen kommt das bereits bestehende OPC UA-Informationsmodell OPC 10000-210 (Industrial Automation - Relative Spatial Location) [33] zum Einsatz. Fehlende Informationen, wie Einstekttiefe und Rückzugskraft, müssen anderweitig erfasst werden (z.B. durch den Maschinenbediener oder die Arbeitsvorbereitung).

4.2.3.5 Fazit

Unter Verwendung des VEC-Datenmodells und OPC UA-Informationsmodells OPC 10000-210 [33] kann ein hoher Grad der Automatisierung im Bereich der Gehäusebestückung erreicht werden. Voraussetzung hierfür ist, dass die CAD-Modelle sich an den Vorgaben des VEC orientieren und über alle Hersteller einheitlich darstellen, oder die Verwaltungsschale eine Möglichkeit zur Querreferenzierung der übermittelten Daten bietet (z.B. Steckrichtung Kontakt = -X).

4.3 AP 3.3 - Erforschung von Teilmodellen für Rückmeldedaten

Das Arbeitspaket 3.3 „Erforschung von Teilmodellen für Rückmeldedaten“ beschäftigt sich mit den Teilmodellen für Rückmeldedaten zum Abgleich der im Fertigungsprozess gemessenen Ist-Werte.

4.3.1 Leitungssatzproduktion

Bei der Herstellung eines Leitungssatzes werden eine Reihe von zugelieferten Rohmaterialien, Bauteilen und Halbfertigprodukten verarbeitet, die in komplexen Produktionsprozessen zum Endprodukt eines Leitungssatz in der gewünschten Variante verarbeitet werden.

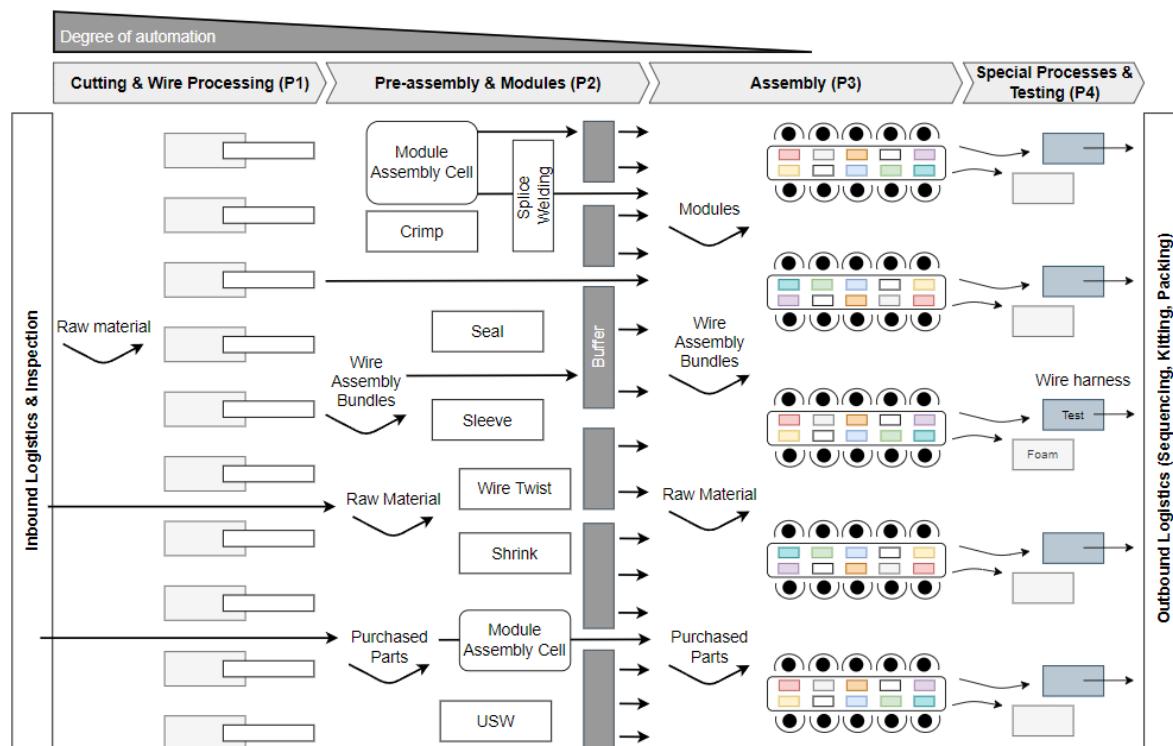


Abbildung 4-15: Produktionsprozesse beim Konfektionär

Bevor die eigentliche Kabelbaumproduktion beginnt, treffen alle notwendigen (Roh-)Materialien in der Eingangslogistik oder im Lagerbereich ein, wo diese gescannt werden. Auch findet eine Qualitätsprüfung der angelieferten Materialien statt.

Die Leitungssatzproduktion beginnt mit dem **Schneidprozess und der Leitungsverarbeitung (P1)**. Diese Prozesse werden im sog. Schneidraum durchgeführt und weisen einen sehr hohen Automatisierungsgrad auf. Der Schneidprozess wird derzeit von Leitungsverarbeitungsmaschinen durchgeführt, die von einem MES-System gesteuert werden. In der Regel übernehmen die Maschinen in diesem Schritt die Prozesse des Ablängens, des Abisolierens, des Aufbringens von Dichtungen und Kontaktteilen und ggf. der Beschriftung durch Etikettierung, Farbcodierung oder maschinenlesbare Markierungen.

Der **Vormontageprozess (P2)** ist teilautomatisiert. In diesem Schritt finden verschiedene Prozesse statt, wie z. B. manuelles Konsolidieren, Ultraschallschweißen (USW), Abdichtung und viele andere, siehe Abbildung 4-15.

Der **Endmontageprozess (P3)** wird überwiegend in Handarbeit durchgeführt, wobei die Teile und Module aus P2 als Input für die Leitungssatzmontage dienen, in der diese Inputs montiert, gewickelt, abgedeckt und das Layout fertiggestellt werden.

In der Stufe "Spezielle Verfahren und Prüfungen" (P4) ist es möglich, dass spezielle Verfahren wie z.B. das Schäumen durchgeführt werden. Am Ende dieser Phase wird der Testprozess durchgeführt, um die Qualität des Leitungssatzes zu überprüfen, wobei die Qualität des gesamten Leitungssatzes und

all seiner Komponenten überprüft wird, einschließlich des Vorhandenseins von Komponenten, elektrischen Anschlüssen, Widerständen, Kondensatoren, Dioden, Drahtfarben, Pinpositionen, Glasfaserdrähten, Airbags, Thermistoren und mehr.

Schließlich wird der Produktionsprozess abgeschlossen, so dass die Kommissionierung und die Verpackung erfolgen können, bevor der Leitungssatz das Werk verlässt.

Jeder dieser Bereiche und die Produktionsschritte benötigen und generieren Informationen über das Produkt (Bauteile, Baugruppen, Kabelbäume usw.), die Produktionsprozesse (Schneiden, Crimpeln, Blockladen, Prüfen usw.) und die Produktionsressourcen (Maschinen, Werkzeuge, Bediener usw.). Für eine generalisierte Prozessbeschreibung müssen die Informationselemente in einer standardisierten Beschreibung zugänglich gemacht werden. Die in Abbildung 4-16 dargestellte Vorgehensweise soll die Aggregation der einzelnen Attribute für die Ausführung des Prozesses aus verschiedenen Quellen verdeutlichen. Der Fokus liegt auf der Abhängigkeit der verschiedenen Bereiche.

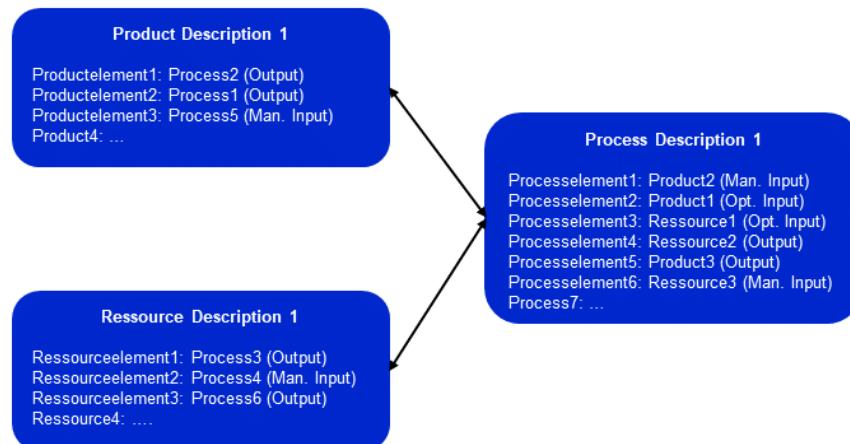


Abbildung 4-16: PPR-Modell

4.3.2 Produktionsprozesse

Entsprechend der Vorhabensbeschreibung zum AP 3.3 wurden die Produktionsprozesse der Leitungssatzfertigung identifiziert und dokumentiert. Nachfolgend soll die Entwicklung dieser Ausarbeitung und der dazugehörigen Ergebnisse dargestellt werden, bis hin zur finalen Version dieser Dokumentation. Im letzten Abschnitt dieses Kapitels wird noch darauf eingegangen, wie die Ausarbeitungen dazu als Basis weiterer Standardisierungsvorhaben genutzt werden konnten.

4.3.2.1 Definitionsphase der Einzelprozesse in der Produktion eines Leitungssatzes

Initial wurde eine erste Auswahl an Produktionsprozessen erarbeitet, inklusive der dazugehörigen Input- und Output-Parameter. Ziel war es die relevanten Informationen zur Leitungssatzfertigung zusammen zu fassen. Ursprünglich umfasste der fertige Erstentwurf der Liste 25 Prozesse, 76 Input-Parameter und 73 Output-Parameter (Abbildung 4-17). Diese Daten sollen hier nur, als Randnotiz erwähnt werden, um später im Dokument die Entwicklung dieser Liste nachvollziehen zu können. Die abgebildete Excelliste ist in der Datei „[Initial_Process_List.xlsx](#)“²⁰ zu finden

²⁰https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP03/Beispieldaten/Initial_Process_List.xlsx

| Prozess | Spezialisierung | Inputdaten | Outputdaten |
|----------------|----------------------|--|--|
| Cut | Cut Wire | Leitungslänge, -zuschläge, Gehäuseinformationen, Materialnummer (Artikelnummer) | Ist-Leitungslänge (Musterleitung), Chargennummer Leitungsbündel (Artikelnummer) |
| Cut | Tube | Länge, Zustandsbeurteilung Schlitzen (ja/nein), Materialnummer (Artikelnummer), Soll-Losgroße für Bündel | Ist-Länge (Muster), Chargennummer Schlachtmutterware, Chargennummer Produkt, Beziehung Chargennummer |
| Crimp | Terminal | Crimpkraft, Crimpbereich (Breite +/-tol.), Materialnummer Terminal + Leitung, Werkzeug, Abzugskraft | Eingangsmaterial zu Ausgangsmaterial (Bsp. Schneidekartennummer 1000 Eingang in 20*50 Ausgang), Ist-Losgroße |
| Crimp | Splice | Crimpkraft, Materialnummer Leitung + Crimpnähte, Werkzeug | Ist-Leitungslänge (Musterleitung --> Crimpföhre /-breite, Abzugskraft), Werkzeugseriennummer, Crimpkurven, Ausschuss |
| Print | Wire | Text, Druckfarbe, Position, Abstand | WEZAG-Crimpzangen oder Maschinenautomatisierung, Chargennummer Crimpnähte, Werkzeug |
| Melt | | Maximaltemperatur, Dauer, Menge Lötzinn, Materialnummer Lötzinnmaterial | |
| Strip | Mittenausisolierung | Absisolierlängen, Position | Ist-Temperatur, Ist-Dauer, Chargennummer Eingangsmaterial, Ausgangsmaterial, Werkzeugseriennummer |
| Strip | Leitungsende | Absisolierlängen, Tiefabzug | Ist-Leitungslänge (Musterleitung), Chargennummer Leitungsbündel (Artikelnummer), Ausschuss |
| Seal | | Leitungsaufendurchmesser, Kompatibilität zu Kontaktteil, Materialnummer Seal, Werkzeug | Ist-Chargennummer Eingangsmaterial, Ausgangsmaterial, Seal Applicator (Werkzeugseriennummer) |
| USW | Verbinder | Leitungsanzahl, Leitung mit Querschnitt und Typ, Lage der Leitung | Schwelldruck, -amplitude, -zeit, -energie; (Musterleitung --> Schwellbreite, Schälwert) |
| USW | Kontaktteil | Lage der Leitung, Querschnitt und Typ, Materialnummer Kontaktteil | Schwelldruck, -amplitude, -zeit, -energie; (Musterleitung --> Schwellhöhe und -breite, Mindestzugwert, Mindestschälwert) |
| Shrink | | Nennquerschnitt, Materialnummer (Schrumpffaktor Schlauch, mit vs. ohne Kleber), Zeit- und Temperaturvorgabe | Ist-Temperatur, Ist-Dauer, Chargennummer Eingangsmaterial, Ausgangsmaterial |
| Twist | | Schlaglänge, Endrilllänge, Vorgabe Endrillschutz (Bündchenposition), Materialnummer der Leitungen und Anzahl, Verdrillkraft, Elastizität, Leitungsumstand (Länge und welches Leitungsende) | Chargennummer Eingangsmaterial, Maschinenseriennummer |
| NC | | Non-order pins, Identifizierung von belegten/unbelegten Kammern --> Suche Steckfehler | Ist-kontaktierte Pin's, Ist zuviel -gesteckte Pin's gegen Order Pin's, Positionsnummern |
| Offnetest | | Prüfung aller Leitungen inkl. Netzwerkbildung | Ist-alle Leitungen der Order, Fehlerdaten |
| Kurzschluss | | Prüfung der Netzwerke auf Kurzschlüsse, gegenseitige Prüfung der Netzwerke | Ist-Auflistung der Verbindungen welche Kurzschluss zueinander haben |
| Schaltertest | | Prüfung Codierungen, Sekundärverriegelung, Deckel, Druckprüfung, usw. | Ist-Auflistung zu testender Zusatzauteile |
| Post Open Test | | Fehlerloser Kabelbau --> Rückverbindungen --> Anlieferzustand von 2 Gehäusen zusammengesteckt --> Allgemein | |
| Route | | Weiterleitweisen nach der Prüfung | |
| Mount | | Verlegenweg | |
| Cover | Fuse Box | Position, Sicherungstyp (Materialnummer), Steckkraft und -tiefe Bewickungsart, Materialnummer Tape, Strecke (Start- und Endpunkt), Steigung | Steckkraft und -tiefe, Chargennummer (Sicherung), Seriennummer (Fuse Box) |
| Foam | | Menge/Volumen, Leitungsanzahl | Chargennummer Eingangsmaterial, Maschinenseriennummer |
| Test | Camera Test Fuse Box | Bestückung = Position + Sicherungstyp | Chargennummer Eingangsmaterial, Werkzeugseriennummer, Maschinenseriennummer, Schäumtemperatur und -zeit, Durchfluss, Werkzeugtemperatur, Schaumdruck |
| Schrauben | | Materialnummer der Mutter, Drehmoment, Drehwinkel, Schraubposition | Kamerabildd als Testnachweis |
| | | | Materialnummer der Mutter, Chargennummer Eingangsmaterial, Ist-Drehmoment, Ist-Drehwinkel, Schraubposition, Seriennummer der Station |

Abbildung 4-17: Initiale Prozessliste

Diese Liste bildete die Grundlage für alle nachfolgenden Aktivitäten.

4.3.2.1.1 Darstellung als UML-Diagramm

Aufbauend auf der initialen Prozessliste wurde wegen der beschränkten Übersichtlichkeit des Formates diese Liste in ein UML-Klassendiagramm übertragen (Abbildung 4-18). In mehreren Arbeitsterminen konnten weitere Prozesse mit den notwendigen Parametern angereichert werden, d.h. sowohl Soll- als auch Ist-Werte und dabei auch schon, wo es möglich war, ersten Datentypen zu den Parametern definieren.

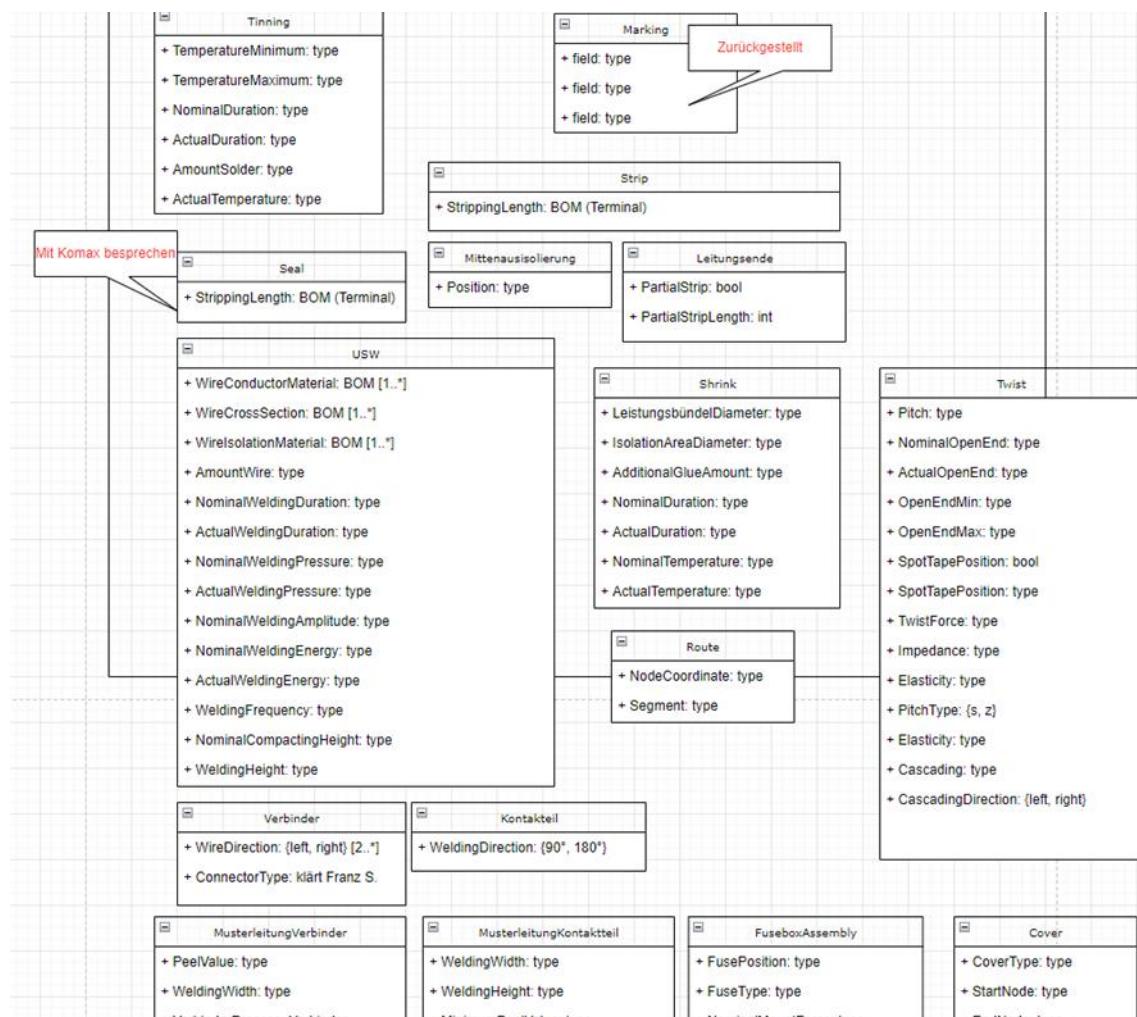


Abbildung 4-18: UML-Klassendiagramm für Produktionsprozesse

Bei der Erstellung und weiteren Ausarbeitung des UML-Diagramms wurden initial zwei generische Klassen definiert (Abbildung 4-19) und in den sogenannten allgemeingültigen Bereich „Common“ überführt. Der Hintergrund dieser Vorgehensweise war, dass diese Datenmodelle für jeden Prozess relevant sind und möglichst Redundanzen bezüglich der Parameter in den Prozessdefinitionen vermieden werden sollte. Durch dieses Vorgehen konnten die jeweiligen Prozessdefinitionen auf die wesentlichen spezifischen Parameter beschränkt gehalten werden.

Die Hauptklasse wurde „Manufacturing Process“ genannt und soll alle übergreifenden Parameter enthalten, die zur Durchführung und Dokumentation des Prozesses erforderlich sind. Dazu gehören beispielsweise Start- und Endzeit der Prozessausführung, das Ergebnis des Prozesses usw. Des Weiteren wird zwischen dem geplanten Materialeinsatz (*Consumption*) und ungeplantem Mehrverbrauch (*AdditionalConsumption*) unterschieden.

Die zweite Klasse „Material Consumption“ enthält die notwendigen Informationen hinsichtlich des Materialeinsatzes, die für die Ausführung des zugehörigen Prozesses zur Erzeugung des Produktes notwendig sind, beispielsweise die Parameter Material_Id, Amount, Unit usw.

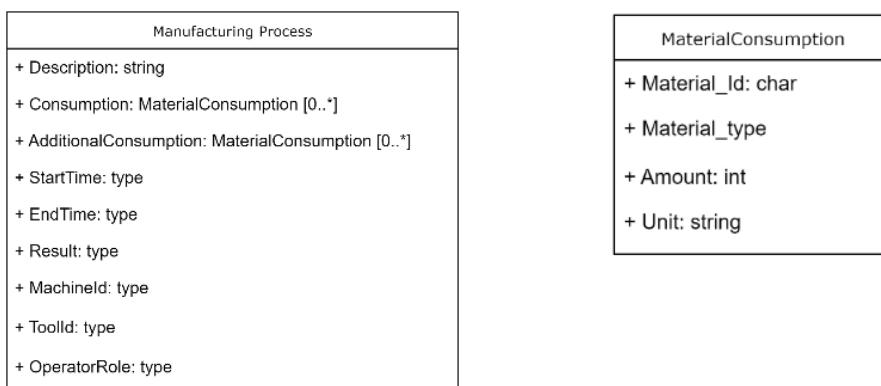


Abbildung 4-19: Allgemeingültige Klassen

Im weiteren Verlauf wurden das Thema „Testing“, das bis dato eher rudimentär mitgeführt wurde, im UML-Diagramm spezifischer ausgearbeitet und eingetragen. Dabei wurden verschiedene Testprozesse, die beispielsweise bei der End-of-Line-Prüfung eines Leitungssatzes anfallen können, mit den entsprechenden Parametern erarbeitet.

Das Layout und die Benamungen der Prozesse inklusive der dazugehörigen Parameter mit Datentypen wurden im Laufe der Arbeitstermine immer weiter verfeinert, bis das UML-Diagramm in seiner finalen Version abgeschlossen wurde. Die finale Version des UML-Diagramms ist im Anhang hinterlegt.

Im nachfolgenden Unterkapitel werden noch die Gründe der Verwendung des ECLASS-Standards für das weitere Vorgehen bei der Erstellung der Prozessübersicht erläutert.

4.3.2.1.2 Semantische Interoperabilität

Um eine interoperabel agierende Wertkette zu ermöglichen, müssen die einzelnen Informationselemente mittels eindeutiger semantischer Beschreibungen für den Mensch als auch für die Maschine eindeutig interpretierbar gemacht werden.

Zu diesem Zweck existiert der [ECLASS-Standard](#) [8] zur Klassifizierung und Beschreibung von Produkten und Dienstleistungen (Datenwörterbücher basieren auf der ISO 13584-42 und IEC 61360-2). Die dort eingepflegten Informationen erhalten eine global eindeutige ID (eine sog. IRDI) und sind somit eindeutig für alle Teilnehmer, die ebenfalls den ECLASS-Standard nutzen, interpretierbar. Das Datenmodell in Abbildung 4-20 gibt einen Überblick, welche Informationen für die Beschreibung der einzelnen Informationselemente notwendig sind. Für eine detaillierte Erläuterung des Datenmodells wird auf die [technische Spezifikation von ECLASS](#) [34] verwiesen. In unserem Anwendungsfall wurde vorwiegend auf die Struktur-Elemente **Property**, **Value** und **Unit** zurückgegriffen.

Property: Ein **Property** (Eigenschaft) beschreibt eine bestimmte Eigenschaft und weist dieser Eigenschaft durch eine Beziehung einen definierten **Value** zu (Bspw. Eigenschaft: Leitungsfarbe, Value: Rot).

Value: Der Value ist eine Aufzählung innerhalb einer PROPERTY, die die Menge der möglichen Werte für die Eigenschaft einschränken.

Unit: Die Unit setzt eine Maßeinheit in Beziehung zur Basisdimension des zugrunde liegenden Einheitsystems.

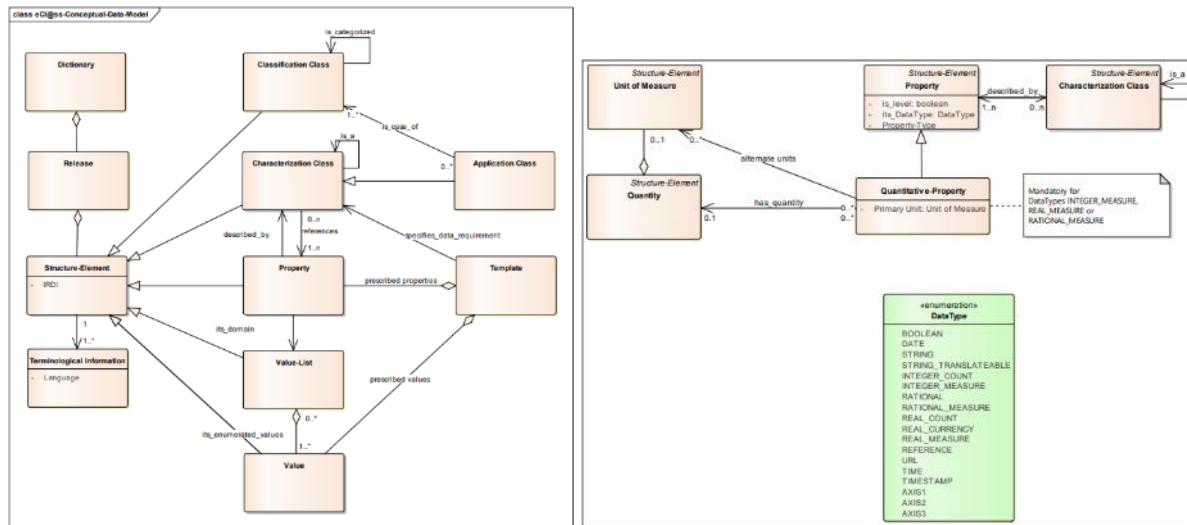


Abbildung 4-20: ECLASS Datenmodell und Detaillierung des Properties (Quelle: [ECLASS](#) [34])

Die hierarchische Struktur, die in ECLASS verwendet wird, kann ohne großen Aufwand auf das UML-Diagramm gemappt werden. Der Fokus zu diesem Zeitpunkt lag auf den Prozessen *Common*, *Cut* und *Crimp*. Für diese Prozesse wurden ECLASS Klassen angelegt. Während der Übertragung mussten den einzelnen Informationselementen noch weitere Details hinzugefügt werden, um eine semantische Beschreibung zu ermöglichen:

- Definition
- Data-Type
- Preferred-Name
- Property-Data Type

Nach der Anreicherung wurde für jedes Informationselement eine International Registration Data Identifier (IRDI) erzeugt, die global von dem System lediglich einmal vergeben wird. Der Aufbau eines solchen Identifiers kann in den Spezifikationen von ISO/IEC 11179-6, ISO 29002 and ISO 6532 näher betrachtet werden.

4.3.2.2 Struktur der finalen Prozessliste

Die [Prozessliste](#)²¹ bildet für das weitere die Grundlage. Wie im letzten Unterkapitel beschrieben, musste eine Anreicherung an Informationen gemacht werden, um die Daten für eine Interoperabilität zu präparieren. Aufgrund der begrenzten Möglichkeiten dies auf eine übersichtliche Weise im UML-Diagramm darzustellen, wurde wieder auf eine Listenform übergegangen. Diese wurde in ihrer Form zur ursprünglichen Liste optimiert und erweitert, um die zusätzlichen Angaben darin in strukturierter Weise abbilden zu können.

4.3.2.2.1 Ableitung der Prozessliste aus UML-Diagramm

Die im UML-Diagramm begonnene Kategorisierung von ähnlichen Prozessen mit dem Herauslösen von identischen Parametern in die übergeordnete Kategorie wurde in der neuen Vorlage der Prozessliste konsequent weitergeführt und noch verfeinert. Ebenso wurde die Parameter in einzelne Zeilen geschrieben, damit die Informationsanreicherung in den dahinterliegenden Spalten vorgenommen werden konnte.

²¹ <https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP03/Beispieldaten/Prozessliste.xlsx>

Die Entwicklung der Darstellung und Aufbereitung der Daten zu den Prozessen, soll in der nachfolgenden Abbildung 4-21 am Beispiel des Prozesses „Cut“ gezeigt werden.

| Initiale Prozessliste | | | | | | |
|---|-----------------|--|--|--|--|--|
| Prozess | Spezialisierung | Inputdaten | Outputdaten | | | |
| Cut | Cut Wire | Leitungslänge, -zuschläge, Gehäuseinformationen, Materialnummer (Artikelnummer) | Ist-Leitungslänge (Musterleitung), Chargennummer Leitungsbündel (Artikelnummer) | | | |
| Cut | Tube | Länge, Zusatzbearbeitung Schlitzen (ja/nein), Materialnummer (Artikelnummer), Soll-Losgröße für Bündel | Ist-Länge (Muster), Chargennummer Schlauchmeterware, Chargennummer Produkt, Beziehung Chargennummer Eingangsmaterial zu Ausgangsmaterial (Bsp. Schneidekartennummer 1000 Eingang in 20*50 Ausgang), Ist-Losgröße | | | |
| UML-Diagramm | | | | | | |
| <pre> graph TD Cut --> CutTube Cut --> CutWire CutTube CutWire </pre> | | | | | | |
| Finale Prozessliste | | | | | | |
| Category | Submodel | Submodel Property | Description | | | |
| Cut | Cut | NominalLength | Soll-Länge | | | |
| | | ActualLength | Ist-Länge (Muster) Cut wire Cut tube | | | |
| | CutWire | | | | | |
| | CutTube | | | | | |

Abbildung 4-21 Entwicklung der Darstellung zu den Prozessen

Auf diese Weise wurden alle im UML-Diagramm enthaltenen Prozesse inklusive der Parameter in einem ersten Schritt in die neue Form der Prozessliste übertragen. Mit der Übernahme aus dem UML-Diagramm wurden auch die einzelnen Parameter und deren Zuordnung nochmals diskutiert und ggf. korrigiert. Zudem wurden in diesem Zuge auch fehlende Datentypen und Properties noch ergänzt.

4.3.2.2.2 Ergänzung der Prozessliste

Nach der Übertragung der Prozesse und der dazugehörigen Daten wurden für die einzelnen Prozesse entsprechende Prozessexperten eingeladen. Mit denen wurden die bisher erarbeiteten Ergebnisse diskutiert und bei Bedarf wurden Änderungen und Ergänzungen vorgenommen.

Um den Reifegrad der Prozessliste weiter zu erhöhen, wurden mit den Experten auch weitere Properties zu den Parametern befüllt. Speziell ging es um die Zuordnung, ob es sich um Input- oder Output-Daten handelt. Eine weitere wichtige Eigenschaft war die Bestimmung, ob es sich bei dem entsprechenden Parameter um einen optionalen oder mandatorischen handelt, sowohl auf der Input-Seite für die Prozessausführung als auch beim Output für die Rückmeldewerte. Abschließend wurden auch nochmal der Datentyp und die entsprechende Einheit abgestimmt (Abbildung 4-22).

| Direction | Man/Opt | Submodel Property Type | Unit |
|-----------|-----------|------------------------|------|
| In | Optional | double | mm |
| Out | Mandatory | double | mm |
| In | Mandatory | double | mm |
| In | Optional | double | N |
| In | Optional | double | N |
| Out | Optional | double | N |
| Out | Optional | double | N |

Abbildung 4-22 Erweiterte Properties zu den Prozessparametern

Nach der Definition aller Prozesse und der zugehörigen Parameter wurde bei der Benamung auf eine einheitliche Nomenklatur bei ähnlich zusammenhängenden Bedeutungen geachtet. Um auch die semantische Standardisierung voranzutreiben, wurden die Einheiten der Parameter mit ECLASS abgeglichen und mit verfügbaren IRDIs entsprechend verknüpft (Abbildung 4-23).

| Unit | IRDI |
|-----------------|----------------------|
| mm | 0173-1#05-AAA480#003 |
| mm ² | 0173-1#05-AAA295#003 |
| N | 0173-1#05-AAA561#003 |
| degree celsius | 0173-1#05-AAA567#004 |
| s | 0173-1#05-AAA203#003 |
| mm/s | 0173-1#05-AAA068#003 |
| degree | 0173-1#05-AAA117#003 |
| rad | 0173-1#05-AAA012#003 |
| m/s | 0173-1#05-AAA434#003 |
| Hz | 0173-1#05-AAA351#003 |
| bar | 0173-1#05-AAA044#002 |
| percent | 0173-1#05-AAA129#003 |
| ms | 0173-1#05-AAA114#003 |
| kg | 0173-1#05-AAA731#003 |
| cm ³ | 0173-1#05-AAA100#003 |
| g/s | 0173-1#05-AAA385#003 |
| px | 0173-1#05-AAA028#002 |
| F | 0173-1#05-AAA540#003 |
| m | 0173-1#05-AAA551#003 |
| V | 0173-1#05-AAA153#003 |
| Ohm | 0173-1#05-AAA049#003 |
| mbar | 0173-1#05-AAA466#003 |
| g | 0173-1#05-AAA728#003 |
| ml/s | 0173-1#05-AAA285#004 |
| pt | 0173-1#05-AAB597#002 |
| Ws | 0173-1#05-AAB252#002 |
| Nm | 0173-1#05-AAA212#004 |
| dB | 0173-1#05-AAA715#004 |

Abbildung 4-23 Einheiten mit den dazugehörigen ECLASS IRDIs

Nachdem die Definitionen rundum die Prozesse und Parameter abgeschlossen waren, wurde der Common-Bereich nochmals überarbeitet und die in der Abbildung 4-24 dargestellten Kategorien inklusive der dazugehörigen Parameter definiert.

| Category | Submodel |
|----------|----------------------------|
| Common | ProductionOrder |
| | Batch |
| | Lot |
| | Process |
| | Process/ProductParameters |
| | Process/ProcessParameters |
| | Process/ResourceParameters |
| | Process/ProcessBom |

Abbildung 4-24 Common-Bereich der finalen Prozessliste

Der Common- Bereich unterteilt sich in die Kategorien, denen das Präfix „Process“ vorangestellt ist, d.h. Informationen zu einem Prozess. Demgegenüber beziehen sich die Kategorien „ProductionOrder“, „Batch“ und „Lot“ auf ein Produkt, was ein End- oder Halbfabrikat sein kann. Diese Kategorien können einen aber auch mehrere Prozesse beinhalten. Eine ausführlichere Beschreibung dazu gibt es im folgenden Kapitel 0.

Die Begriffe werden oft unterschiedlich verwendet²², in unserem Kontext jedoch wird unter einem **Lot** (dt. Los) eine kleinere, oft gleichartige Produktionsmenge verstanden, die unter gleichen Bedingungen gefertigt wurde. Ein **Batch** hingegen beschreibt eine größere, organisatorisch definierte Menge, die oft mehrere Lots umfassen kann. Das Lot ist immer identifizierbar, hingegen das Batch eine logische Einheit.

Um den Arbeitsaufwand rundum die Prozessliste zu verdeutlichen, ist in der nachfolgenden Tabelle die Entwicklung der Umfänge von der initialen zur finalen Liste dargestellt. Bei der finalen Prozessliste sind die Testing- und Common-Prozesse mitberücksichtigt.

| | Initiale Prozessliste | Finale Prozessliste |
|---------------------|-----------------------|---------------------|
| Input-Daten | 76 | 405 |
| Output-Daten | 73 | 138 |
| Insgesamt | 149 | 552 |

Tabelle 4-1 Vergleich zwischen initialer und finaler Prozessliste

4.3.2.3 Basis für OPC UA Companion Specification

Dieser Abschnitt beschreibt die Überführung von Teilbereichen der Prozessliste in die OPC UA Companion Specification "Wire Harness Manufacturing" [18].

In TP 1 wurde im Laufe der Zeit erkannt, dass eine Verallgemeinerung und Standardisierung der Maschinenschnittstellen von entscheidender Bedeutung sind. Daher wurde mit Unterstützung des VDMA (Verband Deutscher Maschinen- und Anlagenbauer) die OPC UA Arbeitsgruppe 'Wire Harness Manufacturing' ins Leben gerufen. Durch diese Initiative können Maschinen und Anlagen standardisiert ihren "Datenbeitrag" zur VWS4LS leisten.

Im ersten Schritt wurden die relevanten Prozesse für die Version 1.0 diskutiert und festgelegt. Dies war notwendig, da der bestehende Detailgrad der Prozessliste zu umfangreich für eine erste Version war. In der Version 1.0 der Companion Specification werden daher folgende Prozesse zu finden sein:

- Cut
- Strip
- Slit
- Crimp
- Seal
- Sleeve

Nach der Festlegung der Prozesse war außerdem eine erneute "Synchronisation" zwischen der OPC UA-Arbeitsgruppe und TP3 erforderlich, da aus Sicht der Maschinen der ein oder andere Prozess mit zusätzlichen Daten angereichert werden musste. Ebenso erfolgte eine Aufgliederung der Prozesse auf Maschinenseite in die Bereiche "Spezifikation", "Validierung" und "Überwachung".

Wie auch in der Prozessliste dargestellt, sind für den Input und Output auf der OPC UA-Seite unterschiedlichste Datentypen erforderlich. Diese werden als Strukturen repräsentiert, die sowohl in Methodenaufrufen (wie „Store“, „Start“ usw.) als auch für „ResultTransfer“ verwendet werden können. Der Aufbau dieser Datenstrukturen orientiert sich selbstverständlich an der Prozessliste und spiegelt diese wider um eine effektive Integration zwischen OPC UA "Wire Harness Manufacturing" [18] und VWS4LS zu gewährleisten.

²² <https://de.wikipedia.org/wiki/Losfertigung>

4.3.3 Prozess VWS

In diesem Kapitel wird die Umsetzung der Prozess VWS (siehe Datei „[AAS_Prozesse_2024_01_26_V3 - Demonstrator.aasx](#)“²³) im Kontext des PPR-Modells beschrieben. Die im vorherigen Kapitel vorgestellte Prozessliste bildete dabei die zentrale Grundlage zur Strukturierung der Prozess VWS mit ihren einzelnen Bestandteilen.

Im VWS-Kontext wird zwischen Typ und Instanz unterschieden. Für den Leitungssatz lässt sich diese Unterscheidung auf die Phasen bzw. Prozesse (Entwicklung, Produktion) abbilden. Auf Typenebene wird im Engineering zunächst ein sog. „150%-Leitungssatz“ definiert, der alle Funktionsmodule des Leitungssatzes beschreibt, die in einem Fahrzeugprojekt aus rein funktionaler Sicht existieren können. Das heisst, der 150%-Leitungssatz beinhaltet alle möglichen Varianten, inkl. sich gegenseitig ausschließende Funktionsmodule und Konstellationen, die somit nie produziert werden. Aus diesem Grund wird für einen konkret zu produzierenden Leitungssatz von einem sog. „100%-Leitungssatz“ gesprochen. Dieser wird aus dem 150%-Leitungssatz abgeleitet und beinhaltet die notwendigen Funktionsmodule für ein konkretes Fahrzeug.

Grundsätzlich wurde bei der Strukturierung der VWS für die Produktion eines Leitungssatzes in Typ und Instanz unterschieden. Die VWS „*ProductType*“ beschreibt dabei das zu fertigende Produkt bzw. Halbfabrikat auf Typebene. Die Instanzebene beginnt, sobald ein Produktionsauftrag (VWS *ProductionOrder*) für das Produkt bzw. Halbfabrikat existiert. Dieses Produkt bzw. Halbfabrikat kann in der Produktion in mehreren Batches (VWS *Batch01*) und mehreren Lots (VWS *Lot01*) gefertigt werden. Nachfolgend werden die Struktur und Inhalte der einzelnen VWS genauer beschrieben.

4.3.3.1 ProductType

Die VWS *ProductType* ist eine VWS auf Typebene und beinhaltet alle Informationen, um das Produkt bzw. Halbfabrikat zu fertigen (Abbildung 4-25).

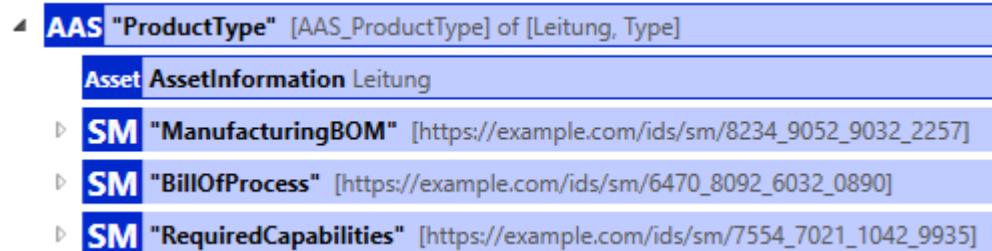


Abbildung 4-25: VWS *ProductType*

Die VWS *ProductType* beinhaltet die Submodelle „*ManufacturingBOM*“, „*BillOfProcess*“ und „*Required Capabilities*“. In der *ManufacturingBOM* werden alle notwendigen Materialien als Entities gelistet. Analog werden im Submodell *BillOfProcess* alle notwendigen Prozesse in einer SMC *ProcessSequence* aufgeführt.

Die einzelnen Prozesse sind wiederum in eigenen SMCs strukturiert und besitzen eine Namenskonvention, bei der vor jedem Prozess das Präfix „Nominal“ vorangestellt wird (z.B. *NominalCutProcess01*).

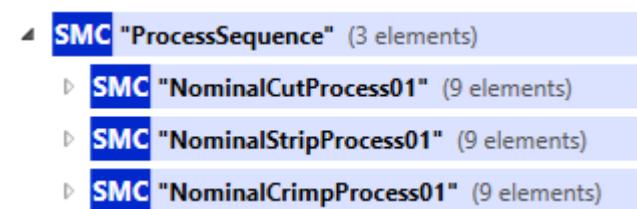


Abbildung 4-26: SMC *Product Sequence*

Die Nominalprozesse bzw. Sollprozesse beinhalten eine einheitliche Struktur mit den folgenden Properties (Abbildung 4-27):

²³https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP03/Beispieldaten/AAS_Prozesse_2024_01_26_V3 - Demonstrator.aasx²³https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP03/Beispieldaten/AAS_Prozesse_2024_01_26_V3 - Demonstrator.aasx

- *ProcessId*: Jeder Prozess soll eine eindeutige ID besitzen
- *ProcessName*: Der Name des Prozesses
- *OperatorRole*: Die Rolle des Maschinenbedieners (bspw. Worker, Wartungsmitarbeiter)
- *ProcessDescription*: Prozessbeschreibung
- *PlannedProcessTime*: Die Planzeit des Prozesses

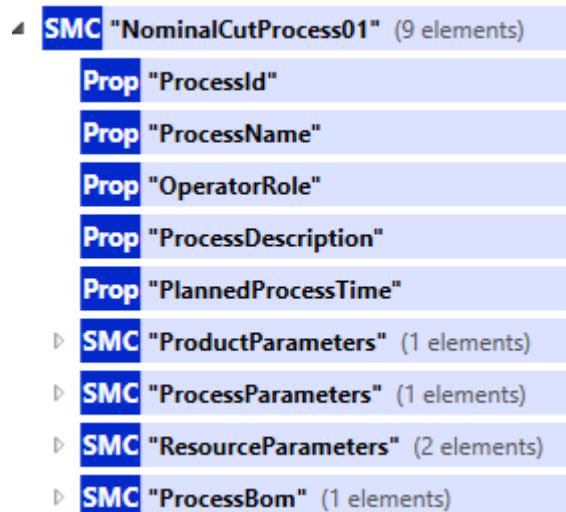


Abbildung 4-27: Nominalprozess Cut

Des Weiteren sind folgende SMC enthalten (Abbildung 4-28):

- *ProductParameters*: Listet alle Einzelmaterialien mit ihren Produkteigenschaften auf
- *ProcessParameters*: Beinhaltet die Prozessparameter (z.B. Schneiden auf Solllänge (*NominalLength*)
- *ResourceParameters*: Beinhaltet die Referenzen auf die eingesetzte Maschine und gerüstete Werkzeuge
- *ProcessBom*: Beinhaltet Referenzen auf alle zu verarbeitenden Materialien für den Prozess auf Planungsebene für den Materialverbrauch

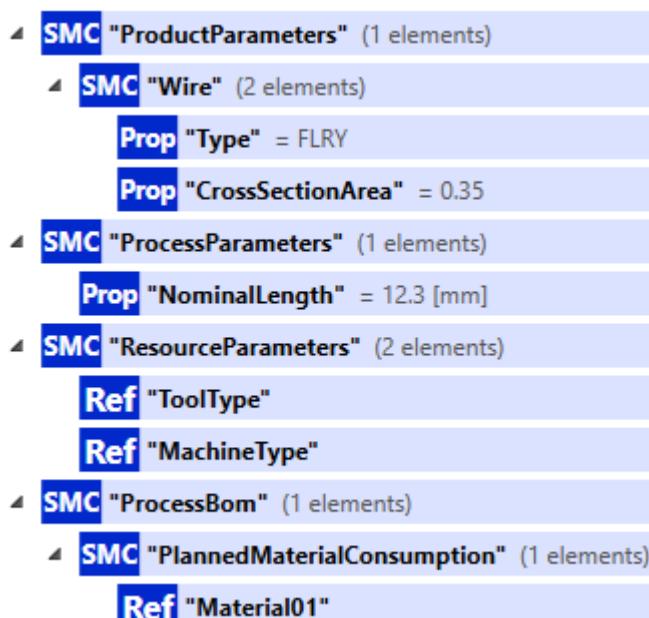


Abbildung 4-28: SMC PPR, ProcessBom

Anschließend wurde innerhalb der VWS *ProductType* das Submodell „RequiredCapabilities“ definiert. Die Struktur folgt der offiziellen Spezifikation und unterteilt die *RequiredCapabilities* in eine SMC *CapabilitySet* und dieses wird wiederum in einem oder mehreren *CapabilityContainern* untergliedert (Abbildung 4-29). Jeder einzelne *CapabilityContainer* beinhaltet eine Capability „Cap“ mit der entsprechenden Bezeichnung und einer SMC *PropertySet*.

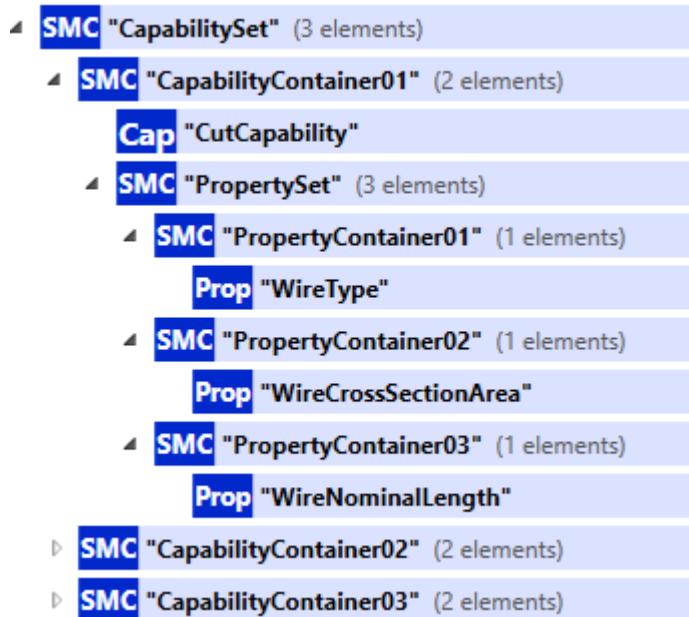


Abbildung 4-29: Required Capabilities

4.3.3.2 ProductionOrder

Die Instanzenebene beginnt mit der Erstellung eines konkreten Produktionsauftrages. Hierzu wurde die VWS *ProductionOrder* („[AP3.5_AAS_Prozesse_2024_04_12_V3.aasx](#)“²⁴) definiert (siehe Abbildung 4-30).

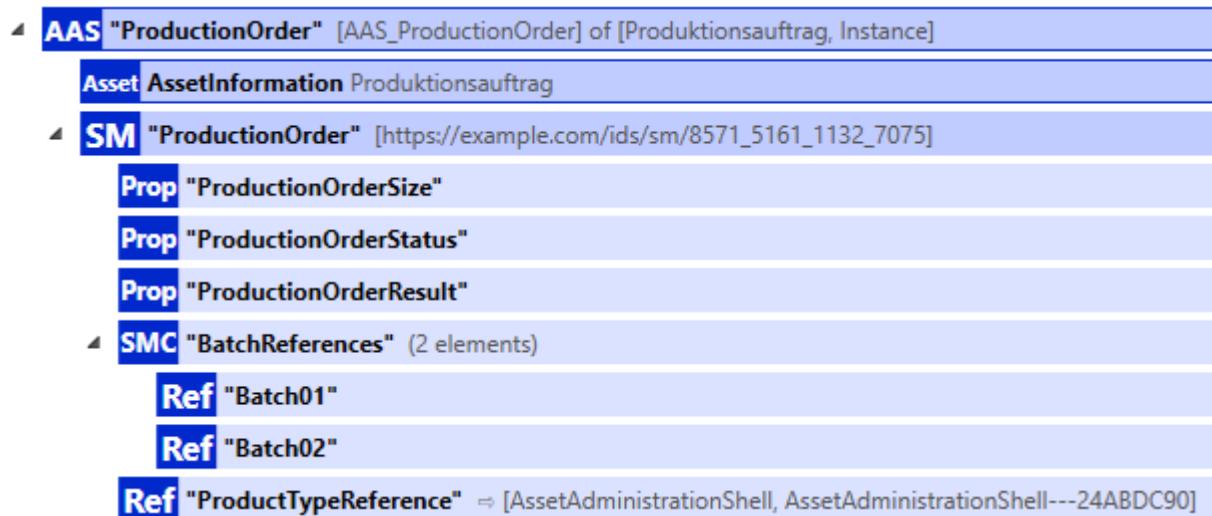


Abbildung 4-30: VWS ProductionOrder

Die VWS *ProductionOrder* enthält ein Submodell „*ProductionOrder*“ mit den folgenden Inhalten:

- *ProductionOrderSize*: Die zu produzierende Stückzahl für den Produktionsauftrag

²⁴

https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP03/Beispieldaten/AP3.5_AAS_Prozesse_2024_04_12_V3.aasx

- *ProductionOrderStatus*: Der Status des Produktionsauftrags, bspw. „erstellt“, „freigegeben“, „gestartet“, „abgeschlossen“ oder „Fehlermeldung“
- *ProductionOrderResult*: Das Ergebnis des Produktionsauftrages, bspw. „OK“ oder „NOK“ (Not OK)
- *SMC BatchReferences*: Verlinkt die einzelnen Batches des Produktionsauftrages mit der VWS *Batch* (im Beispiel „Batch01“ und „Batch02“)
- *ProductTypeReference*: Referenz auf den zu fertigenden Produkttyp

4.3.3.3 IDTA Standardisierung BoP Submodel

Die in diesem Dokument entworfenen Prozess-VWS diente als Grundlage für die Standardisierung des Submodells „IDTA 02031: Bill of Process“ [26].

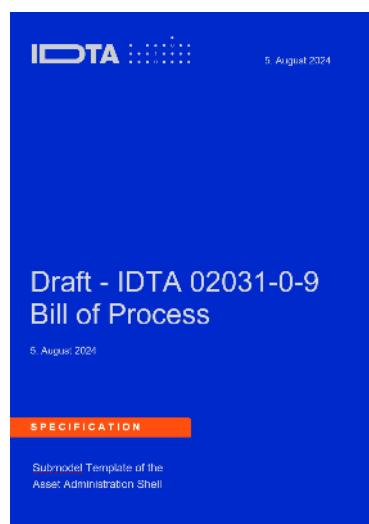


Abbildung 4-31: IDTA 2031 Bill of Process

4.3.4 Capabilities aus Prozesssicht

Als typisches Muster in der konventionellen Informationsmodellierung sind nach dem PPR-Paradigma die Prozess- und Ressourcendimensionen eng miteinander verbunden. Für jeden Vorgang in einem Produktionsprozess weist die klassische Fertigungsplanung die für die Durchführung des Vorgangs am besten geeignete Maschine als Ressource zu. Daher sind Produktdesign, Fertigungsplanung und -ausführung in der konventionellen Planung oft eng miteinander gekoppelt. Um ein Szenario der flexiblen Fertigung nach den Ideen von Industrie 4.0 zu realisieren, muss diese enge Kopplung in der Planungsphase aufgelöst werden, damit die Zuordnung von Prozessen zu Ressourcen flexibel entschieden und automatisiert werden kann.

Um die bislang statische Verknüpfung zwischen dem Prozess und den zugehörigen Ressourcen aufzubrechen, benötigt man daher ein zusätzliches Element, das zwischen den beiden vermittelt. Ein solches zusätzliches Element auf Basis des PPR-Modells wird als Fähigkeit (Capability) oder Fertigkeit (Skill) bezeichnet. Das daraus resultierende konzeptionelle Modell ist eine Erweiterung des etablierten PPR-Darstellungsparadigmas um die Konzepte "Capability", "Skill" und "Service", welche die zugrunde liegende produktionsrelevante Funktion aus einer anderen Perspektive erfassen. Daher wird das resultierende Modell das als Capability-Skill-Service-Modell oder kurz CSS-Modell bezeichnet, welches die grundlegenden formalen Elemente für die Realisierung von flexiblen Produktionsszenarien liefert [35].

Das CSS-Modell umfasst vier Bereiche, die in Abbildung 4-32 farblich hervorgehoben wurden. Gelb kennzeichnet die einzelnen Elemente des PPR-Modells. Orange bezeichnet Funktionen, die sich in Produktionsprozessschritten widerspiegeln, die einerseits für die Herstellung von Produkten erforderlich sind und andererseits an Produktionsressourcen ausgeführt werden. Erforderliche und bereitgestellte Fähigkeiten (Required und Offered Capabilities) müssen abgeglichen werden, um eine geeignete Abfolge

von Produktionsschritten für gegebene Anforderungen zu finden. Dies kann zunächst auf einer deskriptiven Ebene geschehen - z.B. durch den Vergleich von Fähigkeitstypen und deren Eigenschaften - unabhängig davon, welche tatsächlichen Ressourcen diese Prozessschritte später ausführen.

Um die durch eine Fähigkeit beschriebene Funktion in einem bestimmten Produktionsprozessschritt anzuwenden, wird ein so genannter "Skill" (Fähigkeit) aufgerufen, d.h. eine Implementierung dieser Funktion, die von einer bestimmten Ressource bereitgestellt wird (blau in Abbildung 4-32). Ein Skill wird über eine Schnittstelle aktiviert und gesteuert, die von der Skill-Implementierung bereitgestellt wird, um externen Einheiten den Zugriff auf einen Skill zu ermöglichen, ohne interne Implementierungsdetails preiszugeben. Skills werden Ressourceninstanzen zugewiesen.

Zusätzlich zu den funktionalen Aspekten, die durch die Fähigkeiten beschrieben werden, müssen auch weitere organisatorische und kommerzielle Aspekte, wie z.B. Zeitplanung, Qualität oder Kosten, berücksichtigt werden. Zu diesem Zweck werden im Modell erforderliche und angebotene Services (grün in Abbildung 4-32) definiert [36]. Services (Dienstleistungen) beschreiben das Angebot von Fähigkeiten in einem breiteren Rahmen größerer Lieferkettennetzwerke, die über die lokale Produktionseinrichtung hinausgehen. Daher sind Services in der Regel nicht Teil einer Ressource, sondern von übergeordneten Softwarekomponenten wie Enterprise Resource Planning (ERP)-Systemen.

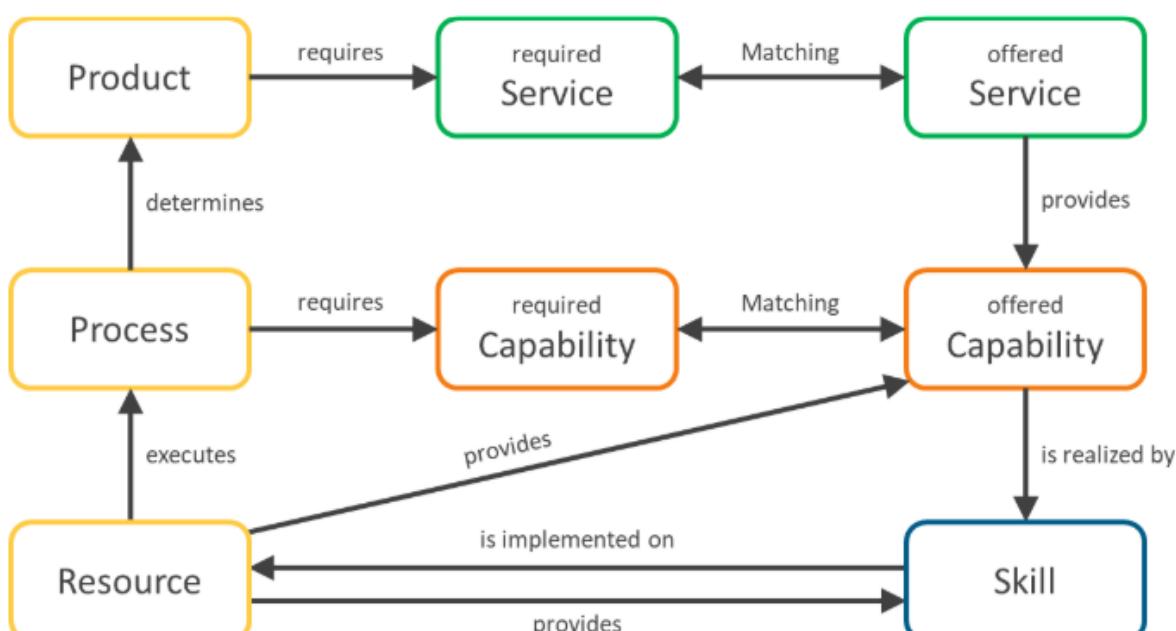


Abbildung 4-32: CSS-Modell [36]

4.3.4.1 Definition der Capabilities

Im Teilprojekt 3 liegt der Fokus auf den Produktionsprozessen eines Leitungssatzes, weshalb der Bereich „Process“ und die dazugehörigen „Required Capabilities“ anhand des CSS-Modells näher analysiert wurden. Hierzu wurden an einem typischen Beispielprodukt (Abbildung 4-33) die „Required Capabilities“ abgeleitet, um das Matching mit den „Offered Capabilities“ durchführen zu können.

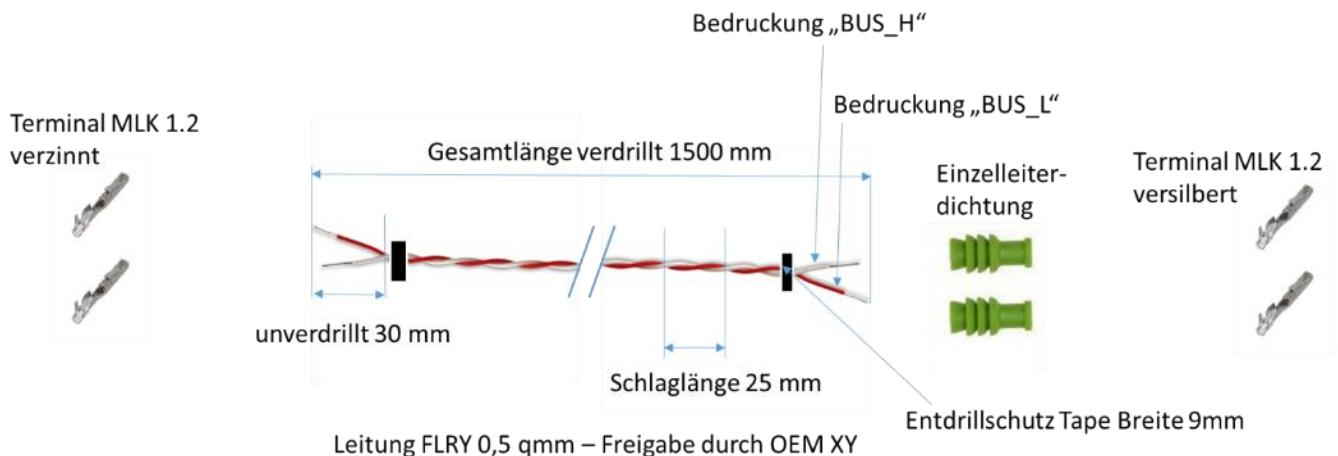


Abbildung 4-33: Beispielprodukt zur Ableitung der Required Capabilities

Das Beispielprodukt besteht aus einem verdrillten Leitungspaar mit jeweils einem Entdrillschutz auf jeder Seite. An beiden Enden werden Terminals gecrimpt und eine Einzelleiterdichtung angebracht. Für das Beispielprodukt wurden folgende Prozesse identifiziert, um eine Herstellung durchführen zu können:

- Cut
- Strip
- Crimp
- Twist
- Montage Einzelleitungsabdichtung (ELA)
- Marking

In den nachfolgenden Unterkapiteln werden zu den einzelnen Prozessen die benötigten Parameter aus Fähigkeitenperspektive (Required Capabilities) vorgestellt.

4.3.4.1.1 Cut

Für den Prozess „Cut“ werden folgende Parameter benötigt:

- Leitungslänge
- Leitungsquerschnitt
- Isolationsdurchmesser
- Leitermaterial (Keine Mischung der Materialien (z.B. Alu, Kupfer) bei Kabelschneid-Maschinen)

4.3.4.1.2 Strip

Für den Prozess „Strip“ werden folgende Parameter benötigt:

- Isolationswandstärke
- Abisolierungslänge
- Startposition für Abisolierung
- Leitungsaußendurchmesser
- Leitungsquerschnitt
- Überwachung für Berührung des Messers beim Abisolieren (Verfügbarkeit der Überwachungseinrichtung)

4.3.4.1.3 Crimp

Für den Prozess „Crimp“ werden folgende Parameter benötigt:

- Crimpkraftüberwachung als Qualitätsanforderung
- Leitungsquerschnitt
- Leitungstyp
- Terminaltyp (Freigegebene Terminaltypen sind in der Ressource (Werkzeug) hinterlegt)
- Crimpbreite
- Crimphöhe
- Abfrage: Werkzeug passt in Maschine

4.3.4.1.4 Twist

Benötigte Parameter:

- Leitungstyp (gibt den Außendurchmesser vor)
- Min und Max für Leitungslänge (Annahme: Verdrillergebnis wird vorgegeben und Einstellparameter werden von der Maschine berechnet)
- Min und Max für Leitungsquerschnitt
- Max. Anzahl der verdrillten Leitungen
- Schlaglänge + Toleranz
- Min und Max Länge für unverdrilltes offenes Ende (Komax: Offene Kabelenden Verdrillseite max./min.)
--> 2x vorhanden für jedes Ende
- Unterschiedliche offene Enden max.
- Endlänge verdrillter Leitungen min./max.
- Vorhandensein der Funktion "Entdrillschutz anbringen" + Position zum Anbringen

4.3.4.1.5 Montage Einzelleitungsabdichtung (ELA)

Für den Prozess „Montage Einzelleitungsabdichtung“ werden folgende Parameter benötigt:

- Leitungstyp (gibt den Außendurchmesser vor)
- Min Leitungslänge
- Min und Max Leitungsquerschnitt
- Passendes Werkzeug für ELA vorhanden
- Abfrage: Werkzeug passt in Maschine

4.3.4.1.6 Marking

Für den Prozess „Marking“ werden folgende Parameter benötigt:

- Leitungstyp (nicht auf jedes Material kann man gleich drucken (Bsp. Silikon))
- Min Leitungslänge
- Min und Max Leitungsquerschnitt
- Markingverfahren (Bsp.: Tintenstrahl, Stempel, Hot Stamp)
- Position der Bedruckung (erstmögliche Bedruckung, z.B. 50mm nach Leitungsende)
- Druckvarianten (Bsp.: Alphanumerisch + Barcode)
- Druckfarbe (nur schwarz oder mehrfarbig / mindestens hell und dunkel)
- Min und Max Schriftgröße (abhängig von Leitungsaußendurchmesser)
- Wiederholungen (1x/2x/n-mal)

4.3.4.2 Ableitung der Capabilities für Produktionsprozesse

Die notwendigen *Required Capabilities* wurden identifiziert, indem die Prozessliste mit allen Prozessen und den einzelnen Submodel Properties analysiert wurde, um zu identifizieren welche Submodel Properties notwendig für ein Matching mit den *Offered Capabilities* sind.

Die gesamte Liste der Prozesse mit den dazugehörigen Submodel Properties für die *Required Capabilities* ist im Tabellenblatt „*Capabilities*“ der [Prozessliste](#) zu finden.

4.3.5 Prozesssteuerung

Ein Job besteht aus den notwendigen Arbeitsschritten bzw. Prozessen, um einen diskreten Fertigungsschritt abzuschließen. Jeder Prozess verwendet ein Produkt und mindestens eine Ressource, die für die Ausführung des Arbeitsschritts erforderlich sind. Ziel der Prozesssteuerung ist es, die Abfolge der Einzelprozesse eines Jobs zu bestimmen. Abbildung 4-34 zeigt exemplarisch die Arbeitsschritte, die in einem fiktiven Job ausgeführt werden. Dabei werden die einzelnen Prozesse in der Regel sequenziell nacheinander abgearbeitet. Wenn die Umstände es zulassen, können einzelne Arbeitsschritte oder Teilsequenzen, bestehend aus mindestens zwei Prozessen, parallelisiert werden. Auch sind optionale Prozesse möglich, also Prozesse, die nur unter bestimmten Voraussetzungen in dem aktuellen Job ausgeführt werden. Darüber hinaus können einzelne Prozesse eines Jobs auch effektiv unabhängig voneinander sein. Beispielsweise könnte ein Prozess die Dauer eines Jobs messen und diese nach Abschluss dokumentieren.

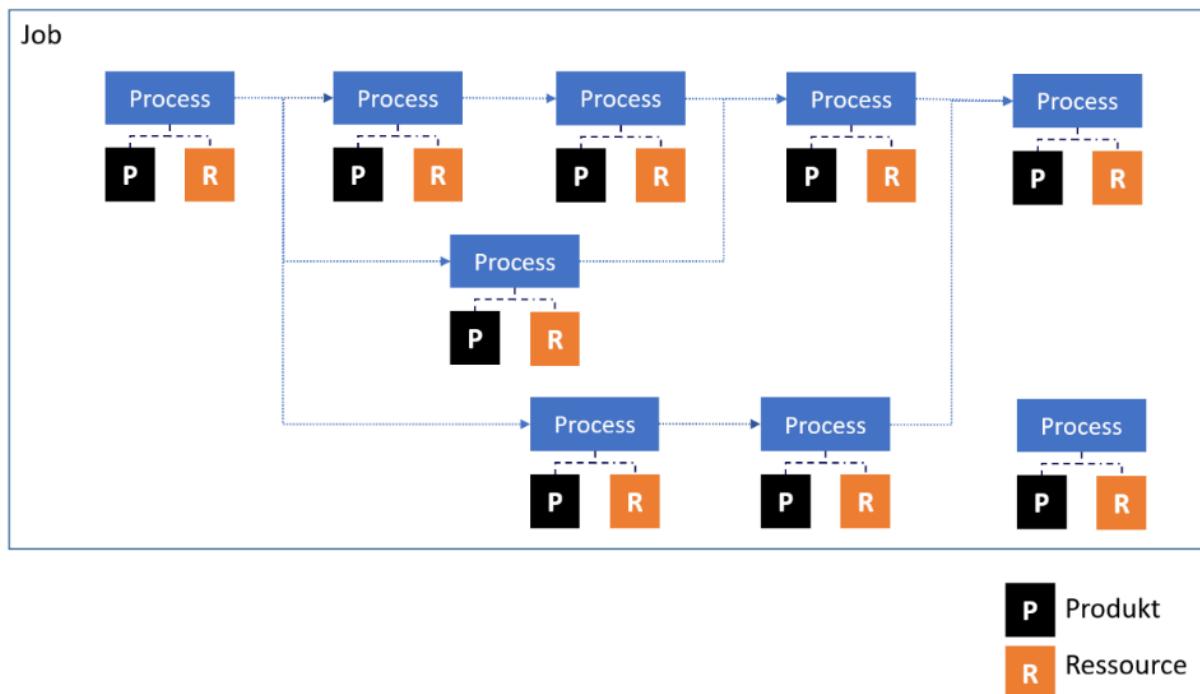


Abbildung 4-34: Prozessequenz

4.3.5.1 Problemstellung

In modernen Produktionsprozessen ergeben sich zahlreiche Herausforderungen, die eine effiziente und flexible Prozesssteuerung erfordern. Eine der zentralen Schwierigkeiten liegt in der **Parallelisierung von Arbeitsschritten**, die eine gleichzeitige Durchführung mehrerer Tätigkeiten ermöglicht, um die Produktionszeit zu verkürzen und die Effizienz zu steigern. Dabei muss eine dynamische Anpassbarkeit gewährleistet sein, was bedeutet, dass flexibel entschieden werden kann, welcher Schritt zu welchem Zeitpunkt durchgeführt wird. Diese Anpassungsfähigkeit ist essenziell, um auf Veränderungen in der Produktion etwa durch Auftragsanpassungen oder Engpässe bei Material oder Personal schnell reagieren zu können.

Eine weitere Herausforderung besteht im Tracking und der Darstellung der Produktionsabläufe. Es ist notwendig, den tatsächlichen Ablauf mit der digitalen Repräsentation sowie möglichen Visualisierungen zu synchronisieren, um eine genaue und aktuelle Übersicht zu gewährleisten. Ebenso sollen Flaschen-

hälse in der Produktion durch die Beobachtung des Prozesses identifizierbar gemacht werden, um Engpässe frühzeitig zu erkennen und zu beheben. Hierbei spielt die Dokumentation des Prozesses in digitaler Form eine wichtige Rolle. Die Prozesse müssen maschinenlesbar und ausführbar dokumentiert werden, wozu etwa Daten aus dem Teilmodell „IDTA 02031-1-0 Bill of Process“ [26] genutzt werden, aber auch eine Grundlage für eine Ergebnis-Dokumentation ermöglichen.

Die Konfiguration der Prozesssteuerung stellt ebenfalls eine komplexe Aufgabe dar. Es wird angestrebt, Prozessexperten durch Automatisierung zu ersetzen, um weniger Grundlagenwissen zu den jeweiligen Fertigungsprozessen zu benötigen und gleichzeitig eine anwenderfreundliche Nutzung gewährleisten zu können.

Die Bedeutung einer effizienten Prozesssteuerung ist nicht zu unterschätzen. Sie ermöglicht ein umfassendes Monitoring, Tracking und die Darstellung der Produktionsprozesse. Durch die Vereinfachung der Komplexität wird es möglich, schnell einzutreten und Anpassungen vorzunehmen, was besonders in dynamischen Umgebungen von Vorteil ist. Die dynamische Konfigurierbarkeit spielt insbesondere bei der Produktion in Losgröße 1 eine wichtige Rolle, da hier individuelle Anpassungen notwendig sind. Zudem wird in Zusammenhang mit den in TP6 erarbeiteten Verhandlungsprozessen auch in der Prozesssteuerung die Verbindung von Aufgaben, Produkten und Ressourcen erzeugt, was die Effizienz und Transparenz weiter erhöht.

Eine Integration der Prozesssteuerung in den digitalen Zwilling der Produktion bringt zahlreiche Vorteile mit sich. Der digitale Zwilling fungiert als Single Source of Truth, wodurch alle relevanten Daten und Informationen an einer zentralen Stelle gebündelt werden. Dies ermöglicht einerseits die Automatisierung durch Ausführungsschritte als Teil des digitalen Zwillinges sowie andererseits auch schnelle Anpassungen der Informationen, was die Effizienz und Reaktionsfähigkeit weiter steigert. Darüber hinaus bietet ein digitaler Zwilling eine hohe Konfigurierbarkeit, wodurch Prozesse flexibel an neue Anforderungen angepasst werden können. Dazu werden in diesem Teilprojekt bestehende Konzepte und Lösungen verwendet, um eine Kompatibilität nach außen gewährleisten zu können.

Zusammenfassend lässt sich sagen, dass die Herausforderungen im Produktionsprozess durch eine effiziente und flexible Prozesssteuerung gemeistert werden können. Die Integration dieser Steuerung in den digitalen Zwilling bietet dabei zusätzliche Vorteile, die zur Optimierung und Zukunftsfähigkeit der Produktion beitragen.

4.3.5.2 Digitale Prozessabbildung

Zunächst wurde ein digitales Prozessabbild am Beispiel einer Sicherungsbox (siehe Abbildung 4-35) erstellt. Die Sicherungsbox stellt ein überschaubar komplexes Beispiel dar, da verschiedene Prozessschritte parallel ablaufen müssen, während gleichzeitig Schritte grundsätzlich gleich sein könnten. So müssen für jede Sicherung grundsätzlich zwei Verbindungen durch Schrauben befestigt werden. Dieser Prozessschritt ist für alle Positionen gleich, erfolgt aber nicht in einer vordefinierten Reihenfolge, so dass theoretisch Position 3 vor Position 1 verschraubt werden könnte. Ebenso sind im Gesamtprozess parallele Prozesse eingearbeitet. So wird ein elektrischer Test mit jedem Schraubvorgang ausgeführt, um potenzielle Probleme frühzeitig zu erkennen, während der Gesamtprozess einen Überwachungsschritt hat, um diese Informationen zu sammeln und ggfs. zu verwenden.

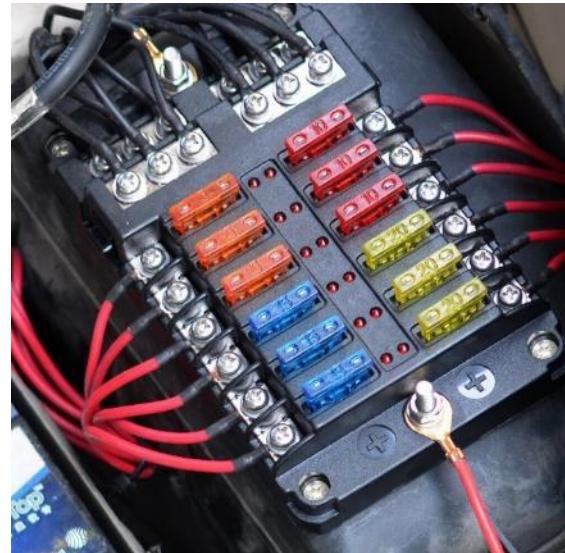


Abbildung 4-35: Beispiel der Sicherungsbox

Abbildung 4-37 zeigt die einfache Übersicht zum Prozess auf der linken Seite, sowie die Erweiterung um einen Scan-Vorgang auf der rechten, welcher als Sicherheitsschritt eingebaut wird. Als Endausbau des Prozessbeispiels zeigt Abbildung 4-37 die gesamte Prozessequenz inkl. der parallel auszuführenden Prozesse. Ebenso sind die Aufgaben für zwei Positionen, bestehend aus einem Scan- sowie Schraub-Vorgang und dem parallel ausgeführten elektrischen Test, sequenziell dargestellt, um die rekursiven

Aspekte des Prozessschrittes abzubilden. Diese drei Einzelschritte werden für jegliche Positionen ausgeführt, während der Überwachungsprozess aktiv sein muss, und können in zufälliger Reihenfolge existieren.

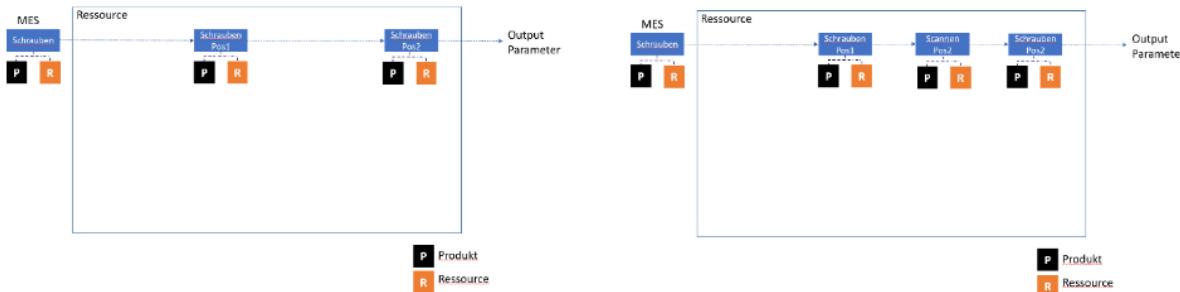


Abbildung 4-36: Simple Prozesssequenz zum Schraubvorgang an zwei Positionen

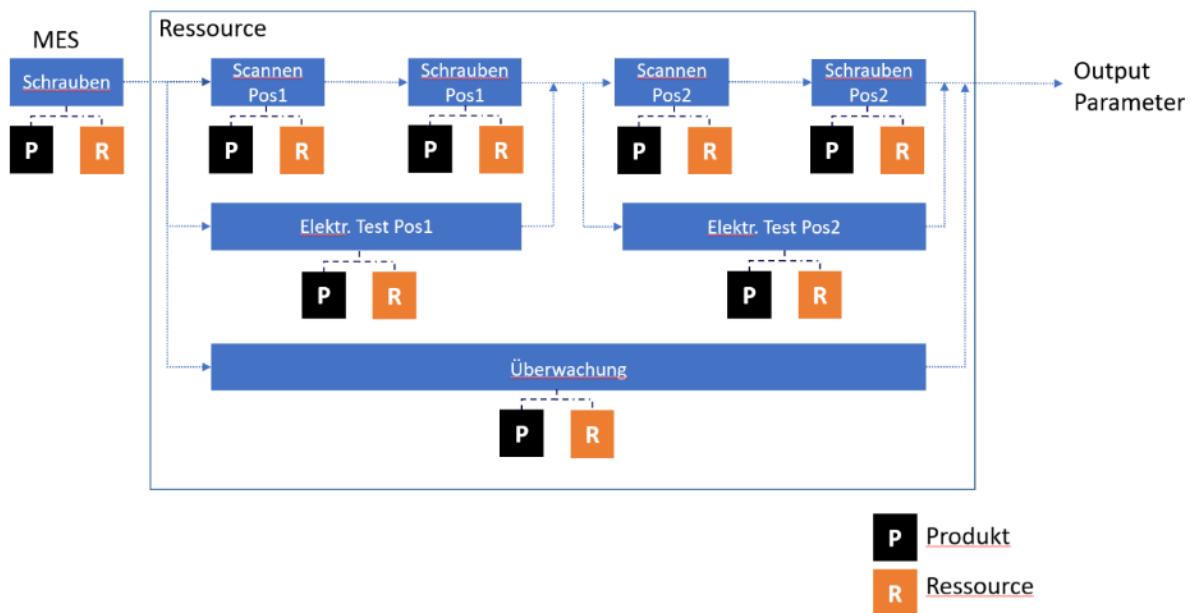


Abbildung 4-37: Prozessequenz am Beispiel Sicherungsbox

Die „Business Process Model and Notation“ (BPMN) ist in ISO/IEC 19510:2013 [37] standardisiert und erlaubt die grafische Modellierung komplexer Prozesse. Der Standard wird von der gemeinnützigen [Object Management Group \(OMG\)](#) verwaltet und international in vielen Organisationen eingesetzt, da die visuelle Natur von BPMN ein besseres Verständnis komplexer Abläufe ermöglicht.

Aus diesen Überlegungen wurde für den vorliegenden Fall zunächst manuell ein BPMN-Dokument erstellt (siehe Abbildung 4-38), um den Gesamtprozess über eine automatisierte Steuerung mittels der BPMN Engine und Prozess-Orchestration CAMUNDA²⁵ auszuführen. In der Abbildung sind die automatisierten Prozessschritte zur Überwachung, dem elektrischen Test und den Scan- sowie Schrauben-Vorgängen, besonders gekennzeichnet. Ebenso sind manuelle Schritte hinzugekommen, um zunächst die Position festzulegen und anschließend zu prüfen, ob weitere Schraubvorgänge notwendig sind. Diese Schritte sind ebenfalls anhand des Auftrags automatisierbar.

In die Struktur des BPMN-Protokolls wurden Konnektoren eingefügt, um Prozesse parallel starten und ablaufen zu lassen, sowie Entweder-Oder-Fälle abhandeln zu können. Diese Konnektoren werden entsprechend vom ausführenden Softwaresystem erkannt und ggfs. Entscheidungen aus den existierenden Daten der Verwaltungsschalen der jeweiligen Geräten, Produkten und Prozessen abgeleitet.

Die BPMN-Struktur bringt auch Vorteile in repetitiven Aufgaben, da Prozessschritte einfach wiederholt werden können und auch Rekursive Prozesse abgebildet werden können. Dies ist im Beispiel in Abbildung 4-38 durch die Abfrage „Schraubprozess fertig?“ gegeben, wo nach einer entsprechenden Abfrage

²⁵ <https://camunda.com/>

der Daten der gleiche Schraubprozess für weitere Positionen durchlaufen werden kann, ohne den gesamten Prozess als lineare Reihenfolge durchzuplanen.

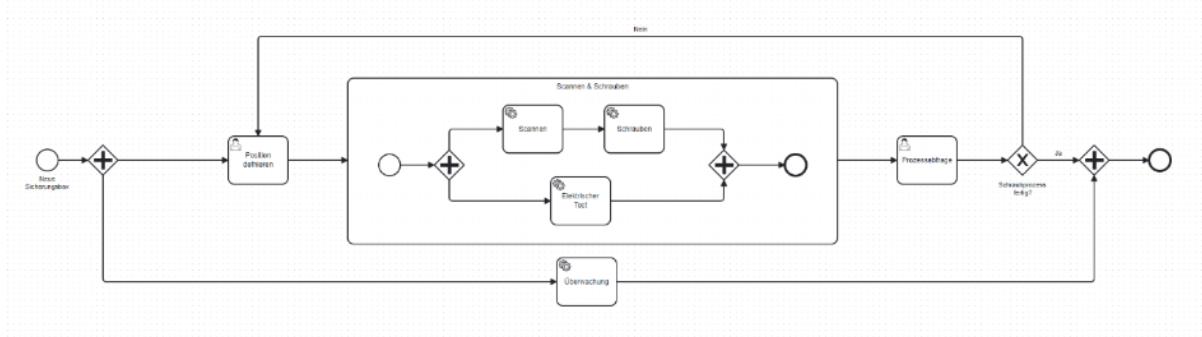


Abbildung 4-38: Darstellung des Prozessbeispiels als BPMN

4.3.5.3 Prozessausführung als Teil der Produkt-Verwaltungsschale

Im Projekt wurde zur automatisierbaren Ausführung des Prozesses dieser als Teil der Produkt-Verwaltungsschale grundsätzlich integriert. Dies beinhaltet sowohl das BOP-Teilmodell als auch ein Konzept zur Erweiterung der Komponenten der Open-Source Middleware Eclipse BaSyx²⁶ um eine Komponente zur automatischen Generierung von BPMN-Strukturen aus dem Teilmodell.

In Verbindung mit den in TP6 entwickelten Verhandlungsprozessen werden zu jedem auszuführenden Schritt des Prozesses entsprechende Ressourcen, also Maschinen und Materialien zugewiesen. Somit werden Prozessschritte und digitale Zwillinge der ausführenden Ressource dynamisch verknüpft, so dass zwar eine Vorbelegung stattfindet, diese jedoch vollständig automatisierbar ist.

Abbildung 4-39 zeigt diese Verknüpfung zwischen den Aufgaben und jeweiligen Ressourcen anhand der vorher dargestellten Prozesse schematisch. So wird in der Prozessequenz zunächst Position 1 von Ressource 1 gescannt und verschraubt während Ressource 3 den elektrischen Test und Ressource 5 die generelle Überwachung parallel ausführen. Anschließend werden Ressourcen 1 und 3 durch Ressourcen 2 und 4 ersetzt, um den gleichen Arbeitsschritt für Position 2 auszuführen.

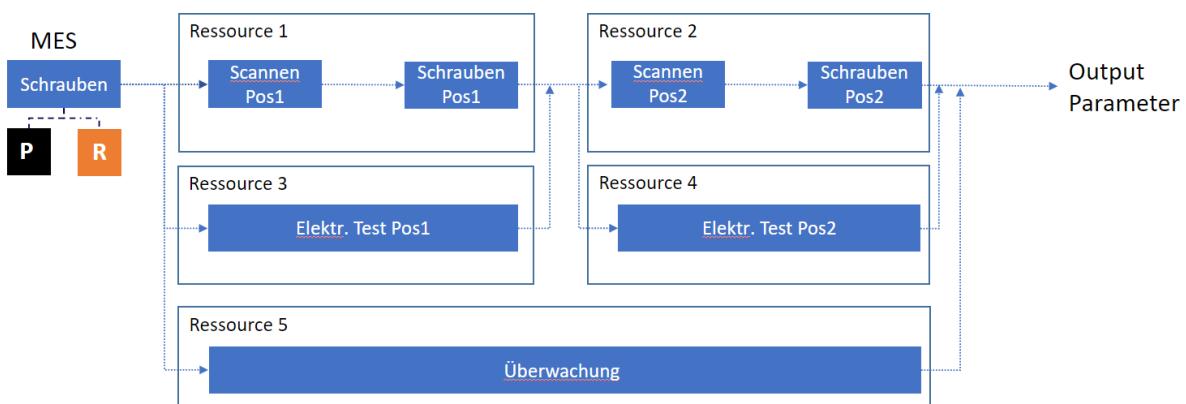


Abbildung 4-39: Prozessequenz am Beispiel Sicherungsbox inkl. Ressourcenzuteilung

Grundsätzlich wurde dadurch ein direktes Mapping zwischen verfügbaren Funktionalitäten von Ressourcen und benötigten Ressourcen durch den Prozess angelegt, welches als Grundlage für komplexe und größere Prozesse genutzt werden kann. Durch die in der BPMN-Struktur vorgesehenen Prozessgruppen kann ebenfalls ein Prozess hierarchisch aufgebaut werden, um komplexe Herstellungsprozesse zu kapseln und spezifische Arbeitsschritte als Pakete zu verwenden.

²⁶ <https://github.com/eclipse-basyx/>

Mit dem Einsatz virtueller Ressourcen, die mögliche verknüpfte Prozessschritte bearbeiten können, indem sie diese als eigenen Gesamtprozess bearbeiten, kann so die Komplexität verringert und die dynamische Anpassbarkeit, etwa durch minimale Veränderungen in Prozessteilen aufgrund von Messwerten, die während vorheriger Schritte genommen wurden, erhöht werden.

4.3.5.4 Dokumentation der Prozessparameter

Neben der Ausführung der Prozesse werden die Funktionalitäten der Ressourcen genutzt, um Prozessparameter als Teil der Produkt-Verwaltungsschale zu dokumentieren. Abbildung 4-40 zeigt dazu einen abgelaufenen Prozess in BPMN-Notation in CAMUNDA, die gewählten Pfade sind blau markiert. So ist auch in der Visualisierung erkennbar, dass ein Scan- und Schraubprozess durchgeführt wurde.

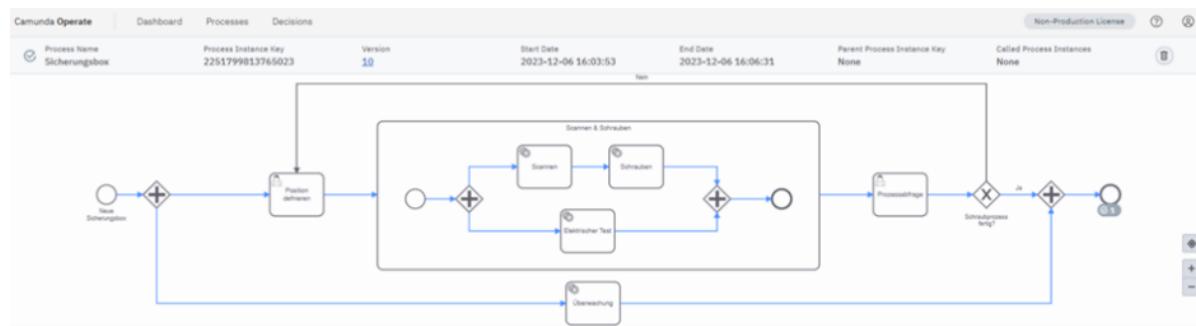


Abbildung 4-40: Darstellung eines Prozessablaufs als BPMN

Alle Komponenten in Form der Verwaltungsschale werden genutzt, um die jeweiligen Prozessdaten, wie bspw. die Crimp-Kraft, produktgenau als Teil des digitalen Produktzwillings festzuhalten. So können Entscheidungen und auch Fehler rückverfolgbar dokumentiert bzw. Abbruchkriterien bereits im laufenden Prozess betrachtet werden. Neben der möglichst frühzeitigen Beendigung des Fertigungsprozesses im Fehlerfall, könnten diese Daten auch für weitergehende Analysen oder auch KI-Fragestellungen verwendet werden, um so die Herstellung noch weiter zu optimieren, bspw. mittels neuer und effizienterer Scheduling-Möglichkeiten eine bessere Maschinenauslastung zu erreichen.

4.3.5.5 Anwendung im Demonstrator

Im Demonstrator des Projektes VWS4LS wird die beschriebene Prozesssteuerung als Ersatz für ein herkömmliches Manufacturing-Execution-System (MES) eingesetzt. Für die Ende-zu-Ende Digitalisierung mit Verwaltungsschalen für alle genutzten Assets können so die notwendigen Daten erhoben und verarbeitet werden.

Der Demonstrator beginnt seine Ablaufsequenz mit dem simulierten MES-System, das als User Interface fungiert und den Gesamtprozess startet, indem zunächst ein Produkt ausgewählt wird. Daraufhin werden die entsprechenden Verwaltungsschalen für *ProductionOrder*, *Batch* und *Lot* angelegt. Anschließend werden die über Node-RED ausgeführten Verhandlungsprozesse aus TP6 angestoßen. Die Ergebnisse dieser Verhandlungen werden dem Nutzer zurückgespielt, damit er zwischen den möglichen Ressourcen auswählen kann. Anschließend wird der Rüstvorgang der Ressourcen gestartet, was ebenfalls in den Ressourcen-Verwaltungsschalen reflektiert wird. Der tatsächliche Fertigungsprozess auf den Produktionsressourcen wird anschließend automatisch gestartet und entsprechend der Auswahl durchgeführt. Schließlich werden die Ergebnisse dokumentiert, bevor der Demonstrationsprozess von Neuem beginnt, um den kontinuierlichen Ablauf und die Anpassungsfähigkeit des Systems zu demonstrieren.

Die während des Prozesses anfallenden Daten werden, wie in Kapitel 4.3.5.4 beschrieben, dokumentiert und eine Visualisierung der Prozessschritte wird als Live-Darstellung in CAMUNDA bereitgestellt.

4.4 AP 3.4 - Proof of Concept

Quellsysteme liefern Daten unterschiedlich strukturiert, d.h. es kann sich um un- bzw. semistrukturierte Daten handeln. Für die zugrunde liegende Datenstruktur in der VWS wurde daher ein Datenmapping Service entwickelt, der die Kompatibilität zwischen bestehenden Systemlandschaften (Brownfield) und der VWS ermöglicht, um in beide Richtungen kommunizieren zu können. Um die Konvertierung zu gewährleisten, ist eine Validierung zwingend erforderlich, welche in verschiedenen Schritten der Datenerhebung und Datenverarbeitung durchgeführt wird.

Die Ziele des generischen Ansatzes für Datenmapping mit anschließender Validierung:

- Konvertierung von un- bzw. semistrukturierten Daten
- Unterbrechungen im Datenfluss verhindern
- Strukturelle Veränderungen erkennen
- Vermeidung unvollständiger Daten
- Steigerung der Datenqualität
- Reduzierung des manuellen Aufwands

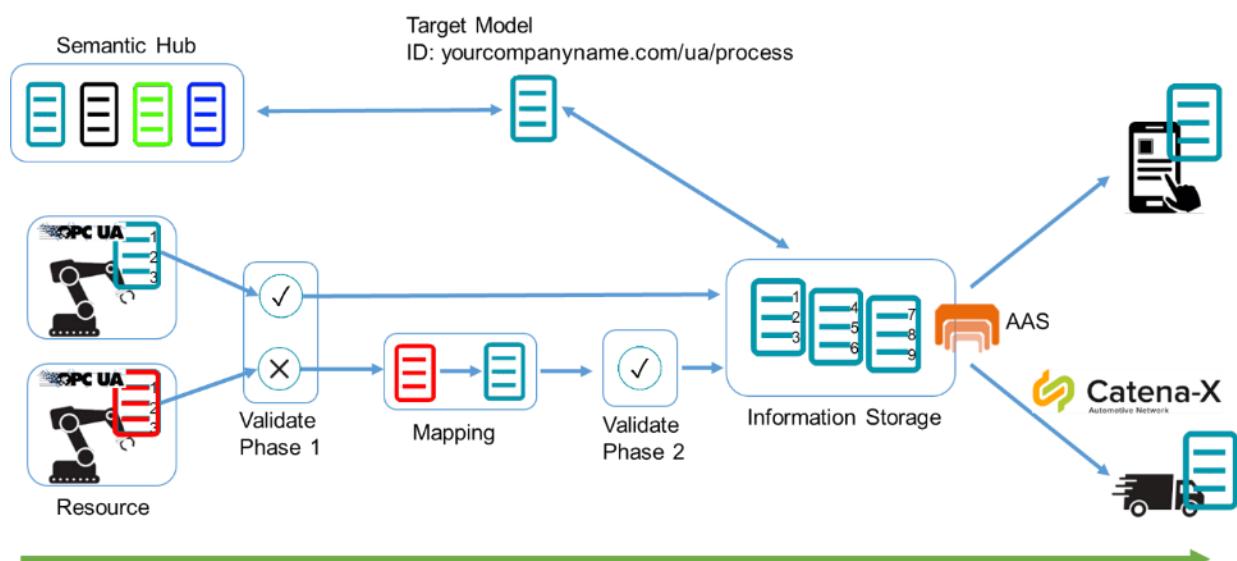


Abbildung 4-41: Exemplarische Darstellung des Data Mapping Services mit VWS als Zielsystem

Abbildung 4-41 zeigt einen fiktiven Ablauf eines typischen Anwendungsfalls. Eine aus Datenqualitätsicht unsichere Quelle, in diesem Schaubild eine Ressource, erzeugt semistrukturierte Daten in Form von einem OPC UA Nodeset (XML-Format). Bevor diese Daten in das Zielsystem (Information Storage in Form der Verwaltungsschale) weggeschrieben werden, werden diese validiert. Im positiven Fall erfolgt direkt das Persistieren im Zielsystem. Im negativen Fall erfolgt ein Mapping der Daten zur Zielstruktur. Diese Daten werden erneut validiert und im positiven Fall weggeschrieben.

Die Verwaltung von gültigen Strukturen der Daten von Quell- und Zielsystem erfolgt im Semantic Hub. Der Semantic Hub beinhaltet alle benötigten Konfigurationen, sowohl aus Datensicht aber auch die Systemkonfiguration von Quell und Zielsystem.

4.4.1 Datenvalidierung

4.4.1.1 Quellformate

In der Praxis ist eine Validierung direkt an der Maschine nicht sinnvoll. Die Vielzahl der unterschiedlichen Maschinen Protokolle und eine in der Regel immer später stattfindende Transformation zur Übertragung der Daten, sprechen gegen eine Validierung an der Maschine.

Aufgrund der Tatsache das in BaSyx die Daten der Verwaltungsschale per REST API in einem JSON-Format übertragen wird, und JSON als Standard am Markt angesehen werden kann, macht es Sinn im POC die Validierung- und die Mapping Logik auf das JSON-Format zu beschränken.

Nach der Validierung wird ausschließlich das validierte/gemappte JSON zur Verfügung gestellt.

4.4.1.2 Semantic Hub

- Die Referenz-Struktur eines Submodels muss vorhanden sein.
- Das Identifizieren des Submodels über eindeutige Identifier muss gegeben sein.
- Eine Versionierung der Referenzstrukturen muss möglich sein.
- Die Semantik des Referenzmodels muss definiert sein.

4.4.1.3 Micro-Service „Datenvvalidierung“

- Der Web-Service ist in der Lage die richtige Zielverwaltungsschale zu finden.
- Der Web-Service ist in der Lage die Namen, die Struktur und den Datentyp der JSON-Objekte zu validieren.
- Im Fehlerfall soll eine für das IT-System verwendbare Fehlerbeschreibung mit Fehlercode zurückgesendet werden.

4.4.1.4 Submodel InformationModelDescription für die Datenvvalidierung

Das Submodel InformationModelDescription hat folgende Anforderungen zu erfüllen:

- Das Identifizieren des Submodels über eindeutige Identifier.
- Eine Versionierung der Referenzstrukturen.
- Die Semantik des Referenzmodels muss definierbar sein.
- Die notwendigen Bearbeitungsfunktionalitäten müssen zur Verfügung stehen.
- Die Daten müssen persistierbar sein.

Die Abbildung 4-42 zeigt eine Ressourcen VWS mit dem verlinkten Submodel *InformationModelDescription*. Das Submodel enthält die SemanticId des Informationsmodells, welche im Semantic Hub als Suchkriterium für die Beschreibung des Informationsmodells dient.

Das Submodel ‚*InformationModelDescription*‘ wird offen gehalten, d.h. die enthaltenen Informationen und Formate der bereitgestellten Informationsmodelle können je nach UseCase erweitert werden. Beispielhaft wurden 3 SMC *InformationModelDescriptionSets* angelegt. Mandatorisch enthält das Submodel die *ModelVersion*

Das in der Abbildung 4-43 dargestellte SMC *InformationModelDescriptionSetJSONSchema* wurde für die Benutzung der Validierung über das JSON-Format angelegt. Das JSON-Schema wurde als String Property mit SemanticId ‚*JSONSchema*‘ hinterlegt.

In der Abbildung 4-44 erfolgte die Umsetzung mit der SMC *InformationModelDescriptionSetAASElement*. Die in einem Template dargestellten Properties werden ohne den eigentlichen Content dargestellt, da die Validierung nur auf die Struktur und nicht auf den Content durchgeführt wird.

AAS "AAS_WDT_UP150" V1.0 [IRI, http://demo2023.vws4ls.com/ids/aas/7031_8082_3022_7912] of [IRI, www.wezag.de/maschine/crif/1/0]

- ▷ **SM "Nameplate"** [IRI, http://demo2023.vws4ls.com/ids/sm/0200_3113_3022_3161]
- ▷ **SM "ArticleInformation"** [IRI, http://demo2023.vws4ls.com/ids/sm/4010_3113_3022_1716]
- ▷ **SM "ContactInformation"** [IRI, http://demo2023.vws4ls.com/ids/sm/9110_3113_3022_8966]
- ▷ **SM "HandoverDocumentation"** [IRI, http://demo2023.vws4ls.com/ids/sm/6210_3113_3022_2805]
- ▷ **SM "TechnicalData" V1.1** [IRI, http://demo2023.vws4ls.com/ids/sm/0310_3113_3022_6366]
- ▷ **SM "MCAD"** [IRI, http://demo2023.vws4ls.com/ids/sm/5410_3113_3022_4726]
- ▷ **SM "Job" V1.0.0** [IRI, http://demo2023.vws4ls.com/submodel/Job/0987654321]
- ▷ **SM "OperationalData" V1.0.0** [IRI, http://demo2023.vws4ls.com/submodel/OperationalData/0987654321]
- ▷ **SM "ExecutionSteps" V1.0.0** [IRI, http://demo2023.vws4ls.com/submodel/ExecutionSteps/0987654321]
- ◀ **SM "InformationModelDescription"** [IRI, https://example.com/ids/sm/6215_1140_9032_0115]

Prop "SemanticId" = https://www.arena2026.com/ids/informationmodeldescription/informationmodel/crimp/1/0



AAS "InformationModelDescriptionShellCrimp" [IRI, http://www.arena2036.com/InformationModelDescriptionShell/Crimp/1/0]

- ◀ **SM "InformationModelDescription"** [IRI, https://www.arena2026.com/ids/sm/informationmodeldescription/informationmodel/crimp/1/0]
- ▷ **SMC "InformationModelDescriptionSetAASElement"** (1 elements)
- ▷ **SMC "InformationModelDescriptionSetAASSerialized"** (1 elements)
- ▷ **SMC "InformationModelDescriptionSetJsonSchema"** (1 elements)

Prop "ModelVersion" = 1.0.0

Abbildung 4-42: AAS Resource mit AAS und SM

AAS "InformationModelDescriptionShellCrimp" [IRI, http://www.arena2036.com/InformationModelDescriptionShell/Crimp/1/0]

- ◀ **SM "InformationModelDescription"** [IRI, https://www.arena2026.com/ids/sm/informationmodeldescription/informationmodel/crimp/1/0]
- ▷ **SMC "InformationModelDescriptionSetAASElement"** (1 elements)
- ▷ **SMC "InformationModelDescriptionSetAASSerialized"** (1 elements)
- ◀ **SMC "InformationModelDescriptionSetJsonSchema"** (1 elements)
- Prop "JsonSchema" = {"\$schema": "http://json-schema.org/draft-04/schema#", "type": "object", "properties": {"Executed": {"type": "boolean"}, "LastRun": {"type": "string"}, "ModelVersion": {"type": "string"}, "Status": {"type": "string"}, "Type": {"type": "string"}}}**
- Prop "ModelVersion" = 1.0.0**

Abbildung 4-43: AAS und SM und SMC InformationModelDescriptionSetJSONSchema

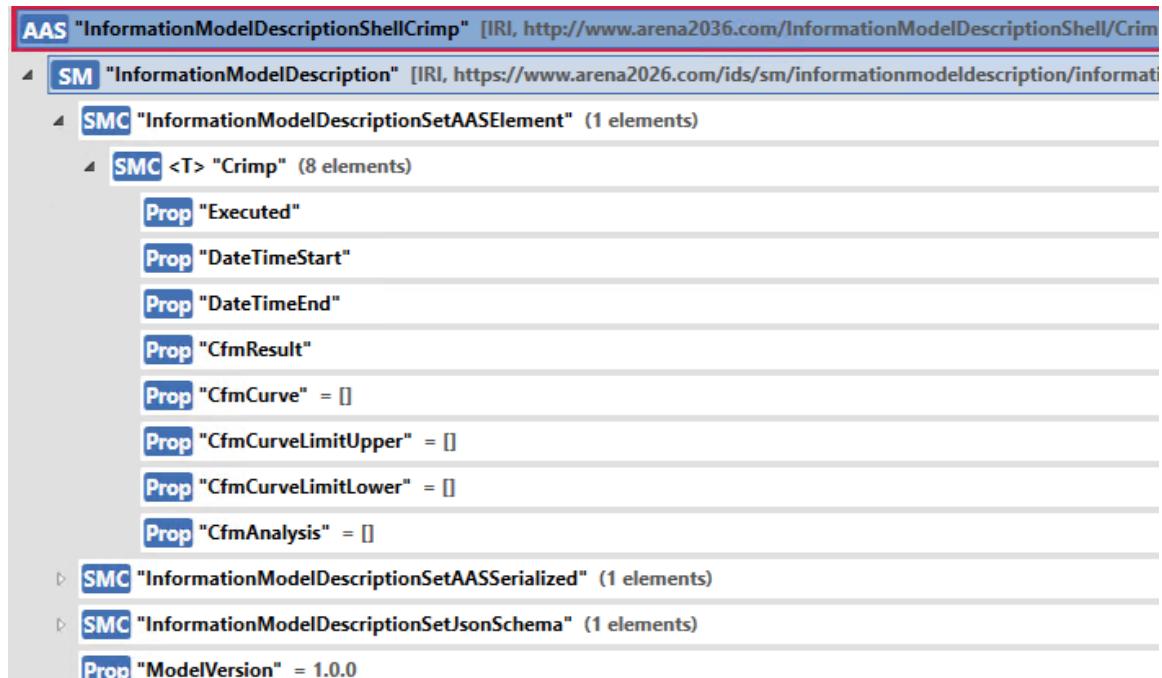


Abbildung 4-44: AAS und SM und SMC InformationModelDescriptionSetAASElement

4.4.1.5 Out of Scope

- Alle Themen, die in der Implementierung in Richtung **Security** gehen, werden ausgeklammert und in den fachlichen Konzepten nicht berücksichtigt. Das Thema Security würde den Rahmen des POC's sprengen.
- Die **Transformation der unterschiedlichen Maschinenschnittstellen** in das Übertragungsformat JSON ist nicht Bestandteil des Daten Mapping Konzeptes. Die Transformation kann im Demonstrator in das existierende MES-Modul eingebaut werden.
- Das **Speichern** der validierten und gemappten Daten in das **Zielsystem** ist nicht Bestandteil des Konzeptes. Dies kann im Demonstrator in das existierende MES-Modul eingebaut werden.

4.4.2 Datenmapping

4.4.2.1 DataBridge: Keine Datensenkenzuordnung im Transformer

Der Data Mapping Service nutzt die [Eclipse BaSyx DataBridge](#), um Daten von einer distinkten Datenquelle an eine oder mehrere Verwaltungsschalen Endpunkt(e) zu transferieren. Dazu ist die DataBridge für das Zuordnen von (physischen) Daten zur logischen Geräteinformationsstruktur der Verwaltungsschale und die Übertragung zuständig. Der Begriff „Datenmapping“ beschreibt hierbei den Prozess dieser Zuordnung, die anhand von vorher festgelegten Mustern erfolgt.

Eine Instanz der DataBridge kann Daten aus einer einzelnen Datenquelle auf mehrere Datensenken übertragen und währenddessen die Daten umwandeln. Diese grundlegende Funktion kann in drei Hauptschritten erklärt werden:

1. **Datenquelle:** Die Datenquelle ist das physische Gerät oder System, das die Rohdaten generiert. Dies können beispielsweise Sensoren, Aktoren oder Maschinen sein, die verschiedene Informationen wie Temperatur, Druck oder Zustandsdaten liefern. Die Datenquellen liefern kontinuierlich oder in Intervallen ihre Messwerte oder Statusinformationen.
2. **Transformer:** Der Transformer ist eine zentrale Komponente, die für das Datenmapping verantwortlich ist. Er transformiert die Rohdaten aus der Datenquelle in ein Format, das für die jeweilige Anwendung oder Datensenke geeignet ist. Somit können komplexe Datenobjekte vereinfacht und auf die jeweiligen semantischen Modelle heruntergebrochen werden. Diese Transformation kann verschiedene Formen annehmen und auch verschachtelt werden:
 - a. Skalierung oder Normalisierung der Daten: Rohdaten werden in einem standardisierten Format bereitgestellt.
 - b. Filterung oder Aggregation: Unnötige Informationen werden herausgefiltert, oder Daten werden auf Basis von bestimmten Regeln zusammengefasst.
 - c. Einheitenumrechnung: Daten können in andere Einheiten umgerechnet werden, um Kompatibilität zu gewährleisten (z.B. von Fahrenheit in Celsius). Der Transformer stellt sicher, dass die Daten der Quelle in einem konsistenten, verwertbaren Format an die nächste Komponente weitergegeben werden.
3. **Datensenke:** Die Datensenke ist das Zielsystem, das die transformierten Daten empfängt und weiterverarbeitet als Teil der Verwaltungsschale.

Zusammengefasst:

- Eine **Datenquelle** erzeugt Rohdaten.
- Die **Transformer** wandeln diese Rohdaten in das passende Format um.
- Die **Datensenken** empfangen die transformierten Daten zur weiteren Nutzung.

Durch diese Pipeline wird eine reibungslose und standardisierte Kommunikation zwischen verschiedenen Systemen und Geräten gewährleistet.

Das nachfolgend aufgeführte Beispiel zeigt die Konfiguration einer Route in der `routes.JSON`, in welcher ein Datenpunkt der Quelle `datasource` auf drei Datensenken (`sink1`, `sink2`, `sink3`) zugewiesen wird. Dabei werden in diesem Fall immer alle drei definierten Transformatoren (`transformer1`, `transformer2`, `transformer3`) durchlaufen.

Datei: routes.JSON

```
[
  {
    "datasource": "mqttSource",
    "transformers": [
      "transformer1",
      "transformer2",
      "transformer3"
    ],
    "datasinks": [
      "sink1",
      "sink2",
      "sink3"
    ],
    "trigger": "event"
  }
]
```

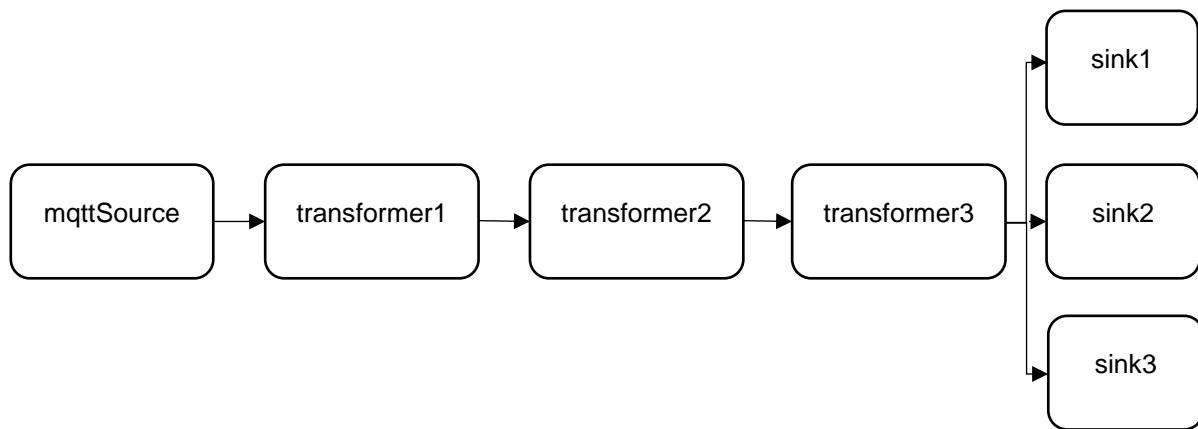


Abbildung 4-45: Keine Datensenkenzuordnung im Transformer

4.4.2.2 DataBridge: Mit Datensenkenzuordnung im Transformer

Im Unterschied zum vorangegangenen Beispiel werden nun für jede Datensenke die zu durchlaufenden Transformatoren angegeben. Im Fall von *sink3* wären dies z.B. *transformer2* und *transformer3*.

Datei: routes.JSON

```
[
  {
    "datasource": "mqttSource",
    "transformers": [
      "transformer1",
      "transformer2",
      "transformer3"
    ],
    "datasinks": [
      "sink1",
      "sink2",
      "sink3"
    ],
    "datasinkMappingConfiguration": [
      {
        "sink1": ["transformer1"],
        "sink2": ["transformer2"],
        "sink3": ["transformer2", "transformer3"]
      }
    ],
    "trigger": "event"
  }
]
```

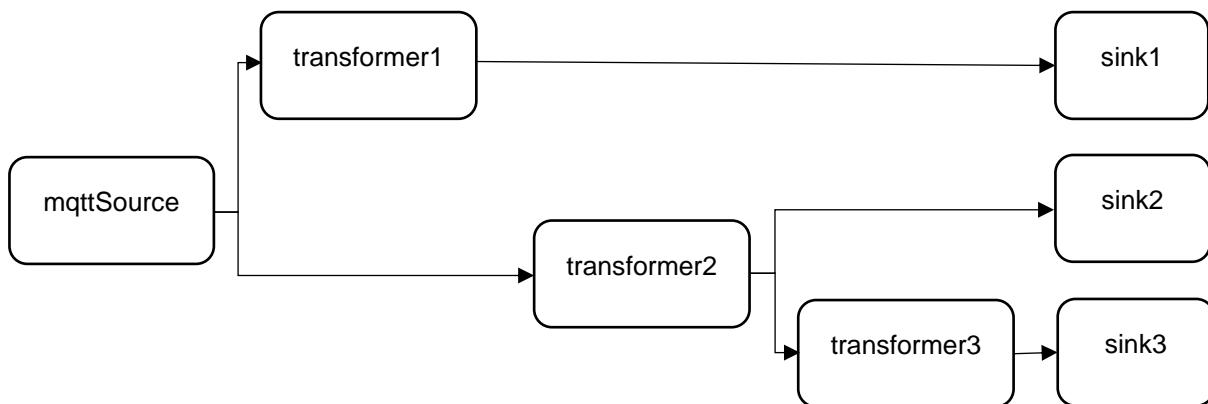


Abbildung 4-47: Mit Datensenkenzuordnung im Transformer

Wurden Routen mit Datensenken (*datasinks*) definiert, die nicht Teil der Datensenkenzuordnung (*datasinkMappingConfiguration*) sind, so durchlaufen diese keine Transformatoren und werden somit nicht innerhalb der Route modifiziert. Wurden mehr Transformatoren definiert, als in der Datensenkenzuordnung (*datasinkMappingConfiguration*) verwendet werden, dann werden diese ignoriert.

4.4.2.3 Mögliche Integration des Datenmapping im Demonstrator

Die Integration der Ergebnisse des Data Mapping Service in den Demonstrator des Projektes wäre möglich, indem die Maschinendaten sowohl über ein MES (Manufacturing Execution System) als auch über OPC UA Knoten ausgelesen werden. Diese Datenquellen ermöglichen es, in Echtzeit auf Produktionsdaten zuzugreifen und diese dynamisch für die weitere Verarbeitung, etwa als Teil der Prozesssteuerung, einzusetzen. Hierbei übernimmt der Data Mapping Service die Aufgabe, die Rohdaten der Maschinen in ein konsistentes, verständliches Format zu transformieren, sodass sie in den digitalen Zwillingen standardisiert abgebildet werden können. Nachfolgende Abbildung zeigt den Ablauf als Teil des Demonstrators.

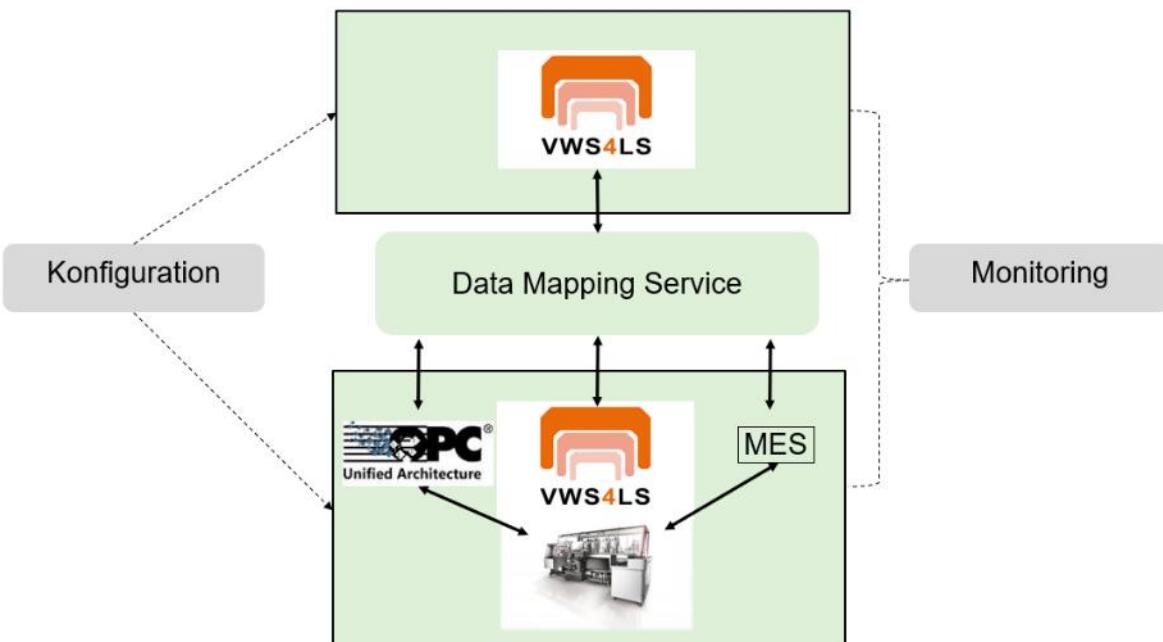


Abbildung 4-48: Mögliche Datenmapping-Integration

4.5 AP 3.5 - Validierung der Prinzipien

Das AP3.5 „Validierung der Prinzipien“ fokussierte sich auf die systematische Erfassung, Gliederung und digitale Beschreibung der Produktionsprozesse, unter Anwendung des VWS-Teilmodells „IDTA-02031: Bill of Processes“ (BoP) [26], der dynamischen Prozessmodellierung mittels BPMN und der dynamischen Prozessteuerung mittels OPC UA [18].

4.5.1 Prozessablaufsteuerung für Produktionsprozesse

Die Produktion eines Leitungssatzes ist ein komplexer Gesamtprozess und erfordert mehrere zu berücksichtigende Teilbereiche. Sie setzt sich aus mehreren Produktionsschritten zusammen, von denen einige manuelle Tätigkeiten umfassen (siehe Abbildung 4-49).

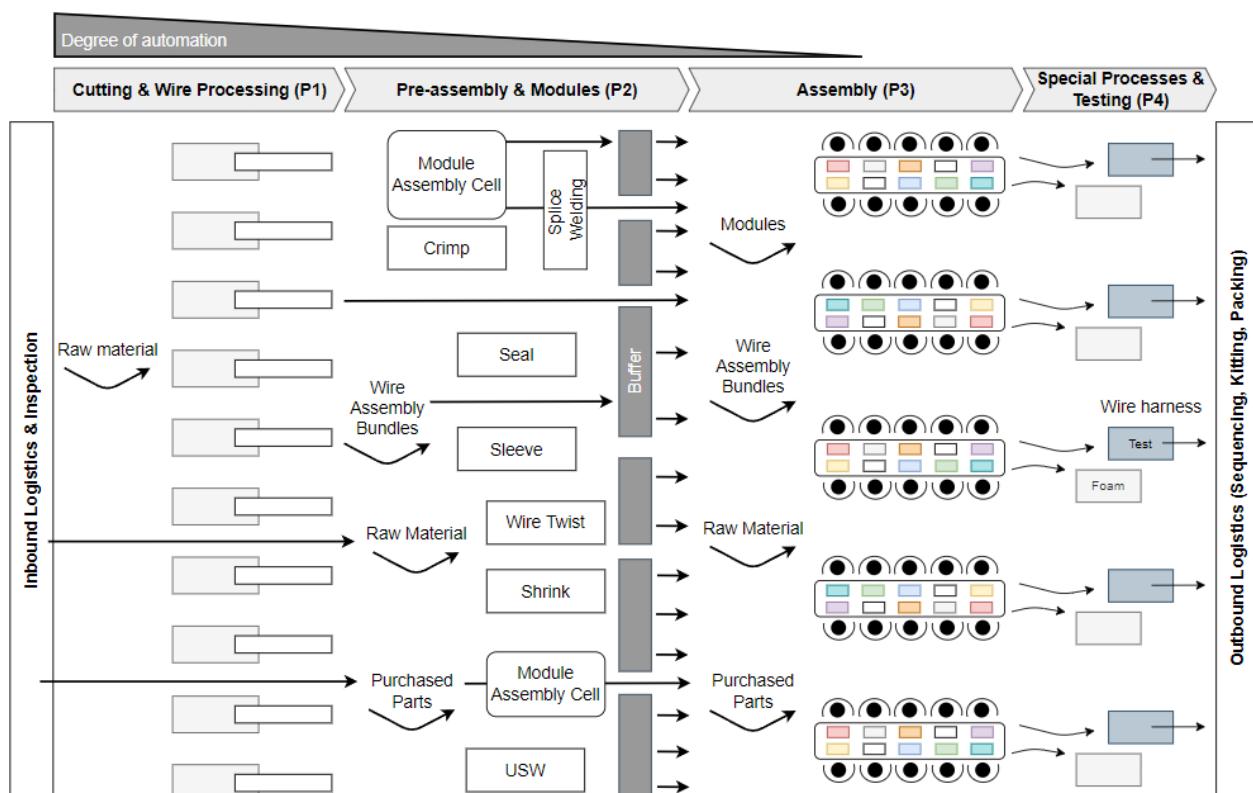


Abbildung 4-49: Übersicht der Produktionsschritte in der Leitungssatzproduktion

Bevor die eigentliche Leitungssatzproduktion beginnt, kommen alle notwendigen (Roh-)Materialien in der Eingangslogistik oder im Lagerbereich an, wo Komponenten und Rohstoffe gescannt werden.

Der **Schneidprozess und die Leitungsverarbeitung (P1)** weisen einen sehr hohen Automatisierungsgrad auf und erfolgen nach dem Eingang des Rohmaterials. Der Schneidprozess wird derzeit von Leitungsverarbeitungsmaschinen durchgeführt, die von einem MES-System gesteuert werden. In der Regel übernehmen die Maschinen in diesem Schritt die Prozesse des Ablängens, des Abisolierens, des Aufbringens von Dichtungen und Kontaktteilen und bei Bedarf der Kennzeichnung durch Etikettierung, Farbcodierung oder maschinenlesbare Markierungen.

Der **Vormontageprozess (P2)** ist teilautomatisiert. In diesem Schritt finden verschiedene Prozesse statt, wie z. B. Schrumpfen oder Ultraschallschweißen (USW) und viele andere, wie in Abbildung 4-49 dargestellt. Häufig werden auch sog. Produktionsmodule in größerer Stückzahl gefertigt, die im späteren Produktionsverlauf dann in den Gesamtleitungssatz verbaut werden.

Der **Endmontageprozess (P3)** wird überwiegend in Handarbeit durchgeführt, wobei die Teile und Module aus P2 als Input für die Leitungssatzmontage dienen, in der diese Vorprodukte montiert, gewickelt, gesteckt und das Layout fertiggestellt werden.

In der Stufe „**Spezielle Prozesse und Prüfung**“ (P4) ist es möglich, dass spezielle Prozesse wie das Schäumen durchgeführt werden. Am Ende dieser Phase wird der Testprozess durchgeführt, um die

Qualität des Leitungssatzes zu überprüfen, wobei die Qualität des gesamten Leitungssatzes und aller seiner Komponenten überprüft wird, einschließlich des Vorhandenseins von Komponenten, elektrischen Anschlüssen, Widerständen, Kondensatoren, Leitungsfarben und Pin-Positionen.

Schließlich wird der Produktionsprozess mit Kommissionierung und Verpackung abgeschlossen, bevor das Produkt das Werk verlässt.

4.5.1.1 Ablaufsteuerung im Sinne einer Sequenzierung (BOP in AAS Product Type)

Bei der Verwaltungsschale kann man zwischen Typen-, also einer abstrakten Beschreibung der Art des Assets, und Instanzen-AAS, also des tatsächlich entstandenen Assets, unterscheiden (siehe Abbildung 4-50).

Bevor die tatsächliche Produktion gestartet wird, befindet man sich auf der Typenebene. D.h. das Produkt ist mit seinen Bestandteilen und den dazugehörigen Produktionsprozessen definiert, jedoch noch nicht produziert.

Sobald ein Produktionsauftrag für einen Leitungssatz und den dazugehörigen Halbfabrikaten angelegt wurde, beginnt die Instanzebene. Das bedeutet, zu diesem Zeitpunkt wird die Produktion eines konkreten Produktes gestartet und alle relevanten Attribute durch die Verwaltungsschale dokumentiert. In der Abbildung 4-50 beschreibt die Verwaltungsschale „Batch01“ eine physische oder logische Unterteilung eines Produktionsauftrages während „Lot01“ Informationen zu allen produzierten Einheiten beinhaltet.

- ▷ **AAS "ProductType"** [AAS_ProductType] of [Leitung, Type]
- ▷ **AAS "ProductionOrder"** [AAS_ProductionOrder] of [Produktionsauftrag, Instance]
- ▷ **AAS "Batch01"** [AAS_Batch01] of [Batch01, Instance]
- ▷ **AAS "Lot01"** [AAS_Leitungsbündel01] of [Leitungsbündel01, Instance]

Abbildung 4-50: Typ- und Instanzen-AAS

4.5.1.1.1 Alle Prozesse für 150%-LS und Ableitung der Fähigkeiten (Required Capabilities)

Der „150%-Leitungssatz“ beschreibt alle Funktionsmodule des Leitungssatzes, die in einem Fahrzeugprojekt existieren rein aus funktionaler Sicht. Dieser Gesamtumfang kann auch nicht in einem konkreten Fahrzeug verbaut werden, da sich Funktionsmodule gegenseitig ausschließen können, wie beispielsweise Xenon- und LED-Scheinwerfer.

Daher muss der 150%-Leitungssatz auch nie in dieser Form produziert werden. Es ist viel mehr die Gesamtheit der technischen Dokumentation, um alle möglichen Funktionsmodule abzubilden und damit später auch produzieren zu können.

Die Daten der Verwaltungsschale für den 150%-Leitungssatz werden von dem Leitungssatzkonfektionär, der mit der Entwicklung des Leitungssatzes betraut ist, erzeugt. Das Vorgehen zur Erzeugung des 150%-Leitungssatzes wurden im Teilprojekt 2 beschrieben.

Diese Zusammensetzung des 150%-Leitungssatz ist also die datentechnische Ausgangsbasis, aus der sich die für die Herstellung notwendigen Prozesse ableiten lassen. So impliziert bspw. ein Crimpkontaktteil an einem Leitungsende den Prozess „Crimp“ oder eine Einzeladerabdichtung an einem Leitungsende den Prozess „Seal“. Mit einer derartigen Zuordnung von Prozessen entsteht dann die Bill of Process (BoP) [26].

Um einen Leitungssatz produktionsoptimiert herstellen zu können, wird neben der Zuweisung von Prozessen des 150%-Leitungssatzes beim Konfektionär eine Gliederung in Baugruppen durchgeführt, also in für die Produktion besser geeignete kleinere Einheiten. In den nachfolgenden Kapiteln wird anhand einer konkreten Baugruppe die Abbildung in einer Verwaltungsschale dargestellt. Die Baugruppe entspricht einer geschnittenen Einzelleitung mit Kontaktteilen auf beiden Seiten inklusive einer Einzelleitungsabdichtung (ELA) auf einer Seite (siehe Abbildung 4-51).



Abbildung 4-51: Baugruppe einer geschnittenen Einzelleitung mit Kontaktteilen inklusive ELA

Die Prozesse, die zur Herstellung aller Module des 150% Leitungssatzes nötig sind, stellen also die Gesamtheit an benötigten Prozessfähigkeiten dar, die ein Leitungssatzkonfektionär bieten muss, um die Kabelsätze produzieren zu können. Enthält eine Baugruppe des Leitungssatzes bspw. einen Crimpkontakt an einer 4mm²-Leitung, so muss dies als Prozessfähigkeit (required capability) hervorgehen. Im Teilprojekt 5 wird das Fähigkeitenmodell inkl. Prozessfähigkeit detaillierter beschrieben, in Teilprojekt 6 die Verhandlungsprozesse.

4.5.1.1.2 Prozesssequenz anhand von Baugruppen

Die in Kapitel 4.5.1.1.1 beschriebene Beispielbaugruppe einer geschnittenen Einzelleitung mit Kontaktteilen auf beiden Seiten inklusive einer Einzelleitungsabdichtung (ELA) auf einer Seite weist durch die BoP bereits die zur Herstellung notwendigen Prozessen auf. Da einzelne Prozessarten (z.B. Strip) mehrfach auftreten können, wird der Prozessbezeichnung ein Zähler beigelegt (Strip01, Strip02, usw.):

- Einzelleitung mit gewünschter Länge → Prozess „Cut01“
- Abisiertes Isolationsmaterials am Leitungsende, Seite 1 → Prozess „Strip01“
- Abisiertes Isolationsmaterials am Leitungsende, Seite 2 → Prozess „Strip02“
- Aufgezogene Einzelleitungsabdichtung am Leitungsende 2 → Prozess „Seal01“
- Angecrimptes Kontaktteil am Leitungsende 1 → Prozess „Crimp01“
- Angecrimptes Kontaktteil am Leitungsende 2 → Prozess „Crimp02“

Die sich aus den Prozessen ergebende Prozesssequenz einer Baugruppe wird später in der AAS des Produktionsauftrages abgebildet inklusive der notwendigen Informationen, wie beispielsweise der Prozessparameter, zu den einzelnen Prozessen.

4.5.1.1.3 Struktur eines Produktionsauftrags

Für die Produktion werden Produktionsaufträge erstellt und – je nach Anforderung – in kleinere Produktionseinheiten aufgegliedert (vgl. Abbildung 4-52). Ein Produktionsauftrag beinhaltet allgemeine Kopfdaten:

- Auftragsgröße
- Auftragsstatus
- Gesamtergebnis der Auftragsbearbeitung
- Referenz auf den Produkttyp

Vom Produktionsauftrag wird auf Produktionslose verwiesen. Jedes Produktionslos besteht aus einer Teilmenge des Produktionsauftrags und beinhaltet Informationen über:

- Losgröße
- Status der Losabarbeitung

Jedes Produktionslos verweist wiederum auf Chargen. Jede Charge besteht aus einer Teilmenge eines Produktionsloses und beinhaltet Informationen über:

- Chargen-Größe
- Status der Chargenabarbeitung
- Anzahl produzierter Einzelerzeugnisse

Zudem verweist jede Charge auf die ausgeführten Jobs der Produkt-Instanz.

Die eindeutige Identifikation der Produkte erfolgt dabei auf Chargenebene. Für Produkte, die nur als Bündel identifizierbar sind (bspw. Einzelleitung), ist die Chargengröße >1. Für Produkte, die einzeln identifizierbar sind (bspw. Leitungssatz), ist die Chargengröße 1.

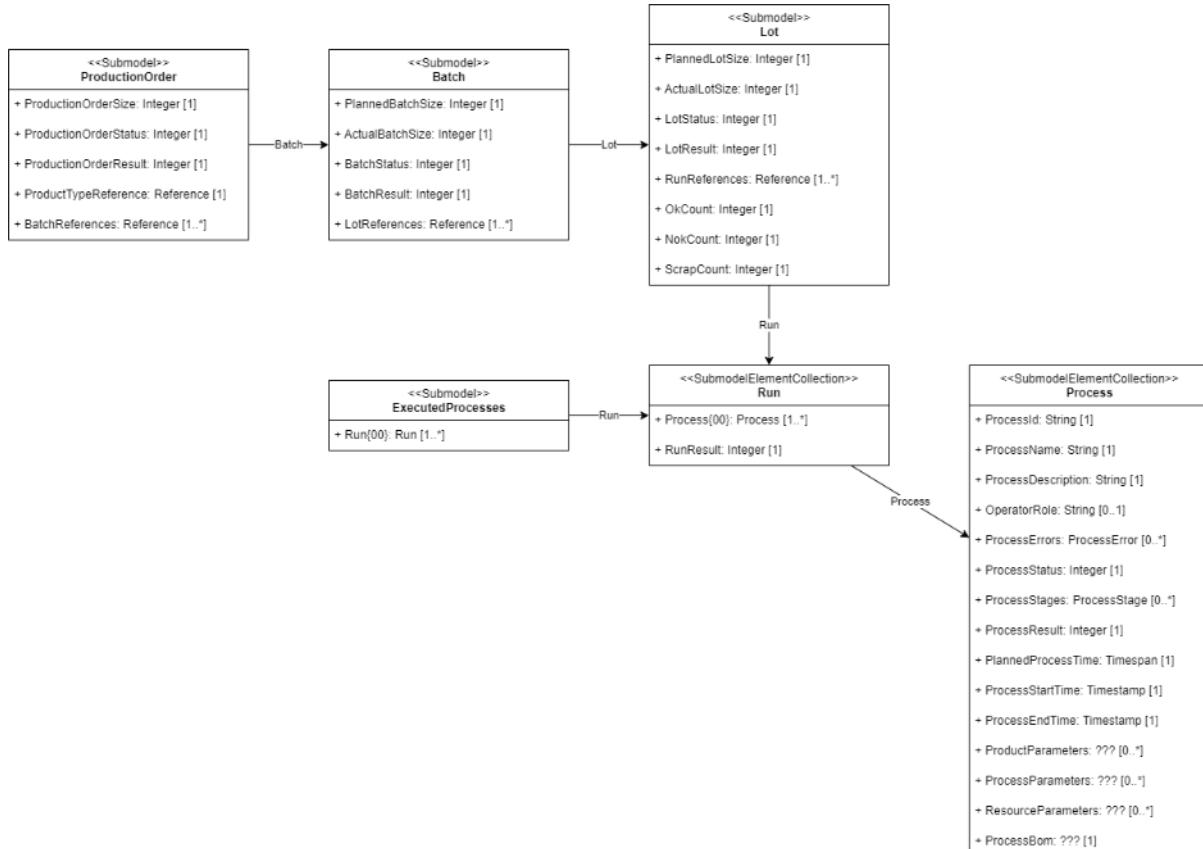


Abbildung 4-52: Struktur eines Produktionsauftrags. (Eigene Darstellung)

4.5.1.2 Beispiele für Produktionsaufträge

4.5.1.2.1 Beispiel 1: Physische Unterteilung eines Produktionsauftrages in Transporteinheiten

Für die Produktion von Leitungen soll ein Produktionsauftrag mit einer Auftragsmenge von 1000 Stück hergestellt werden. Produktionsbedingt werden die Leitungen mit einer Bündelgröße von 10 hergestellt und in Kisten mit je 10 Bündeln verpackt.

Zunächst wird ein Produktionsauftrag mit einer Auftragsmenge von 1000 Stück erstellt. Für den Gesamtauftrag werden 10 Kisten benötigt, die in Form von Produktionslosen dem Produktionsauftrag zugeordnet werden.

Jedes Produktionslos besteht wiederum aus 10 Leitungsbündeln, die in Form von Chargen dem jeweiligen Produktionslos zugeordnet werden.

Der Produktionsauftrag wird an die Fertigung übergeben, wo die Produktionsmaschinen mit der Herstellung der Leitungen beginnen. Für jede Charge wird eine Produktinstanz erstellt, in der die Produktionsdaten hinterlegt sind und alle ausgeführten Jobs der Produktionsmaschinen dokumentiert werden. Für jede produzierte Leitung werden die Produktionsdaten in der Produktinstanz hinterlegt und mit der Charge verknüpft.

Sobald ein Leitungsbündel (eine Charge) fertiggestellt ist, wird dieses in der Kiste platziert. Wenn alle Leitungsbündel für eine Kiste (ein Los) hergestellt wurden, wird die Kiste auf die Palette gestellt. Sobald sich alle Kisten auf der Palette befinden, ist der Produktionsauftrag abgeschlossen.

4.5.1.2.2 Beispiel 2: Logische Unterteilung eines Produktionsauftrages auf mehrere Maschinen

Für die Produktion von Leitungen soll ein Produktionsauftrag mit einer Auftragsmenge von 1000 Stück hergestellt werden. Es erfolgt eine logische Unterteilung der Auftragsmenge zu je 200 Stück, die auf 5 gleichartigen Maschinen parallel produziert werden. Produktionsbedingt werden die Leitungen mit einer Bündelgröße von 20 hergestellt.

Zunächst wird ein Produktionsauftrag mit einer Auftragsmenge von 1000 Stück erstellt. Da dieser Produktionsauftrag auf 5 Maschinen ausgeführt wird, wird eine logische Unterteilung des Produktionsauftrages in 5 Batches zu je 200 Stück durchgeführt.

Jedes dieser Batches wird auf einer Maschine in 10 Produktionslosen hergestellt. Jedes Produktionslos wird in Form eines Leitungsbündels (Charge) zu je 20 Leitungen produziert. Jedes Leitungsbündel ist eindeutig identifizierbar (z.B. Barcode) und kann dem jeweiligen Produktionslos zugeordnet werden.

Der Produktionsauftrag wird an die Fertigung übergeben, wo die 5 Produktionsmaschinen mit der Herstellung der Leitungen beginnen. Für jedes Leitungsbündel (Charge) wird eine Produktinstanz erstellt, in der die Produktionsdaten hinterlegt sind und alle ausgeführten Jobs der Produktionsmaschinen dokumentiert werden. Für jede produzierte Leitung werden die Produktionsdaten in der Produktinstanz der zugehörigen Charge hinterlegt. Eine Charge ist identifizierbar (z.B. Barcode), während einzelne Leitungen nicht identifizierbar ist (kein Identifikationsmerkmal an der Leitung vorhanden). Der Produktionsauftrag ist abgeschlossen, wenn alle Batches auf allen Maschinen abgeschlossen sind.

4.5.1.3 Funktionsweise eines Produktionsablaufs

Grundsätzlich kann man zwischen einer Einzelteil- und einer Batchfertigung unterscheiden. Das SM ProductionOrder (Produktionsauftrag) beinhaltet hierfür die Möglichkeit über sog. BatchReferences eine Referenz auf ein zu fertigendes Batch (Batch01) zu setzen (Abbildung 4-53). An dieser Stelle ist es auch möglich den Produktionsauftrag in mehrere Batches (Batch01, Batch02, ...) zu unterteilen wie in Kapitel 0 bereits beschrieben.



Abbildung 4-53: Submodel ProductionOrder

Die oben beschriebene Referenz Batch01 zeigt nun auf eine Verwaltungsschale Batch01, die alle Informationen zu dem gefertigten Batch enthält (Abbildung 4-54). In der SMC LotReferences ist zu erkennen, dass ein Lot mit einer Referenz zu Lot01 gefertigt wurde.

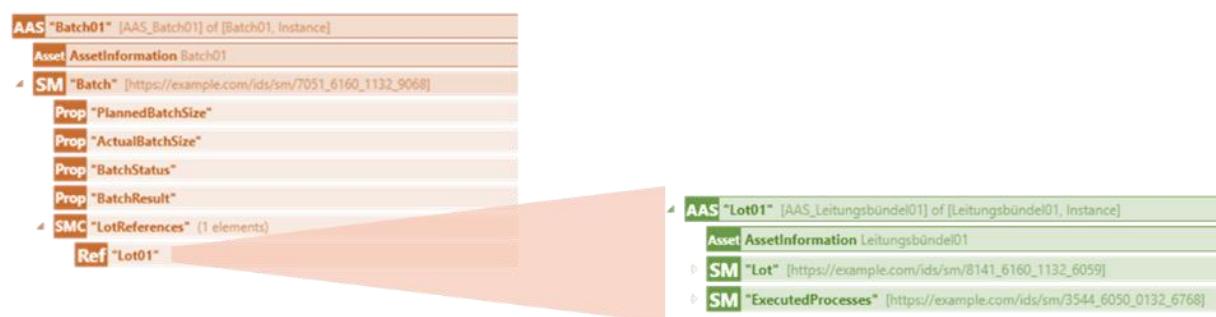


Abbildung 4-54: AAS & Submodel Batch

Die Referenz Lot01 zeigt auf die AAS Lot01, die alle Informationen zu dem gefertigten Lot enthält (Abbildung 4-56). Die AAS Lot01 enthält die beiden Submodels „Lot“ und „ExecutedProcesses“.

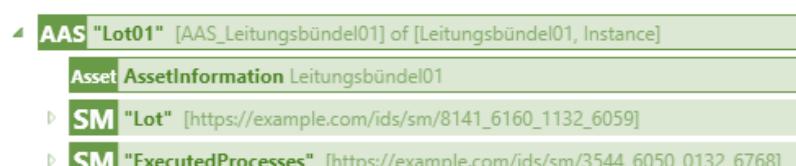


Abbildung 4-55: AAS Lot / Submodel Lot & Executed Processes

Das Submodel Lot beinhaltet zum Lot01 diverse Properties zum Lot und eine SMC RunReferences, die Referenzen zu einzelnen Runs (Run01, Run02, Run03) enthält. Das Submodel ExecutedProcesses enthält die einzelnen referenzierten Runs in Form von SMCs. Jeder einzelne Run beschreibt ein gefertigtes Produkt innerhalb des Lots inklusive aller Produktionsprozesse und dem RunResult. Die Produktionsprozesse und deren Parameter sind Bestandteil der Prozessliste aus AP3.3 und werden dort im Ergebnisdokument detailliert beschrieben.

| | |
|------------|--|
| AAS | "Lot01" [AAS_Leistungsbündel01] of [Leistungsbündel01, Instance] |
| Asset | AssetInformation Leistungsbündel01 |
| SM | "Lot" [https://example.com/ids/sm/8141_6160_1132_6059] |
| Prop | "PlannedLotSize" = 3 |
| Prop | "ActualLotSize" = 3 |
| Prop | "OkCount" = 3 |
| Prop | "NokCount" = 0 |
| Prop | "ScrapCount" = 0 |
| Prop | "LotStatus" = Finished |
| Prop | "LotResult" = OK |
| SMC | "RunReferences" (3 elements) |
| Ref | "Run01" |
| Ref | "Run02" |
| Ref | "Run03" |
| SM | "ExecutedProcesses" [https://example.com/ids/sm/3544_6050_0132] |
| SMC | "Run01" (7 elements) @{Leitung01} |
| SMC | "Run02" (7 elements) @{Leitung01} |
| SMC | "Run03" (7 elements) @{Leitung01} |

Abbildung 4-56: AAS & Submodel Lot

Für eine Einzelteilefertigung wird ein Batch (Batch01) und in diesem Batch nur ein Lot (Lot01) verwendet. Im Lot01 definiert man für das Property „PlannedLotSize“ den Wert „1“, damit nur ein einzelnes Produkt für dieses Lot gefertigt wird. Somit lässt sich diese Struktur sowohl für Einzelteil- als auch Batchfertigung nutzen.

4.5.1.4 Ablaufsteuerung für parallel ausgeführte Prozesse

Das Teilmodell „Bill of Process“ (BoP) [26] spielt eine zentrale Rolle bei der Ablaufsteuerung parallel ausgeführter Prozesse. Es liefert die notwendigen Grundlagen für die jeweiligen auszuführenden Prozessschritte, wie beispielsweise *Cutting* oder *Crimping*. Diese Prozessbeschreibungen aus dem BoP-Teilmodell müssen in physikalische Prozesse umgewandelt werden, was eine automatische Transformation des BoP-Teilmodells in ausführbare Prozessmodelle erfordert.

Im allgemeinen Ablauf der Prozesssteuerung (siehe Kapitel 4.5.1.1) wird beschrieben, wie ausgehend von den im BoP-Teilmodell definierten Prozessschritten konkrete Arbeitsabläufe abgeleitet werden. Diese Abläufe werden durch eine detaillierte Planung und Analyse der benötigten Ressourcen, der Abhängigkeiten zwischen den Prozessschritten und der notwendigen Reihenfolge der Ausführung erstellt. Die hier beschriebenen Techniken und Methoden finden auch in der Ablaufsteuerung für parallele und sequenzielle Prozesse Anwendung.

Die Handhabung von parallelen und sequenziellen Prozessen erfordert eine präzise Koordination und Abstimmung der verschiedenen Prozessschritte. Diese Prozesse werden grundsätzlich in Form eines BPMN-Diagramms (Business Process Model and Notation) abgebildet und für eine leichtere Handhabung visualisiert wie in Abbildung 4-57 dargestellt. Das Ziel ist es, auf dem bestehenden BoP-Teilmodell aufzubauen und dieses als Basis für das Zielbild der Digitalisierung mittels BPMN zu verwenden.

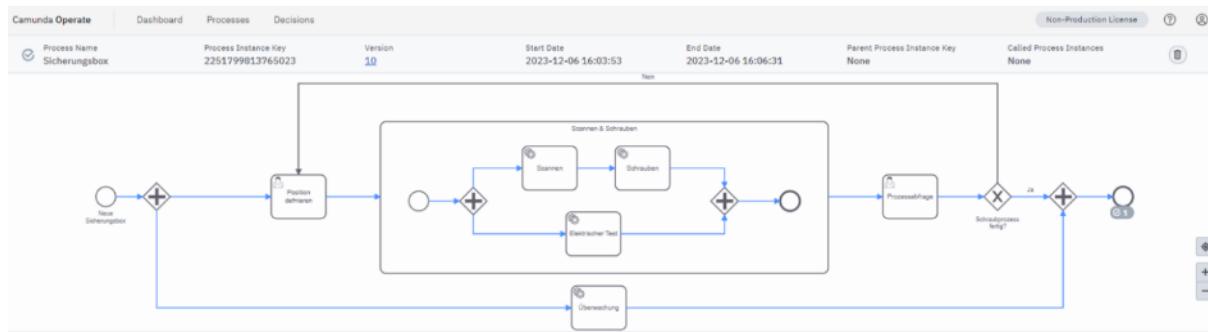


Abbildung 4-57: Überwachung des Schraubprozesses mit parallel abgearbeiteten Aufgaben

Die Modellierung des Prozesses mithilfe von BPMN beschreibt die einzelnen Prozessschritte zunächst abstract. Diese jeweiligen Arbeitsschritte werden durch eingehende und ausgehende „Flows“ verbunden. Konnektoren wie Parallel- oder Exklusiv-Gateways verbinden mehrere Prozessschritte und ermöglichen die Steuerung des Ablaufs.

- Parallele Prozesse:** Müssen aufeinander abgestimmt werden. Beispielsweise darf die Überwachung erst enden, sobald alle parallelaufenden Prozesse abgeschlossen sind. Hierfür sind spezielle Koordinationsmechanismen erforderlich als Teil der Implementierung der Ausführungslogik der jeweiligen Schritte notwendig.
- Sequentielle Prozesse:** Diese Prozesse folgen einer definierten Reihenfolge. In diesem Fall muss die Reihenfolge vordefiniert und strikt eingehalten werden.

Im Projekt VWS4LS werden Prozessschritte im digitalen Zwilling abgebildet und koordiniert. Dazu werden alle Schritte automatisiert mit Operationen der Verwaltungsschale verknüpft. Dies erfolgt über dynamisch festgelegte Schlüssel, welche auf Verhandlungsprozesse aus Teilprojekt 6 aufbauen sollen, um entsprechende Geräte und Aufgaben dynamisch verbinden zu können.

Zusätzlich werden manuelle Teilschritte, etwa für Eingaben von Werkern, mit abgebildet. Dadurch kann etwa, wie in Abbildung 4-57 dargestellt, eine Abfrage zur Position durch den Werker behandelt werden. Automatisierte Teilschritte beinhalten Aufgaben für einen bestimmten Teilprozess wie etwa die Steuerung von Maschinen. Dabei handelt es sich um „Microservices“, welche vorher zentrale Services wie ein Manufacturing-Execution-System (MES) ersetzen und die jeweiligen Geräte mit mehr „Intelligenz“ ausstatten. Alle Prozessschritte werden automatisch angesteuert und besitzen einen Automatismus, um eine erfolgreiche (oder auch fehlerhafte) Beendigung des Schrittes zu dokumentieren und darüber zu informieren.

Parallele Prozesse benötigen zusätzliche Events zur Koordination mit den anderen Prozessen. Diese Events sorgen dafür, dass alle parallelaufenden Prozesse synchronisiert sind und korrekt abgeschlossen werden können.

Der Prozess selbst wird durch das Anlegen eines neuen Auftrags gestartet. Dieses führt zu einem Event, welches von einem weiteren Service erkannt wird. Ein Beispiel zur Veranschaulichung ist ein Rundtakt, der Prozessschritte nur in einer bestimmten Reihenfolge abarbeiten kann. In diesem Fall muss die Reihenfolge im Vorfeld genau definiert sein, um einen reibungslosen Ablauf zu gewährleisten.

4.5.1.5 Technische Beschreibung der Prozesssteuerung

In der entwickelten prototypischen Erweiterung von [Eclipse BaSyx](#) werden neben dem standardmäßig eingesetzten MQTT-Broker als Event-Mechanismus und der open-source BPMN-Engine Zeebe ([CAMUNDA](#)) auch zwei weitere Services, einer „ProcessFactory“, um den Prozess als dynamisches Element in Verbindung mit den Ergebnissen der Verhandlungen aus TP6 verfügbar zu machen und zu starten, und einem „WorkerManager“, um alle notwendigen Operationen der Geräte-Zwillinge (inklusive der virtuellen Geräte-Zusammenschlüsse) mit einem ausführbaren „Worker“ zu versehen. Abbildung 4-58 zeigt die dazugehörige Software-Architektur und die Interaktionen der verschiedenen Services.

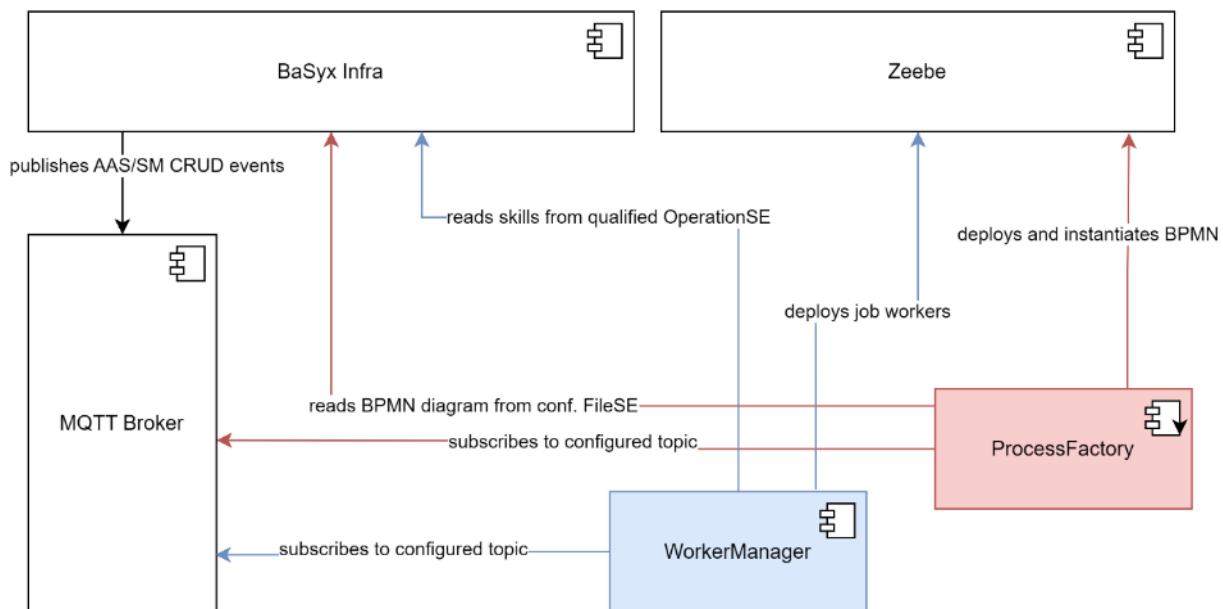


Abbildung 4-58: Architektur der Prozesssteuerung

Um ein Beispiel-Szenario auszuführen, müssen zunächst die notwendigen Services auf Basis von Eclipse BaSyst erstellt werden. Im Projekt geschieht das im Rahmen von Docker-Images und dadurch Containern, die nach Ausführung alle Services der ProcessFactory und des WorkerManager verfügbar machen. Technisch wird zum Starten eines Prozesses über das Eventing-System eine „*deploy_operation*“ Nachricht versendet, welche alle registrierten Verwaltungsschalen nach ausführbaren Prozessen durchsucht und jeweils eine Start-Operation zum jeweiligen Teilmodell hinzufügt. Dadurch wird sichergestellt, dass nur beschriebene Prozesse im System ausgeführt werden können, aber auch Änderungen (inkl. Erstellen von neuen und Löschen von existierenden) an Prozessen durchgeführt werden können.

Gleichzeitig gibt es eine „*update_skills*“ Nachricht, welche die Worker alle Geräte aktualisiert und entsprechende Verknüpfungen zwischen der Prozess-Engine Zeebe und der Verwaltungsschale herstellt. Dies ist die Grundlage, um die Prozessschritte automatisiert ausführen zu können, da durch Zeebe automatisch die jeweiligen Worker zu bestimmten Schritten angesteuert werden.

4.5.1.5.1 ProcessFactory

Die ProcessFactory als eigenständiger Service erfüllt die Aufgabe Prozesse aus Verwaltungsschalen zu erkennen und in der Verwaltungsschalenumgebung ausführbar zu machen. Abbildung 4-59 zeigt das dazugehörige Sequenzdiagramm und die Interaktion mit den BaSyst-Core-Services und der BPMN Engine selbst. So wird aus dem Digitalen Zwilling ein BPMN-Prozess-Template ausgelesen, welches zuvor aus dem BOP-Teilmodell definiert wurde, während eine *deployProcess*-Operation hinzugefügt wird, um den jeweiligen Prozess zu starten. Sobald ein Nutzer die entsprechende Operation, etwa wie im Demonstrator über das Interface des MES-Systems, startet, wird eine Instanz des Prozesses in der BPMN-Engine registriert und gestartet. Dieser Prozess greift automatisiert auf die Operationen der Geräte/Resourcen-Verwaltungsschalen in Eclipse BaSyst über den WorkerManager zu.

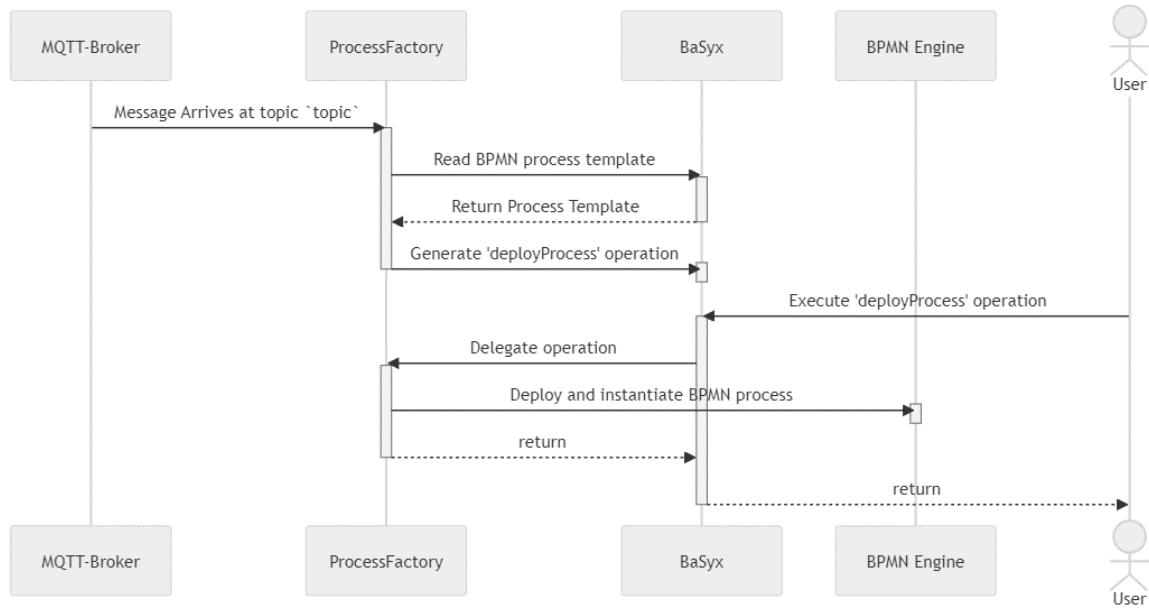


Abbildung 4-59: ProcessFactory Sequenzdiagramm

Aktualisierungen der Prozesse können durch eine erneute Nachricht zum Remapping der vorhandenen Prozesse angesteuert werden.

4.5.1.5.2 WorkerManager

Der WorkerManager, ähnlich zur ProcessFactory, ist ein zunächst autonomer Service auf Basis der Verwaltungsschale und der BPMN-Engine. Abbildung 4-60 zeigt auch hier ein Sequenzdiagramm zur Funktionalität des Services. Gleich zur ProcessFactory wird durch ein Event, welches automatisiert, aber auch manuell angesteuert werden kann, aus allen in BaSyst registrierten Verwaltungsschalen von Ressourcen, sofern vorhanden, ausführbare Operationen für potentielle Arbeitsschritte extrahiert. Zu jeder Operation wird automatisch ein Worker in der BPMN-Engine registriert, um nach Prozessstart automatisch die Ressource anzusteuern.

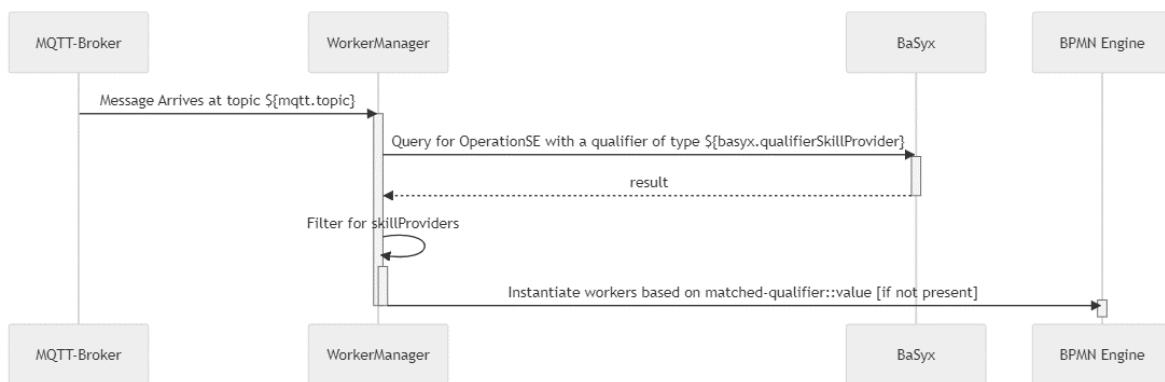


Abbildung 4-60: WorkerManager Sequenzdiagramm

4.5.2 Datenmapping mit Validierung

Quellsysteme im Shopfloor (z.B. Produktionsanlagen) liefern Daten unterschiedlich strukturiert, d.h. es kann sich um un- bzw. semi- strukturierte Daten handeln. Um die Konvertierung zwischen Strukturen und einem digitalen Datenthread zu gewährleisten, ist eine Validierung und ein eventuell notwendiges Mapping zwingend erforderlich. Die Validierung muss in verschiedenen Schritten der Datenerhebung und Datenverarbeitung durchgeführt werden.

Die Ziele des generischen Ansatzes für die Validierung mit Datenmapping:

- Konvertierung von un- bzw. semi-strukturierten Daten
- Vermeidung inkonsistenter Daten
- Unterbrechungen im Datenfluss verhindern
- Strukturelle Veränderungen erkennen
- Vermeidung unvollständiger Daten
- Steigerung der Datenqualität
- Reduzierung des manuellen Aufwands

Die Validierung betrachtet nicht den Content der Daten.

4.5.3 Micro-Service „Datenvvalidierung“

Der Validierungsdienst ist als REST-Service mit Patch-Endpoint implementiert. Der Service benötigt eine JSON-Datei als Input. Das JSON wird validiert und als Ergebnis wird ein serialisiertes JSON-Objekt zurückgegeben.

Die Abbildung 4-61: Testszenario Validation Service zeigt den Aufbau des Testszenarios.

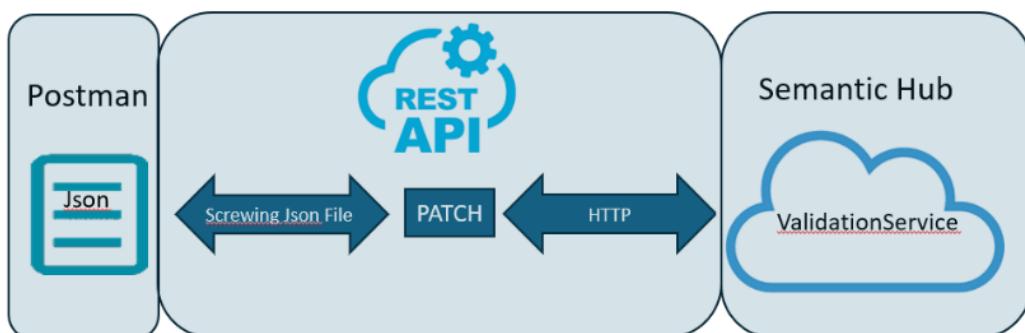


Abbildung 4-61: Testszenario Validation Service

Input: JSON-Datei eines Schraubprozesses

Returns: Der Dienstauftrag wird mit [HTTP-Statuscode](#) 200 oder 404 zurückgegeben. Wenn der Validierungsdienst erfolgreich ausgeführt wird, gibt der Dienst eine JSON-Zeichenkette mit den Ergebnissen zurück, ansonsten einen Fehlermeldungstext.

Beispiel 1 (Validierung mit positiven Ergebnis):

```
{"resultValue":true, "valueType":"boolean", "message":""}
```

In der JSON-Zeichenkette ist der resultValue true, wenn die Validierung erfolgreich war. Der Wert von valueType ist per default Boolean. Die message ist leer, wenn der Dienst erfolgreich ausgeführt wurde

Beispiel 2 (Validierung mit negativen Ergebnis):

```
{"resultValue":false, "valueType":"boolean", "message":("$.sealing_30091.ProcessData.MaterialPressure.AV: boolean found, string expected ")}
```

Die Validierung schlägt fehl, weil die JSON-Datei nicht mit dem JSON-Schema übereinstimmt. Der Dienst gibt ebenfalls 200 (HTTP_OK) zurück. Der Parameter "resultValue" liefert *false* zurück. Im Attribut "message" wird die Fehlermeldung des Validierungsdienstes angezeigt.

Beispiel 3 (Semantic ID wird nicht gefunden):

```
{"The semantic id semantic4567 is not found."}
```

Wenn der Dienst mit einer nicht bekannten Semantic ID fehlschlägt, gibt der Dienst 404 mit einer Fehlermeldung zurück.

Der Source Code des Web-Services und eine Installationsanleitung ist abgelegt unter <https://github.com/VWS4LS/vws4ls-data-validation/>.

4.5.4 Traceability

Die Rückverfolgbarkeit aller Aspekte des Leitungssatzes sind ein elementarer Bestandteil der Leitungssatzentwicklung und -produktion. Im Architekturkapitel „Rückverfolgbarkeit“ werden die Konzepte und Use Cases ausführlich beschrieben. Es wird auch auf die Einhaltung gesetzlicher Aufbewahrungsfristen sowie die Datenvorhaltung eingegangen.

4.5.5 Transfer in andere Produktserien

4.5.5.1 Coroplast

Coroplast ist auf die Herstellung von technischen Klebebändern, Kabeln und Leitungen oder Leitungssätze spezialisiert. Zur Validierung der in dieser Arbeit vorgestellten Prinzipien soll der Prozess der Kabelherstellung dargestellt werden. Dabei soll vor allem überprüft werden, inwieweit dieses Konzept in der Praxis anwendbar ist. Der zu bewertende Anwendungsfall befasst sich mit der Bereitstellung und Produktion der Materialien, die für die Herstellung eines Kabels erforderlich sind.

Wie in Abbildung 4-62 zu sehen ist, kann das Produkt Kupfer von einem unserer Lieferanten bezogen werden, während das Silikon auf unseren Maschinen hergestellt wird. Aus diesem Grund wird das Silikonprodukt als Typ dargestellt, während das Kupfer durch die Daten des Lieferanten dargestellt wird. Für die Herstellung des Kabels ist es im nächsten Schritt notwendig, das Silikon auf das Kupfer zu extrudieren. Des Weiteren ist zu erkennen, dass das Kabelprodukt aus Silikon und Kupfer besteht und als Typ definiert ist, wobei die Materialien in der Stückliste beschrieben werden.

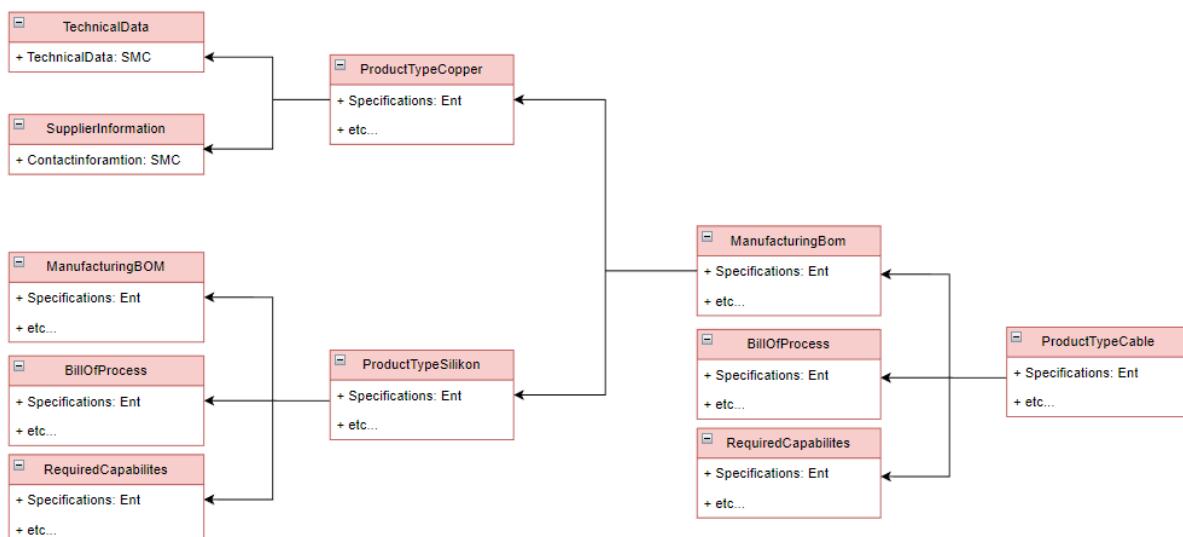


Abbildung 4-62: Repräsentation des Kabelherstellungsprozesses

Um mit der Validierung der Prinzipien fortzufahren, werden im Folgenden die Konzepte im Ökosystem Verwaltungsschale dargestellt. Abbildung 4-63, zeigt die Darstellung der wichtigsten Materialien, wobei

Kabel und Silikon, wie sie von Coroplast hergestellt werden, direkt als "Typ"-Produkte dargestellt werden. Andererseits ist zu erkennen, dass der Werkstoff Kupfer, da er nicht von Coroplast hergestellt wird, aus zwei Teilmödellen besteht: technische Informationen und Lieferantendaten. Darüber hinaus ist jeder Ressource eine Verwaltungsschale zugeordnet, wie z.B. bei der Knetmaschine und der Extrusionsmaschine, die durch die technischen Spezifikationen und die Kapazitäten der jeweiligen Maschine dargestellt werden.

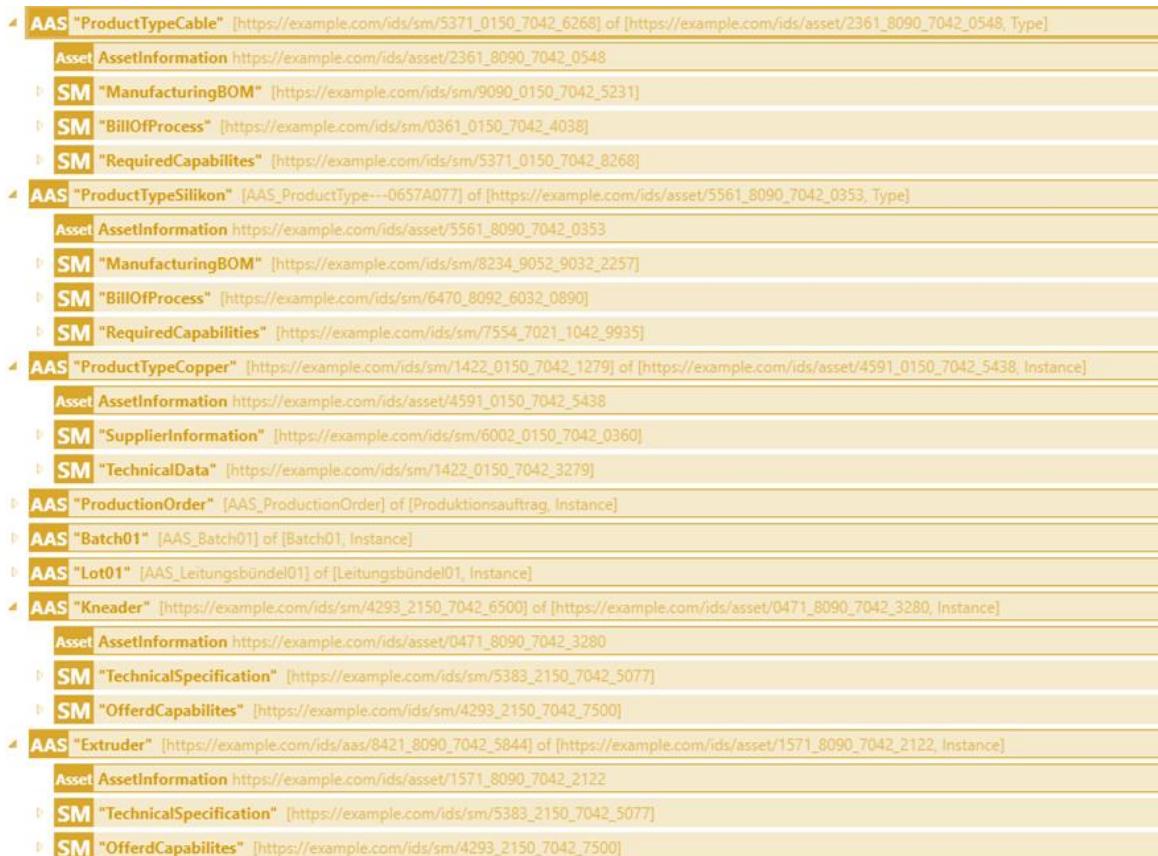


Abbildung 4-63: Repräsentation des Kabelherstellungsprozesses in der VWS

4.5.5.1.1 Prototypische Implementierung zur Konzeptvalidierung

Um das Konzept weiter zu validieren, haben wir eine prototypische Implementierung vorgenommen. Abbildung 4-64 zeigt die Architektur zur Konzeptvalidierung. Für die Darstellung und Speicherung der Verwaltungsschalen nutzen wir die Implementierung des Fraunhofer-Instituts namens BaSyx. Die BaSyx-Implementierung umfasst eine Reihe von essenziellen Services, die zusammen ein leistungsfähiges Ökosystem für die Industrie 4.0 bilden. Im Zentrum steht der *aas-env* Service, der als Asset Administration Shell Environment fungiert und die grundlegende Infrastruktur für die Verwaltung digitaler Zwillinge bereitstellt. Ergänzend dazu operiert der *aas-registry* Service als Asset Administration Shell Registry, der die Registrierung und das Auffinden von Asset Administration Shells ermöglicht. Für die Verwaltung von Submodellen ist der *sm-registry* Service zuständig, der als Submodel Registry dient. Um die Auffindbarkeit von Asset Administration Shells zu gewährleisten, kommt der *aas-discovery* Service zum Einsatz. Abgerundet wird das Serviceportfolio durch die *aas-web-ui*, eine Asset Administration Shell Web User Interface, die eine benutzerfreundliche Oberfläche für die Interaktion mit den verschiedenen BaSyx-Komponenten bietet. Diese Services arbeiten nahtlos zusammen, um eine umfassende Lösung für die digitale Repräsentation und Verwaltung von industriellen Assets im Kontext der Industrie 4.0 zu schaffen.

Zur persistenten Speicherung der Daten wird eine MongoDB eingesetzt. Diese NoSQL-Datenbank bietet die nötige Flexibilität und Skalierbarkeit für die Verwaltung der komplexen Strukturen der VWS.

Die VWS werden mittels eines Python-Scripts unter Verwendung des basyx-python-sdk Packages erstellt. Der Prozess läuft wie folgt ab:

- Erstellung der VWS durch das Python-Script
- Serialisierung der VWS in JSON-Dateien
- Übertragung der serialisierten Daten über die Web-API des aas-env Service von BaSyst

Anschließend können die Daten über die aas-web-ui betrachtet und evaluiert werden.

Die Quellen für die Daten in unserem System sind vielfältig. In der prototypischen Implementierung werden Daten primär aus zwei Quellen bezogen:

- OPC-UA-Protokoll
- Maschinendatenbank

Für Testzwecke und zur Vereinfachung werden in unserem Prototyp Daten durch das Python-Script simuliert.

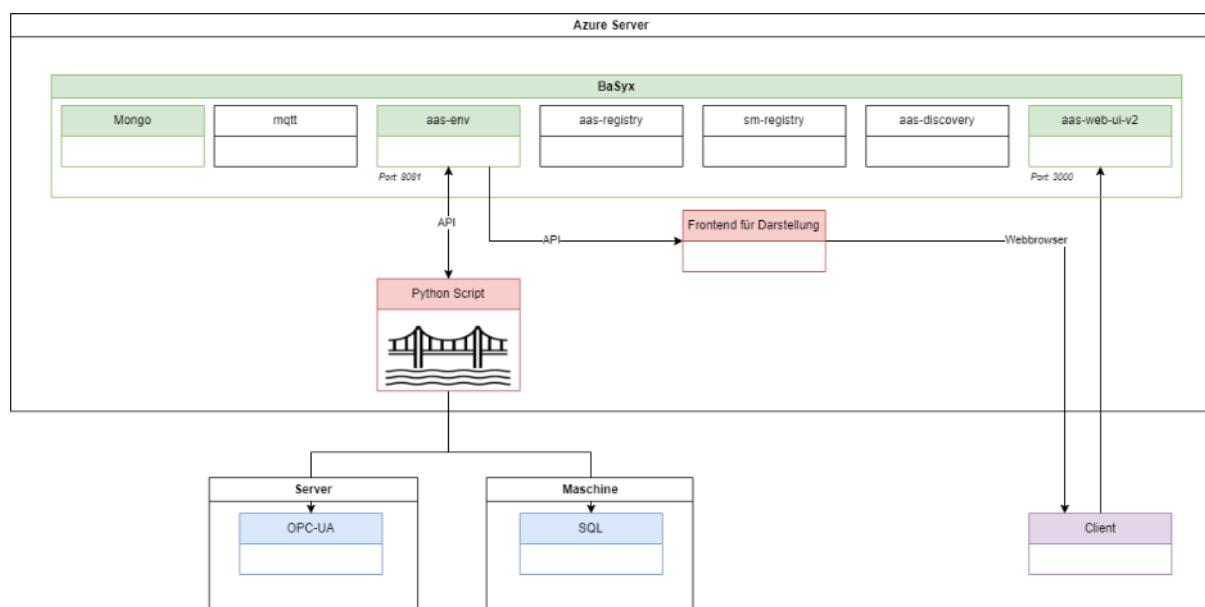
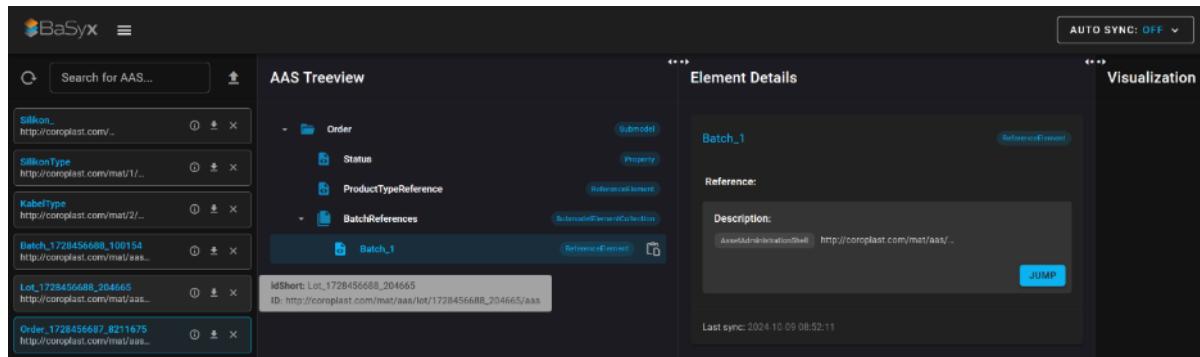


Abbildung 4-64: Architektur für die Konzeptvalidierung

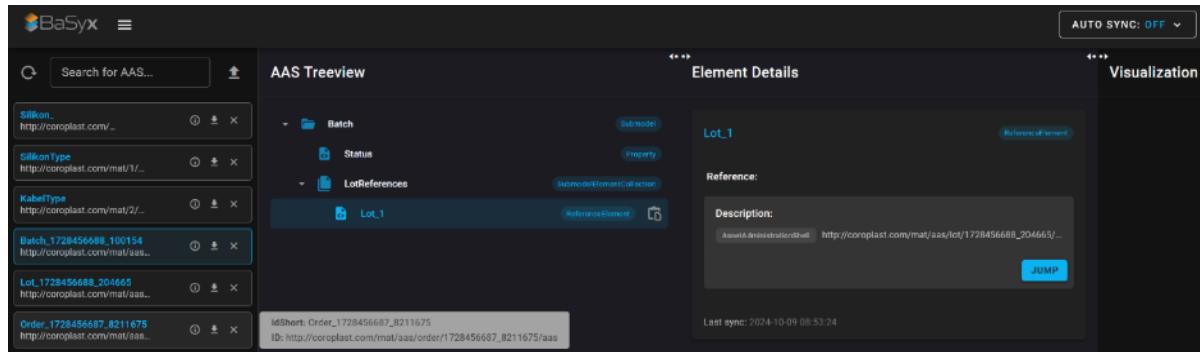
4.5.5.1.2 Anwendungsbeispiel 1

Durch Erstellung einer Order mit verzeigertem Batch mit verzeigertem Lot mit Verweisen auf ein erstelltes Produkt wurde prototypisch die Produktionsplanung und das Speichern der Ergebnisse simuliert. Alle Produktionsschritte sind geloggt und im Lot in den Runs hinterlegt. Abbildung 4-67 zeigt, wie die VWS vom Lot in BaSyst aussieht. Hier ist eine Referenz auf eine VWS mit Informationen des entsprechenden Batches zu finden. Theoretisch können hier beliebig viele Batches hinterlegt sein. In der Abbildung 4-66 ist eine VWS mit dem Batch zu sehen, in welchem wiederrum ein Verweis auf das Lot zu finden ist. Die Abbildung 4-67 zeigt das Lot mit einem Submodell für allgemeine Informationen, aber auch ein Submodell für die Runs. In Jedem Run sind alle Prozess-Steps abgebildet, die sich auch in dem Typ wiederfinden, aber nun mit realen Werten aus der Produktion.



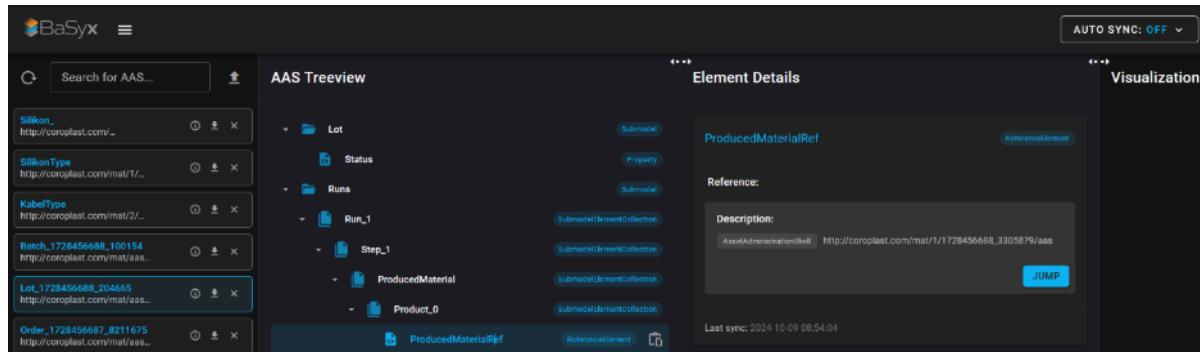
The screenshot shows the BaSyx AAS Treeview interface. On the left, the treeview displays various elements under the 'Order' category, including 'Status', 'ProductTypeReference', and 'BatchReferences'. Under 'BatchReferences', 'Batch_1' is selected, which is highlighted in blue. The 'Element Details' panel on the right shows 'Batch_1' with its reference information. The 'Visualization' panel on the far right is currently empty.

Abbildung 4-65: Darstellung des Produktionsauftrags in BaSyx



This screenshot shows the BaSyx AAS Treeview with the 'Lot' category selected in the treeview. 'Lot_1' is selected and highlighted in blue. The 'Element Details' panel shows 'Lot_1' with its reference information. The 'Visualization' panel is empty.

Abbildung 4-66: Darstellung des Batchs in BaSyx



The screenshot displays the BaSyx AAS Treeview with the 'Run' category selected. 'Run_1' is selected and highlighted in blue. The 'Element Details' panel shows 'ProducedMaterialRef' with its reference information. The 'Visualization' panel is empty.

Abbildung 4-67: Darstellung des Lots in BaSyx

4.5.5.1.3 Anwendungsbeispiel 2

Im zweiten Anwendungsbeispiel haben wir die Rückverfolgbarkeit über mehrere Lots hinweg simuliert. Dafür haben wir die Produktion von einem Kabel, wie in Abbildung 4-68 zu finden, realisiert. Entscheidend dafür sind zwei Lots. Eines, dass die Prozessschritte beinhaltet, die zur Herstellung von dem Silikon benötigt wurden. Der Einfachheit halber ist das hier mit einem Schritt dargestellt.

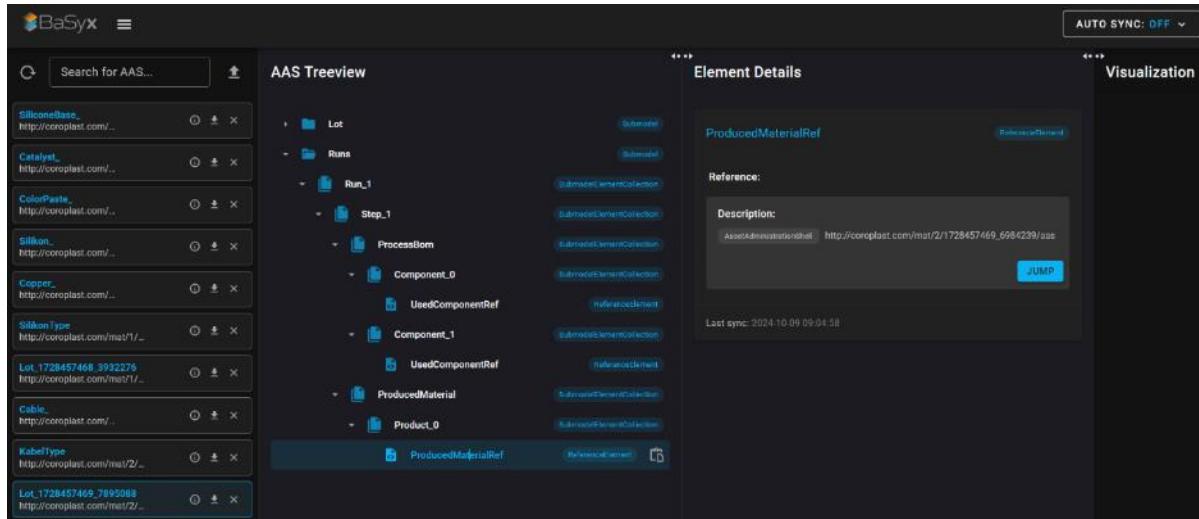


Abbildung 4-68: Darstellung des Lots zur Kabelproduktion

In der Abbildung 4-69 ist zu sehen, wie aus dem Silikon und einer weiteren Komponente schließlich ein Kabel hergestellt wurde. Informationen zur Chargennummer sind in der VWS von dem Produkt zu finden.

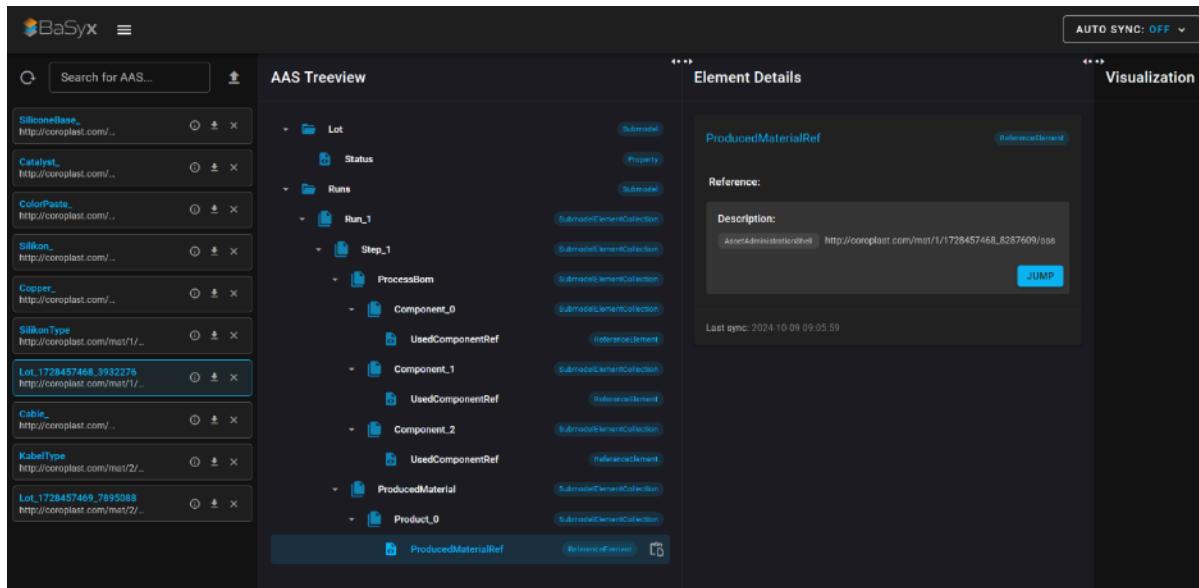


Abbildung 4-69: Darstellung des Lots zur Silikonproduktion

Stellt ein Nutzer also fest, dass bei seinem Kabel das Silikon gebrochen ist, so kann er die Chargennummer davon beim Hersteller angeben. Dieser kann schauen in welchem Lot dieses Kabel produziert wurde und anschließend die Chargennummer des Silikons herausfinden, dass verwendet wurde. Sobald diese bekannt ist, kann herausgefunden werden in welchem Lot das Silikon produziert wurde. Hier findet er alle relevanten Daten in den Prozessschritten, um nachvollziehen zu können, wie es zu der Schwäche im Material kam. Z.B. kann festgestellt werden, dass die Temperatur, während der Knetzeit lange zu hoch war, oder es kann aufgeklärt werden, dass die Materialschwäche aufgrund von fehlerhaften Komponenten entstanden ist.

4.5.5.1.4 Zusammenfassung und Ausblick

Diese prototypische Implementierung ermöglicht es uns, die Funktionalität und Effizienz des Konzepts in einer realitätsnahen Umgebung zu testen. Durch die Verwendung von BaSyx und standardisierten Protokollen wie OPC-UA stellen wir sicher, dass unser System interoperabel und zukunftssicher ist. Die Flexibilität bei den Datenquellen erlaubt eine einfache Integration in bestehende Produktionsumgebungen.

4.5.5.2 KOSTAL

KOSTAL Kontakt Systeme ist auf die Entwicklung und Produktion von Steckverbindern spezialisiert. Zur Validierung der in dieser Arbeit vorgestellten Prinzipien soll der Prozess der Steckverbinderherstellung dargestellt werden. Dabei soll vor allem überprüft werden, inwieweit dieses Konzept in der Praxis anwendbar ist.

Der zu bewertende Anwendungsfall befasst sich mit der Bereitstellung und Produktion der Materialien, die für die Herstellung eines Kontaktes erforderlich sind.

Wie in Abbildung 4-70 zu sehen ist, werden die Bänder für den Grundkörper und für die Lamelle meist von verschiedenen Lieferanten bezogen.

Für die Herstellung des Kontaktes ist es im nächsten Schritt notwendig, die Lamelle und den Grundkörper aus dem Band umzuformen sowie beide zusammen zu fügen. Beide Materialien sind in der Stückliste eindeutig beschrieben.

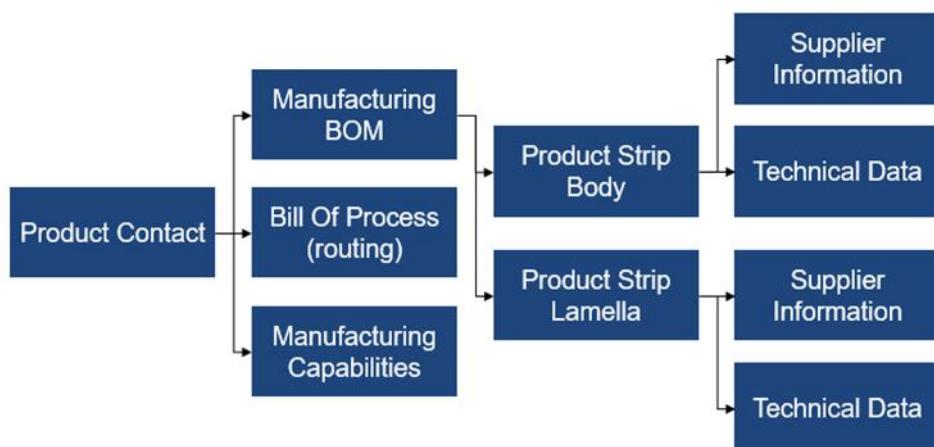


Abbildung 4-70: Repräsentation des Kabelherstellungsprozesses

Um mit der Validierung der Prinzipien fortzufahren, werden im Folgenden die Konzepte im Ökosystem Verwaltungsschale dargestellt. Nachfolgend sind die wichtigsten Materialien aufgelistet, wobei von den Lieferanten der Bänder nur die technischen Informationen und Lieferantendaten übermittelt werden.

Product Contact

- Asset Information
- Manufacturing BOM
- Bill Of Process (routing)
- Manufacturing Capabilities

Product Strip Body

- Asset Information
- Supplier Information
- Technical Data

Production Order

- Material Number + Revision Level + Charge
- Routing Specification
- Single Materials
- Equipment
- Packaging

Darüber hinaus ist die Ressource der Verwaltungsschale zugeordnet, die durch die technischen Spezifikationen und die Kapazitäten der jeweiligen Maschine dargestellt werden (PPS-Fertigungsauftrag, siehe Abbildung 4-71).

PPS-Fertigungsauftrag (BDE) (int.Nr)
Fertigungsauftrag
Original

KOSTAL

| | | | | | |
|--|--|--|--|------------------|-------------------------|
| Auftrags-Nr. (BPO) | 62617175 | Materialeinheit (S05) | 10140499100020 | Rev Stand | 02 |
|  3P062617175 | |  3OS10140499100020-02 | | | |
| Auftragsmenge | 720.000 ST | Auftragsart | PP06 | Werk | Materialeinheit |
| | | | | 0003 - K KKS | Stecker SLK 2,8 , - - - |
| Entstehungsdatum | Eck-Start | Eck-Ende | Fertigstellungszeitraum | Depotort | Reservierungsnr. |
| | 19.08.2024 | 21.08.2024 | KM0 | KM0 | 0138949480 |
| Status | | | Änderungsnr. Restbestand des Artikels | Charge | |
| FREI DRUC VOKL ABRV CHPF CHZG NMVP | | | REVLV | M02 | |
| Sequenznummer | 0000000000000000 | | | | |
| Routing | Stanzen 2-fach WZG 200041834 | | | | |
| Folge | 0 | | | | |
| Stammfolge | Folge | Beifolge | Aus Vorgang | Zu Vorgang | |
| Vorgang | 0010 | | Vg.-Menge 720.000 ST | Folge | 0 |
| Arbeitsplatz | Arbeitsplatzbezeichnung | | | | |
| 9623480 | Stanzautomaten, 80 to. | LA 480 | | | |
| Werk | Start | Ende | Rückverfolgungsnummer | | |
| 0003 | 19.08.2024 | 21.08.2024 | 0022757006 | | |
| v | In 1.000 | Lohngruppe | Kostenst. | Steuerschl. | Bemerkung |
| 3 H | 0,024 H | | 3032323 | LKPE | 1.000 |
| Beschreibung | Bandware stanzen 2- fach Werkzeug 200041834 Werkzeugvorschub 26 mm Werkzeughub 26 mm Auf Spule 7193900500000 zu 4.000 aufhaspeln, Spulenende festkleben. Etikettieren. | | | | |
| Flt. Schüssel | LK02 | | | | |
| HasMin. | 451 | | | | |
| KwMin | 2,000 | | | | |
| Komponenten für Vg. | 0010 | | Folge | 0 | |
| Pos.Nr. | Material | Rez. | Beschreibung | Menge | Charge |
| 0010 | 42112403420001 | 00 | Bd 0,32x 32 LK4112-21 CuSn4 R540 verzinn | 849,6 | KG |
| 0040 | 49980000250000 | 00 | Bd Tin cover. CuSn | 489,6 | KG |
| 0050 | 71951012000001 | 00 | Bd 125gr.. 39x 100 PAPER | 9.360 | M |
| Fertigungshilfsmittel für Vg. | 0010 | | Folge | 0 | |
| Pos.Nr. | Werk | Fertigungshilfsmittel | Menge | Technische Pktz: | |
| 1000 | 200041834 | Progression tool modular standard MH | 1 | 03-01-01-A | |
| 1010 | 200045279 | Progression tool modular standard MH | 1 | 04-05-04-C | |
| 1020 | 200045280 | Progression tool modular standard MH | 1 | 04-05-04-C | |
| 1030 | 200045281 | Progression tool modular standard MH | 1 | 04-05-04-C | |
| 1040 | 200045282 | Progression tool modular standard MH | 1 | 04-05-04-C | |
| 1050 | 200045283 | Progression tool modular standard MH | 1 | 04-05-04-C | |
| 1060 | 200045284 | Progression tool modular standard MH | 1 | 04-04-02-A | |
| 1070 | 200045285 | Progression tool modular standard MH | 1 | 04-04-02-A | |
| Packvorschrift | 50929 | | | | |
| HFlag | Material | | Menge | | |
| X | 7193905000000 | KSP02 Spule KSP02PS-Massivba Ox 0 | 1ST | | |
| | 71999999 | Referenzmaterial für Packvorschrift *ST* | 4.000ST | | |

Abbildung 4-71: Repräsentation des PPS-Fertigungsauftrages

Das Mapping eines Fertigungsauftrags in eine AAS wird in Abbildung 4-72 schematisch dargestellt. Es wird gezeigt an welchen Stellen die einzelnen Bestandteile eines Fertigungsauftrags im pdf-Format in einer AAS („KKS_Product_TERMINAL_SLK2.8_Type.aaxs“²⁷) hinterlegt sind.

²⁷ https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP03/AP3.5/KKS_Product_TERMINAL_SLK2.8_Type.aaxs

| Element Content | | Fertigungsauftrag | |
|--|--|---|--|
| Kontextreferenz | | Original | |
| AAS "TAB.SLK_2,8" [https://www.arena2036.de/aas/8074_8002_2042_7117] of [https://www.arena2036.de/asset/3174_8002_2042_7736] | | Werknummer (WZ) 62617175 | |
| SM "Product_BOM" [https://www.arena2036.de/sm/7374_8002_2042_4915] | | 10140499100020 02 | |
| BOM Bill of Material - Graph display ready | | Liefertermin (LIE) 19.08.2024 | |
| Ent "Strip" | | Lieferort (LIE) KMO KMO | |
| Ent "Paper" | | Referenznummer 0138649 | |
| SM "BillOfProcess" [https://www.arena2036.de/sm/3574_8002_2042_5469] | | Lieferanweisung E-Metallteile E-Metallteile | |
| SMC "Processes" (2 elements) | | Anlieferungszeitraum REVLY M02 | |
| SMC "PunchProcess01" (8 elements) @{SMT/Cardinality=ZeroToMany} | | Anlieferungszeitraum REVLY M02 | |
| Prop "ProcessId" @{SMT/Cardinality=One} | | Anlieferungszeitraum REVLY M02 | |
| Prop "ProcessName" = Stanzen 2 -fach WZG 200041834 @{SMT/Cardinality=One} | | Anlieferungszeitraum REVLY M02 | |
| MLP "ProcessDescription" → @{SMT/Cardinality=One} | | Anlieferungszeitraum REVLY M02 | |
| Prop "PlannedProcessTime" @{SMT/Cardinality=One} | | Anlieferungszeitraum REVLY M02 | |
| SMC "ProductParameters" (1 elements) @{SMT/Cardinality=One} | | Anlieferungszeitraum REVLY M02 | |
| SMC "Strip" (2 elements) | | Anlieferungszeitraum REVLY M02 | |
| Prop "Type" = www.coroplast.de/asset/7594_8002_2042_4617 | | Anlieferungszeitraum REVLY M02 | |
| Prop "CrossSectionArea" = 0,5 | | Anlieferungszeitraum REVLY M02 | |
| SMC "ProcessParameters" (2 elements) @{SMT/Cardinality=One} | | Anlieferungszeitraum REVLY M02 | |
| Prop "ToolFeed" = 26 mm | | Anlieferungszeitraum REVLY M02 | |
| Prop "ToolStroke" = 26mm | | Anlieferungszeitraum REVLY M02 | |
| SMC "ResourceParameters" (2 elements) @{SMT/Cardinality=One} | | Anlieferungszeitraum REVLY M02 | |
| Ref "MachineType" → [GlobalReference, 9623480 Stanzautomat] | | Anlieferungszeitraum REVLY M02 | |
| Ref "ToolType" → [GlobalReference, Werkzeug 200041834] | | Anlieferungszeitraum REVLY M02 | |
| SMC "ProcessBoM" (2 elements) @{SMT/Cardinality=One} | | Anlieferungszeitraum REVLY M02 | |
| Ent "Strip" | | Anlieferungszeitraum REVLY M02 | |
| Ent "Paper" | | Anlieferungszeitraum REVLY M02 | |
| SMC "PackagingProcess01" | | Anlieferungszeitraum REVLY M02 | |
| SMC "CapabilityRelationships" (1 elements) | | Anlieferungszeitraum REVLY M02 | |
| Rel "PunchCapability" | | Anlieferungszeitraum REVLY M02 | |
| SM "RequiredCapabilities" [https://www.arena2036.de/sm/7084_8002_2042_5] | | Anlieferungszeitraum REVLY M02 | |

Abbildung 4-72: Mapping der Daten eines beispielhaften Fertigungsauftrags in die VWS

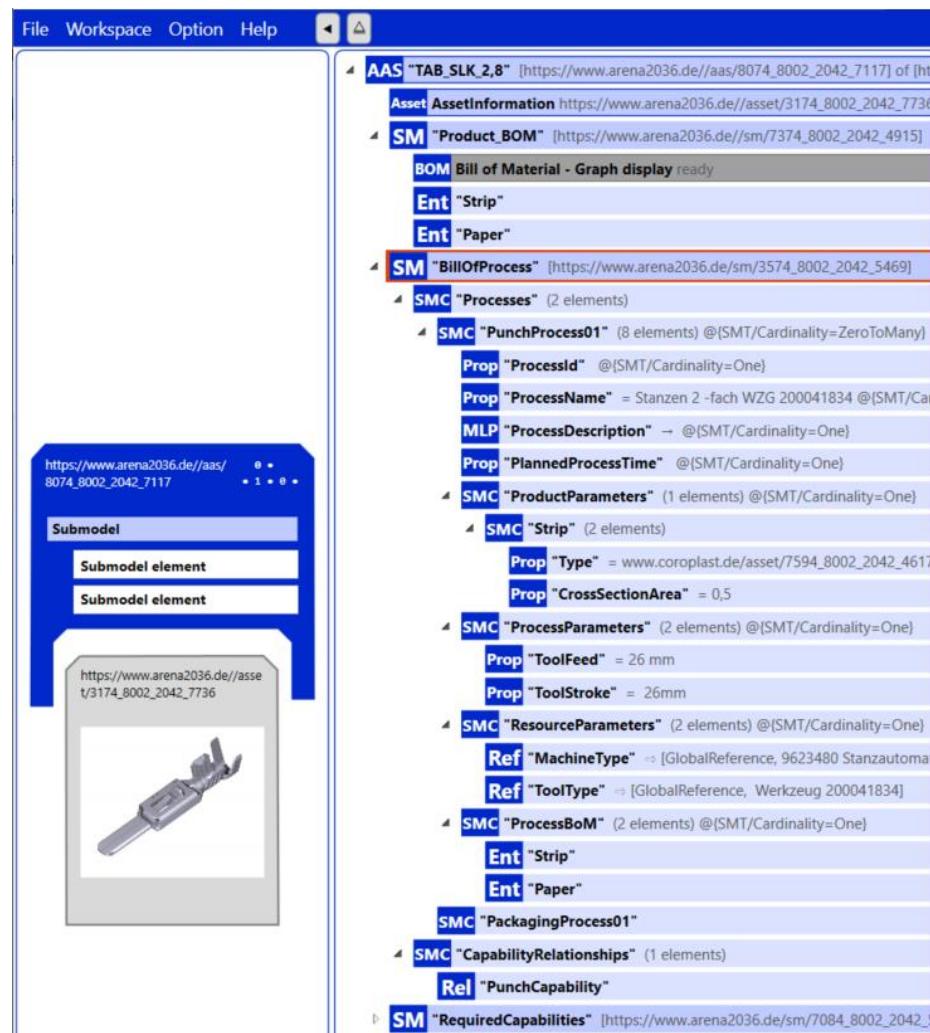


Abbildung 4-73: Repräsentation des Steckers (Terminal) der Verwaltungsschale

4.6 Fazit

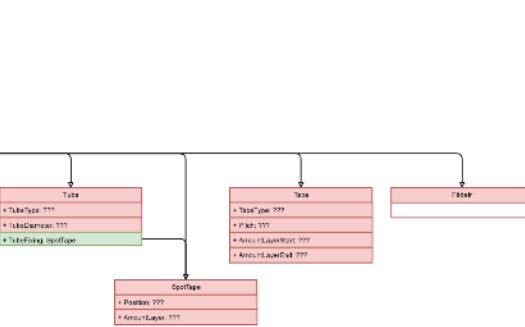
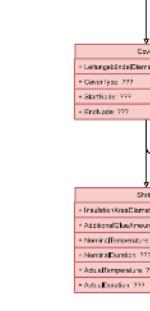
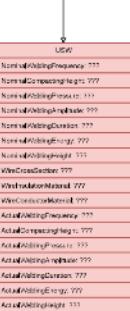
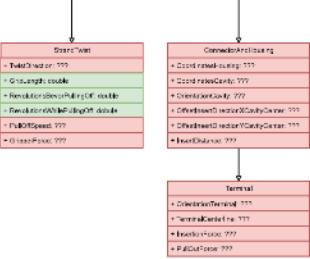
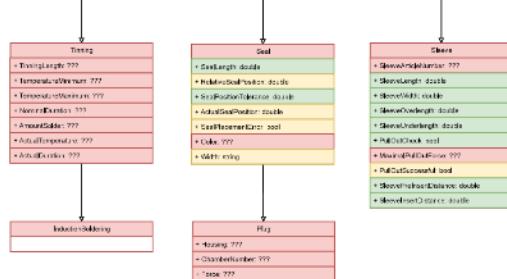
Im Rahmen des TP3 wurden verschiedene Aspekte beleuchtet, um die Produktionsprozesse für Leitungssätze zu automatisieren. Ein wichtiger Grundstein wurde mit der Prozessliste gelegt, die an verschiedenen Stellen im Gesamtprojekt genutzt werden konnte, um erstens eine Sammlung aller identifizierten Prozesse in einer konsolidierten Liste und zweitens gemeinsam mit den dazugehörigen Parameter in feingranularer Form für eine potenzielle Automatisierung zu haben.

Auf dieser Grundlage wurde anschließend ein Referenzmodell für die Produktionsprozesse erstellt und das VWS-Teilmodell „Bill of Process“ entwickelt sowie seine Anwendung beispielhaft demonstriert. Das Teilmodell enthält alle wesentlichen Strukturen, um die erforderlichen Daten aus dem Teilmodell auszulesen (z.B. Maschinenparameter) bzw. zurückzumelden (z.B. Prozessstatus). Somit dient es als zentrale Datendrehscheibe für alle Produktionsprozesse.

Ein weiterer wichtiger Baustein konnte mit der Entwicklung eines Data Mapping und Validation Service erreicht werden. Der Data Mapping Service stellt sicher, dass die Kompatibilität zwischen bestehenden Systemlandschaften und der VWS besteht, um in beide Richtungen kommunizieren zu können. Hierzu benötigt man einen Validation Service, um bspw. Namen, die Struktur und den Datentyp zu validieren, bevor Daten in die Ziel-VWS geschrieben werden. Im Ergebnis trägt dieser Service zur Vermeidung unvollständiger Daten, zur Steigerung der Datenqualität und zur Reduzierung des manuellen Aufwands bei.

Die Ergebnisse aus TP3 wurden abschließend validiert. An dieser Stelle stand die Anwendung der entwickelten Prinzipien für reale Anwendungsfälle im Vordergrund. Es konnte dadurch bspw. gezeigt werden, dass das „Bill of Process“ Teilmodell nicht nur für die Leitungssatzproduktion Anwendung finden kann, sondern auch bei der Produktherstellung von Zulieferteilen. Hierdurch konnte nachgewiesen werden, dass das Teilmodell eine generelle Anwendungsmöglichkeit für Prozesse in der Produktherstellung bietet.

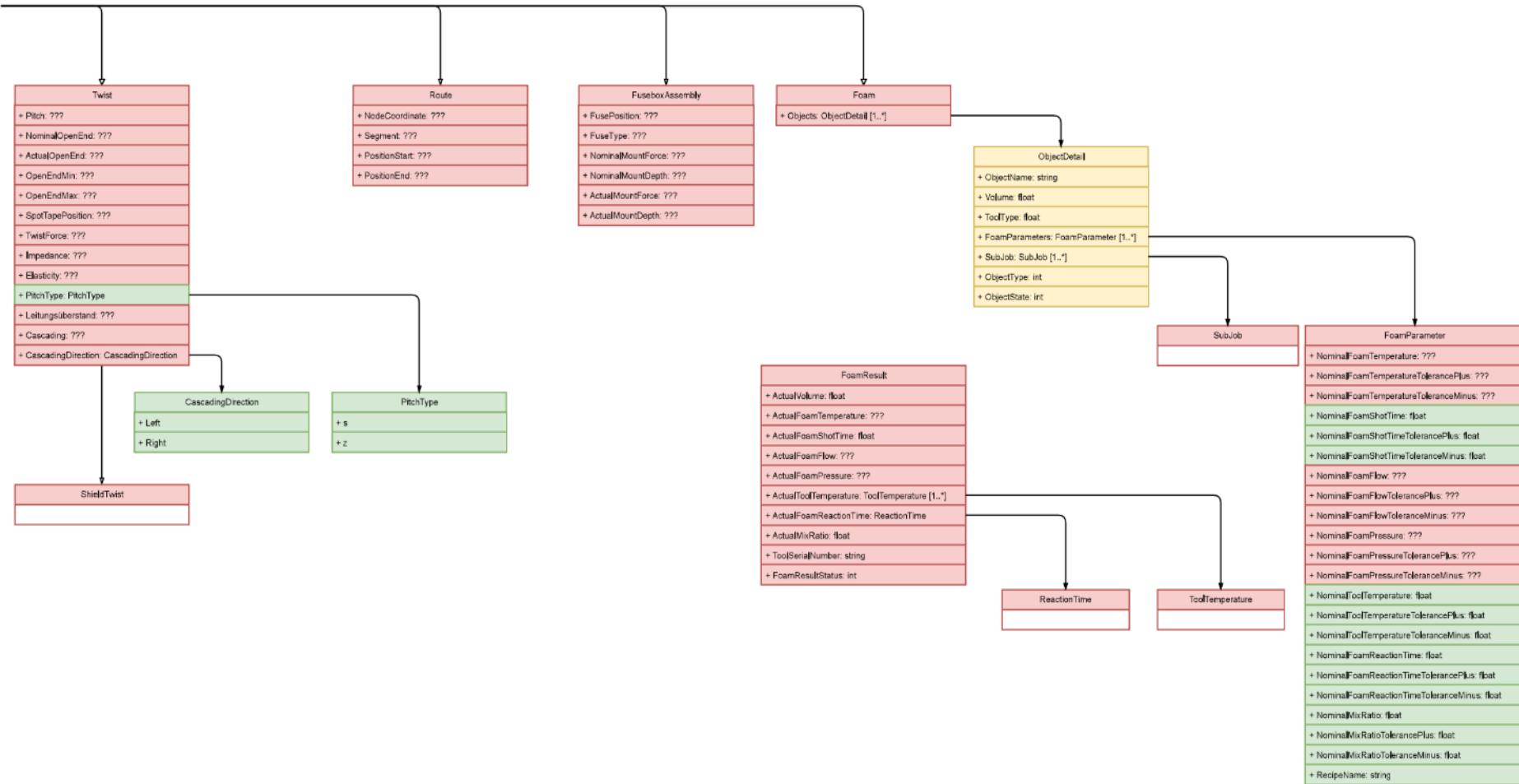
A

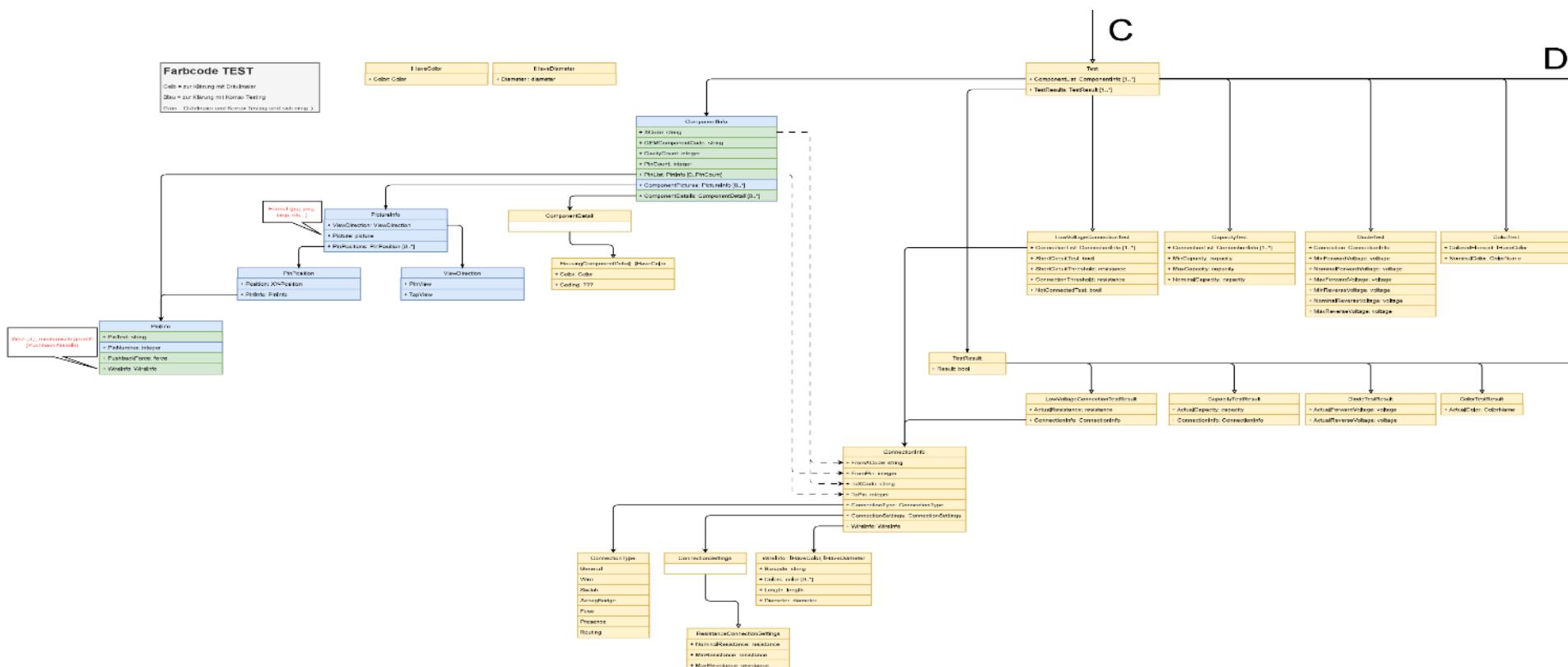


B

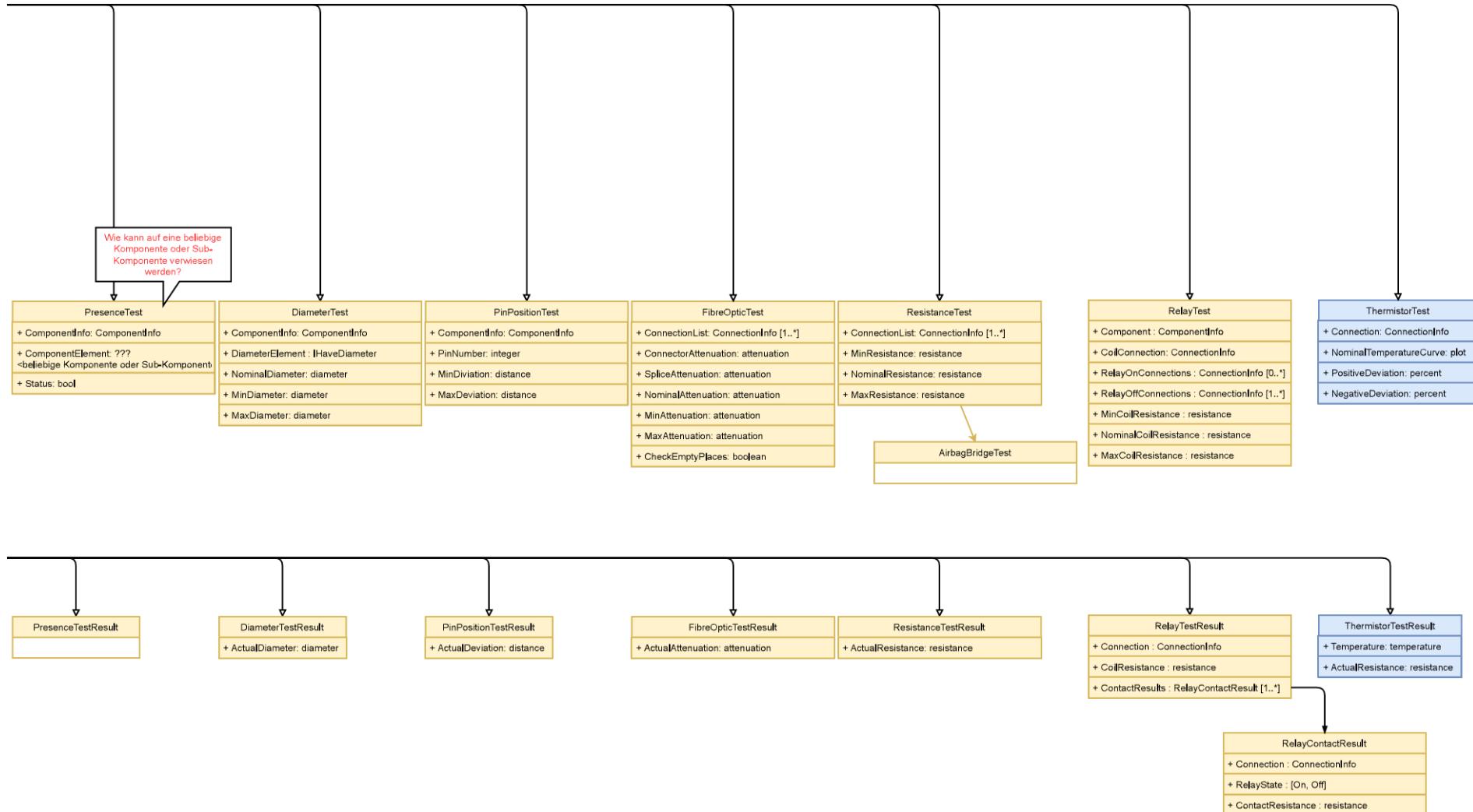


B





D



5 TP4 - Montage des Leitungssatzes

Im TP 4 "Montageprozesse des Leitungssatzes" wurden Möglichkeiten untersucht, die Montageprozesse beim OEM über Verwaltungsschalen als standardisierte Digitale Zwillinge abzubilden. Dazu wurden die einzelnen Teilprozesse der Montage eines Leitungssatzes in der Fahrzeugkarosserie analysiert, relevante Informationen abgeleitet und semantisch beschrieben, um diese in einem Teilmodell der Verwaltungsschale zu hinterlegen und für die Prozesse automatisiert abrufbar bereitzustellen.

Der Leitungssatz besitzt die Aufgabe, die Endverbraucher innerhalb eines Fahrzeugs miteinander zu verbinden und somit eine gezielte Ansteuerung einzelner Komponenten zu ermöglichen. Er ist ein Produkt, welches eine hohe Varianz in seinen Einzelteilen besitzt, kundespezifisch (Losgröße 1) produziert wird und schwer handzuhaben ist, da er keine eigene Steifigkeit in sich selbst besitzt.

Das Teilprojekt war in folgende Arbeitspakete aufgeteilt, deren Ergebnisse in diesem Dokument zusammenfassend erläutert werden:

- AP 4.1 - Verwaltungsschalen der relevanten Montage-Komponenten
- AP 4.2 - Auswahl der zu automatisierender Prozesse
- AP 4.3 - Dokumentation der Anforderungen zu diesen Prozessen
- AP 4.4 - Konzept zur Montage des Leitungssatzes

Das Zielbild insgesamt war, die automatisierte Montage des Leitungssatzes im Fahrzeug möglichst vollständig zu automatisieren. Das bedeutet, dass alle Datenbedarfe, wie bspw. Informationen zur Bahnplanung, Geometrie etc. semantisch beschrieben, digital bereitgestellt, abgerufen und bei Bedarf aktualisiert werden müssen. Als Startpunkt wurde die Anlieferung der Packtasche inklusive des Leitungssatzes an der Montagezelle beim OEM angenommen. Als Grundlage zur Ermittlung der Datenbedarfe für die Prozessschritte wird auf die [VDI2860](#) zurückgegriffen werden. Innerhalb der [VDI2860](#) werden grundlegende Handhabungs- und Montagetechniken untergliedert und beschrieben.

Von der Vorgehensweise her wurde der komplexe Montageprozess in automatisierbare Teilprozesse unterteilt, die in den weiteren Kapiteln einzeln betrachtet werden. Dabei soll der Fokus auf notwendige Informationsbedarfe und Fähigkeiten gelegt werden, die für die einzelnen Prozessschritte benötigt und anschließend in einem Submodell der Verwaltungsschale hinterlegt werden sollen.

Für die Vereinfachung des Prozesses werden aufgrund der Komplexität einige Annahmen getroffen, die im folgenden Kapitel erläutert und anschließend in den einzelnen Kapiteln berücksichtigt werden.

5.1 Definition der Randbedingungen

In diesem Kapitel sollen einige Randbedingungen definiert werden, um einen einheitlichen Handlungsrahmen für die Analyse jedes Teilprozesses zu erzielen.

Da eine robotische Anwendung nicht den vollständigen Montagevorgang binnen eines Prozessschrittes durchführen kann, wird der gesamte Prozess in mehrere Teilprozesse untergliedert. Für die Umsetzung der entsprechenden Prozesse wird ein weiterer Detailierungsgrad in Grob- und Feinauslegung durchgeführt.

- **Grobauslegung:** Das Einbringen, öffnen und grobe Auslegen des Leitungssatzstrangs in der Karosserie. Die richtige Ausrichtung von Clipsen und Steckern, spielen in diesem Prozessschritt noch eine untergeordnete Rolle
- **Feinauslegung:** Fokus wie die einzelnen Clipse und Stecker richtig orientiert und formschlüssig in das Gegenstück oder in die Montagevorrichtung gebracht wird.



Abbildung 5-1 Maße der Packtasche eines Leitungssatzes

In Abbildung 5-1 ist eine Packtasche zusehen, die verwendet wird, um den Leitungssatz nach der Produktion beim Konfektionär zum OEM an das Montageband zu liefern. Die Packtasche (in der Realität gibt es unterschiedliche) und eine vorgegebene Packanleitung für den Leitungssatz wird in der Branche vorgegeben und unterscheidet sich je nach OEM. Um einige Basisparameter (Abmessungen und Öffnungsmechanismus) zu identifizieren, wird die Packtasche als Referenz herangezogen und analysiert. Diese Angaben sind für die Auslegung des Arbeitsraums und des später zu verwendenden Roboters notwendig. Für das vorliegende Beispiel wird eine Höhe von 50 cm, eine Breite von 110cm und eine Tiefe von 75 cm angenommen. Der Leitungssatz wird mit einem Gewicht von 60 Kg (inkl. Packtasche) angenommen. Der abgebildete Öffnungsmechanismus ist ein Klinkenverschluss und somit herausfordernd für den Roboter diesen zu betätigen. Um eine erste Vereinfachung anzunehmen, wird davon ausgegangen, dass es sich beim Öffnungsmechanismus der Packtasche um einen Klettverschluss handelt, da dies mehrere Angriffspunkte für den Roboter bietet.

Die Reihenfolge, wie die einzelnen Stränge des Leitungssatzes aus der Packtasche entnommen werden müssen, ist durch die Packanleitung vorgegeben. Eine farbliche Markierung am Ende der einzelnen Stränge macht eine Unterscheidung möglich.

In Abbildung 5-2 sind die Abmessungen der Karosserie dargestellt, die für das Einbringen der Packtasche in das Fahrzeug betrachtet werden.

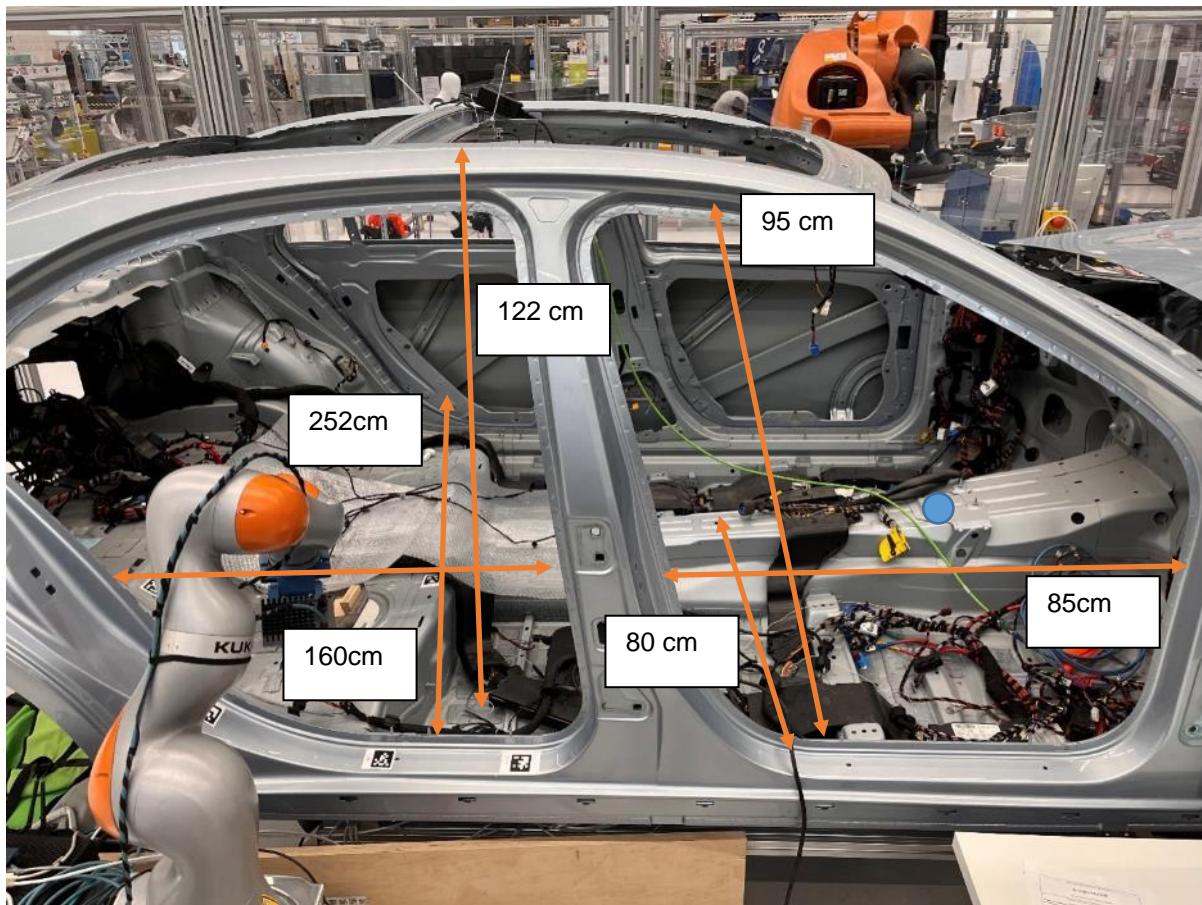


Abbildung 5-2 Maße der Karosserie (S-Klasse)

Die folgenden Maße für die Karosserie und deren Öffnung wurden gemessen und werden für die weitere Betrachtung verwendet.

Karosserieinnenraum:

- *Länge: 252cm*
- *Breite: 160cm*
- *Höhe: 122cm*

Türausschnitt:

- *Breite: 85cm*
- *Höhe: 95cm*

Nullpunkt:

Um einen grundsätzlichen Nullpunkt festzulegen, wird die geometrische Lage des Schaltknaufs als 0-Punktkoordinatensystem innerhalb der Karosserie verwendet (gelber Punkt Bild 2).

5.2 Einbringen der LS-Tasche in die Karosserie

Für eine typische Montagezelle wie in *Abbildung 5-3* schematisch dargestellt, werden die notwendigen Szenarien analysiert. Zum einen müssen die relativen Positionen der einzelnen Prozessteilnehmer (Roboter, Karosserie und Packtasche) bekannt sein. Hierzu müssen die jeweiligen Koordinatensysteme der bestehenden Teilnehmer bekannt sein, damit die relativen Positionen errechnet werden können. Hierfür wird angenommen, dass wie bereits in Kapitel 2 erwähnte, das Koordinatensystem am Schaltknauf das absolute Koordinatensystem ist. Die weiteren Prozessteilnehmer sind relativ zu diesem Punkt zu sehen und in der Lage die Bahnplanung entsprechend zu berechnen.

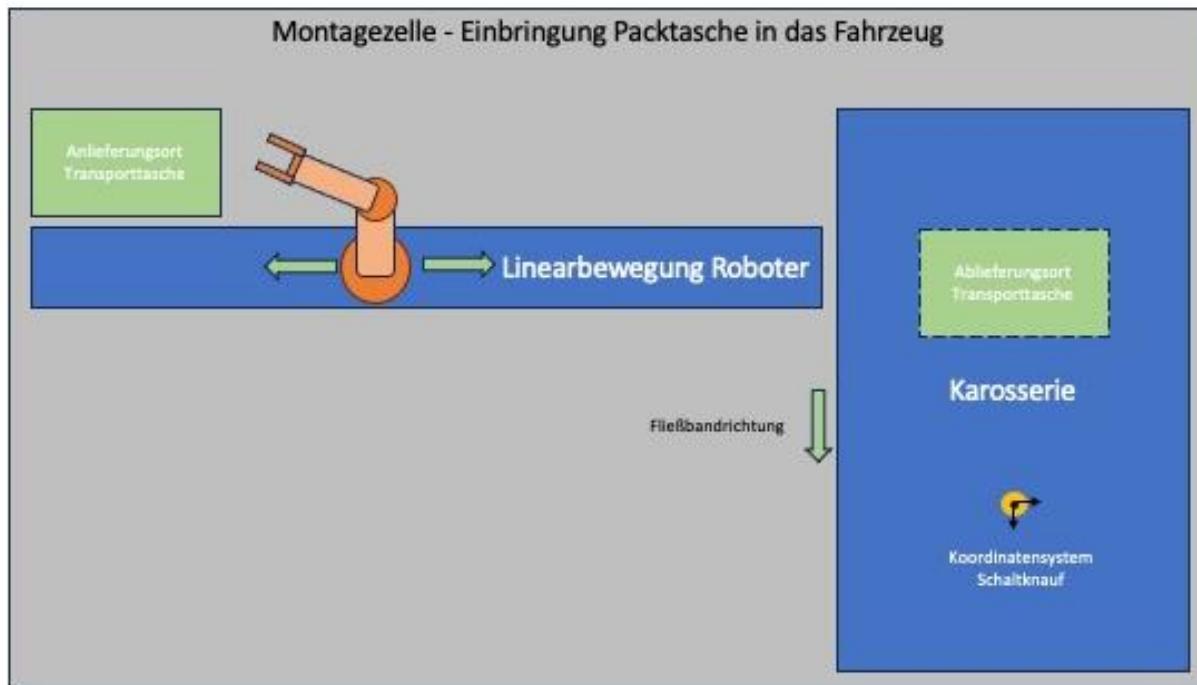


Abbildung 5-3 Montagezelle für das Einbringen des Leitungssatzes

Die Teilprozesse, die mit dem Einbringen der Packtasche in die Karosserie einhergehen, sind:

1. Das Anfahren des Anlieferungspunktes
2. Die Grobausrichtung der Packtasche im Innenraum
3. Ablage der Transporttasche innerhalb der Karosserie

5.2.1 Das Anfahren des Anlieferungspunktes

Der Montageprozess wird durch die Anlieferung der Packtasche ausgelöst, hierbei wird ein Endschalter betätigt, der signalisiert, dass die Packtasche an einem definierten Ort innerhalb der Montagezelle abgelegt wurde. Somit kann der Roboter, der sich auf einer Konstruktion befindet, die es ihm erlaubt, den Weg vom Anlieferungspunkt zur Karosserie zu überbrücken, mit dem Anfahrprozess beginnen. Die Kontrolle, ob es sich hierbei um die „richtige“ Packtasche mit dem richtigen Leitungssatz handelt, wird bereits bei der Anlieferung an das Montagewerk überprüft, und muss somit nicht erneut überprüft werden, d.h. die Anlieferung der Komponente Leitungssatz erfolgt nach der Just-in-Sequence Methode (JIS)²⁸.

Nachdem das Startsignal ausgelöst wurde, fährt der Roboter mit entsprechenden Greifern die Henkel der Packtasche an. Für die Bahnplanung zu den Transporthenkeln werden einige Inputparameter benötigt. Grundsätzlich muss überlegt werden, ob für die Bahnplanung statische Parameter hinterlegt, sein müssen, oder diese automatisiert durch ein Kamerasytem erkannt und berechnet werden oder ob eine Kombination von Beidem sinnvoll ist. Beispielsweise können statische Parameter für die Anfahr-, Transport- und Abbremsgeschwindigkeit festgelegt werden und das Erkennen von Position und Orientierung der Transporthenkel (welche formstabil ausgeführt sein müssen), kann durch ein Kamerasytem erkannt werden. Wird davon ausgegangen, dass ein Kamerasytem auf dem Endeffektor

²⁸ <https://de.wikipedia.org/wiki/Just-in-sequence-Produktion>

montiert ist, welches einen Abgleich von CAD-Daten und der Ist-Situation ermöglicht, können formstabile Transporthenkel erkannt, und ein dynamisches Anfahren umgesetzt werden. Hierfür sind spezielle Greifer zu entwickeln, die in der Lage sind die formstabilen Transporthenkel zu greifen. Eine beispielhafte Darstellung ist in Abbildung 5-4 dargestellt. Beide Ausprägungen sind starr, durch die Form der unteren Ausprägung, kann das Einfädeln der Transporthenkel jedoch vereinfacht werden.

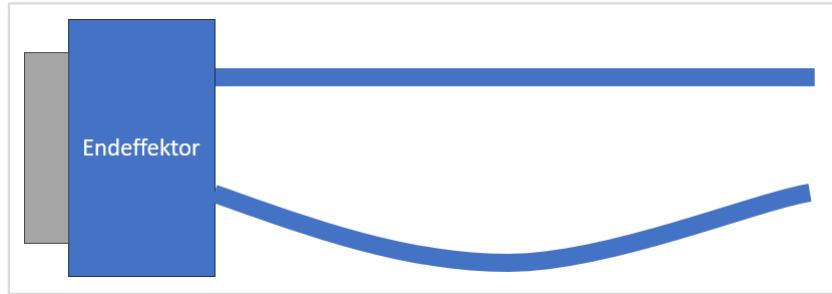


Abbildung 5-4 Greifer zum Greifen der Transportlaschen

Datenbedarfe: Für das Greifen der Transporthenkel an der Packtasche müssen die geometrischen Abmessungen ($hbxt$) von Packtasche und der explizite Greipunkt der Transporthenkel durch Koordinaten oder Bildgebung beschrieben bzw. erkennbar sein. Für die Bahnplanung werden Input-Parameter wie Beschleunigung, Fahrgeschwindigkeit, Orientierung und Positionierung des Tool Center Point (TCP) benötigt. Die CAD-Modelle müssen kinematische Daten und Störkonturen bereitstellen.

Fähigkeiten: Positionieren (Beschreibung der kartesischen Koordinaten einer Komponente im Raum), Orientierung (Beschreibung der rotatorischen Achsen im Raum einer Komponente), Greifen (Ein Produkt mit definierten Maßen durch einen Roboter neu zu arretieren oder positionieren zu können, bspw. anheben) und Greipunktermittlung.

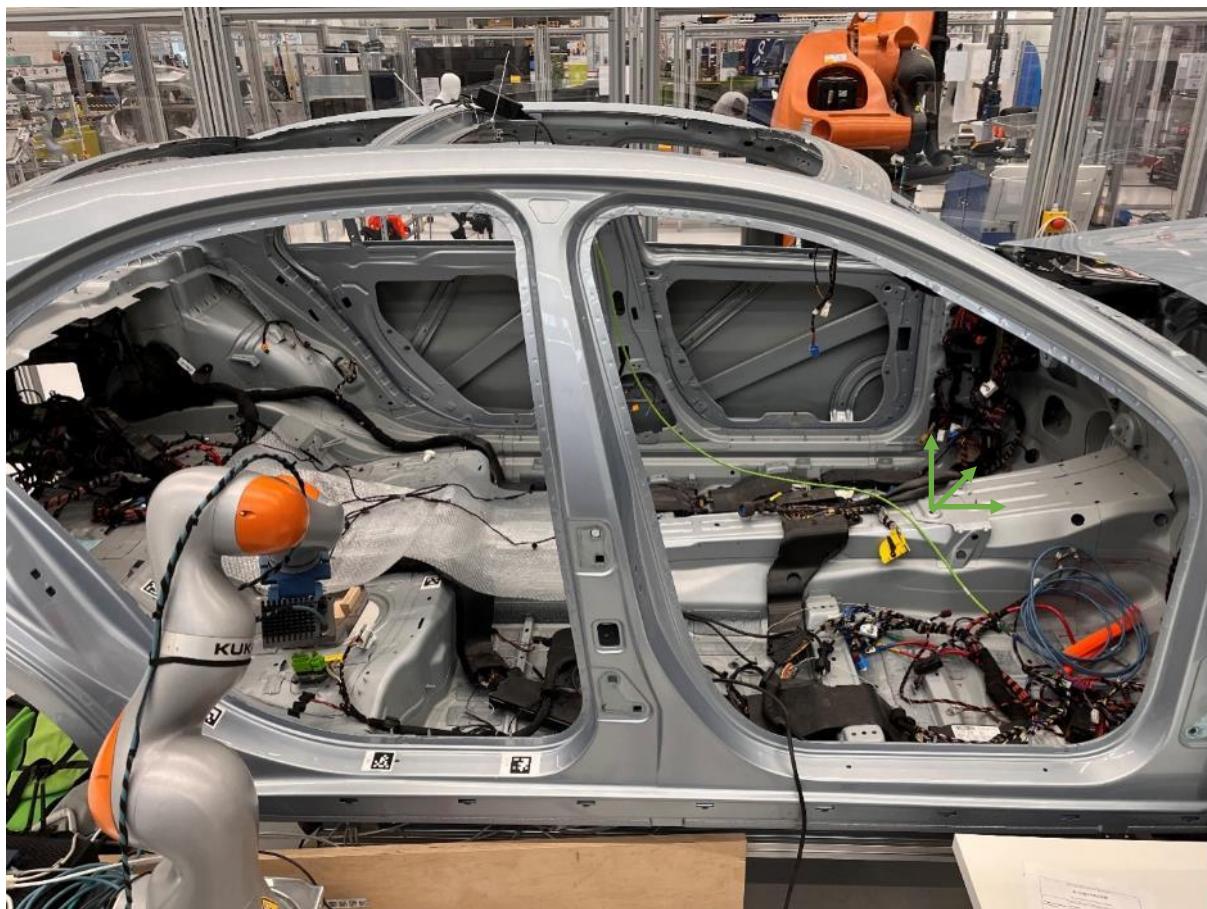


Abbildung 5-5: Hauptkoordinatensystem der Montagezelle

Nachdem der Greifer die Transporthenkel der Tasche gegriffen hat, kann das Anheben der Packtasche beginnen. Hierfür werden die Parameter für die Bahnplanung erneut benötigt. Um eine Kollision beim Einbringen der Packtasche in die Karosserie zu vermeiden, wird die Packtasche in einer mittig zur Türöffnung ausgerichteten Bahn in die Karosserie transportiert. Hierfür kann bspw. das CAD-Modell der Karosserie und die Geometrien der Türöffnung in Verbindung mit dem 0-Punktkoordinatensystem verwendet werden. Für die Ausführung des Prozesses spielen einige Produktstammdaten (Abmessungen (Packtasche und Karosserie, Gewicht etc.) und Maximalwerte für den Transportparameter (Geschwindigkeiten) eine Rolle.

Datenbedarfe: Für die Bahnausführung werden gewisse Input Parameter wie Beschleunigen, Fahrgeschwindigkeit, Orientierung und Positionierung des TCP benötigt. Des Weiteren Kinematische Daten, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasystem).

Fähigkeiten: Führen (Das Durchführen eines Produktes durch eine Öffnung in der Karosserie von zwei verschiedenen Seiten), Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren (Beschreibung der rotatorischen Achsen im Raum einer Komponente.)

Nun ist das Anheben und die Positionierung relativ zur Karosserieöffnung durchgeführt, und der Roboter kann die Packtasche in den Karosserieinnenraum fahren.

5.2.2 Grobausrichtung der Transporttasche im Innenraum

Die Packtasche befindet sich nun im Innenraum des Fahrzeugs und der Roboter soll die Packtasche an einem vorgegebenen Ort innerhalb der Karosserie absetzen. Für die Ausführung dieses Prozesses müssen die Zielkoordinaten des Abstellortes, mögliche Störkonturen, sowie die Input Parameter für die Bahnplanung bekannt sein.

Datenbedarfe: Bahnplanung (Zielkoordinaten des Ablageorts, Input Parameter wie Beschleunigen, Fahrgeschwindigkeit, Orientierung und Positionierung des TCP benötigt). Kinematische Daten, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasystem).

Fähigkeit: Führen (Das Durchführen eines Produktes durch eine Öffnung in der Karosserie von zwei verschiedenen Seiten) Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren (Beschreibung der rotatorischen Achsen im Raum einer Komponente.)

5.2.3 Ablage der Transporttasche innerhalb der Karosserie

Die Packtasche wird nun an den Zielkoordinaten abgesetzt und der Roboter fährt ohne eine Kollision kontrolliert aus der Karosserie heraus und begibt sich in die Eingangsposition.

Datenbedarfe: Bahnplanung (Zielkoordinaten Eingangsposition, Input Parameter wie Beschleunigen, Fahrgeschwindigkeit, Orientierung und Positionierung des TCP benötigt). Kinematische Daten, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasystem).

Fähigkeiten: Führen (Das Durchführen eines Produktes durch eine Öffnung in der Karosserie von zwei verschiedenen Seiten) Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren (Beschreibung der rotatorischen Achsen im Raum einer Komponente.).

5.3 Grobauslegung des Leitungssatz

In diesem Kapitel soll sich der Grobauslegung des Leitungssatzes im Fahrzeug gewidmet werden. Der Startzustand sieht wie folgt aus, die LS-Tasche liegt in der Karosserie muss jedoch, bevor die Auslegung durchgeführt werden kann, noch geöffnet werden. Es wird angenommen, dass es sich bei dem Verschluss um einen Klettverschluss handelt, der erkannt und entsprechend geöffnet werden muss. Die einzelnen Stränge sind durch eine Markierung erkennbar, die eine Aussage geben, ob der Strang in Richtung des Motors oder des Kofferraums gelegt werden muss und auf welcher Seite (Fahrer/Beifahrer) der Strang abgelegt werden muss. Es wird angenommen, dass ein zweiter Roboter für die Tätigkeiten innerhalb der Karosserie notwendig ist, da die Anforderungen an Wendbarkeit und Traglast von einander abweichen.

5.3.1 Transporttasche Öffnen

Die Tasche liegt durch einen Klettverschluss verschlossen in der Mitte der Karosserie. Um den Verschluss anfahren zu können, müssen die Verschlüsse für den Roboter erkennbar (bspw. durch Apriltag oder Datamatrix als visuelles Referenzsystem) und die notwendigen Informationen zur Öffnung bekannt sein. Dies können Bewegungsparameter und -informationen sein, in welcher Richtung der Greifer zur Öffnung des Verschlusses bewegt werden muss, o.ä. Der Prozess kann je nach Anzahl der Verschlüsse mehrmals wiederholt werden, dies soll bei der Grundsätzlichen Beschreibung jedoch keine Rolle spielen. Somit können folgende Informationsbedarfe und Fähigkeiten abgeleitet werden:

Datenbedarfe: Zielkoordinaten, Bahnplanung (Input Parameter wie Beschleunigen, Fahrgeschwindigkeit, Orientierung und Positionierung des TCP benötigt). Kinematische Daten, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasytem), Greifpunkttermittlung

Fähigkeiten: Führen (Das Führen des Klettverschlusses zum Öffnen der Tasche), Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren (Beschreibung der rotatorischen Achsen im Raum einer Komponente.)

5.3.2 Leitungssatzstränge im Innenraum auslegen

Nach der Öffnung der Transporttasche liegen die einzelnen Stränge entsprechend einer Packanleitung in der geöffneten Packtasche. Das Entfernen der Packtasche kann erstmal vernachlässigt werden, da der Fokus auf dem Auslegen der einzelnen Stränge liegt. Nun muss ein Abgleich mit den möglichen Erkennungsmerkmalen der einzelnen Stränge und dem jeweiligen Zielort umgesetzt werden. Es wird davon ausgegangen, dass die Stränge in wiederum einzelnen Schläuchen verpackt sind, die ein verknoten der einzelnen Abgänge verhindern und somit das Auslegen vereinfachen soll. Die Informationen, an welchen Zielort die einzelnen Stränge gezogen werden müssen, müssen durch ein übergeordnetes System zur Verfügung gestellt werden. Diese sind kein Bestandteil der Produktverwaltungsschale. Der Greifer fährt nun die einzelnen Stränge an, muss die einzelnen Greifpunkte der jeweiligen Stränge identifizieren und diese in die Zielposition und -orientierung bringen. Grundsätzlich soll das Vorgehen auch für weitere Auslegeprozesse angewendet werden können, bspw. das Verschieben von Kanälen im Cockpitinnenraum. Bei der Auslegung in Richtung des Kofferraums, müssen aufgrund der fehlenden Metallwand im Bereich der Rücksitze die Überzieher der ausgelegten Stränge nach der Grobauslegung bereits entfernt werden.

Datenbedarfe: Zielkoordinaten, Bahnplanung (Input Parameter wie Beschleunigen, Fahrgeschwindigkeit, Orientierung und Positionierung des TCP benötigt). Kinematische Daten, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasytem), Greifpunkttermittlung, Produktspezifikationen

Fähigkeiten: Führen (Das Durchführen eines Produktes durch eine Öffnung in der Karosserie von zwei verschiedenen Seiten), Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren (Beschreibung der rotatorischen Achsen im Raum einer Komponente.)

5.3.3 Leitungssatz am Fahrzeugboden montieren

Nach Fertigstellung der Grobauslegung, wird mit der Befestigung der Hauptverlegewege des Leitungssatzes im Fahrzeuginnenraum begonnen.

Datenbedarfe: Zielkoordinaten (zum Stecken der Befestigungselemente), Bahnplanung (Input Parameter wie Beschleunigen, Fahrgeschwindigkeit, Orientierung und Positionierung des TCP benötigt). Kinematische Daten, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasytem), Greifpunkttermittlung, Produktspezifikationen (bspw. Abmessung der jeweiligen Produkte)

Fähigkeiten: Führen (Stecken der einzelnen Befestigungselemente in die in der Karosserie vorgesehenen Befestigungselemente), Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren (Beschreibung der rotatorischen Achsen im Raum einer Komponente.).

5.4 Durchführung von Fädelprozessen

Unter Fädeln wird üblicherweise das Durchführen von Leitungssatzsträngen durch Karosseriedurchbrüche verstanden.

Grundsätzlich kann beim Fädeln zwischen Ziehen und Schieben unterscheiden werden. Für einen „einarmigen“ Roboter sind beide Techniken problematisch. Das Ende des Leitungsstranges mit dem jeweiligen Steckergehäuse bis zu dem Durchbruch bspw. in Richtung des Motorraums zu manövrieren, ist grundsätzlich kein Problem. Herausfordernd werden die zur Seite abzweigenden Leitungsstränge während des fädeln durch die Öffnung in Richtung des Motorraums. Die Stecker können am Durchbruch verkanten und zu einer Beschädigung des Leitungssatzes und Funktionsverlust führen.

Möglicher Lösungsansatz:

- Die Leitungsstränge, welche durch Karosseriedurchbrüche geführt werden, sind bereits in verschiedenenfarbige Packsäcke verpackt

In diesem Dokument soll vermehrt auf die technische Umsetzbarkeit und die dafür notwendigen Informationen bzw. Parameter eingegangen werden. Wird davon ausgegangen, dass der Leitungssatz im Cockpitraum vollständig durch den vorherigen Prozessschritt „Grobauslegung“ ausgebretet wurde, muss sich nun der Umsetzung des Fädelprozesses gewidmet werden. Das bedeutet, dass sich im nächsten Schritt der Erschließung der jeweiligen Leitungsstrangenden und der jeweiligen Angriffspunkte gewidmet werden muss.

5.4.1 Lokalisieren des Einzelstrangs

Für die Lokalisierung des zu greifenden Leitungsendes muss das Kamerasystem auf dem Greifarm den richtigen Leitungsstrang erkennen. Das wäre durch eine Erkennung des jeweiligen Produktes oder ein Vision System umsetzbar. Hierbei wird ein Abgleich mit der durch das Kamerasystem erzeugten Punktwolke und dem originalen CAD-Modell in der richtigen Position und Orientierung durchgeführt. Die grobe Positionierung des jeweiligen Leitungsendes ist bereits durch die Grobauslegen bekannt.

Datenbedarf: *Bündelquerschnitt bzw. Bündeldurchmesser, Masse des Teilleitungsstrangs, Angriffspunkte der jeweiligen Leitungen, Zielkoordinaten des Durchbruchs, Durchmesser der Karosserieöffnung, Bahnplanung*

Fähigkeiten: *Führen (Das Durchführen eines Produktes durch eine Öffnung in der Karosserie von zwei verschiedenen Seiten), Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren (Beschreibung der rotatorischen Achsen im Raum einer Komponente.), Greifpunktermittlung, Greifen*

5.4.2 Greifen des Einzelstrangs

Das Greifen und schlussendlich anheben, kann nicht an jeder Stelle entlang des Stranges durchgeführt werden. Die Angriffspunkte müssen optisch oder durch definierte Koordinaten kenntlich gemacht werden, damit der Roboter nicht an Gehäusen oder anderen unvorteilhaften Positionen angreift, diese anhebt und schlussendlich durch die Aufgebrachte Zugkraft gesteckte Pins löst oder Gehäuse zerstört. Die Ermittlung des Greifpunktes muss durch den Abgleich zwischen Punktwolke und CAD-Modell geschehen.

1. Der Greifer fährt zum definierten Aufenthaltsort und erkennt dort optisch den gesuchten Leitungsstrang. Dies kann durch unterschiedliche farbige Netzsäume unterstützt werden.
2. Greifen des Stranges in einem Sicherheitsabstand hinter dem Gehäuse

Datenbedarf: *Produktstammdaten (Durchmesser, Gewicht), Greifkoordinaten*

Fähigkeiten: *Greifpunktermittlung, Greifen (Ein Produkt mit definierten Maßen durch einen Roboter neu zu arretieren oder positionieren zu können (bspw. anheben))*

5.4.3 Leitungsende zum Durchbruch bewegen

Die grobe Position eines Durchbruchs ist durch die 3D-Daten der Karosserie bekannt. Der Roboter bewegt sich mit dem geöffneten Leitungssende und einem Sicherheitsabstand zur Karosserie in Richtung des Durchbruchs. Die genaue Position der Durchbruchsmitte wird durch den Abgleich der CAD-Datei und des Kamerasystems am Endeffektor korrigiert. Der herbeizuführende Zielzustand muss eine möglichst mittige und rechtwinklige Ausrichtung des Leitungsstranges vor dem Durchbruch herbeiführen. Dazu ggf. Position und Winkel des Greifarmes korrigieren, um das Gehäuse mittig vor dem Durchbruch zu positionieren und das Ende des Leitungsstranges annähernd senkrecht zur Oberfläche der Trennwand auszurichten.

Datenbedarf: Mittelpunkt des Bündels, Mittelpunkt des Durchbruchs, Kinematische Informationen, Bahnplanung, Zielkoordinaten, Produktstammdaten (Geometrische Informationen des Strangs), Bahnplanung

Fähigkeiten: Greifen, Greifpunkttermittlung, Bahnplanung

5.4.4 Leitungsende einfädeln

Der Fädelprozess kann in der Regel nicht von einem einzelnen Greifarm bewältigt werden, da die beiden beteiligten Bauräume (Cockpit und bspw. Heck- oder Motorraum) durch eine Trennwand voneinander separiert sind. Es muss somit ein zweiter Roboter das Ende des Leitungsstranges auf der anderen Seite des Durchbruchs annehmen und durch eine aufgebrachte Zugkraft in eine vorgegebene Position bringen und anschließend ablegen. Wichtig hierbei ist, dass die Zugkraft nicht größer als die Schubkraft sein darf, sonst würde der Greifer inkl. Strang innerhalb der Karosserie in Richtung der Karosseriewand „gezogen“ werden.

Während des Greifens, ist es wichtig, dass der Leitungsstrang nicht an einem Steckergehäuse gegriffen und anschließend daran gezogen wird. Dies könnte dazu führen, dass mechanische Verbindung von Pins und Stecker oder anderen Verbindungen gelöst, eine Beschädigung und somit ein Fehlerfall produziert wird. Der Leitungsstrang inklusive des Überzugs muss durch das Kamerasytem auf dem Greifarm lokalisiert werden. Der Strang muss in ausreichendem Abstand hinter dem Strangende gegriffen werden. Das Ziehen muss langsam erfolgen, dabei muss permanent die Schub- und Zugkraft bzw. der Widerstand überwacht werden. Das Ende des Prozesses, muss durch eine Überwachung des zurückgelegten Weges erfolgen. Das Ende des Stranges (außerhalb des Cockpits) wird an einer definierten Position abgelegt.

Datenbedarf: Zielposition des Greifers: Die grobe Position des Durchbruchs ist durch die 3D Daten der Karosserie bekannt. Der Greifarm bewegt sich in einem Sicherheitsabstand vor die erwartete Position des Durchbruchs. Die genaue Position wird durch optische Erkennung des Durchbruchs durch die Kamera am Greifer korrigiert.

Fähigkeiten: Bahnplanung, Greifpunkttermittlung, Greifen

Ein innovativerer Ansatz, könnte die Entwicklung eines Fördergreifers sein (siehe Abbildung 5-6). Hierbei würde die Leitung (in der Abbildung Kupferfarben dargestellt) von einem Greifer (in der Abbildung Blau dargestellt) umschlossen werden. Auf der Innenseite des Greifergehäuses hat der Greifer zwei oder mehr Förderrollen, die durch eine Drehbewegung den Strang in eine definierte Richtung befördern können. Somit könnte eine Schubbewegung auf den Strang ausgeübt und dieser durch den Durchbruch geschoben werden. Die Konstruktion würde den zweiten Roboter jedoch nicht ersetzen, lediglich das Fädeln vereinfachen. Denn der zweite Greifer müsste das Ende des Leitungsstrangs immer noch außerhalb des Cockpits annehmen und an die Zielposition führen.

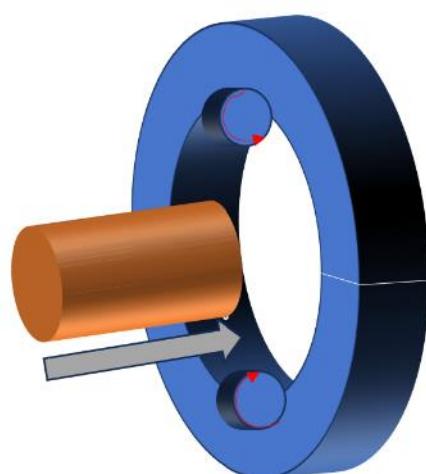


Abbildung 5-6: Idee: Fördergreifer als mögliche Lösung

5.5 Dichtungselemente befestigen

An den Durchbrüchen dient das Dichtungselement oder die Kabeltülle als Schutz gegen mechanischen Verschleiß zwischen Kabelbündel und Karosserieteil sowie Schutz gegen Eindringen von Wasser zwischen zwei Karosseriebereiche. Während des Herstellungsprozess des Leitungssatzes, werden die notwendigen Leitungen durch die Kabeltülle eingebracht und befestigt. Je nach seiner Einbaurichtung kann die Befestigung eines Dichtungselements an den Durchbrüchen auf zwei Arbeitsprozessen unterschieden werden.



Abbildung 5-7: Kabeldichtungen (Quelle: <https://www.woco-kkw.com>)

Axiale Einbaurichtung zum Durchbruch an der Karosserie:

Diese Art von Einbaurichtung ist ein Folgeschritt nach dem Fädelprozess (siehe Kapitel Abbildung 5-7). Daher ist die grobe Positionierung der Dichtung durch den Abschluss des Fädelprozesses bereits erfolgt.

5.5.1 Lokalisieren des Dichtungselements

Die Dichtung liegt somit vor dem Durchbruch der Karosserie, in einer undefinierten Position und Orientierung. Als Grundlage müssen die Ist-Position und Orientierung des Elements erkannt werden.

Datenbedarf: Koordinaten der Ist-Position des Karosseriedurchbruches, Bahnplanung, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasytem),

Fähigkeiten: Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren, Führen.

5.5.2 Grobpositionierung des Dichtungselements

Der Roboterarm greift das Dichtungselement an den vordefinierten Angriffspunkten und bringt es in die definierte Orientierung und Position vor dem Durchbruch.

Datenbedarf: Koordinaten des Greif-Punkt, max. zul. Greifkraft, Bahnplanung, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasytem), Zielkoordinaten Karosseriedurchbruch

Fähigkeiten: Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren.

5.5.3 Feinpositionierung des Dichtungselements

Ist die Grobpositionierung des Dichtungselementes relativ zum Karosseriedurchbruch erfolgreich umgesetzt worden, muss die Dichtung in den dafür vorgesehenen Durchbruch eingesetzt werden. Hierfür greift der 2. Roboterarm die bereits durch den Durchbruch gefädelte Leitung und zieht die Dichtung, bis Kontakt zwischen der nach außen gerichteten Dichtungslippe und der Karosserie besteht. Die Parallele

und zentrische Ausrichtung von Dichtung und Durchbruch muss hierbei während der ausgeführten Bewegung durch das Kamerasystem überprüft und wenn notwendig justiert werden.

Datenbedarf: Zielposition Dichtungselement, Produktabmessung der Dichtung, Bahnplanung

Fähigkeiten: Führen, Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren.

5.5.4 Befestigung des Dichtungselements

Der tatsächliche Befestigungsprozess erfolgt von der „End“-Seite der Karosserie. Der Roboterarm greift die vordere Seite des Dichtungselementes und zieht dieses, bis die Dichtung in der dafür vorgesehenen Aussparung sitzt. Das Dichtungselement hat mehrere „Dichtungsslippen“, die i.d.R. einen größeren Durchmesser als die Karosserieöffnung besitzen. Diese Dichtungsslippen müssen durch die Öffnung gezogen werden, um das Abschotten von Innenraum (Trockenraum) zu bspw. Motor (Nassraum) gewährleisten zu können.

Datenbedarf: Zielposition Dichtungselement, Koordinaten vom Greif-Punkt, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasystem), Soll-Druck- bzw. Zugkraft.

Fähigkeiten: Führen, Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren, Kraft-Überwachung

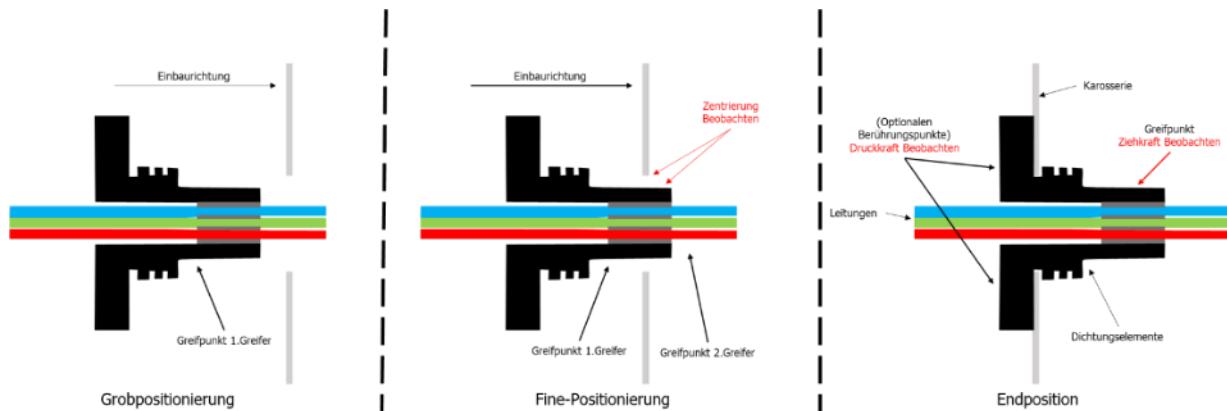


Abbildung 5-8: Dichtungselement mit axialem Einbaurichtung.

5.5.5 Senkrechte Einbaurichtung zur Aussparung der Karosserie

Diese Art von Einbaurichtung ist ein Folgeschritt nach dem Fädelprozess (siehe Kapitel 5.4). Daher ist die grobe Positionierung der Dichtung durch den Abschluss des Fädelprozesses bereits erfolgt. Für den senkrechten Einbau der Dichtung, wird die Ist-Position und Orientierung der Dichtung festgestellt. Anschließend wird der Strang gegriffen und in die notwendige Zielposition oberhalb des Durchbruchs, siehe gebracht. Somit wird auch bei der senkrechten Einbauweise zu Beginn die Ist-Position und Orientierung berechnet.

Datenbedarf: Koordinaten der Soll-Position des Dichtungselements, Bahnplanung, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasystem), Zielkoordinaten des Karosseriedurchbruchs

Fähigkeiten: Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientierung.

Ist das Dichtungselement in der richtigen Position und Orientierung, wird es über die Aussparung positionierte. Die Position und Orientierung des Dichtungselements relative zur Aussparung wird optisch noch einmal überprüft. Falls keine Abweichungen erkannt werden, wird das Dichtungselement in die Aussparung gesenkt.

Datenbedarf: Koordinaten der Soll-Position des Dichtungselements, Bahnplanung, Störkonturen (CAD-Modelle oder Erkennung durch das Kamerasystem), Zielkoordinaten der Aussparung

Fähigkeiten: Führen, Positionieren (Beschreibung der Position (X-, Y-, oder Z-Achse) einer Komponente im Raum), Orientieren, Kraft-Überwachung.

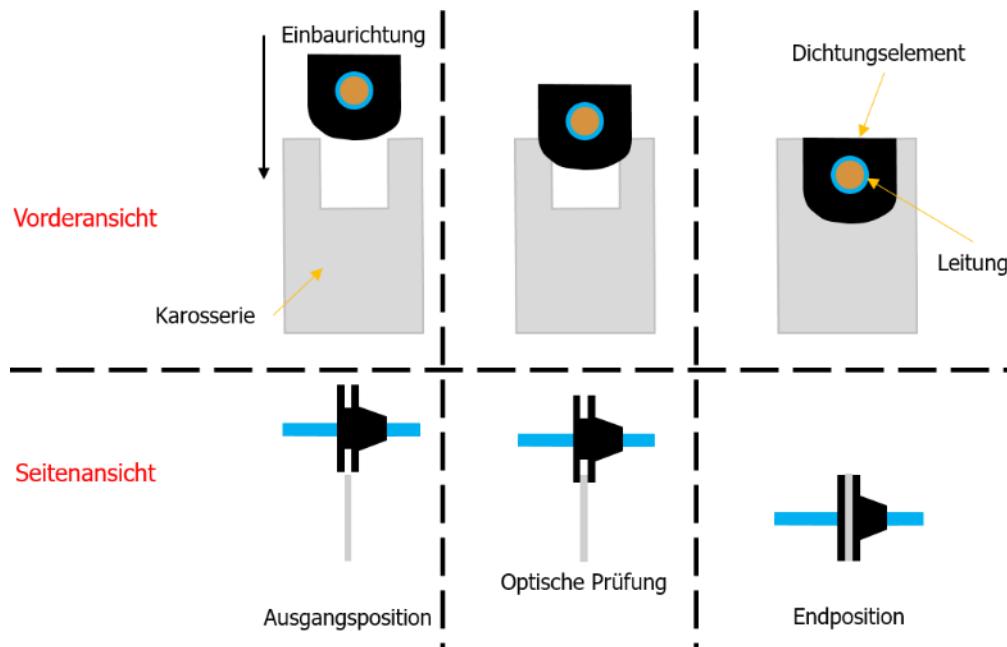


Abbildung 5-9: Radiale Einbaurichtung

5.6 Strangbefestigung

Die Absprungbasis, für die in dem nachfolgenden Absatz beschriebenen Tätigkeiten ist, dass die unter Kapitel 5.1 beschriebene Grobauslegung und Befestigung des Leitungsstrangs bereits erfolgt ist. Das heißt, alle Hauptabgriffe sind bereits in einer der Endlage angenäherten Position verlegt.

Die folgenden Schritte zur Befestigung des Leitungsstranges werden in drei Kategorien unterteilt, um die spezifischen Anforderungen, die diese Vorgänge aufweisen, in den entsprechenden folgenden Unterkapiteln detaillierter darzustellen.

Die Reihenfolge wie der Leitungssatz in der Karosserie befestigt wird lässt sich laut Aussage der Teilnehmer in diesem TP nicht weiter ausarbeiten. Dazu würden mehr Daten benötigt, wann und in welchen Montageabschnitten welche Verlegearbeiten am Leitungssatz gemacht werden. Wenn diese Daten vorhanden wären -am besten über mehrere Baureihen hinweg- könnte evtl. dazu eine Aussage gemacht bzw. Richtlinie erarbeitet werden.

Gewisse Verlegearbeiten werden immer individuell je Fahrzeugderivat sein und müssen ohnehin immer für das entsprechende Derivat eigens geplant werden. Es ist auch wahrscheinlich, dass erst Steckverbindungen hergestellt werden müssen, bevor die letzten Clipse des Abgriffes befestigt werden dürfen.

5.6.1 Clipse und Halterungen befestigen

Clipse gibt es in unterschiedlichen Ausführungen. Zum einen unterscheiden sie sich in der Art wie diese am Leitungssatz befestigt werden (welche hier aber nicht genauer betrachtet wird), und zum anderen, wie die Art der Befestigung an der Karosserie realisiert wird. Anschließend werden die gängigsten Ausprägungen zur Befestigung an der Karosserie gezeigt.

- Kante
- Bolzen
- Langloch
- Rundloch
- Schraubbefestigung
- Rohr- und Schlauchhalterung

Tabelle 5-1 Bilder aus dem Onlinekatalog von [Hellermann Tyton](#)

| Kante | Bolzen | Langloch | Rundloch | Schraubbefestigung | Rohr und Schlauchhalterung |
|---|---|---|---|--|---|
|  |  |  |  |  |  |

Im CES-Arbeitskreis wurde ein Leitfaden zur Erstellung von Elektrik-Bauteilen mit CATIA V5 (Version 2.6 Stand: 18.05.2010) erstellt. Die darin aufgeführten Konstruktionsrichtlinien, speziell für die Clips, sollen als Grundlage herangezogen werden für die standardisierte Kategorisierung dieser Bauteile. Folgendes ist im Leitfaden dazu beschrieben:

Die Positionierung der Bauteilgeometrie im CATPart (natives Dateiformat von Catia V5) ist wie folgt vorzunehmen:

- Der Ursprung des Bauteils liegt im Schnittpunkt der Auflagefläche des Clips (im montierten Zustand) mit der Mittelachse des Rastzapfens, der Bolzenaufnahme oder des Lochs bzw. Langlochs (je nach Befestigungsart).
- Die Y-Achse liegt in der Ebene der Auflagefläche, und ist parallel zu der Richtung orientiert, in der der Kabelbaum durch den Clip geführt wird.
- Die Z-Achse steht normal zur Ebene der Auflagefläche und zeigt entgegen der Montagerichtung des Clips.
- Bei Befestigungselementen, die auf eine Kante aufgesteckt werden (siehe Abbildung 5.41), liegt der Ursprung des Bauteils im Schnittpunkt der Auflagefläche des Bauteils (im montierten Zustand) mit der Mitte der seitlichen Anlagefläche des Bauteils am Blech



Abbildung 5.39: Clip mit Bolzenaufnahme

2 = Bauteil-Ursprung
3 = Achsenystem

Abbildung 5.10: Clip mit Bolzenaufnahme

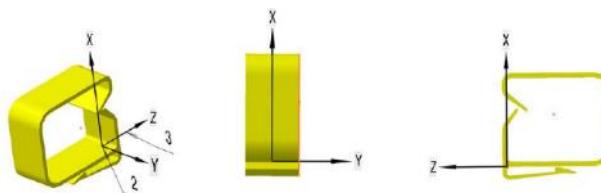


Abbildung 5.41: Kantenbefestiger

2 = Bauteil-Ursprung
3 = Achsenystem

Abbildung 5.11: Kantenbefestiger

Mit Hilfe dieser einheitlichen konstruktiven Vorgaben lässt sich eine einfache Montage der Befestigungselemente realisieren, durch die Standardisierung des Ursprungspunktes/Nullbezugspunkt und des festgelegten Achsenystems.

Aus den CAD-Modellen sind die Positionen der Clipse sowie deren Orientierung im Bezug zur Karosserie und dessen Achsen-/Koordinatensystem bekannt. Diese Informationen müssen über die Verwaltungsschale transportiert werden.

Bei der Leitungssatzkonstruktion bekommen die Clipse eine eindeutige ID zugewiesen. Diese ID ist wiederum mit einem Stammdatum verknüpft, welche das exakte Bauteil (Instanz) identifiziert. Diese Bauteilstammdaten sollten zur einfacheren Handhabung mit einem Merkmal versehen werden, um welche Karosseriebefestigungsart es sich handelt, damit der Roboter das Handling auf den spezifischen Clip einstellen kann. Ein Offset eines Steckers, kann durch ein Vision System erkannt und angefahren werden. Hierbei zählt lediglich die Richtung des Kraftvektors.

Annahme:

Analog zum Leitungssatz haben auch Bolzen, Löcher usw. eine eindeutige ID, diese müsste vorhanden sein, um eine automatisierte prozessuale Fertigung bei der Karosseriefertigung realisieren zu können. Die IDs der Karosseriebefestigungen und der Clipse müssen eine Referenz zueinander aufweisen, diese dann für die Zuordnung beim Montageprozess verwendet werden kann.

Zwei wesentliche Herausforderungen bei der Befestigung des Leitungsstranges müssen noch geklärt werden:

- Identifizierung der am Leitungssatz verbauten Clipse und die entsprechende Zuordnung zu den in der CAD-Konstruktion vergebenen IDs.
- Ermittlung der Greipunkte, die sich durch sich ändernde Geometrien des kundenspezifischen Leitungsstranges ergeben und je nach Varianten des Clips unterschiedlich sein können. Zusätzlich müssen Angriffspunkte definiert werden, damit die Montagekräfte ohne Schädigung des Bauteils ausgeübt werden können. Die notwendigen Informationen sind in der 100%-Variante hinterlegt.

Datenbedarf: Koordinaten inkl. IDs der Befestigungen der Karosserie; Art des Clips zu jeder ID; Verarbeitungsrichtung (Verarbeitungskoordinaten) des Clips und der Karosseriebefestigung; Montagekräfte

Fähigkeiten: Identifizieren des Clips, Erkennen der aktuellen Ausrichtung; Greifen von sich ändernden Gegebenheiten, Fügen des Bauteils in Endlage; Kräftemessung

5.6.2 Kabelkanäle und komplexe Halterungen befestigen

Die Befestigungsarten von Kabelkanälen, die bei Anlieferung des Leitungssatzes am Leitungssatz befestigt sind, lassen sich im Grunde, bezüglich der Befestigungsvarianten an der Karosserie, analog zu den Clips ableiten. Der Unterschied besteht darin, dass es mehrere Befestigungspunkte an einem Bauteil geben kann. Und dass unterschiedliche Ausprägungen von Befestigungen an einen Bauteil vorliegen können, z.B. ein Teil der Befestigungen ist für Rundlöcher und anderer Teil für Bolzen.

In der Leitungssatzkonstruktion hat der Kabelkanal als Ganzes eine eindeutige ID aber nicht jeder Befestigungspunkt.

Datenbedarf: Druckpunkte für die Befestigung; Montagekräfte; evtl. eine Reihenfolge zur Montage; Koordinaten inkl. IDs der Befestigungen der Karosserie; Art des Befestigungspunktes an den Kabelkanälen zu jedem Befestigungspunkt; Verarbeitungsrichtung (Verarbeitungskoordinaten) des Kabelkanals als Ganzes oder für jeden Befestigungspunkt einzeln und der Karosseriebefestigung, Output Informationen (Kräftemessung)

Fähigkeiten: Identifizieren des Kabelkanals, Erkennen der aktuellen Ausrichtung; Greifen; Fügen des Bauteils in Endlage;

5.6.3 Einlegen in Kabelkanäle und Schließen des Deckels

In diesem Punkt soll die Verlegung in Kabelkanälen betrachtet werden, die nicht wie im Kapitel 5.6.2 am Leitungssatz befestigt sind, sondern in der Karosserie montiert sind bevor der Leitungssatz darin verlegt wird.

Nicht näher betrachtet wird, wie die Kabelkanäle in die Karosserie befestigt werden. Es ist aber davon auszugehen, dass dies ähnlich wie in Punkt 7.1.2 auch im Vorfeld dann von einem Roboter übernommen werden könnte.

Für das Einlegen in den Kabelkanal müssen entweder das vollständige Routing oder Routingpunkte festgelegt werden. Hierfür müssen zusätzlich die EIntriis bzw. Austrittspunkte der Stränge definiert werden.

Nach dem Einlegen kann es eine Herausforderung sein, dass der Leitungssatz nicht in der gewünschten „Verlegebahn“ bleibt. Denn es kann je nach der Dicke des Bündels wegen der erhöhten Steifigkeit oder fehlendem Eigengewicht etc. vorkommen, dass das Leitungsbündel die angedachte Verlegeposition wieder verlässt.

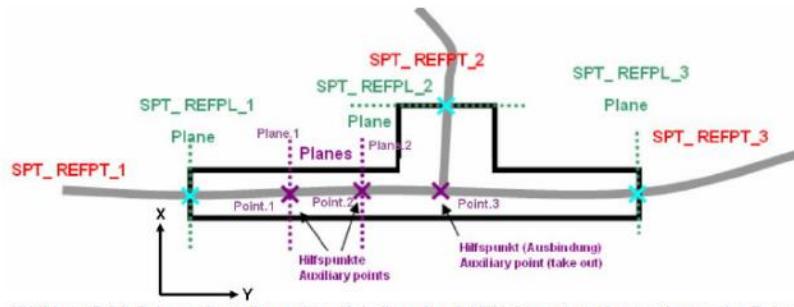


Abbildung 5.84: Schemadarstellung eines Kabelkanals mit Hilfselementen; Nummerierung der Points und Planes aufsteigend in positiver Achsrichtung

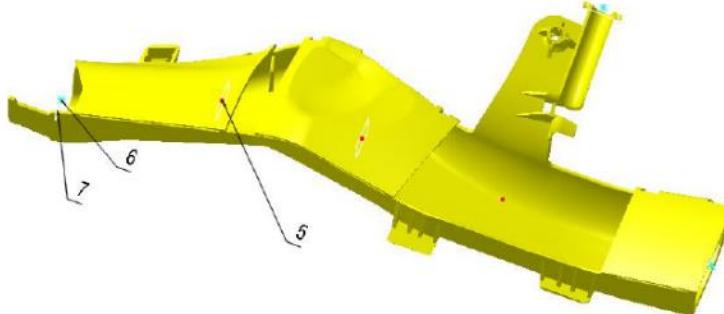


Abbildung 5.85: Beispiel eines Kabelkanals mit Hilfselementen

5 = Hilfspunkt
6 = Referenzpunkt
7 = Referenzplane

Abbildung 5.12: Abbildung eines Kabelkanals mit Hilfselementen (Quelle: Dräxlmaier)

Der Leitfaden zur Erstellung von Elektrik-Bauteilen mit CATIA V5 (Version 2.6 Stand : 18.05.2010) zeigt, wie in der Abbildung oben zu sehen ist, dass Kabelkanäle bereits mit Hilfs- und Referenzpunkten sowie Referenzpläne konstruiert werden.

Datenbedarf: *Routingweg/Routingpunkte; Befestigungsmechanismus des Leitungsstrangs im Kabelkanal falls vorhanden z.B. Kabelbinder; Befestigungsmechanismus Kabelkanaldeckel; Kräfte für Schließung des Deckels;*

Fähigkeiten: *Greifen; Fügen des Bauteils in Endlage; Kräftemessung; Erkennung von Abgriffen für mehrere Ausbindemöglichkeiten*

5.7 Steckverbindung herstellen

Nachfolgend wird über den Aufbau und die Lage von Steckverbindungen eingegangen und wie diese Definition bei der automatisierten Montage des Leitungssatzes verwendet werden kann. Dies stellt nur eine Möglichkeit dar dies zu realisieren. Weitere Möglichkeiten sind denkbar werden aber hier nicht betrachtet.

Um Steckerverbindungen zwischen verschiedenen Leitungssätzen (Trennstellen) oder Aggregaten, wie beispielsweise Steuergeräte, Sensoren, Aktoren usw., und dem Leitungssatz herzustellen, sind verschiedene Daten notwendig. Hierzu müssen einheitliche konstruktive Richtlinien gegeben sein, um nicht für jede mögliche Verbindung individuelle Daten zu erzeugen. Des Weiteren müssen produktbezogene Daten vorhanden sein, um eine prozesssichere Herstellung der Steckverbindung zu gewährleisten. Im TP3 wurde hierfür bereits die Grundlagen (CAD2BOP) erarbeitet.

5.7.1 Konstruktive Gegebenheiten

Zur Konstruktion von Steckerverbindungen wird auf den Leitfaden des CES-Arbeitskreises zur Erstellung von Elektrik-Bauteilen mit CATIA V5 (Version 2.6 Stand: 18.05.2010) Bezug genommen, der folgende Richtlinien enthält:

Die Positionierung der Bauteilgeometrie im CATPart ist wie folgt vorzunehmen:

- Der Ursprung des Bauteils liegt in der Ebene der elektrischen Anlagefläche in der Lochmitte der Kammer/Pin mit dem kleinsten alphanumerischen Zeichen, z.B. Kammer/Pin 1 oder A1.
- Die Y-Achse liegt in der Ebene der elektrischen Anlagefläche und zeigt in Richtung der Kammer/Pin mit dem nächsthöheren alphanumerischen Zeichen, z.B. Kammer/Pin 2 oder A2, bei einpoligen Steckern in Richtung der Kodiernase, falls vorhanden.
- Die Z-Achse steht normal zur Ebene der elektrischen Anlagefläche und zeigt in Richtung des Buchsengehäuses nach innen und vom Stiftgehäuse nach außen.

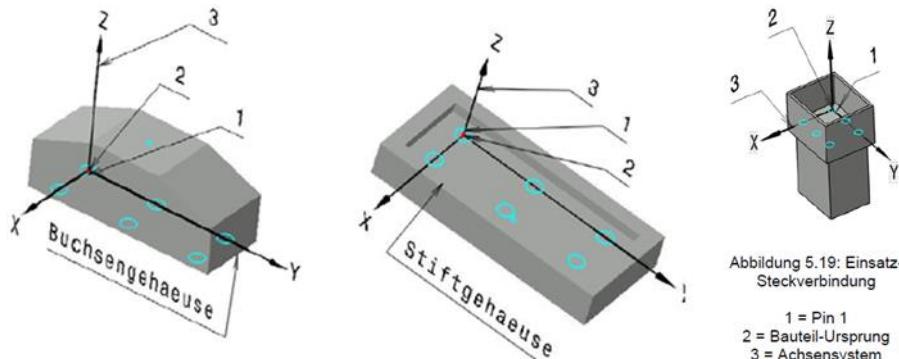


Abbildung 5.13: Konstruktionsmerkmale von Steckverbindungen

Diese konstruktiven Richtlinien müssen auch auf die Gehäuseaufnahmen von Aggregaten übertragen werden.

Die zuvor beschriebenen Informationen, bezüglich des Nullpunkts, Achssysteme, usw., sind für alle Steckverbindungen ausreichenden, wenn sich diese durch Fügung in Y-Richtung herstellen lassen und keine weiteren Elemente des Gehäuses betätigt werden müssen, um das Gehäuse in Endposition zu bringen (dazu nähere Informationen in Punkt 8.2).

Ausnahmen gelten für Steckverbindungen wie beispielsweise in der nachfolgenden Abbildung dargestellt. Hierbei ist das Fügen in Y-Richtung nicht möglich, sondern Bedarf einer speziellen Montagevorgabe, bei der erst die „Rastnase“ eingehakt werden muss. Bei diesen Ausnahmen muss dann für die jeweilige Steckverbindung separat eine Beschreibung erstellt werden.

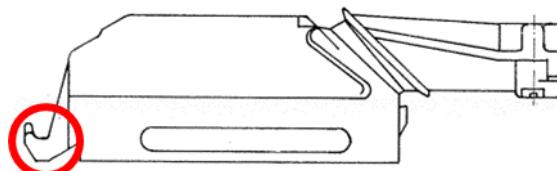


Abbildung 5.14: Steckverbindung mit Rastnase

5.7.2 Umsetzung der Steckverbindung im Fahrzeug

Wenn die Nullpunkte der Gehäuse und die Achssysteme entsprechend den Vorgaben wie unter Punkt 8.1.1 beschrieben konstruiert sind, ist die Endlage der Steckverbindung bereits definiert. Die Nullpunkte müssen übereinander liegen ebenso die Achssysteme der beiden Komponenten.

Der Nullpunkt und das Achssystem der zu kontaktierenden Komponente muss in Relation zum Achssystem und Nullpunkt der Karosserie sein, damit ein Roboter weiß, wo sich die Komponenten befinden. Die Komponenten haben eine eindeutige ID, so dass zu jeder ID ein eindeutiger Nullpunkt und Achssystem zugewiesen ist. Eine Ausnahme bilden Komponenten, die mehrere Gehäusepositionen haben, bei denen gibt es dann entsprechend der Anzahl der Positionen diese Zuordnungen. Diese Informationen müssen aus der VWS oder der CAD-Datei abgeleitet werden.

Die Gehäuse am Leitungssatz haben ebenfalls eine eindeutige ID, die den der Komponenten zugeordnet sind. In der Instanz-VWS dürfen dann auch nur die Gehäuse-IDs wieder zu finden sein, die der Konfiguration des Fahrzeuges entsprechen.

Es werden noch weitere Daten auf Stammdatenebene benötigt, damit eine automatische Fügung der Steckverbindungen hergestellt werden kann. Dazu zählen beispielsweise

- Kragenhöhen, die den Mindestabstand der beiden Komponenten bestimmen, um die Fügung durchzuführen
- Greifpunkte an denen ein Roboter das Gehäuse greifen darf mit Angaben zur maximalen Greifkraft
- Bestückungshub/-weg (ggf. nicht notwendig, da die Steckverbindung hergestellt ist wenn die Nullpunkte übereinander/aneinander liegen)
- Bestückungskraft, wird wahrscheinlich ein schwieriges Thema da je nach Anzahl an Terminals, die im Gehäuse verbaut sind, voraussichtlich auch wo die Terminals verbaut sind (einseitig oder gleichmäßig verteilt), andere Kräfte aufzubringen sind
- Rückzugskraft, zur Prüfung ob die Steckverbindung richtig hergestellt wurde
- Für nicht linear in Y-Richtung verfügbare Steckverbindungen muss eine Montagevorgabe gemacht werden die maschinenlesbar ist

Datenbedarf: siehe zuvor genannte Punkte; Gehäuse-IDs; Nullpunkt und Achsensystem zu jeder Komponente; Komponenten-IDs inkl. Nullpunkt und Achssystem in Relation zum Fahrzeug;

Fähigkeiten: Greifen; Fügen des Bauteils in Endlage; Kräftemessung; Erkennung von Abgriffen und der Gehäuse

5.7.3 Verriegelungselemente von Gehäusen (z.B. CPA – Connector Position Assurance)

Es gibt verschiedene Verriegelungsarten bei Gehäusen, die meisten besitzen ausschließlich nur eine sogenannte Primärverrastung die sich bei einem korrekten herstellen der mechanischen Steckverbindung automatisch schließt. Beim Schließen bzw. Einrasten der Primärverriegelung gibt es eine akustische und haptische Rückmeldung, kann aber auch optische kontrolliert werden. Diese Primärverriegelungen lassen sich meist durch einfache Betätigungen einer Lasche o.ä. lösen, so dass sich die Steckverbindung wieder lösen lässt.

Für besonders sicherheitskritische Steckerverbindungen gibt es eine Sicherung, dass diese Primärverrastung nicht durch ein einfaches Betätigen von z.B. einer Lasche gelöst werden kann. Die Primärverriegelung wird durch ein zusätzliches Element gesichert und kann erst betätigt werden, wenn die Connector-Position-Assurance (CPA) wieder gelöst wird. Es gibt verschiedene Varianten von CPAs,

- solche die sich jederzeit in Endposition schieben lassen, unabhängig ob eine Steckverbindung hergestellt ist oder nicht,
- und die andere Variante, die sich erst betätigen lässt, wenn eine korrekte Steckverbindung hergestellt wurde (dient somit noch zusätzlich der Kontrolle), dass richtig gesteckt wurde.

5.7.3.1 Was ist eine CPA

Die CPA ist ein Mechanismus, der sicherstellen soll, dass ein Paar zusammengesteckter Steckverbinder zusammenbleibt, sobald sie zusammengesteckt sind. Das in Abbildung 5-15 dargestellte Steckverbinder-systems verfügt über eine kabelmontierte Buchse und über eine primäre Verriegelung. Die Verriegelung gleitet unter einer Halteklemme am Gegenstecker und fixiert die Steckverbinder in der gesteckten Position. Die Steckverbinder können nur getrennt werden, indem die primäre Verriegelung gedrückt wird, um sie vom Halteclip des Gegensteckers zu lösen.

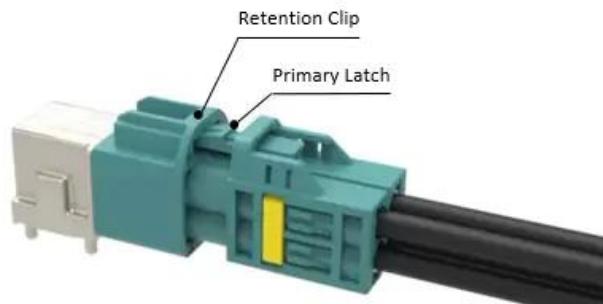


Abbildung 5-15: Mini-Fakra Steckverbinder System

Um zu verhindern, dass sich die Steckverbinder lösen, gibt es eine einfache Lösung (siehe

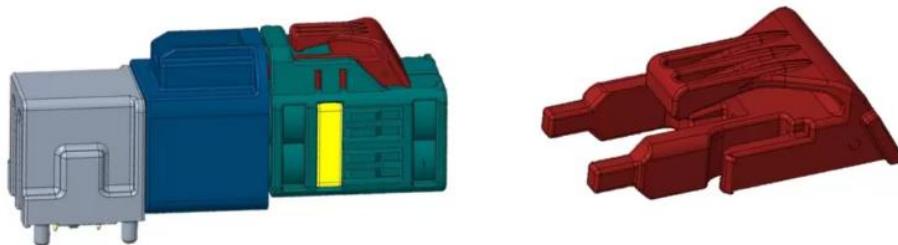


Abbildung 5-16). Indem die Bewegung der Verriegelung mit einer CPA-Vorrichtung eingeschränkt wird, kann die Verriegelung nicht betätigt werden und der Steckverbinder verbleibt in der gesteckten Position.

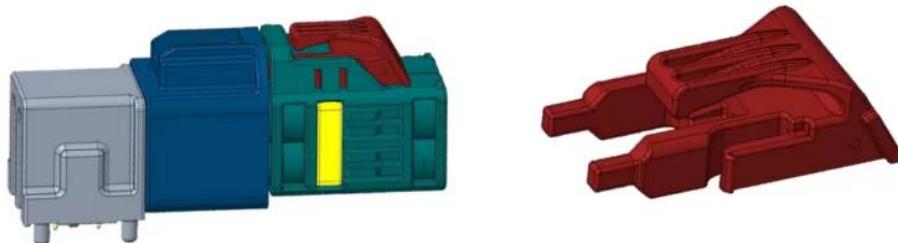


Abbildung 5-16: Funktionsweise CPA-Vorrichtung

Die Positionssicherung ist für die Installation von Steckverbindungen in der Automobilindustrie von entscheidender Bedeutung. Sie stellt sicher, dass die gesteckten Steckverbinder verbunden bleiben (CPA) und die Kontaktkörper vollständig in der richtigen Position sitzen (TPA). (Quelle: [What is Position Assurance?](#))²⁹

5.7.3.2 Herstellen der Steckverbindung mit CPA

In der Regel werden bei einem Leitungssatz die Gehäuse, die eine CPA besitzen, mit geöffneten CPA angeliefert. Das heißt, die Steckverbindungen können hergestellt werden und anschließend bzw. während der Herstellung der Steckverbindung geschlossen werden.

In Ausnahmefällen wird die CPA im geschlossenen Zustand angeliefert, dies wird aber explizit vom OEM gefordert. Dadurch ist im Anlieferungszustand des Leitungssatzes die Position einer CPA, falls diese vorhanden ist, immer definiert und kann an einen Roboter übertragen werden.

Datenbedarf: Gehäuse ID mit der Information hat das Gehäuse eine CPA; Position der jeweiligen CPA im Anlieferungszustand des Leitungssatzes (geschlossen oder offen); wo befindet sich die CPA am Gehäuse; wie lässt sich die CPA betätigen; wann muss eine CPA betätigt werden (während der Herstellung der Steckverbindung oder erst danach);

²⁹ <https://www.amphenolrf.com/news/what-is-position-assurance>

Fähigkeiten: Greifen; Kräftemessung; Fügen (CPA in Endposition bringen und erst lösen falls der Anlieferungszustand der CPA geschlossen ist)

5.8 Fazit

Im TP 4 "Montageprozesse des Leitungssatzes" wurden Möglichkeiten untersucht, die Montageprozesse beim OEM über Verwaltungsschalen als standardisierte Digitale Zwillinge abzubilden. Es wurde ein Überblick über die Montage des Leitungssatzes beim OEM erstellt, mit einem besonderen Fokus auf die prozessualen Herausforderungen und Notwendigkeiten aus der Sicht der Bereitstellung, der Nutzung und des automatisierten Abrufens von Daten zur automatisierten Montage. Es wurde untersucht, welche Fähigkeiten der Ressourcen erforderlich sind, um diese Prozesse erfolgreich auszuführen. Diese Betrachtung betont die Notwendigkeit, die spezifischen Datenbedarfe und notwendigen Fähigkeiten der Roboter für die einzelnen Prozesse und Teilprozesse in der Montage des Leitungssatzes durch Roboter abzudecken.

Ein zentrales Ergebnis der Untersuchungen in TP 4 war die Feststellung, dass es derzeit oft an digital bereitgestellten Informationen aus den Bereichen Produkt, Prozess und Ressourcen mangelt. Diese Lücken erschweren die Effizienz und Genauigkeit der automatisierten Prozesse erheblich. Der Leitungssatz selbst stellt eine besondere Herausforderung dar, da er in Losgröße 1 produziert wird, was eine hohe Varianz und Anpassungsfähigkeit der Produktion erfordert.

Die hohe Produktausprägung und das komplexe Design der Leitungssätze stellen weitere Hürden für eine vollständig automatisierte Montage dar. Jeder Leitungssatz kann individuell in Länge, Anschlussarten und Anordnung der Komponenten variieren, was eine flexible und adaptive Montageanlage erfordert. Diese Variabilität erhöht die technischen Anforderungen an die Automatisierungslösungen und macht den Einsatz hochspezialisierter Roboter und intelligenter Steuerungssysteme notwendig.

Zudem werden die Leitungssätze in der Regel in sog. „Best-Cost-Countries“ gefertigt und müssen anschließend über Logistiknetzwerke zum Montageort transportiert werden, was zusätzliche Komplexität in die Planung und Ausführung der Montageprozesse einbringt.

Zusammenfassend wurde als Ergebnis festgestellt, dass für eine vollständige automatisierte Montage des Leitungssatzes aus heutiger Sicht noch technische Hürden zu lösen sind. Die Losgröße 1 bei hoher Produktausprägung und das komplexe Design der Karosserie erschweren eine automatisierte Montage erheblich. Eine Verbesserung könnte jedoch durch eine grundlegende Neugestaltung der Architektur des Leitungssatzes erfolgen, um eine einfachere und kosteneffizientere Automatisierung zu ermöglichen.

6 TP5 - Integration der Verwaltungsschale

Im Teilprojekt 5 "Integration von Verwaltungsschalen (Verbundkomponente)" wurden ausgehend vom Prinzip der Verbundkomponente der Plattform Industrie 4.0 [38] [39], Umsetzungskonzepte zu definieren, wie die Verwaltungsschale einer Teilkomponente, z.B. Steckverbinder, in die Verwaltungsschale des gesamten Leitungssatz einfließt („Matroschka-Prinzip“).

Das Teilprojekt wurde in folgende Arbeitspakete aufgeteilt, deren Resultate in diesem Dokument zusammenfassend erläutert werden:

- AP 5.1 – Anforderungsdefinition
- AP 5.2 – Verbundkomponente Produkt (Leitungssatz)
- AP 5.3 – Verbundkomponente Ressource (Produktionsmittel) und VIBN
- AP 5.4 – Integration der Verwaltungsschalen
- AP 5.5 – Mapping zwischen Produktionsressource und Leitungssatz
- AP 5.6 – Erfassung von Qualitätsdaten

6.1 AP 5.1 - Anforderungsdefinition

Im AP 5.1 wurden die Anforderungen an das TP5 „Integration von Verwaltungsschalen“ auf Basis des PPR-Konzepts ermittelt.

Die Anforderungen wurden auf Basis einer Workshop-Reihe ermittelt, um ein gemeinsames Verständnis für den Fokus und Inhalt von TP5 zu erlangen. Das Ziel der Anforderungen ist es, sicherzustellen, dass das Teilprojekt gemäß den Bedürfnissen und Erwartungen aller Beteiligten geplant und umgesetzt wird. Sie sollen als Leitfaden für das gesamte Teilprojekt dienen und eine erfolgreiche Umsetzung gewährleisten.

Die Workshops wurden auf Basis des PPR-Konzepts durchgeführt, um sicherzustellen, dass keine für die Integration von Verwaltungsschalen relevanten Beziehungen und Anforderungen übersehen werden. Das Ergebnis der Workshops ist Abbildung 6-2 in zu sehen.

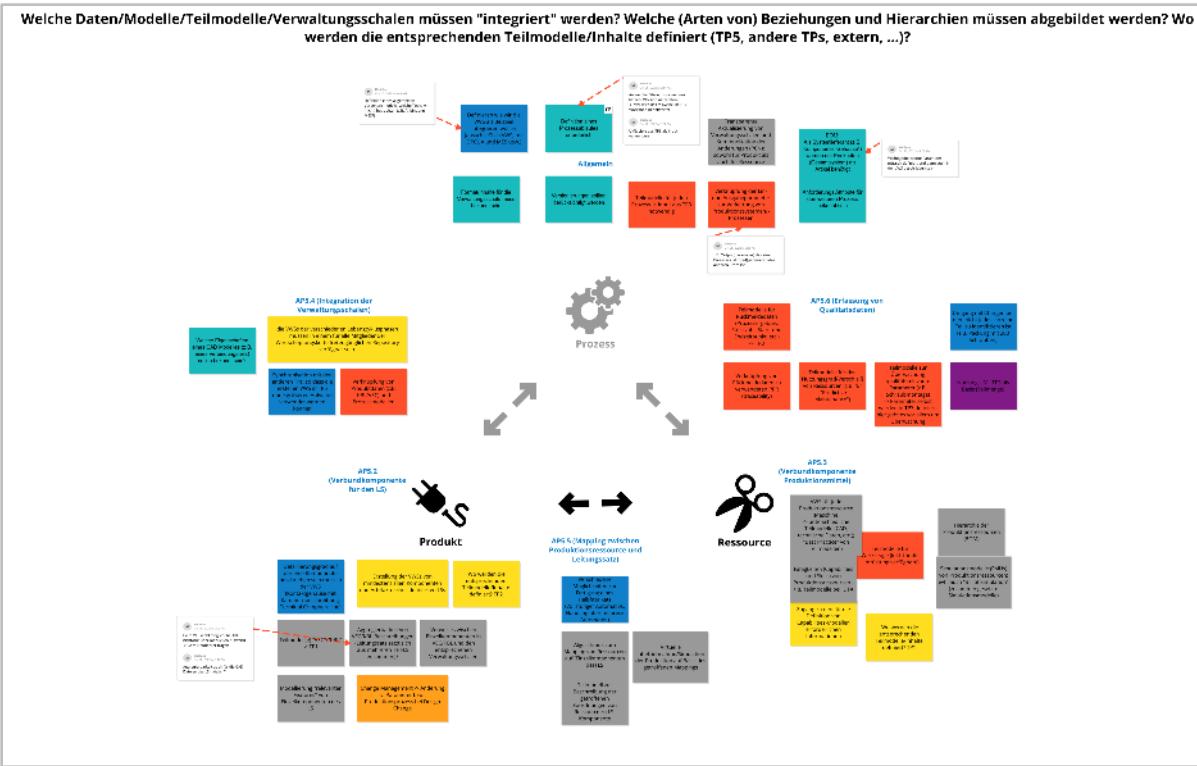


Abbildung 6-1: Ergebnis der Workshop-Reihe

Die Workshop-Ergebnisse wurden anschließend in einem strukturierten Prozess in Anforderungen überführt und den jeweiligen Arbeitspaketen innerhalb von TP5 zugeordnet. Zu jeder Anforderung wurden darüber hinaus sogenannte „Definitions of Done“ (DoDs) formuliert, die die Überprüfbarkeit der Erfüllung einer Anforderung gewährleisten. Einen weiteren wichtigen Punkt stellte die Identifikation von Schnittstellen zu anderen Teilprojekten bzw. Arbeitsgruppen statt. Da im Rahmen von TP5 Verwaltungsschalen vorgelagerter Teilprojekte integriert werden, beschreiben diese Schnittstellen vor allem Randbedingungen an die Arbeit dieser anderen TPs.

Im Folgenden sind die einzelnen Anforderungen inklusive DoD sowie identifizierter Schnittstellen aufgeführt.

6.1.1 Allgemeine Anforderungen

6.1.1.1 Gesamtarchitektur

Anforderung 1: Kopplung bzw. Integration von Verwaltungsschalen mit anderen Tools

Es muss definiert werden, wie bzw. ob die Verwaltungsschalen im Projekt mit anderen Tools gekoppelt/integriert werden (z.B. Anbindung MES, OPC UA-Server, etc.) oder ob die VWS nur für sich genutzt werden.

Definition of Done:

- Übersicht über Anbindungs möglichkeiten erstellt (bspw. REST API)
- Vor- und Nachteile der verschiedenen Möglichkeiten dargestellt
- Definition, welche Anbindungen an Werkzeuge im Projekt notwendigerweise berücksichtigt/untersucht werden müssen

Schnittstellen:

- AP3.2 und AP3.3: Architekturübersicht mit allen beteiligten Werkzeugen/Werkzeugtypen erstellt (bspw. MES, ...)

Anforderung 2: Definition Prozessablauf

Es muss ein Prozessablauf definiert werden, welcher Teilnehmer (OEM, Tier x) wann welche Daten generiert bzw. ergänzt.

Definition of Done:

- Übersichtsdiagramm mit allen beteiligten Prozessteilnehmern (Entwicklung, Produktion, Montage), Prozessschritten sowie ausgetauschten Artefakten erstellt

Schnittstellen:

- TP2/PPR-Workshop: Übersicht über den Entwicklungsprozess des Leitungssatzes mit allen beteiligten Akteuren und ausgetauschten Daten
- UC3: Definition eines Beispiel-Szenarios für das Änderungsmanagement

6.1.1.2 Inhalte von Verwaltungsschalen

Anforderung 3: Lieferung von Komponenten-Daten

Ein Komponentenlieferant (Tier2) muss für eine gelieferte Komponente (z.B. Gehäuse) eine Beschreibung des Gesamtsystems mit allen benötigten Inhalten liefern. Die Auflistung enthaltener Teil-Komponenten in Form einer vollständigen BOM ist nicht notwendig.

Definition of Done:

- Notwendige Teilmodell(e) für Komponenten sind definiert
- Beschreibung, welche Teilmodelle (in welcher Ausprägung/Detailtiefe) für die Nutzung in einer Verbundkomponente mindestens bereitgestellt werden müssen

Schnittstellen:

- TP2: Übersicht über den Entwicklungsprozess des Leitungssatzes mit allen ausgetauschten Daten
- TP1/AP2.4/PPR-Workshop: Definition und Entwicklung benötigter Teilmodelle

Anforderung 4: Definition von Verwaltungsschalen-Typ und Inhalt

Es ist im Projekt zu definieren, welches Format (z.B. Typ 1, 2, 3) und welchen Inhalt (z.B. welche Teilmodelle) die verschiedenen Verwaltungsschalen besitzen müssen

Definition of Done:

- Übersicht über alle benötigten Verwaltungsschalen erstellt
- Benötigte Teilmodelle für alle Verwaltungsschalen definiert
- Inhalte für alle benötigten Teilmodelle definiert
- Interaktionsmuster zwischen Tools und/oder Verwaltungsschalen definiert
- Ableitung getroffen, welche Verwaltungsschale (mindestens) in welchem Typ (1, 2 oder 3) ausgeführt sein muss; aufgegliedert nach Lebenszyklusphase der einzelnen Verwaltungsschale

Schnittstellen:

- PPR-Workshop: Übersicht über alle benötigten Verwaltungsschalen sowie deren Inhalte
- AP2.4/PPR-Workshop: Definition benötigter Teilmodelle

Anforderung 5: Definition von Teilmodellen für Prozesse

Für jeden Prozess (z.B. Crimpen) ist ein Teilmodell/Daten zu definieren. Es muss möglich sein, Ein- und Ausgabeparameter von verschiedenen Prozessen/Produktionssystemen miteinander zu verknüpfen, um Datenströme/Abhängigkeiten zu kennzeichnen

Definition of Done:

- Betrachtete Prozesse (z.B. Crimpen) definiert
- Relevante Daten/Parameter für jeden Prozess definiert; sowohl für die Entwicklungs- als auch für die Produktionsphase
- Teilmodell(e) für jeden Prozess definiert
- Konzeptbeschreibung:
 - Der Ausgabeparameter eines Prozesses kann als Eingabeparameter eines anderen Prozesses verwendet werden.
 - Der Eingabeparameter eines Prozesses kann auf Grundlage von Ausgabeparametern anderer Prozesse berechnet werden.

Schnittstellen:

- TP3: Beschreibung des Produktionsprozesses des Leitungssatzes sowie Entwicklung zugehöriger Teilmodelle

6.1.1.3 Änderung und Versionierung**Anforderung 6: Transparente Aktualisierung von Verwaltungsschalen**

Verwaltungsschalen müssen transparent aktualisiert werden. Änderungen (sowohl auf Seiten eines Produkts als auch auf Seiten einer Ressource) müssen kommuniziert werden (PCN), sodass abhängige Daten aktualisiert werden können.

Definition of Done:

- Analyse durchgeführt, welche Arten Synchronisierung unterstützt werden müssen (Push/Pull)
- Konzept für Synchronisierung von Änderungen mit anderen Verwaltungsschalen definiert

Schnittstellen:

- Architekturteam: Synchronisierung von Verwaltungsschalen sowie Änderungsmanagement
- TP6: Änderungsmanagement als Verhandlungsprozess

Anforderung 7: Versionierung

Die Versionierung von VWS bzw. enthaltenen Daten ist zu berücksichtigen. Dies bedeutet, dass ein Ansatz für die Versionierung von Verwaltungsschalen und/oder Teilmodellen zu entwerfen ist.

Definition of Done:

- Konzept zur Versionierung von VWSen beschrieben, sodass Änderungen zwischen zwei Versionen einer VWS bzw. eines Teilmodells nachvollzogen werden können
- Ein Teilmodell das Informationen zu Gültigkeiten (Ist dies die aktuelle Version des Teilmodells?), Vor- und Nachgänger, Änderungsbeschreibung der Revision usw. enthält, ist definiert. Definiert ist ein Common-Umfang, der ein übergreifendes Mindestmaß an Information darstellt
- Falls das Konzept eine strukturelle Erweiterung des bestehenden Konzepts der VWS erfordert, Kommunikation des Konzepts zu relevanten Gremien (bspw. AG1 der PlattformI4.0)

Schnittstellen:

- Architekturteam: Versionierung von Verwaltungsschalen bzw. -inhalten

6.1.2 Verbundkomponenten für den Kabelbaum (AP5.2)

6.1.2.1 Produktmodell

Anforderung 8: Integration von KBL/VEC

Es sind Teilmodelle für die Integration von Produktmodellen in Form von KBL-/VEC-Dateien zu definieren.

Definition of Done:

- Teilmodell für VEC definiert
- Teilmodell für KBL definiert

Schnittstellen:

- Architekturteam: Modularisierung und Verlinkung von Inhalten in Verwaltungsschalen

Anforderung 9: Erstellung prototypischer Verwaltungsschalen

Es sind (prototypische) VWS von allen relevanten Komponenten und Artefakten eines definierten LS zu erstellen. Die Verwaltungsschalen sollen über die gemäß Anforderung 3: definierten Inhalte verfügen.

Definition of Done:

- Beispielprodukt/-leitungssatz inklusive aller relevanten Halbfabrikate, Komponenten und Varianten definiert
- Benötigte Verwaltungsschalen definiert; Berücksichtigung unterschiedlicher „Sichten“ z.B. zwischen OEM, Tier1, Tier2
- Verwaltungsschalen erstellt und relevante Teilmodelle befüllt; Verknüpfungen zwischen Verwaltungsschalen angelegt

Schnittstellen:

- PPR-Workshop: Definition eines Beispiel-Leitungssatzes, benötigter Verwaltungsschalen
- TP1/AP2.4/PPR-Workshop: Definition und Entwicklung benötigter Teilmodelle

6.1.2.2 Hierarchie und Verweise

Anforderung 10: Integration der Informationsmodelle von KBL/VEC

Es ist zu definieren, wie/welche Verweise zwischen Einzelkomponenten in VEC/KBL und den entsprechenden Verwaltungsschalen definiert werden können/müssen

Definition of Done:

- Referenzziele aus den Informationsmodellen von VEC und KBL definiert (auf welche Elemente/Elementtypen soll verwiesen werden, um Beziehungen zu anderen Teilmodellen herzustellen)

- Referenzierungs-Mechanismus definiert

Schnittstellen:

- Architekturteam: Modularisierung und Verlinkung von Inhalten in Verwaltungsschalen

Anforderung 11: Aggregation von KBL/VEC-Beschreibungen

Es ist zu untersuchen, inwiefern VEC/KBL-Beschreibungen „aggregiert“ werden können, sodass sich ein LS aus mehreren Teil-LS zusammensetzen kann.

Definition of Done:

- Konzept zur Modularisierung von VEC-Beschreibungen definiert
- Konzept zur Modularisierung von KBL-Beschreibungen definiert

Schnittstellen:

- Architekturteam: Modularisierung und Verlinkung von Inhalten in Verwaltungsschalen

6.1.2.3 Änderungsmanagement

Anforderung 12: Kommunikation und Integration von Änderungen

Es ist zu definieren, wie Änderungen von Parametern bzw. Produktionsprozessen auf Basis eines Design Changes kommuniziert und konsistent aktualisiert werden können (Change Management).

Definition of Done:

- Konzept für Änderungsfreigabe/-übernahme definiert

Schnittstellen:

- Architekturteam: Synchronisierung von Verwaltungsschalen sowie Änderungsmanagement

6.1.3 Verbundkomponenten Produktionsmittel (AP5.3)

6.1.3.1 Modellierung von Produktionsressourcen

Anforderung 13: Definition von Teilmodellen für Ressourcen

Es ist zu definieren, welche Teilmodelle für eine Produktionsressource geliefert werden müssen (z.B. CAD, technische Daten, Teilmodelle für Werkzeuge und Werkzeugumrüstung, Simulationsmodelle, Fähigkeiten und Skills, ...).

Definition of Done:

- Benötigte Teilmodelle für Ressourcen-VWSen definiert
- Beschreibung, welche Teilmodelle (in welcher Ausprägung/Detailtiefe) für die Nutzung in einer Verbundkomponente mindestens bereitgestellt werden müssen --> „Best Practice“-Dokument

Schnittstellen:

- TP1: Definition von Informationsmodellen

6.1.3.2 Hierarchie

Anforderung 14: Hierarchien von Produktionsressourcen

Es ist zu definieren, wie/welche Hierarchien von Produktionsressourcen (BOM) definiert werden können/müssen.

Definition of Done:

- Konzept zur Modellierung zusammengesetzter Ressourcen definiert (bspw. Maschine und verfügbare Werkzeuge/Erweiterungsmodule)

- Konzept zur Modellierung von Capabilities unter Berücksichtigung zusammengesetzter Ressourcen beschrieben

Schnittstellen:

- Architekturteam: Modularisierung und Verlinkung von Inhalten in Verwaltungsschalen

Anforderung 15: Aggregation von Simulationsmodellen

Es ist zu prüfen, ob/wie Simulationsmodelle für Maschinen aus mehreren Simulationsmodellen für Einzelkomponenten einer Maschine aggregiert werden können („Bill of Simulation“).

Definition of Done:

- Beschreibung anhand eines oder mehrerer konkreten/r Use Cases, welche Art von Simulationsmodellen benötigt werden
- Beschreibung, welche Art von Simulationsmodellen für die Nutzung in einer Verbundkomponente mindestens bereitgestellt werden müssen
- Konzept zur automatischen Ableitung einer Gesamt-/Systemsimulation aus Simulationsmodellen von Einzel-/Teilressourcen beschrieben

Schnittstellen:

- ---

6.1.4 Integration der Verwaltungsschalen (AP5.4)**6.1.4.1 Allgemeines****Anforderung 16: Synchronisation von Anforderungen**

Die Anforderungen aus TP5 sind mit den anderen TPs zu synchronisieren, sodass die dort erstellten Teilmodelle/Verwaltungsschalen in TP5 ohne größeren Aufwand integriert werden können.

Definition of Done:

- Benötigte Teilmodelle sind aufgelistet.
- Für alle benötigten Teilmodelle sind Anforderungen aus Sicht der Verbundkomponente formuliert.

Schnittstellen:

- TP1/2/3/4: Definition von Informations- und Teilmodellen

Anforderung 17: Bereitstellen von Verwaltungsschalen in einem Repository

Die VWS der verschiedenen Lebenszyklusphasen müssen in einem für alle Mitglieder der Wertschöpfungskette frei zugänglichen Repository verfügbar sein.

Definition of Done:

- Es ist eine prototypische Infrastruktur aufgesetzt, in der alle bestehenden Verwaltungsschalen über einen oder mehrere Server abgerufen werden können und auf die alle Mitglieder zugreifen können.
- Über die Infrastruktur können neue Verwaltungsschalen an-/abgelegt werden.
- Die Infrastruktur unterstützt die Definition und Auswertung von Zugriffsrechten für einzelne VWSen/Teilmodelle/Elemente.

Schnittstellen:

- ---

6.1.4.2 Verknüpfung von Teilmodellen und Verwaltungsschalen

Anforderung 18: Verknüpfung von CAD-Modellen mit anderen Teilmodellen

Es ist zu definieren, welche Eigenschaften eines CAD-Modells (z.B. eines Verbindungsteils) bekannt sein müssen, damit dieses mit anderen Teilmodellen/Verwaltungsschalen in Beziehung gesetzt werden kann.

Definition of Done:

- Anhand eines bestimmten Beispiels (z.B. automatisiertes Stecken von Terminals in Gehäuse bei der Leitungssatzfertigung) ist definiert, welche Parameter aus dem CAD-Modell lesbar/ableitbar sein müssen, um dies mit geringem Aufwand realisieren zu können
- Eine Systematik zur Verallgemeinerung des Beispiels ist definiert.

Schnittstellen:

- AP3.2: Betrachtung der benötigten Prozesse bei der LS-Fertigung sowie relevanter Parameter

Anforderung 19: Verknüpfung von Produkt- und Prozessmodell

Es ist zu definieren, wie Daten aus dem Produktmodell (z.B. KBL/VEC) mit dem Prozessmodell zu verknüpfen sind. Dabei ist auch eine Zuordnung von einzelnen Prozessschritten und resultierenden Halbfabrikaten zu schaffen.

Definition of Done:

- Das Prozessmodell kann direkt auf erforderliche Datenfragmente des Produktmodells verweisen.
- Konzept zur Verknüpfung des Prozess- und Produktmodells ist definiert, z.B. auf Basis von „Required Capabilities“
- Das Konzept ermöglicht die Zuordnung von Prozessschritten zu genutzten bzw. resultierenden Komponenten und Halbfabrikaten.

Schnittstellen:

- TP3: Definition des Prozessmodells sowie der zugehörigen Capabilities inkl. Parametern
- Architekturteam: Verlinkung von Inhalten in Verwaltungsschalen

6.1.5 Mapping zwischen Produktionsressource und Leitungssatz (AP5.5)

6.1.5.1 Verknüpfung von Produktionsprozess und -ressourcen

Anforderung 20: Verknüpfung von Prozessmodell und Ressourcen

Es ist ein Teilmodell zu definieren, in dem Zuordnungen zwischen Ressourcen und Prozessschritten beschrieben werden können. Dabei müssen verschiedenen Varianten zur Fertigung von Halbfabrikaten (z.B. vollstufiger Automat vs. Handling über mehrere Automaten) definiert werden können.

Definition of Done:

- Mithilfe des CSS-Modells (Capabilities-Skills-Services) werden verschiedene Zuordnungen (Fertigungswege) herausgefunden und mit weitere Faktoren aggregiert wie z.B. Durchlaufzeit
- Es existiert eine Spezifikation für ein eigenes Teilmodell oder für die Erweiterung eines existierenden Teilmodells, die beschreibt, wie Prozessschritte mit zugehörigen Ressourcen verknüpft werden können.
- Im Falle modularer Ressourcen (bspw. Werkzeugmaschine + Werkzeug) ermöglicht die Spezifikation die Beschreibung, welche(s) Modul(e) für den Prozessschritt genutzt werden müssen.

Schnittstellen:

- TP6: Betrachtung automatischer Verhandlungsprozesse als Grundlage für die Ermittlung potentieller Fertigungswege

Anforderung 21: Algorithmus zum Fähigkeitenabgleich

Es ist ein Algorithmus zu entwerfen, der auf Basis einer Analyse von Fähigkeiten von Produktionsressourcen eine automatische Abbildung von Ressourcen auf LS-Komponenten/Prozessschritte ableiten kann.

Definition of Done:

- Konzeptbeschreibung für einen Algorithmus zum automatischen Mapping von Prozessschritten auf Ressourcen.
- Der Algorithmus nutzt als Basis ein Modell von „required capabilities“ für einen (Satz von) Prozessschritten sowie ein Modell von „offered capabilities“ eines (Satzes von) Produktionsressourcen.
- Es existiert eine prototypische Implementierung des Algorithmus.

Schnittstellen:

- TP3: Definition des Prozessmodells sowie der zugehörigen Capabilities inkl. Parametern
- TP6: Betrachtung automatischer Verhandlungsprozesse als Grundlage für die Ermittlung potentieller Fertigungswege

Anforderung 22: Virtuelle Inbetriebnahme der Produktion

Die virtuelle Inbetriebnahme/Simulation der Produktion auf Basis der definierten Zuordnungen zwischen Ressource und LS-Komponente/Prozessschritt ist zu untersuchen.

Definition of Done:

- Teilmodelle für die virtuelle Inbetriebnahme sind definiert bzw. vorhandene aus der Produktion werden dafür verwendet.
- Später mit „realen Daten“ aus den Teilmodellen der Produktion vergleichen, um die Datenqualität der virtuelle Inbetriebnahme zu prüfen und zu verbessern
- Es existiert ein Konzept zur Gesamtsimulation einer Produktion bzw. eines Produktionsschrittes auf Basis einzelner Simulationsmodelle, die z.B. eine (Teil-)Schritt abbilden.

Schnittstellen:

- ---

6.1.6 Erfassung der Qualitätsdaten (AP5.6)

6.1.6.1 Teilmodelle

Anforderung 23: Teilmodell für Qualitätsdaten

Es ist ein Teilmodell zu definieren, mit dem relevante Rückmelde-/Qualitätsdaten beschrieben werden können (Stückzahl, Start- und Endzeitpunkt, etc.).

Definition of Done:

- Es existiert eine Teilmodellspezifikation zur Beschreibung von Rückmelde- bzw. Qualitätsdaten eines Produktionsschrittes bzw. einer Gesamtproduktion.
- Im Teilmodell muss ein Vorhalt berücksichtigt sein, falls eine Charge eines Produktionsauftrags nicht abgeschlossen werden kann (neues Material muss gerüstet werden o.ä.)
- Dieser Vorhalt muss auch die Verbindung von mehreren Chargen zu einem Produktionsauftrag vorsehen (Nachproduktion zur Erfüllung des Produktionsauftrags)

Schnittstellen:

- TP3: Definition von Teilmodellen für den Produktionsprozess inkl. Parametern als Grundlage für Rückmelddaten

Anforderung 24: Teilmodell zur Überwachung von Parametern

Es sind Teilmodelle zu definieren, mit denen qualitätsrelevante Parameter überwacht werden können (z.B. bei der Schraubmontage)

Definition of Done:

- Die Spezifikation für ein Teilmodell zur Beschreibung qualitätsrelevanter Parameter liegt vor. Das Teilmodell muss dabei ermöglichen, für bestimmte Parameter eines Prozessschrittes Qualitätsvorgaben zu definieren (bspw. erlaubte Toleranzen).
- Es muss möglich sein, Ereignisse zu definieren, die ausgelöst werden sollen, wenn ein qualitätsrelevanter Parameter seinen definierten Grenzwert über- bzw. unterschreitet. Der Grenzwert kann dabei auch eine Funktion sein, um bspw. Tendenzen zu erkennen. Qualitätsrelevante Parameter können sich entweder auf einzelne Messwerte oder ganze Messreihen beziehen.
- Es existiert eine Implementierung, die demonstriert, dass die entsprechenden Ereignisse ausgelöst werden.

Schnittstellen:

TP3: Definition von Teilmodellen für den Produktionsprozess inkl. Parametern als Grundlage für Rückmeldedaten

Anforderung 25: Teilmodell zur Beschreibung des Nutzungsgrades

Es sind Teilmodelle zur Beschreibung des Nutzungsgrades/Verschleiß von Ressourcen zu definieren (z.B. für Predictive Maintenance).

Definition of Done:

- Es existiert eine Teilmodellspezifikation, mit dem der aktuelle Zustand einer Ressource beschrieben werden kann. Dies kann Werte wie Abnutzungsgrad, Verschleiß oder erwartete Restlaufzeit beinhalten.
Es existiert ein Teilmodell zur Beschreibung von Wartungszyklen o.ä.

Schnittstellen:

- ---

6.1.6.2 Verknüpfung von Daten

Anforderung 26: Traceability

Es ist zu definieren, wie Rückmeldedaten mit anderen Elementen/Daten aus dem PPR-Kontext verknüpft werden können („Traceability“). Ein Beispiel hierfür stellt die Zuordnung eines bestimmten Datensatzes zu dem produzierten Halbfabrikat sowie der produzierenden Maschine dar.

Definition of Done:

- Die Rückmeldedaten können auf die verwendeten Produkt-, Prozess- und Ressourcenmodelle zurückgeführt werden.

Schnittstellen:

- TP3: Definition des Prozessmodells sowie der zugehörigen Capabilities inkl. Parametern
- Architekturteam: Verlinkung von Inhalten in Verwaltungsschalen

Anforderung 27: Unterstützung von Chargen

Es ist zu definieren, wie mit Chargen umzugehen ist, bei denen nicht jedes einzelne Teil zu identifizieren ist (z.B. Packung mit 200 Schrauben). Wie kann eine Zuordnung zwischen Chargen und anderen Elementen (z.B. Qualitätsdaten) erfolgen?

Definition of Done:

- Definition eines Konzeptes, das sowohl die Definition einer Seriennummer als auch einer bzw. mehrerer Chargennummer(n) zulässt/verarbeiten kann und dies kennzeichnet, wobei die Information möglichst automatisiert aus der VWS (wenn vorhanden) des verwendeten Materials übernommen werden soll

Schnittstellen:

- ---

6.2 AP 5.2 - Verbundkomponente Produkt (Leitungssatz)

6.2.1 Zielsetzung

Das AP 5.2 behandelt die Ableitung und Definition der Verbundkomponente (VBK) „Produkt“, d.h. des Leitungssatzes.

Grundlage für die VBK sind dabei zunächst die relevanten Beziehungen zwischen den beteiligten Verwaltungsschalen bzw. zwischen Elementen dieser Verwaltungsschalen. Diese Beziehungen wurden im Rahmen von AP5.2 ermittelt, wobei die Analyse notwendiger Beziehungen von den Ergebnissen des PPR-Workshops ausging, in dem relevante Assets und Verwaltungsschalen (unter anderem) für die Abbildung des Leitungssatzes analysiert wurden. Abbildung 6-2 zeigt eine Übersicht über die relevanten Verwaltungsschalen, welche zur Analyse der Beziehungen herangezogen wurden.

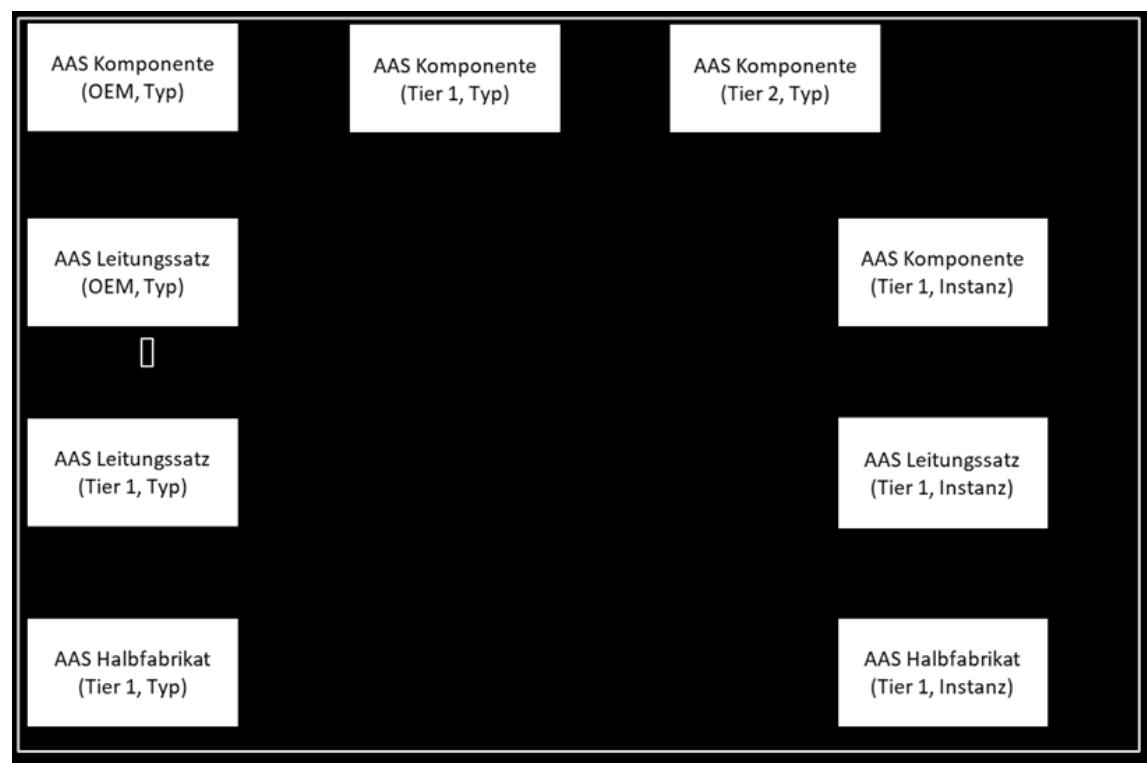


Abbildung 6-2: Unterschiedliche Arten von Verwaltungsschalen als Teil der VBK „Produkt“

Die verschiedenen Beziehungen zwischen den oben abgebildeten Verwaltungsschalen wurden klassifiziert (Quelle, Ziel, Kategorie, Kardinalität) und mit je einem Use Case beschrieben. Dieser Use Case zeigt jeweils auf, warum es sinnvoll ist, die entsprechende Beziehung explizit im Rahmen der VBK „Produkt“ abzubilden.

Abschnitt 6.2.2 führt die einzelnen Beziehungen inklusive Klassifizierung sowie zugeordnetem Use Case auf. Darauf aufbauend erläutert Abschnitt 0 die Umsetzung der verschiedenen Kategorien von Beziehung auf Basis der Verwaltungsschale.

6.2.2 Ermittelte Beziehungen

In diesem Abschnitt werden die einzelnen Beziehungen inklusive Klassifizierung sowie zugeordnetem Use Case in Bezug auf Abbildung 6-2 aufgeführt.

6.2.2.1 Hierarchische Beziehungen

6.2.2.1.1 Leitungssatz enthält Komponenten

Klassifizierung:

- Quelle: AAS Leitungssatz Tier 1
- Ziel: AAS-Komponente Tier 1
- Kategorie: hierarchisch, top-down
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einem LS und den enthaltenen Komponenten

Use Case:

- Engineering-Tool kann erkennen, welche Komponenten in einem LS enthalten sind und direkt die benötigten Daten aus deren AASen auslesen, bspw. für die Produktionsplanung
- MES-Tool kann erkennen, welche Komponenten in einem LS enthalten sind und prüfen, ob die Komponenten auf Lager liegen bzw. deren Beschaffung anstoßen

Kommentar:

- Inverse Beziehung zu 6.2.2.1.4
- Analoge Beziehung zu 6.2.2.1.2, 6.2.2.1.7, 6.2.2.1.11 und 6.2.2.1.15

6.2.2.1.2 (OEM-)Leitungssatz enthält (OEM-)Komponenten

Klassifizierung:

- Quelle: AAS Leitungssatz OEM
- Ziel: AAS-Komponente OEM
- Kategorie: hierarchisch, top-down
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einem LS und den enthaltenen Komponenten

Use Case:

- OEM prüft, ob die verwendeten Komponenten freigegeben und noch gültig sind und nach außen gegeben werden dürfen
- Voraussetzung für die Ersetzung der OEM-Teilenummern durch die Tier1-Teilenummern

Kommentar:

- Inverse Beziehung zu 6.2.2.1.4
- Analoge Beziehung zu 6.2.2.1.1, 6.2.2.1.7, 6.2.2.1.11 und 6.2.2.1.15

6.2.2.1.3 Leitungssatz-Instanz enthält Komponenten-Instanzen

Klassifizierung:

- Quelle: AAS Leitungssatz-Instanz Tier 1
- Ziel: AAS-Komponenten-Instanz Tier 1
- Kategorie: hierarchisch, top-down

- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einer LS-Instanz und den enthaltenen Komponenten/Chargen von Komponenten

Use Case:

- Traceability

Kommentar:

- Inverse Beziehung zu 6.2.2.1.6
- Analoge Beziehung zu 6.2.2.1.8, 6.2.2.1.12 und 6.2.2.1.16

6.2.2.1.4 Komponente wird in Leitungssatz verwendet**Klassifizierung:**

- Quelle: AAS-Komponente Tier 1
- Ziel: AAS Leitungssatz Tier 1
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung der Leitungssätze, in denen eine Komponente verwendet wird

Use Case:

- Komponente hat Lieferengpass -> benachrichtigt automatisch alle Leitungssätze, in denen sie verwendet wird
- Daten einer Komponente werden geändert -> benachrichtigt automatisch alle Leitungssätze, in denen sie verwendet wird

Kommentar:

- Hoher Datenpflege-Aufwand
- Use Cases können auch über Rückwärtssuche der Beziehung 6.2.2.1.1 realisiert werden

6.2.2.1.5 (OEM-)Komponente wird in (OEM-)Leitungssatz verwendet**Klassifizierung:**

- Quelle: AAS-Komponente OEM
- Ziel: AAS Leitungssatz OEM
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung der Leitungssätze, in denen eine Komponente verwendet wird

Use Case:

- Komponente hat Lieferengpass -> benachrichtigt automatisch alle Leitungssätze, in denen Sie verwendet wird
- Daten einer Komponente werden geändert -> benachrichtigt automatisch alle Leitungssätze, in denen Sie verwendet wird

Kommentar:

- Hoher Datenpflege-Aufwand
- Use Cases können auch über Rückwärtssuche der Beziehung 6.2.2.1.2 realisiert werden

6.2.2.1.6 Komponenten-Instanz wird in LS-Instanz verwendet

Klassifizierung:

- Quelle: AAS-Komponenten-Instanz Tier 1
- Ziel: AAS Leitungssatz-Instanz Tier 1
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung der Leitungssätze, in denen eine Komponente verwendet wird

Use Case:

- Traceability

Kommentar:

- Hoher Datenpflege-Aufwand
- Use Cases Könnte auch über Rückwärtssuche der Beziehung 6.2.2.1.3 realisiert werden

6.2.2.1.7 Leitungssatz enthält Halbfabrikate

Klassifizierung:

- Quelle: AAS Leitungssatz Tier 1
- Ziel: AAS Halbfabrikat Tier 1
- Kategorie: hierarchisch, top-down
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einem LS und den vom Tier 1 definierten Halbfabrikaten, aus denen sich dieser zusammensetzt

Use Case:

- MES-Tool kann erkennen, welche Halbfabrikate in einem LS enthalten sind und prüfen, ob das Halbfabrikat auf Lager liegt bzw. dessen Produktion anstoßen
- Wenn ein Halbfabrikat zugekauft wird, kann über dessen AAS die Beschaffung angestoßen werden (-> automatische Verhandlungsprozesse)

Kommentar:

- Inverse Beziehung zu 6.2.2.1.9
- Analoge Beziehung zu 6.2.2.1.1, 6.2.2.1.2, 6.2.2.1.11 und 6.2.2.1.15

6.2.2.1.8 Leitungssatz-Instanz enthält Halbfabrikat-Instanzen

Klassifizierung:

- Quelle: AAS Leitungssatz-Instanz Tier 1
- Ziel: AAS Halbfabrikat-Instanz Tier 1
- Kategorie: hierarchisch, top-down
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einer LS-Instanz und den vom Tier 1 definierten Halbfabrikat-Instanzen, aus denen sich dieser zusammensetzt

Use Case:

- Traceability

Kommentar:

- Inverse Beziehung zu 6.2.2.1.9
- Analoge Beziehung zu 6.2.2.1.3, 6.2.2.1.12 und 6.2.2.1.16

6.2.2.1.9 Halbfabrikat wird in Leitungssatz verwendet**Klassifizierung:**

- Quelle: AAS Halbfabrikat Tier 1
- Ziel: AAS Leitungssatz Tier 1
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung der Leitungssätze, in denen ein Halbfabrikat verwendet wird

Use Case:

- Zugeliefertes Halbfabrikat hat Lieferengpass -> benachrichtigt automatisch alle Leitungssätze, in denen es verwendet wird
- Daten eines Halbfabrikats werden geändert -> benachrichtigt automatisch alle Leitungssätze, in denen es verwendet wird

Kommentar:

- Hoher Datenpflege-Aufwand
- Use Cases können auch über Rückwärtssuche der Beziehung 6.2.2.1.7 realisiert werden

6.2.2.1.10 Halbfabrikat-Instanz wird in Leitungssatz-Instanz verwendet**Klassifizierung:**

- Quelle: AAS Halbfabrikat-Instanz Tier 1
- Ziel: AAS Leitungssatz-Instanz Tier 1
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-1

Beschreibung:

- Zuordnung zur Leitungssatz-Instanz, in der eine Halbfabrikats-Instanz verwendet wird

Use Case:

- Traceability

Kommentar:

- Hoher Datenpflege-Aufwand
- Use Cases können auch über Rückwärtssuche der Beziehung 6.2.2.1.8 realisiert werden

6.2.2.1.11 Halbfabrikat enthält Halbfabrikat**Klassifizierung:**

- Quelle: AAS Halbfabrikat Tier 1
- Ziel: AAS Halbfabrikat Tier 1
- Kategorie: hierarchisch, top-down
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einem Halbfabrikat und den vom Tier 1 definierten Unter-Halbfabrikaten, aus denen sich dieser zusammensetzt

Use Case:

- MES-Tool kann erkennen, welche Halbfabrikate in einem Halbfabrikat enthalten sind und prüfen, ob das Halbfabrikat auf Lager liegt bzw. dessen Produktion anstoßen
- Wenn ein Halbfabrikat zugekauft wird, kann über dessen AAS die Beschaffung angestoßen werden (-> automatische Verhandlungsprozesse)

Kommentar:

- Inverse Beziehung zu 6.2.2.1.13
- Analoge Beziehung zu 6.2.2.1.1, 6.2.2.1.2, 6.2.2.1.7 und 6.2.2.1.15

6.2.2.1.12 Halbfabrikat-Instanz enthält Halbfabrikat-Instanz**Klassifizierung:**

- Quelle: AAS Halbfabrikat-Instanz Tier 1
- Ziel: AAS Halbfabrikat-Instanz Tier 1
- Kategorie: hierarchisch, top-down
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einer Halbfabrikat-Instanz und den Unter-Halbfabrikat-Instanzen, aus denen sich dieses zusammensetzt

Use Case:

- Traceability

Kommentar:

- Inverse Beziehung zu 6.2.2.1.14
- Analoge Beziehung zu 6.2.2.1.3, 6.2.2.1.8 und 6.2.2.1.16

6.2.2.1.13 Halbfabrikat wird in Halbfabrikat verwendet**Klassifizierung:**

- Quelle: AAS Halbfabrikat Tier 1
- Ziel: AAS Halbfabrikat Tier 1
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung der Halbfabrikate, in denen ein Halbfabrikat verwendet wird

Use Case:

- Zugeliefertes Halbfabrikat hat Lieferengpass -> benachrichtigt automatisch alle Halbfabrikate, in denen es verwendet wird
- Daten eines Halbfabrikats werden geändert -> benachrichtigt automatisch alle Halbfabrikate, in denen es verwendet wird

Kommentar:

- Hoher Datenpflege-Aufwand
- Use Cases können auch über Rückwärtssuche der Beziehung 6.2.2.1.11 realisiert werden

6.2.2.1.14 Halbfabrikat-Instanz wird in Halbfabrikat-Instanz verwendet

Klassifizierung:

- Quelle: AAS Halbfabrikat Tier 1
- Ziel: AAS Halbfabrikat Tier 1
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-1

Beschreibung:

- Zuordnung zur Halbfabrikat-Instanz, in der eine Halbfabrikat-Instanz verwendet wird

Use Case:

- Traceability

Kommentar:

- Hoher Datenpflege-Aufwand
- Use Cases können auch über Rückwärtssuche der Beziehung 6.2.2.1.12 realisiert werden

6.2.2.1.15 Halbfabrikat enthält Komponente

Klassifizierung:

- Quelle: AAS Halbfabrikat Tier 1
- Ziel: AAS-Komponente Tier 1
- Kategorie: hierarchisch, top-down
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einem vom Tier 1 definierten Halbfabrikat und den enthaltenen Komponenten

Use Case:

- Engineering-Tool kann erkennen, welche Komponenten in einem Halbfabrikat enthalten sind und direkt die benötigten Daten aus deren AASen auslesen, bspw. für die Produktionsplanung
- MES-Tool kann erkennen, welche Komponenten in einem Halbfabrikat enthalten sind und prüfen, ob die Komponenten auf Lager liegen bzw. deren Beschaffung anstoßen

Kommentar:

- Inverse Beziehung zu 6.2.2.1.17
- Analoge Beziehung zu 6.2.2.1.1, 6.2.2.1.2, 6.2.2.1.7 und 6.2.2.1.11

6.2.2.1.16 Halbfabrikat-Instanz enthält Komponenten-Instanzen

Klassifizierung:

- Quelle: AAS Halbfabrikat-Instanz Tier 1
- Ziel: AAS-Komponenten-Instanz Tier 1
- Kategorie: hierarchisch, top-down
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zwischen einem vom Tier 1 produzierten Halbfabrikat und den enthaltenen Komponenten-Instanzen

Use Case:

- Traceability

Kommentar:

- Inverse Beziehung zu 6.2.2.1.18
- Analoge Beziehung zu 6.2.2.1.3, 6.2.2.1.8 und 6.2.2.1.12

6.2.2.1.17 Komponente wird in Halbfabrikat verwendet**Klassifizierung:**

- Quelle: AAS-Komponente Tier 1
- Ziel: AAS Halbfabrikat Tier 1
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung der Halbfabrikate, in denen eine Komponente verwendet wird

Use Case:

- Komponente hat Lieferengpass -> benachrichtigt automatisch alle Halbfabrikate, in denen sie verwendet wird
- Daten einer Komponente werden geändert -> benachrichtigt automatisch alle Halbfabrikate, in denen sie verwendet wird

Kommentar:

- Hoher Datenpflege-Aufwand
- Use Cases können auch über Rückwärtssuche der Beziehung 6.2.2.1.15 realisiert werden

6.2.2.1.18 Komponenten-Instanz wird in Halbfabrikat-Instanz verwendet**Klassifizierung:**

- Quelle: AAS-Komponenten-Instanz Tier 1
- Ziel: AAS Halbfabrikat-Instanz Tier 1
- Kategorie: hierarchisch, bottom-up
- Kardinalität: 1-zu-n

Beschreibung:

- Zuordnung zur Halbfabrikat-Instanz, in der eine Komponenten-Instanz verwendet wird

Use Case:

- Traceability

Kommentar:

- Beziehung hat die Kardinalität „1-zu-n“, da eine Komponenten-Instanz auch eine Charge repräsentieren kann
- Hoher Datenpflege-Aufwand
- Use Cases können auch über Rückwärtssuche der Beziehung 6.2.2.1.16 realisiert werden

6.2.2.2 Typ-Instanz-Beziehungen**6.2.2.2.1 Leitungssatz ist Instanz aus LS-Typ****Klassifizierung:**

- Quelle: AAS Leitungssatz-Instanz Tier 1
- Ziel: AAS Leitungssatz Tier 1

- Kategorie: Instanz ist von Typ
- Kardinalität: 1-zu-1

Beschreibung:

- Modellierung, dass eine bestimmte LS-Instanz einem bestimmten Typ genügt

Use Case:

- Bei Auftauchen eines Fehlers, Qualitätsproblems, o.ä. bei einer Instanz kann ermittelt werden, welcher Leitungssatz-Typ genau betroffen sind

Kommentar:

- Analoge Beziehung zu 6.2.2.2.3, □ und 6.2.2.2.7

6.2.2.2 Leitungssatz-Typ hat produzierte Instanzen**Klassifizierung:**

Quelle: AAS Leitungssatz Tier 1

- Ziel: AAS Leitungssatz-Instanz Tier 1
- Kategorie: Typ hat Instanz
- Kardinalität: 1-zu-n

Beschreibung:

- Beschreibung, welche Instanzen aus einem Leitungssatz-Typ produziert wurden

Use Case:

- Bei Auftauchen von Fehlern, Qualitätsproblemen, o.ä. kann ermittelt werden, welcher Leitungssatz-Instanzen genau betroffen sind

Kommentar:

- Inverse Beziehung zu 6.2.2.2.1
- Hoher Datenpflege-Aufwand
- Use Case Könnte auch über Rückwärtssuche der Beziehung 6.2.2.2.1 realisiert werden
- Analog zu Beziehung 6.2.2.2.4, 6.2.2.2.6 und 6.2.2.2.8

6.2.2.3 Halbfabrikat ist Instanz aus Halbfabrikat-Typ**Klassifizierung:**

- Quelle: AAS Halbfabrikat-Instanz Tier 1
- Ziel: AAS Halbfabrikat Tier 1
- Kategorie: Instanz ist von Typ
- Kardinalität: 1-zu-1

Beschreibung:

- Modellierung, dass eine bestimmte Halbfabrikat-Instanz einem bestimmten Typ genügt

Use Case:

- Bei Auftauchen eines Fehlers, Qualitätsproblems, o.ä. bei einer Instanz kann ermittelt werden, welcher Halbfabrikats-Typ genau betroffen sind

Kommentar:

- Analoge Beziehung zu 6.2.2.2.1, □ und 6.2.2.2.7

6.2.2.4 Halbfabrikat-Typ hat produzierte Instanzen**Klassifizierung:**

- Quelle: AAS Halbfabrikat Tier 1
- Ziel: AAS Halbfabrikat -Instanz Tier 1
- Kategorie: Typ hat Instanz
- Kardinalität: 1-zu-n

Beschreibung:

- Beschreibung, welche Instanzen aus einem Halbfabrikat-Typ produziert wurden

Use Case:

- Bei Auftauchen von Fehlern, Qualitätsproblemen, o.ä. kann ermittelt werden, welche Halbfabrikats-Instanzen genau betroffen sind

Kommentar:

- Inverse Beziehung zu 6.2.2.2.3
- Hoher Datenpflege-Aufwand
- Use Case Könnte auch über Rückwärtssuche der Beziehung 6.2.2.2.3 realisiert werden
- Analog zu Beziehung 6.2.2.2.2, 6.2.2.2.6 und 6.2.2.2.8

6.2.2.5 Komponente ist Instanz aus Komponenten-Typ

Klassifizierung:

- Quelle: AAS-Komponenten-Instanz Tier 1
- Ziel: AAS-Komponenten Tier 1
- Kategorie: Instanz ist von Typ
- Kardinalität: 1-zu-1

Beschreibung:

- Modellierung, dass eine bestimmte Komponenten-Instanz einem bestimmten Typ genügt

Use Case:

- Scannen einer Komponenten-Instanz vor dem Einbau in einen Leitungssatz -> Anhand des Typs wird die Freigabe für den Leitungssatz überprüft

Kommentar:

- Analoge Beziehung zu 6.2.2.2.1, 6.2.2.2.3 und 6.2.2.2.7

6.2.2.6 Komponenten-Typ hat produzierte Instanzen

Klassifizierung:

- Quelle: AAS-Komponente Tier 1
- Ziel: AAS-Komponenten-Instanz Tier 1
 - Kategorie: Typ hat Instanz
 - Kardinalität: 1-zu-n

Beschreibung:

- Beschreibung, welche Instanzen aus einem Komponenten-Typ produziert wurden

Use Case:

- Bei Auftauchen von Fehlern, Qualitätsproblemen, o.ä. kann ermittelt werden, welche Komponenten-Instanzen/Chargen genau betroffen sind

Kommentar:

- Inverse Beziehung zu □

- Hoher Datenpflege-Aufwand
- Use Case Könnte auch über Rückwärtssuche der Beziehung □ realisiert werden
- Analog zu Beziehung 6.2.2.2.2, 6.2.2.2.4 und 6.2.2.2.8

6.2.2.2.7 Tier2-Komponente ist Instanz aus Tier2-Komponenten-Typ

Klassifizierung:

- Quelle: AAS-Komponenten-Instanz Tier 2
- Ziel: AAS-Komponenten Tier 2
- Kategorie: Instanz ist von Typ
- Kardinalität: 1-zu-1

Beschreibung:

- Modellierung, dass eine bestimmte Komponenten-Instanz einem bestimmten Typ genügt

Use Case:

- Scannen einer Komponenten-Instanz vor dem Einbau in einen Leitungssatz -> Anhand des Typs wird die Freigabe für den Leitungssatz überprüft

Kommentar:

- Analoge Beziehung zu 6.2.2.2.1, 6.2.2.2.3 und □

6.2.2.2.8 Tier2-Komponenten-Typ hat produzierte Instanzen

Klassifizierung:

Quelle: AAS-Komponente Tier 2

- Ziel: AAS-Komponenten-Instanz Tier 2
- Kategorie: Typ hat Instanz
- Kardinalität: 1-zu-n

Beschreibung:

- Beschreibung, welche Instanzen aus einem Komponenten-Typ produziert wurden

Use Case:

- Bei Auftauchen von Fehlern, Qualitätsproblemen, o.ä. kann ermittelt werden, welche Komponenten-Instanzen/Chargen genau betroffen sind

Kommentar:

- Inverse Beziehung zu 6.2.2.2.7
- Hoher Datenpflege-Aufwand
- Use Case Könnte auch über Rückwärtssuche der Beziehung 6.2.2.2.7 realisiert werden
- Analog zu Beziehung 6.2.2.2.2, 6.2.2.2.4 und 6.2.2.2.6

6.2.2.3 Äquivalenz-/Entsprechungs-Beziehungen

6.2.2.3.1 Tier2-Komponente realisiert OEM-Komponente

Klassifizierung:

- Quelle: AAS-Komponente Tier 2
- Ziel: AAS-Komponente OEM
- Kategorie: Abgeleitet aus/Entspricht
- Kardinalität: 1-zu-n

Beschreibung:

- Tier2 beschreibt für seine Komponente, welche OEM-Komponente(n) diese realisiert

Use Case:

- Tier 1 erhält vom OEM einen Leitungssatz mit (OEM-)Komponenten und kann ermitteln, welchen Lieferanten er für die Realisierung nutzen möchte

Kommentar:

- Je nach Komponente legt der OEM entweder einen oder mehrere Lieferanten fest (s.u.), oder er überlässt die Auswahl dem Tier1

6.2.2.3.2 Tier1-Komponente realisiert OEM-Komponente

Klassifizierung:

- Quelle: AAS-Komponente Tier 1
- Ziel: AAS-Komponente OEM
- Kategorie: Abgeleitet aus/Entspricht
- Kardinalität: 1-zu-n

Beschreibung:

- Tier1 beschreibt für seine Komponente, welche OEM-Komponente(n) diese realisiert

Use Case:

- Grundlage für die Ersetzung von OEM-Sachnummern durch Tier1-Sachnummern

Kommentar:

- ---

6.2.2.3.3 Tier1-Leitungssatz setzt OEM-Leitungssatz um

Klassifizierung:

- Quelle: AAS Leitungssatz Tier 1
- Ziel: AAS Leitungssatz OEM
- Kategorie: Abgeleitet aus/Entspricht
- Kardinalität: 1-zu-1

Beschreibung:

- Beschreibung, dass ein LS beim Tier1 aus einem Leitungssatz beim OEM entstanden ist bzw. diesen realisiert

Use Case:

- Bei Druck von Leitungssatz-Etikett muss OEM-Sachnummer und Zeichnungsstand ermittelt und aufgedruckt werden
- Änderungen durch den OEM haben einen Änderungsprozess beim Tier1 zur Folge

Kommentar:

- ---

6.2.2.3.4 Tier1-Komponenten-Instanz entspricht Instanz/Charge bei Tier2

Klassifizierung:

- Quelle: AAS-Komponenten-Instanz Tier 1
- Ziel: AAS-Komponenten-Instanz Tier 2
- Kategorie: Abgeleitet aus/Entspricht

- Kardinalität: 1-zu-1

Beschreibung:

- Zuordnung von Komponenten-Instanzen bzw. Chargen beim Tier 1 zu der entsprechenden Instanz/Charge beim Tier 2

Use Case:

- Chargen-bezogene Daten wie bspw. Haltbarkeit müssen nicht separat in der Tier1-AAS gespeichert werden sondern es kann auf die Tier2-AAS verwiesen werden

Kommentar:

- ---

6.2.2.3.5 Tier1-LS-Instanz firmiert bei OEM als LS-Instanz**Klassifizierung:**

- Quelle: AAS Leitungssatz-Instanz Tier 1
- Ziel: AAS Leitungssatz-Instanz OEM
- Kategorie: Abgeleitet aus/Entspricht
- Kardinalität: 1-zu-1

Beschreibung:

- Tier 1 beschreibt für seine LS-Instanz, auf Basis welcher Order durch den OEM diese produziert wurde

Use Case:

- Datenhaltung beim Tier1, welche Änderungen bei einer LS-Instanz beinhaltet/berücksichtigt sind
- Tier 1 muss auskunftsähig sein, wenn beim OEM Fehler für eine bestimmte LS-Instanz auftreten

Kommentar:

- ---

6.2.2.4 Freigabe-Beziehungen**6.2.2.4.1 Freigabe von Tier2-Komponente für OEM-Komponente****Klassifizierung:**

- Quelle: AAS-Komponente OEM
- Ziel: AAS-Komponente Tier 2
- Kategorie: Freigabe
- Kardinalität: 1-zu-n

Beschreibung:

- OEM beschreibt für seine Komponente, welche Tier2-Komponente(n) zur Verwendung durch den Tier1 freigegeben sind

Use Case:

- Tier 1 erhält vom OEM einen Leitungssatz mit (OEM-)Komponenten und kann ermitteln, welche Lieferanten er für die Realisierung nutzen darf

Kommentar:

- Je nach Komponente legt der OEM entweder einen oder mehrere Lieferanten fest (s.u.), oder er überlässt die Auswahl dem Tier1
- Analoge Beziehung zu 6.2.2.4.2

6.2.2.4.2 Freigabe von Tier2-Komponente für Tier1-Komponente

Klassifizierung:

- Quelle: AAS-Komponente OEM
- Ziel: AAS-Komponente Tier 1
- Kategorie: Freigabe
- Kardinalität: 1-zu-n

Beschreibung:

- Tier 1 beschreibt für seine Komponente, welche Tier2-Komponente(n) zur Verwendung von ihm selbst freigegeben sind

Use Case:

- Nach der Ersetzung der OEM-Teilenummern durch seine eigenen Tier1-Sachnummern kann der Tier 1 ermitteln, welche Lieferanten er nutzen kann. Dazu nutzt er auch die Beziehung "Freigabe von Tier2-Komponente für OEM-Komponente"

Kommentar:

- Analoge Beziehung zu 6.2.2.4.1

6.2.2.4.3 Freigabe von OEM-Komponente für Tier2-Komponente

Klassifizierung:

- Quelle: AAS-Komponente Tier 2
- Ziel: AAS-Komponente OEM
- Kategorie: Freigabe
- Kardinalität: 1-zu-n

Beschreibung:

- Tier2 beschreibt für seine Komponente, für welche(n) OEM(s) bzw. welche OEM-Komponente(n) diese freigegeben ist

Use Case:

- Tier 1 kann ermitteln, ob ein als Lieferant ausgewählter Tier 2 für den aktuellen OEM freigegeben ist

Kommentar:

- Resultat des "Approved Production Part Process", den der Tier2 durchläuft

6.2.3 Typisierte Beziehungen und Merkmalsfestlegungen

6.2.3.1 Zuordnung von Prozessschritten zu verarbeiteten Komponenten

Klassifizierung:

- Quelle: Element aus Teilmodell „Produktionsprozess“ (Tier 1)
- Ziel: Element aus Teilmodell „Produkt“ (Tier 1)
- Kategorie: Typisierte Beziehung
- Kardinalität:

Beschreibung:

- Das Prozessmodell beschreibt die zur Fertigung eines LS/Halbfabrikats durchzuführenden Schritte (bspw. Schneiden, Crimpen, ...). Jeder Schritt bezieht sich dabei auf eine oder mehrere Komponente(n), bspw. Schneiden von Leitung x oder Crimpen von Kontaktteil y an Leitung x.

Use Case:

- MES/Verarbeitungsmaschine kann benötigte Parameter (bspw. Leitungslänge) für einen durchzuführenden Schritt automatisch aus der verlinkten Komponente aus dem Produktmodell auslesen

Kommentar:

- Eigentlich nicht Teil der Verbundkomponente, da lediglich Beziehung zwischen zwei Teilmodellen des gleichen Assets; muss aber trotzdem abgebildet werden

6.2.3.2 Prozessparameter festlegen (bspw. Crimp-Kraft)

Klassifizierung:

- Quelle: Element aus Teilmodell „Produktionsprozess“ (Tier 1)
- Ziel: Element aus Teilmodell „Produkt“ (Tier 1)
- Kategorie: Merkmalsfestlegung
- Kardinalität: 1-zu-1

Beschreibung:

- Bei der Zuordnung von Prozessschritten zu Komponenten (s.o.) sind manche Parameter bereits aus dem Produktmodell ersichtlich (bspw. Länge Leitung x). Andere Parameter werden jedoch explizit festgelegt (bspw. Leitungszugabe oder Crimp-Kraft).

Use Case:

- MES/Verarbeitungsmaschine kann benötigte Parameter (bspw. Crimp-Kraft) für einen durchzuführenden Schritt automatisch aus der Beziehung auslesen

Kommentar:

- Eigentlich nicht Teil der Verbundkomponente, da lediglich Beziehung zwischen zwei Teilmodellen des gleichen Assets; muss aber trotzdem abgebildet werden
- Erweiterung der Beziehung 6.2.2.2

6.2.3.3 Required Capability resultiert aus Prozessschritt

Klassifizierung:

- Quelle: Element aus Teilmodell „Produktionsprozess“ (Tier 1)
- Ziel: Element aus Teilmodell „Required Capabilities“ (Tier 1)
- Kategorie: Typisierte Beziehung
- Kardinalität: 1-zu-1

Beschreibung:

- Modellierung, aus welchem Prozessschritt eine "Required Capability" resultiert

Use Case:

- Bei Änderungen am Prozessmodell kann nachvollzogen werden, dass auch die "Required Capabilities" entsprechend angepasst werden müssen

Kommentar:

- Eigentlich nicht Teil der Verbundkomponente, da lediglich Beziehung zwischen zwei Teilmodellen des gleichen Assets; muss aber trotzdem abgebildet werden

6.2.4 Umsetzung der ermittelten Beziehungen

Um die in Abschnitt 6.2.2 ermittelten Beziehungen im Rahmen der VBK „Produkt“ abbilden zu können, müssen diese im Rahmen der entsprechenden Quell-Verwaltungsschale modelliert werden. Dabei erfordert jede der fünf in Abschnitt 6.2.2 ermittelten Kategorien von Beziehungen eine eigene Art der Umsetzung. Diese sind im Folgenden beschrieben.

6.2.4.1 Hierarchische Beziehungen

Zur Abbildung von hierarchischen Beziehungen bietet sich das Teilmodell „Bill of Material“ [30] der IDTA³⁰ an. Dieses ermöglicht sowohl die Abbildung von Top-Down- als auch von Bottom-Up-Beziehungen. Die gleichzeitige Abbildung sowohl von Top-Down- als auch von Bottom-Up-Beziehungen erfordert allerdings einen erhöhten Datenpflege-Aufwand. Daher wurde im Rahmen von AP5.2 entschieden, auf die explizite Abbildung von Bottom-Up-Beziehungen zu verzichten. Werden diese Informationen benötigt, so können diese stattdessen auch über eine Rückwärtssuche der inversen Beziehung realisiert werden.

Abbildung 6-3 zeigt exemplarisch ein BOM-Modell, welche die Beziehungen „Leitungssatz enthält Komponenten“ umsetzt.

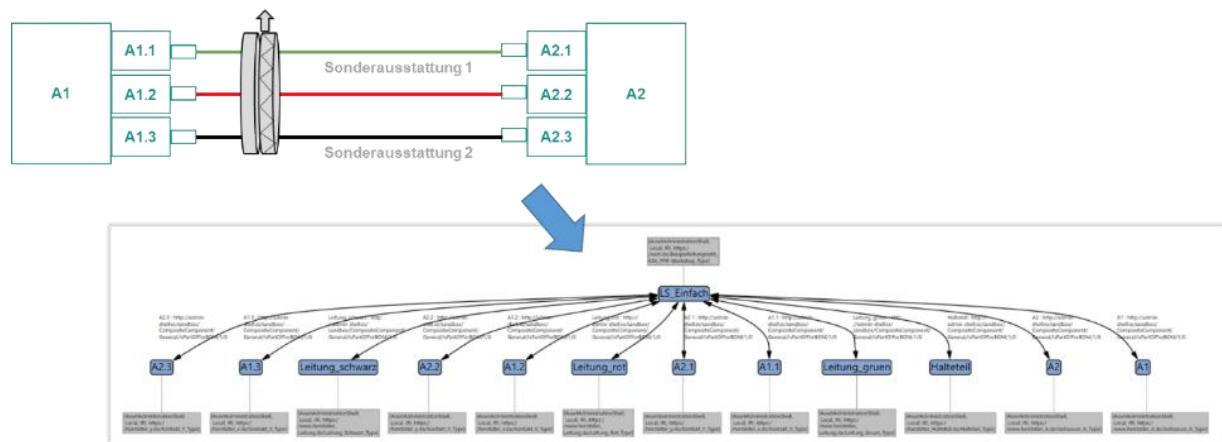


Abbildung 6-3: Umsetzung hierarchischer Beziehungen mit Hilfe des BOM-Teilmodells

Unterschiedliche Hierarchien sollen in unterschiedlichen BOM-Teilmodellen umgesetzt werden, um für jedes Teilmodell einen klaren Fokus herzustellen. Dies bedeutet für den Leitungssatz bei einem Tier1 bspw., dass unterschiedliche BOM-Teilmodelle für die Komponentensicht (aus welchen Einzelkomponenten setzt sich der Leitungssatz zusammen), die Modulsicht (welche bestellbaren Module definiert das Produktmodell) sowie die Fertigungs-BOM (aus welchen produzierbaren Halbfabrikaten wird der Leitungssatz bzw. ein einzelnes bestellbares Modul gefertigt) existieren.

6.2.4.2 Typ-Instanz-Beziehungen

Typ-Instanz-Beziehungen können in der Verwaltungsschale nativ mit Hilfe der „derivedFrom“-Referenz aus dem AAS-Metamodell³¹ (s. Abschnitt 4.2.2) umgesetzt werden. Analog zu den hierarchischen Beziehungen sollen aus Gründen des Datenpflege-Aufwands lediglich die Beziehungen „Instanz ist von Typ“ explizit modelliert werden. Die inversen Beziehungen „Typ hat Instanzen“ lassen sich auch hier wiederum durch eine Rückwärtssuche ermitteln.

6.2.4.3 Äquivalenz-/Entsprechungs-Beziehungen

Die Notwendigkeit der expliziten Modellierung von Ableitungs- bzw. Entsprechungsbeziehungen ergibt sich aus den Tatsachen, dass das gleiche Asset in der Regel bei unterschiedlichen Firmen/Tiers unter unterschiedlichen Bezeichnern/Sachnummern firmiert. Darüber hinaus kann es notwendig sein, dass

³⁰https://industrialdigitaltwin.org/wp-content/uploads/2023/04/IDTA-02011-1-0_Submodel_HierarchicalStructuresEnabling-BoM.pdf

³¹https://industrialdigitaltwin.org/wp-content/uploads/2023/04/IDTA-01001-3-0_SpecificationAssetAdministrationShell_Part1_Metamodel.pdf

jede Firma zu einem Asset (in ihrer eigenen Verwaltungsschale) eigene Daten wie z.B. spezifische Grenzwerte oder Verarbeitungsspezifikationen hinterlegen möchte.

Für die Modellierung solcher Ableitungs- bzw. Entsprechungsbeziehungen bietet Version 3 des Metamodells der Verwaltungsschale³¹ das Konzept der „SpecificAssetID“ (s. Abbildung 6-4) an. Mit solchen „specific asset IDs“ können für ein Asset neben der eigentlichen assetID („GlobalAssetID“), die einem bestimmten Format unterliegen muss (IRI/IRDI), beliebig viele weitere Bezeichner ergänzt werden. Solche Bezeichner können dabei z.B. die Sachnummern sein, unter denen das Asset bei anderen Firmen geführt wird.

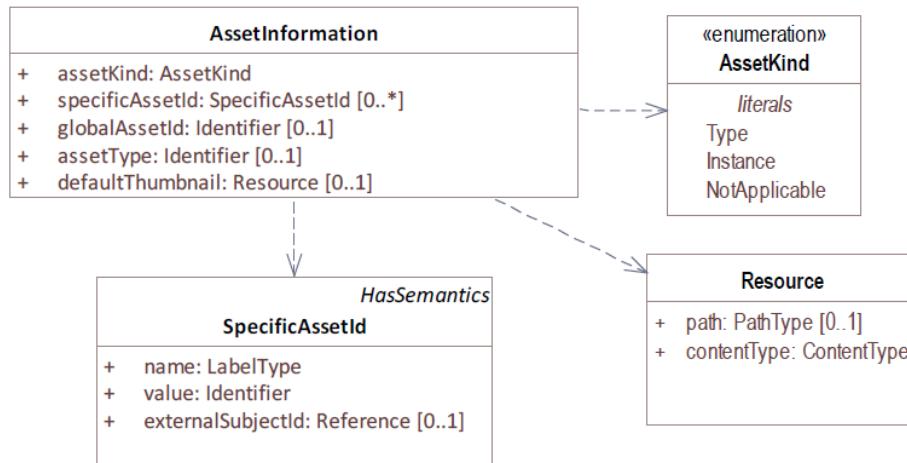


Abbildung 6-4: Konzept der "Specific Asset ID" aus dem Metamodell der VWS (s. Abschnitt 5.3.4)

Für die konkrete Umsetzung bedeutet dies, dass

- für jeden Verweis auf eine externe Sachnummer bzw. auf ein externes Asset eine „SpecificAssetId“ angelegt wird,
- das Attribut „name“ der ID auf einen entsprechenden sinnvollen Wert festgelegt wird (bspw. „partNumberOEM1“),
- das Attribut „value“ auf die entsprechende ID/Sachnummer gesetzt wird und
- (optional) mit Hilfe der „externalSubjectId“ explizit die referenzierte Firma ausgewiesen wird (z.B. über ihre URL).

Abbildung 6-5 zeigt die exemplarische Umsetzung anhand einer Verwaltungsschale für einen Kontakt „MLK 1,2“ bei einem Tier1, die auf die entsprechenden Sachnummern bei zwei OEMs verweist.

| AssetAdministrationShell (according IEC63278) |
|--|
| Referable: |
| idShort: AAS_MLK_1_2_Tier1 |
| description: [en] AAS for an MLK 1,2 at Tier1 |
| HasExtension: |
| Identifiable: |
| id: https://tier1.com/ids/sm/8272_7002_6032_6622 |
| id (Base64): aHR0cHM6Ly90aWVvMS5jb20vaWRzL3NtLzgyNzJfNzAwMI82MDMyXzY2MjI= |
| HasDataSpecification (Reference): |
| AssetInformation |
| Kind (of AssetInformation): |
| kind: Type |
| globalAssetId: |
| globalAssetId: https://tier1.com/ids/asset/9474_8091_6032_5870 |
| assetType: |
| specificAssetId: |
| Pair 1: partNumber |
| semanticId: (GlobalReference) 0173-1#02-AAO676#003 |
| key: partNumber |
| value: tier1-0815 |
| externalSubjectId: (GlobalReference) https://tier1.com |
| Pair 2: partNumberOEM1 |
| semanticId: (GlobalReference) 0173-1#02-AAO676#003 |
| key: partNumberOEM1 |
| value: oem1-1234 |
| externalSubjectId: (GlobalReference) https://oem1.com |
| Pair 3: partNumberOEM2 |
| semanticId: (GlobalReference) 0173-1#02-AAO676#003 |
| key: partNumberOEM2 |
| value: oem2-ABCD |

Abbildung 6-5: Exemplarische Umsetzung des Sachnummern-Mapping mittels „specificAssetId“

Neben solchen „globalen“ Äquivalenzbeziehungen zwischen Assets kann es auch noch „lokale“ Äquivalenzbeziehungen geben, die bspw. eine Äquivalenz zwischen Entitäten unterschiedlicher BOM-Teilmodelle ausdrücken. Ein Beispiel hierfür stellt die Äquivalenz zweier Leitungs-Instanzen in unterschiedlichen Halbfabrikaten dar. Solche Beziehungen lassen sich mit Hilfe von „sameAs“-Beziehungen aus dem BOM-Teilmodell beschreiben.

6.2.4.4 Freigabe-Beziehungen

Die Modellierung von Freigabe-Listen ist rein mit Elementen des Metamodells der Verwaltungsschale oder mit existierenden AAS-Teilmodellen nicht umsetzbar. Aus diesem Grund wurde im Rahmen von AP5.2 ein eigenes Teilmodell zur Modellierung solcher Freigabe-Listen umgesetzt, welches im Folgenden beschrieben wird.

Grundlage des Teilmodells ist, dass jede Freigabe für das Asset, welches die aktuelle Verwaltungsschale beschreibt, in einer eigenen SubmodelElementCollection beschrieben wird. Kern dieser Freigabe ist ein Verweis auf das freigegebene Asset (realisiert über eine entsprechende Referenz). Ergänzt wird diese Beschreibung der Freigabe um Metadaten (Zeitpunkt der Freigabe, Stand der Freigabe sowie Informationen über die freigebende Person). Darüber hinaus können für eine Freigabe beliebig viele Bedingungen bzw. Einschränkungen in Form sog. ConditionSets angegeben werden. Eine typische Einschränkung ist bspw., dass ein Produkt nicht mehr für Neuentwicklungen freigegeben ist, weil es ausläuft. Weitere Einschränkungen können bspw. länderspezifischer oder zeitlicher Natur sein.

Abbildung 6-6 zeigt die gesamte Struktur des Teilmodells.

| | |
|--------------|--|
| SM | <T> "ApprovedAssets" [IRI, https://example.com/ids/sm/3045_3111_5032_7080] |
| SMC | <T> "ApprovedAsset{0:00}" (5 elements) @{SMT/Cardinality=ZeroToMany} @{FormTitle=Approved Asset} |
| Ref | <T> "AssetRef" ~> [Asset, not Local, IRI, https://example.com/ids/asset/5594_7082_1122_7166] |
| SMC | <T> "ApprovalPerson" (3 elements) @{SMT/Cardinality=One} @{FormInfo=The person that can approve} |
| MLP | <T> "NameOfContact" -> Somebody @{SMT/Cardinality=One} @{FormInfo=(Last) Name of contact} |
| MLP | <T> "FirstName" -> Somebody @{SMT/Cardinality=ZeroToOne} @{FormInfo=First name of contact} |
| SMC | <T> "Email{0:00}" (1 elements) @{SMT/Cardinality=OneToMany} @{Multiplicity=OneToMany} |
| Prop | <T> "ApprovalTimestamp" = 2020-01-22T16:38:09Z @{SMT/Cardinality=One} @{FormInfo=The approval timestamp} |
| Prop | <T> "ApprovalState" = APPROVED @{SMT/Cardinality=One} @{SMT/AllowedValue=(APPROVED)} |
| SMC | <T> "ApprovalConditions" (1 elements) @{SMT/Cardinality=ZeroToOne} @{Multiplicity=ZeroToOne} |
| SMC | <T> "ConditionSet{0:00}" (10 elements) @{SMT/Cardinality=OneToMany} @{Multiplicity=Or} |
| Range | <T> "ApprovedTime{0:00}" = 2021-01-01T00:00:00Z .. 2021-05-31T23:59:59Z @{SMT/Cardinality=ZeroToMany} |
| Range | <T> "NotApprovedTime{0:00}" = 2021-01-01T00:00:00Z .. 2021-05-31T23:59:59Z @{SMT/Cardinality=ZeroToMany} |
| Prop | <T> "ApprovedState{0:00}" = NEW_SERIES @{SMT/Cardinality=ZeroToMany} @{SMT/All} |
| Prop | <T> "NotApprovedState{0:00}" = NEW_SERIES @{SMT/Cardinality=ZeroToMany} @{SMT/All} |
| Prop | <T> "ApprovedLocation{0:00}" = Germany @{SMT/Cardinality=ZeroToMany} @{SMT/Entity} |
| Prop | <T> "NotApprovedLocation{0:00}" = Germany @{SMT/Cardinality=ZeroToMany} @{SMT/Entity} |
| Prop | <T> "ApprovedOEM{0:00}" = Mercedes @{SMT/Cardinality=ZeroToMany} @{SMT/Either} |
| Prop | <T> "NotApprovedOEM{0:00}" = Mercedes @{SMT/Cardinality=ZeroToMany} @{SMT/Either} |
| Prop | <T> "ApprovedOEMBusinessUnit{0:00}" = Mercedes-Benz Vans @{SMT/Cardinality=ZeroToMany} |
| Prop | <T> "NotApprovedOEMBusinessUnit{0:00}" = Mercedes-Benz Vans @{SMT/Cardinality=ZeroToMany} |

Abbildung 6-6: Teilmodell "Freigabeliste"

6.2.4.5 Typisierte Beziehungen und Merkmalsfestlegungen

Wie bereits im Rahmen der Definition der einzelnen Beziehungen im Rahmen von Abschnitt 6.2.3 beschrieben, sind diese Beziehungen nicht Teil der Verbundkomponente, müssen aber nichtsdestotrotz abgebildet werden. Diese Abbildung muss dabei im Rahmen des entsprechenden Teilmodells erfolgen, das die Quelle der Beziehung darstellt. Aus diesem Grund wird an dieser Stelle nicht näher auf die konkrete Umsetzung eingegangen.

6.3 AP 5.3 - Verbundkomponente Ressource (Produktionsmittel) und VIBN

6.3.1 Zielsetzung

Das AP 5.3 beschäftigte sich mit der Ableitung und Definition der Verbundkomponente (VBK) „Ressource“, d.h. für die Produktionsmittel.

Zur Ableitung wurden zunächst zwei Beispiel-Ressourcen analysiert, die typische Vertreter der Produktionsmittel in der Domäne „Leitungssatz“ darstellen. Diese Beispiel-Ressourcen werden in Abschnitt 6.3.2 genauer vorgestellt. Basierend auf diesen Ressourcen wurden typische Use Cases analysiert, die auf Basis des Konzepts der Verbundkomponente realisiert werden können bzw. für die die Beziehungen zwischen den einzelnen Teilen der Verbundkomponente relevant sind. Diese werden in Abschnitt 6.3.3 vorgestellt. Basierend auf diesen Use Cases wurde in Abschnitt 6.3.4 das Konzept der Verbundkomponente „Ressource“ definiert.

In Abschnitt 0 wurde das Thema „Virtuelle Inbetriebnahme“ (VIBN) als wesentlicher Bestandteil der Digitalen Fabrik für Produktionsressourcen des Leitungssatzes besprochen.

Im Rahmen von AP5.3 wurden insgesamt die folgenden Use Cases definiert:

1. Finden einer Maschine + Werkzeug für eine bestimmte Fähigkeit
2. Ermittlung des zur Erfüllung einer Fähigkeit notwendigen Werkzeugs
3. Konfiguration und Bestellung einer Maschine
4. VIBN für Antriebseinheit vom Komponentenlieferant
5. VIBN für Prüfmodul

6.3.2 Beispiel-Ressourcen

Verbundkomponenten betrachten den Aufbau sowie die Struktur zusammengesetzter Systeme auf Basis von Teilsystemen und Komponenten. Um die relevanten, mit Hilfe der Verwaltungsschale nachzubildenden Strukturen zu ermitteln, lohnt zunächst eine Analyse typischer, zu betrachtender Systeme. Zu diesem Zweck wurden zwei repräsentative Vertreter von Produktionsmitteln in der Domäne „Leitungssatz“ ausgewählt, die im Folgenden kurz beschrieben werden sollen.



Abbildung 6-7: Beispiel-Ressourcen 2 – Wezag UP 150 (links) und Komax Sigma 688 (rechts)

Als erstes Beispiel wurde die halbautomatische hydraulische Crimpmaschine [UP 150 von Wezag](#) ausgewählt (s. Abbildung 6-7, links). Diese kann per Fußschalter von einem Mitarbeiter bedient werden und Kontakte bis zu einem relativ großen Querschnitt verarbeiten. Die Maschine an sich realisiert dabei im Prinzip lediglich einen Pressvorgang – erst durch Einsatz eines Werkzeugs (Crimpgesenk) sowie eines zugehörigen Adapters wird durch den Pressvorgang ein Crimpvorgang. Crimpvorgänge können bei der Wezag UP 150 zusätzlich durch eine automatische Crimpkraftanalyse begleitet werden, durch die bspw. auf die Qualität bzw. sdas ordnungsgemäße Ausführen des Crimp-Vorgangs geschlossen werden kann.

Für das zweite untersuchte Beispiel wurde die vollautomatische [Sigma 688 der Firma Komax](#) herangezogen (s. Abbildung 6-7, rechts). Je nach enthaltenen Modulen kann sie unterschiedliche Fähigkeiten

wie z.B. Schneiden, Abisolieren oder Crimpen realisieren und dabei vollautomatisiert komplette Chargen von Produkten herstellen. Eine Übersicht über mögliche Fähigkeiten ist in Abbildung 6-8 dargestellt. Zusätzlich zu den in Abbildung 6-8 dargestellten Fähigkeiten ergeben sich je nach verbauten Modulen noch zugehörige Prüf-/Analysefähigkeiten. So kann z.B. überprüft werden, ob bei einem Abisoliervorgang die Leitung beschädigt wurde. Analog zur oben beschriebenen UP 150 benötigt auch die Sigma 688 für das Crimpen ein entsprechendes Werkzeug, welches hier *Applikator* genannt wird.

Processing examples

| | | | |
|---|---|--|---|
| Twisted pair (incl. with open wire ends of different lengths) |  | Seal insertion |  |
| Cutting to length |  | Split cycle function for closed terminals |  |
| Half stripping |  | Cutting pulled strands |  |
| Full stripping |  | Inkjet marking |  |
| Crimping |  | | |

Abbildung 6-8: Übersicht möglicher Fähigkeiten Komax Sigma 688

Für beide Maschinen gilt, dass diese für den Kunden prinzipiell eine Blackbox darstellen – der innere Aufbau ist für den Kunden folglich – mit Ausnahme der Möglichkeiten zum Montieren spezieller Werkzeuge – nicht ersichtlich und auch nicht relevant. Dies bezieht sich vor allem bei der Sigma 688 auch auf die verbauten Module: Diese werden in aller Regel einmal bei der Bestellung der Maschine ausgewählt und sind dann – für den Kunden – fest verbaut und Teil der Maschine.

6.3.3 Betrachtete Use Cases

Bereits im letzten Abschnitt wurde die Tatsache beschrieben, dass die Maschinen zumindest für den Kunden (Leitungssatzkonfektionär) eine Blackbox darstellen. Der innere Aufbau dieser Maschinen bspw. aus unterschiedlichen Komponenten ist daher für den Hauptfokus dieses Projekts – die Auswahl und Nutzung der Maschinen durch den Kunden zur Produktion eines Leitungssatzes – nicht relevant. Eine Ausnahme stellen dabei die Möglichkeiten und Voraussetzungen zur Montage von Werkzeugen dar.

Folglich sind auch die im Rahmen der Verbundkomponente umzusetzenden Beziehungen entsprechend limitiert und beschränken sich auf die sich daraus ergebenden Use Cases:

6.3.3.1 UC1: Finden einer Maschine + Werkzeug für eine bestimmte Fähigkeit

Haupt-Use Case besteht im Finden einer (beim Kunden vorhandenen) Maschine bzw. Maschinen-Instanz, die eine oder mehrere bestimmte Fähigkeiten ausführen kann. Hierbei ist neben dem Vorhandensein der eigentlichen Maschine falls für die entsprechende Fähigkeit notwendig auch das Vorhandensein eines entsprechenden, mit der Maschine kompatiblen Werkzeugs zu prüfen.

6.3.3.2 UC2: Ermittlung des zur Erfüllung einer Fähigkeit notwendigen Werkzeugs

Eng im Zusammenhang mit UC1 steht die Frage danach, welches Werkzeug in einer Maschine montiert werden muss, um eine bestimmte Fähigkeit ausführen zu können. Im Speziellen ist hierbei auch die Frage interessant, welcher Adapter/Modul evtl. benötigt wird, um das Werkzeug montieren zu können (vgl. Abschnitt 6.3.2).

6.3.3.3 UC3: Konfiguration und Bestellung einer Maschine

Einen verwandten und doch erweiterten Use Case stellt die Bestellung einer neuen Maschine durch einen Leitungssatz-Konfektionär bei einem Maschinenlieferanten dar. Hierbei wird der Konfektionär mit

einer oder mehreren benötigten Fähigkeiten an den Maschinenlieferanten herantreten, welche daraufhin alle durch ihn angebotenen Maschinentypen durchsuchen und bewerten muss. Diese Bewertung erstreckt sich dabei vor allem auch auf die möglichen Kombinationen von Maschinen mit Erweiterungsmodulen (vgl. Abschnitt 6.3.2).

6.3.4 Konzept der Verbundkomponente „Ressource“

Die in Abschnitt 6.3.3 betrachteten Use Cases fokussieren auf die Auswertung von durch eine Maschine im Verbund mit einem Werkzeug bereitgestellte Fähigkeiten. Abbildung 6-9 illustriert dies anhand einer Einordnung in den PPR-Kontext.

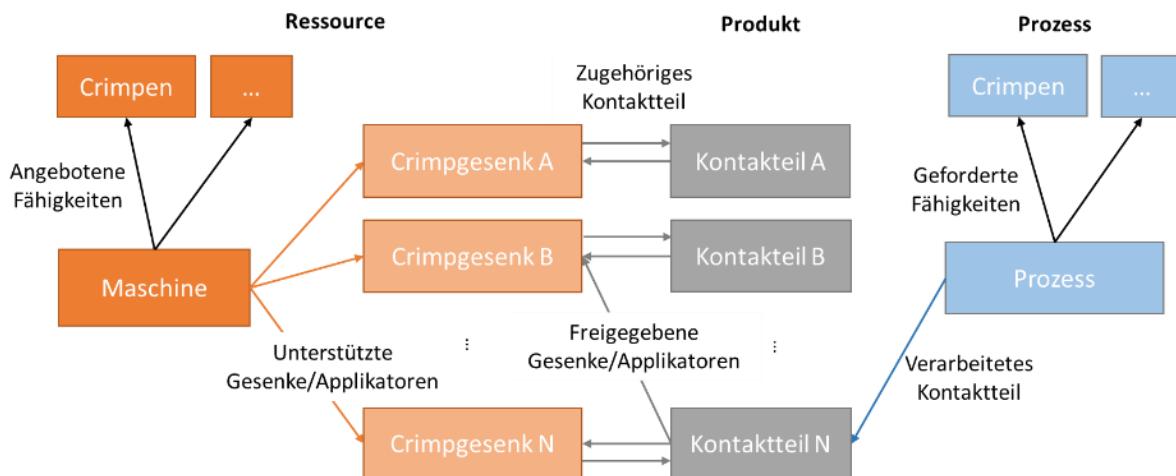


Abbildung 6-9: Einordnung in das PPR-Modell

Für die Bewertung, ob eine Maschine eine bestimmte Fähigkeit ausführen kann, spielen – neben der grundsätzlichen Eignung der Maschine für die Fähigkeit – im Wesentlichen zwei Aspekte eine Rolle:

1. Welches konkrete Werkzeug wird für die Fähigkeit benötigt?
2. Kann das Werkzeug in die gewählte Maschine montiert werden?

Entsprechend sind diese beiden Aspekte im Rahmen der Verbundkomponente „Ressource“ abzubilden.

Für die Auswahl eines passenden Werkzeugs ist dessen formale *Freigabe* das entscheidende Kriterium. So werden z.B. am Beispiel *Crimpen* für jedes Kontaktteil durch dessen Hersteller ein oder mehrere zu nutzende Werkzeuge explizit freigegeben und damit vorgegeben. Diese Beziehungen lassen sich durch das bereits im Rahmen von AP5.2 entwickelte Teilmodell „[Freigabeliste](#)“ abbilden. Basierend auf „[Sigma688_ApprovedApplicators_Submodel_withoutFiles.aasx](#)“³² zeigt Abbildung 6-10 dies exemplarisch anhand des Kontaktteils MLK 14,5 der Firma Kostal: Abgebildet sind hier die Freigaben für zwei unterschiedliche Werkzeuge – zum einen ein Applikator der Firma Hanke/Demirel und zum anderen eine Hand-Crimpzange der Firma EHC.

Der zweite Aspekt – die Kompatibilität eines Werkzeugs mit einer Maschine evtl. unter Nutzung eines entsprechenden Applikators – lässt sich in zwei Teil-Aspekte aufteilen:

Die aktuelle Maschinenstruktur, die eine Aussage über aktuell montierte Werkzeuge und damit über die aktuell, ohne Umbau zu realisierenden Fähigkeiten zulässt, lässt sich einfach mit Hilfe des bereits standardisierten Teilmodells „[IDTA 02011: Hierarchical Structures enabling Bills of Material](#)“ (BOM) [30] abbilden. Abbildung 6-10 zeigt dies Anhand der Maschine *Sigma 688*, welche mit einem Crimp-Modul C1340 ausgestattet ist, in welchem wiederum ein Werkzeug (Applikator) *Schäfer 20* montiert ist. Ob das Crimp-Modul explizit modelliert wird oder nicht, ist dabei abhängig vom konkreten umzusetzenden Use Case (vgl. Abschnitt 6.3.3).

Die Abbildung der theoretisch abbildbaren Maschinenstrukturen ist dagegen komplexer – vor allem da sich je nach konkretem Beispiel eine Vielzahl an möglichen Kombinationen ergeben können. Eine Nutzung des BOM-Teilmodells, wie für den ersten Teilaspekt vorgeschlagen, ist dabei aufgrund des hohen Modellierungsaufwands nicht sinnvoll, wie in Abbildung 6-12 zu sehen ist: So muss bei

³² https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP05/Beispieldaten/Sigma688_ApprovedApplicators_Submodel_withoutFiles.aasx

Einführung eines neuen Crimpmoduls explizit jeder kompatible Applikator angegeben werden bzw. bei Einführung eines neuen Applikators jedes kompatible Cirmp-Modul. Stattdessen ist eine implizite Modellierung sinnvoll, die auf einer Abbildung der – bspw. zwischen Modulen und Applikatoren genutzten – Schnittstellen basiert. Eine solche Modellierung lässt sich bspw. auf Basis der „specificAssetId“ umsetzen, die bereits im Rahmen von AP5.2 genutzt wurde.

| |
|---|
| AAS "CrimpTerminal_PLK_14_5" V1.0 [IRI, https://example.com/ids/aas/8391_4152_5022_5597] of [IRI, https://example.com/ids/asset/5] |
| ↳ SM "Nameplate" [IRI, www.example.com/ids/sm/0200_3113_3022_3161] |
| ↳ SM "ContactInformation" [IRI, www.example.com/ids/sm/9110_3113_3022_8966] |
| ↳ SM "TechnicalData" V1.1 [IRI, www.example.com/ids/sm/0310_3113_3022_6366] |
| ↳ SM "ApprovedApplicators" [IRI, https://example.com/ids/sm/3162_7091_6032_3431] |
| TBD ApprovedAssets ready |
| ↳ SMC "ApprovedApplicator01" (5 elements) |
| Ref "CrimpingApplicator_Hanke_Demirel_10019253" ~> [AssetAdministrationShell, Local, Custom, AssetAdministrationShell] |
| ↳ SMC "ApprovalPerson" (3 elements) |
| Prop "ApprovalTimestamp" = 2020-01-22T16:38:09Z |
| Prop "ApprovalState" = APPROVED |
| ↳ SMC "ApprovalConditions" (1 elements) |
| ↳ SMC "ConditionSet01" (1 elements) |
| Prop "ApprovedWireCrossSection" = 35 [mm^2] |
| ↳ SMC "ApprovedApplicator02" (4 elements) |
| Ref "CrimpingTool_EHC_10_13" ~> [AssetAdministrationShell, Local, Custom, AssetAdministrationShell---4D5413B7] |
| ↳ SMC "ApprovalPerson" (3 elements) |
| Prop "ApprovalTimestamp" = 2020-01-22T16:38:09Z |
| Prop "ApprovalState" = IN_PROGRESS |
| AAS "CrimpingApplicator_Hanke_Demirel_10019253" [Custom, AssetAdministrationShell---7959AEB3] |
| AAS "CrimpingTool_EHC_10_13" [Custom, AssetAdministrationShell---4D5413B7] |

Abbildung 6-10: Modellierung von Werkzeug-Freigaben am Beispiel Crimp-Kontakt

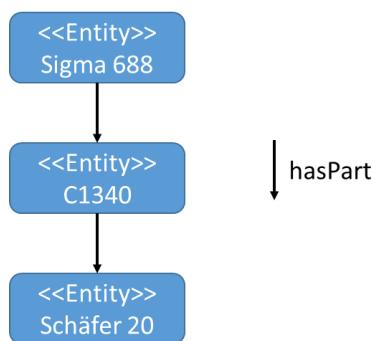


Abbildung 6-11: Abbildung der aktuellen Maschinenstruktur mit Hilfe des BOM-Teilmodells

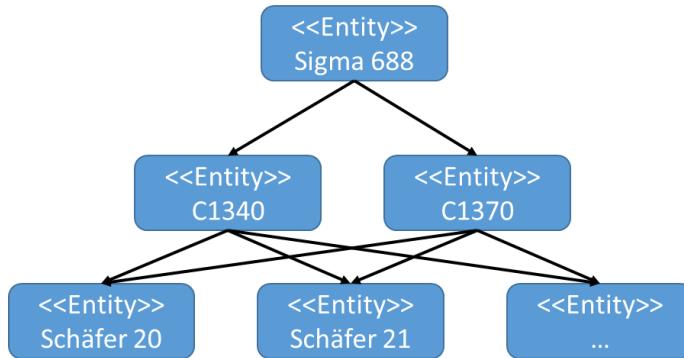


Abbildung 6-12: Abbildung potenzieller Maschinenstrukturen mit Hilfe des BOM-Teilmodells

Abbildung 6-13 zeigt eine mögliche Implementierung, wo die Kompatibilität sowohl zwischen Maschinen und Modulen als auch zwischen Modulen und Werkzeugen mit Hilfe von specificAssetIds „offered/requiredSlot“ abgebildet wird. Die Einführung bspw. eines neuen Werkzeugs gestaltet sich so extrem einfach, da lediglich für das Werkzeug der entsprechende „requiredSlot“ angegeben werden muss. Eine Anpassung der Verwaltungsschalen der verschiedenen Module ist nicht notwendig.

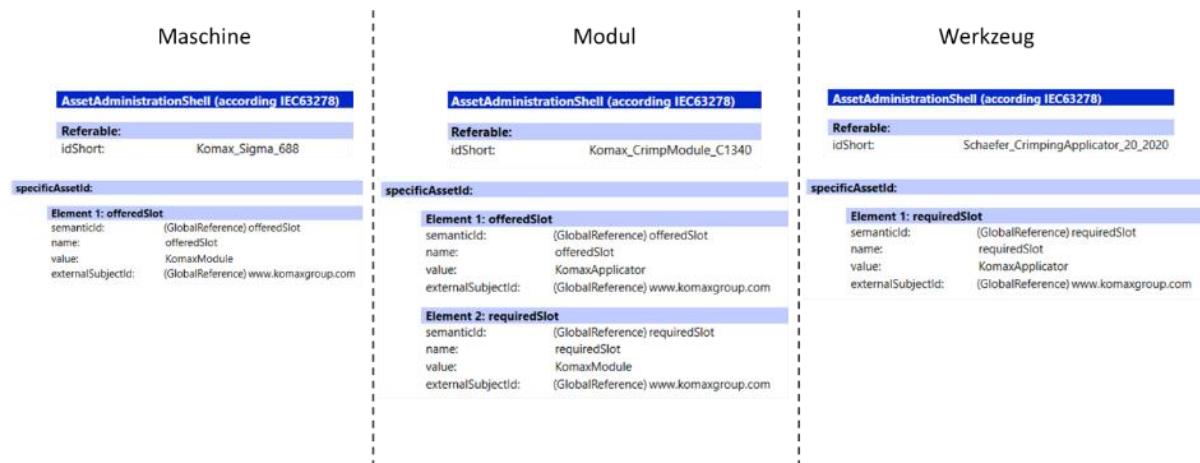


Abbildung 6-13: Modellierung potenzieller Maschinenstrukturen per "specificAssetId"

6.3.5 Virtuelle Inbetriebnahme

Die Virtuelle Inbetriebnahme (VIBN) ist ein wesentlicher Bestandteil der Digitalen Fabrik und beschreibt eine Simulationsmethodik im digitalen Engineering. Die Idee dahinter ist, dass ein virtuelles Abbild der Anlage bzw. der Maschine bereits in der frühen Phase der Entwicklung erstellt wird. Mit diesem virtuellen Abbild, dass das Verhalten der realen Anlage hinreichend gut beschreibt, wird die Steuerungstechnik bzw. der Steuerungsablauf über das Prozessdatenbild einer realen Steuerung getestet. Bei dem klassischen, sequenziellen Engineering (ohne VIBN), beispielsweise „Wasserfallmodell“, werden alle Phasen nacheinander ausgeführt und Prüfungen erst bei dem Factory Acceptance Test (FAT) bzw. bei der Inbetriebnahme (IBN) durchgeführt, was generell zu einer späten Fehlererkennung führt. Eine Studie des Vereins Deutscher Werkzeugmaschinen-fabriken e.V.³³ (VDW) besagt, dass die IBN 15 bis 25% der Gesamtdurchlaufzeit eines Anlagenprojekts ausmacht. Davon ist 90% der Zeit für die IBN von Elektronik und Steuerungstechnik aufzuwenden, wovon wiederum 70% auf Softwarefehlern beruhen. Durch die Einführung der VIBN bereits in der frühen Phase des Entwicklungsprozesses können die Tätigkeiten der Disziplinen Mechanik, Elektronik und Software stärker parallelisiert werden. Es entwickelt sich ein viel früheres Systemverständnis der Anlage. Dies ermöglicht eine frühzeitige Absicherung und Optimierung der Steuerungsprogramme und somit eine Verkürzung der Entwicklungs- und Inbetriebnahmezeiten.

In dem Prozess der VIBN können alle Partner der Wertschöpfungskette involviert sein. Neben dem Maschinenhersteller, der das virtuelle Abbild der Maschine erstellt, und dem Endanwender/OEM, der die reale wie auch virtuelle Anlage betreibt und verwendet, wird die Rolle des Komponentenherstellers zunehmend wichtiger. Für das Erstellen virtueller Anlagenmodelle wird oft das Wissen über die Verhaltensbeschreibung der eingesetzten Komponenten für die Simulation benötigt. Durch die Bereitstellung von Simulationsmodellen durch die Komponentenhersteller hätten Maschinenhersteller weniger Aufwände bei der Erstellung solcher Modelle und würden Modelle in hoher Qualität erhalten, da sie direkt auf das Domänenwissen des Komponentenherstellers zugreifen können.

Um die Bereitstellung und den Austausch von Simulationsmodellen in einer heterogenen Toollandschaft zu ermöglichen, wurde seitens der [Modelica Association](#) der Functional Mock-Up Interface-Standard³⁴ ([FMI](#)) erarbeitet. Der FMI-Standard ist für die Simulationswelt das, was das [STEP-Format](#) in der CAD-Welt ist. Simulationsdateien in Form von Functional Mock-Up Units (FMU) besitzen nach außen hin standardisierte Schnittstellen. Im Inneren befindet sich das Verhaltensmodell als Black-Box. Diese FMU kann von allen Softwaretools verwendet werden, die den FMI-Standard unterstützen.

Um die Entwicklung des Modellaustauschs über die Komponentenebene der FMU/FMI hinaus fortzusetzen, wurde unter dem Dach der Modelica Association ein neues Projekt mit dem Namen "System Structure and Parameterization of Components for Virtual System Design"³⁵ ([SSP](#)) aufgesetzt. Hintergrund war, dass die in vielen Anwendungen bestehende Notwendigkeit zur Simulation eines Verbundes bzw. eines Netzwerks von Simulationsmodellen erkannt wurde.

Der SSP ist ein werkzeugunabhängiger Standard zur Definition kompletter Systeme bzw. Netzwerke, die aus einer oder mehreren FMUs (siehe Functional-Mockup-Interface) einschließlich ihrer Parametrierung bestehen und zwischen Simulationswerkzeugen übertragen werden können.

³³ [VDW-Bericht](#): Abteilungsübergreifende Projektierung komplexer Maschinen und Anlagen. Aachen: WZL 1997.

³⁴ <https://fmi-standard.org/>

³⁵ <https://ssp-standard.org/>

6.3.5.1 Beispiel-Ressourcen

Als exemplarisches Beispiel für die virtuelle Inbetriebnahme wurde das in Abbildung 6-14 dargestellte Prüfmodul von Komax Testing verwendet ([„VWS Komax Pruefmodul ohne Dateien.aasx“³⁶](#)).

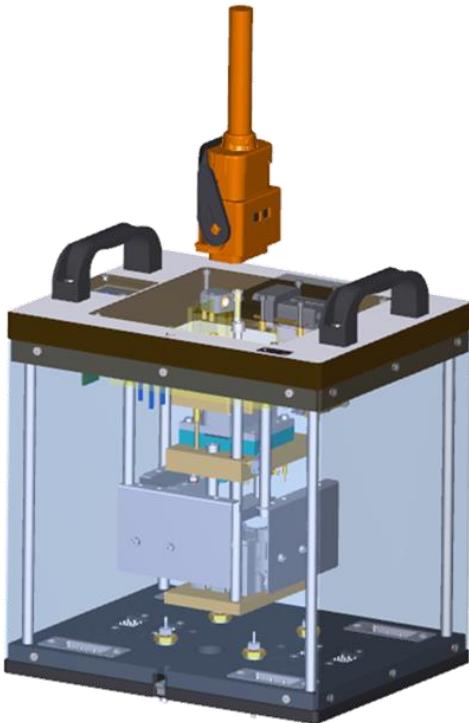


Abbildung 6-14: Prüfmodul mit Gegenstecker (Komax Testing)

Prüfmodule sind mechatronische Baugruppen, die individuell für jeden Stecker und nach Anforderung angefertigt und modular in einer Prüfmaschine eingebaut werden, um vollständige Leitungssätze zu kontaktieren und prüfen. Das Prüfmodul führt diverse Bewegungen aus, die mittels pneumatischer Antriebseinheiten (bestehend aus Ventilen und Zylindern) angetrieben werden. Hierzu gehören:

- Verriegelung des Steckergehäuses in der Aufnahme.
- Vorfahren der Kontaktalebene zur elektrischen Kontaktierung des Steckers.
- Vorfahren von Sensoren für die Erfassung von Steckermerkmalen (z.B. Sekundärverriegelung).

Für diese Baugruppe wurden im Design pneumatische Einheiten von Festo eingesetzt, welche die Bewegungen ausführen.

Mittels „specificAssetId“ (vgl. D5.3 Verbundkomponente Ressource (Produktionsmittel)) verweist die Verwaltungsschale des Prüfmoduls auf die in den verwendeten pneumatischen Einheiten verbauten Komponenten, sodass die zugehörigen Verwaltungsschalen eindeutig identifiziert werden können (s. Abbildung 6-15).

| |
|---|
| Element 1: partNumber |
| semanticId: (GlobalReference) 0173-1#02-AAO676#003 |
| name: partNumber |
| value: 119463 |
| externalSubjectId: (GlobalReference) www.komaxgroup.com |
| Element 2: partNumber |
| semanticId: (GlobalReference) 0173-1#02-AAO676#003 |
| name: partNumber |
| value: 8042543 |
| externalSubjectId: (GlobalReference) www.festo.com |

Abbildung 6-15: „specificAssetId“ einer pneumatischen Komponente im Prüfmodul

³⁶https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP05/Beispieldaten/VWS_Komax_Pruefmodul_ohne_Dateien.aasx

6.3.5.2 Betrachtete Use Cases

Eine virtuelle Inbetriebnahme soll unter anderem ermöglichen, den Funktionsablauf des Prüfmoduls und die Kontaktssicherheit im Zusammenspiel mit den Steckern des Leitungssatzes zu überprüfen, bevor die Baugruppe montiert und in die Prüfmaschine eingebaut wird, sodass Fehler im Design frühzeitig erkannt und abgestellt werden können.

6.3.5.2.1 UC4: VIBN der Antriebseinheit vom Komponentenlieferant

Für die ausgelegten Antriebseinheiten (bestehend aus Ventil + Zylinder) sollen vom Hersteller Verhaltensmodelle für die Simulation in Form von Verwaltungsschalen bereitgestellt werden.

6.3.5.2.2 UC5: VIBN des Prüfmodul

In der Verwaltungsschale des Prüfmoduls sollen Verhaltensmodelle, Verhaltenssimulationen und das kinematische Modell für die virtuelle Inbetriebnahme bereitgestellt und mit den Simulationsmodellen der Antriebseinheiten in Beziehung gesetzt werden.

6.3.5.3 Vorgehen beim Komponentenlieferant

Der Komponentenlieferant stellt zunächst einmal Verwaltungsschalen für seine Einzelkomponenten mit den relevanten Teilmodellen bereit. Im betrachteten Use Case 4 fungiert er jedoch zusätzlich auch als Teilsystem-Lieferant, denn die oben beschriebenen pneumatischen Antriebseinheiten (Verbundkomponenten) setzen sich jeweils aus mehreren Komponenten zusammen: einem Zylinder, der die eigentliche Bewegung ausführt, einem Ventil, welches für die Ansteuerung des Zylinders notwendig ist, sowie weiteren Komponenten wie Schläuche und Fittings, die im Rahmen der virtuellen Inbetriebnahme in aller Regel nicht separat, d.h. in Form von eigenständigen Simulationsmodellen, betrachtet werden.

Solche Teilsysteme werden oftmals durch den Kunden über spezielle Auslegungstools des Komponentenlieferanten konfiguriert. Ein Beispiel hierfür ist das Tool [Festo Pneumatic Sizing](#)³⁷, welches in der Lage ist, für eine gegebene Masse und eine gewünschte Positionieraufgabe automatisch ein geeignetes pneumatisches System zu konfigurieren.

Der Komponentenlieferant bzw. das genutzte Auslegungstool stellt folglich eine System-Verwaltungsschale bereit, die über das Teilmodell „[IDTA 02011: Hierarchical Structures enabling Bills of Material](#)“ (BOM) eine Bill of Material beschreibt, welches die verschiedenen Komponenten-Verwaltungsschalen referenziert. Im Rahmen des betrachteten Use Cases wurden diese Verwaltungsschalen händisch erstellt, z.B. „[VWS Festo Kombinierte Baugruppe.aasx](#)“ (Abbildung 6-16).

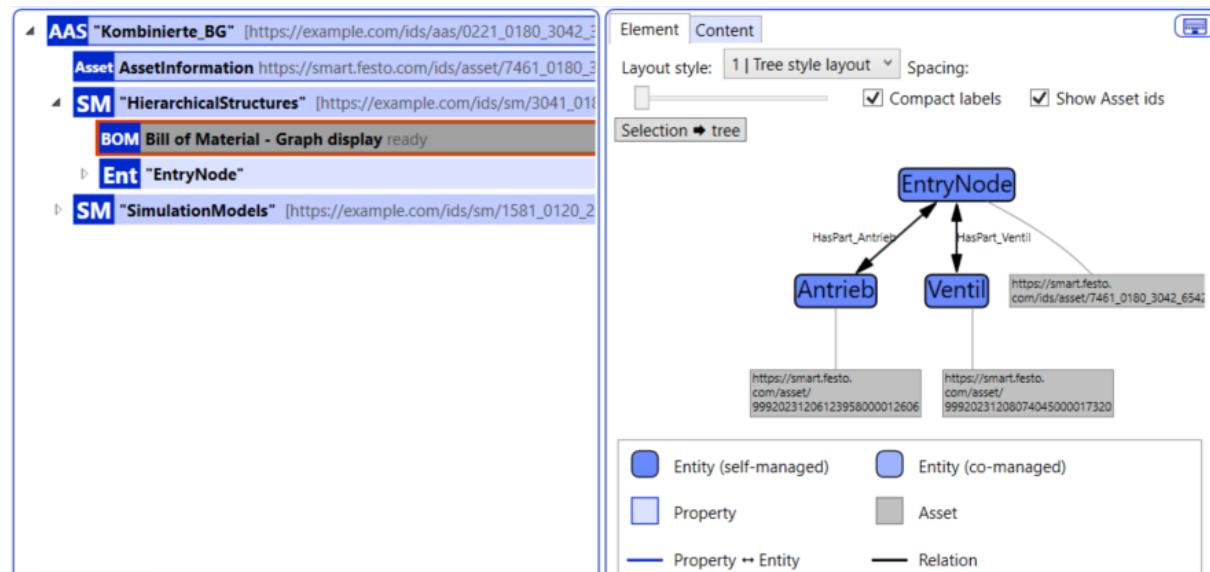


Abbildung 6-16: Verwaltungsschale einer pneumatischen Antriebseinheit mit Bill of Material

Um eine virtuelle Inbetriebnahme für das bereitgestellte Teilsystem zu ermöglichen, ist neben der BOM das Teilmodell „[IDTA 02005: Provision of Simulation Models](#)“ [40] der zentrale Bestandteil dieser System-Verwaltungsschale, denn dieses ermöglicht die Bereitstellung eines Simulationsmodells für das System, welches in Folgeprozessen genutzt werden kann. Es ist dabei grundsätzlich unabhängig von

³⁷ <https://www.festo.com/de/de/s/pneumatic-sizing>

der Art bzw. vom Format des bereitgestellten Simulationsmodells. Um die Verwendung dieses Modells in möglichst vielen Zielumgebungen zu ermöglichen, ist es für den Komponentenlieferanten jedoch sinnvoll, Simulationsmodelle in einem anerkannten Standardformat anzubieten. Hierbei eignen sich für Einzelkomponenten der FMI-Standard und für Verbundkomponenten der SSP-Standard.

Für den betrachteten Anwendungsfall wurden zunächst die Simulationsmodelle für die Magnetventile und Zylinder auf Basis des FMI-Standards beschrieben. Darauf aufbauend, wurde eine Antriebseinheit, bestehend aus einem Magnetventil und einem Zylinder, mithilfe des SSP-Standards beschrieben. In Abbildung 6-17 ist das Netzwerk der FMUs in der Simulationsumgebung sowie die Beschreibung des Netzwerks im SSP-Standard.

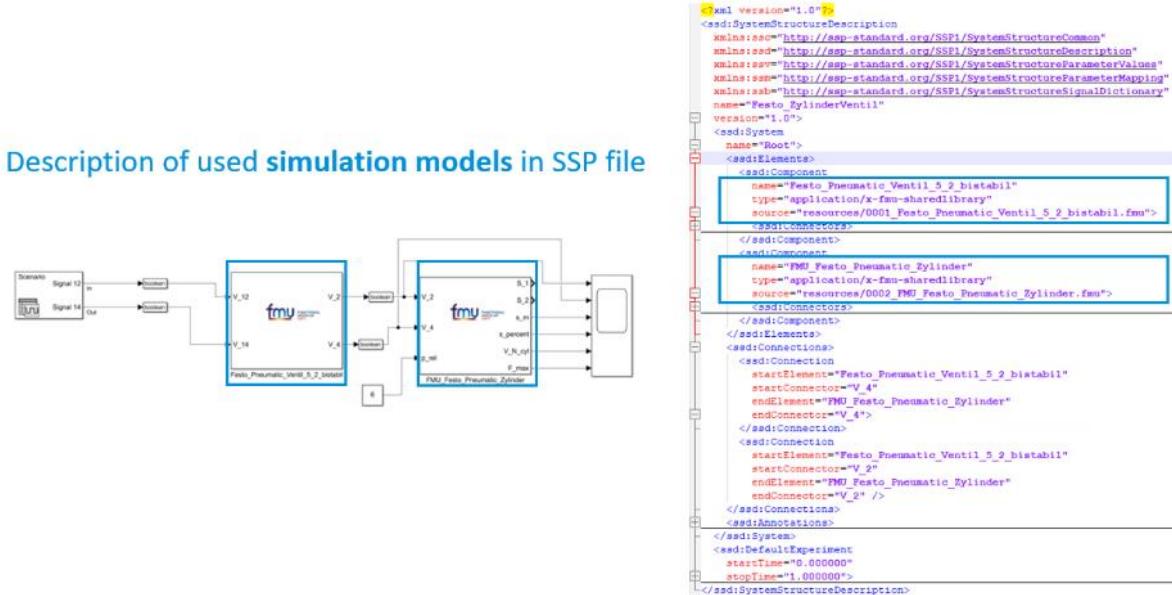


Abbildung 6-17: Netzwerk aus FMU-Modellen für eine pneumatische Antriebseinheit (rechts).

6.3.5.4 Vorgehen beim Maschinenhersteller

Ausgangssituation für die Simulation sind mehrere Verwaltungsschalen für die Einzel- und Verbundkomponenten:

- Magnetventil
- Zylinder
- Antriebseinheit (Verbundkomponente)
- Prüfmodul (Verbundkomponente)

Für die Simulationsumgebung wurden mehrere Teilsysteme beschrieben:

- Verhaltensmodell für Einzelkomponenten auf Basis des FMI-Standards (siehe Kapitel 6.3.5.3)
- Verhaltensmodell für Verbundkomponenten (Netzwerk von Einzelkomponenten) auf Basis des SSP-Standards (siehe Kapitel 6.3.5.3).
- Kinematisches Modell basierend auf dem CAD-Modell
- Logisches Verhaltensmodell in Form einer SPS-Simulation

Eingesetzte Werkzeuge (siehe Abbildung 6-18):

- [Siemens TIA-Portal](#)³⁸: SPS-Programmierung.
- [Siemens SIMATIC S7-PLCSIM Advanced](#)³⁹: SPS-Simulation.
- [EDAG iSiLOG Toolbox](#)⁴⁰: Verhaltenssimulation des Netzwerks auf Basis des SSP-Standards.
- [Siemens SIMIT Simulation Plattform](#)⁴¹: Verhaltenssimulation auf Basis des FMI-Standards.
- [Siemens NX Mechatronics Concept Designer](#)⁴²: kinematische Simulation.

³⁸ <https://www.siemens.com/de/de/produkte/automatisierung/industrie-software/automatisierungs-software/tia-portal.html>

³⁹ <https://xcelerator.siemens.com/global/en/all-offerings/products/s/s7-plcsim-advanced.html>

⁴⁰ <https://smartfactory.edag.com/en/smart-factory/digitalization-solutions/isilog-toolbox/>

⁴¹ <https://www.siemens.com/global/en/products/automation/industry-software/simit.html>

⁴² <https://plm.sw.siemens.com/en-US/nx/cad-online/automation/mechatronic-design/>

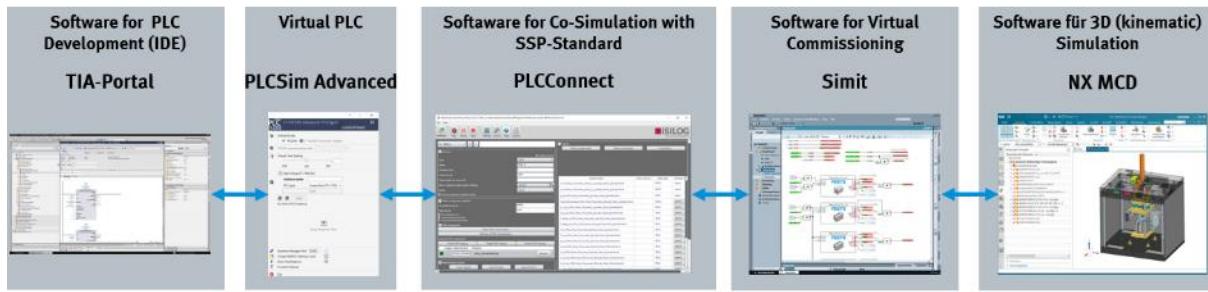


Abbildung 6-18: Eingesetzte VIBN-Tools

6.3.5.5 Darstellung in der Verwaltungsschale

In der Verwaltungsschale des Prüfmoduls werden die Antriebseinheiten über eine Stückliste identifiziert. Hierfür wurde das Teilmodell „[IDTA 02011: Hierarchical Structures enabling Bills of Material](#)“ (BOM) verwendet.

Für die Simulationsmodelle wurde das Teilmodell „[IDTA 02005: Provision of Simulation Models](#)“ [40] eingesetzt. Insgesamt wurden vier Modelle hinterlegt:

- Das kinematische Modell aus Siemens NX Mechatronics Concept Designer
- Das logische Verhaltensmodell aus Siemens SIMIT Simulation Plattform
- Das SPS-Verhaltensmodell aus Siemens SIMATIC S7-PLCSIM Advanced
- Die Konfiguration für die Co-Simulation aus PLCConnect (EDAG iSILOG Toolbox)

Damit wird UC2 erfüllt. Anhand der Stückliste sind die Antriebseinheiten auffindbar und werden eindeutig mit der Verwaltungsschale des Prüfmoduls verknüpft. Alle Simulationsmodelle werden im Teilmodell „SimulationModels“ bereitgestellt.

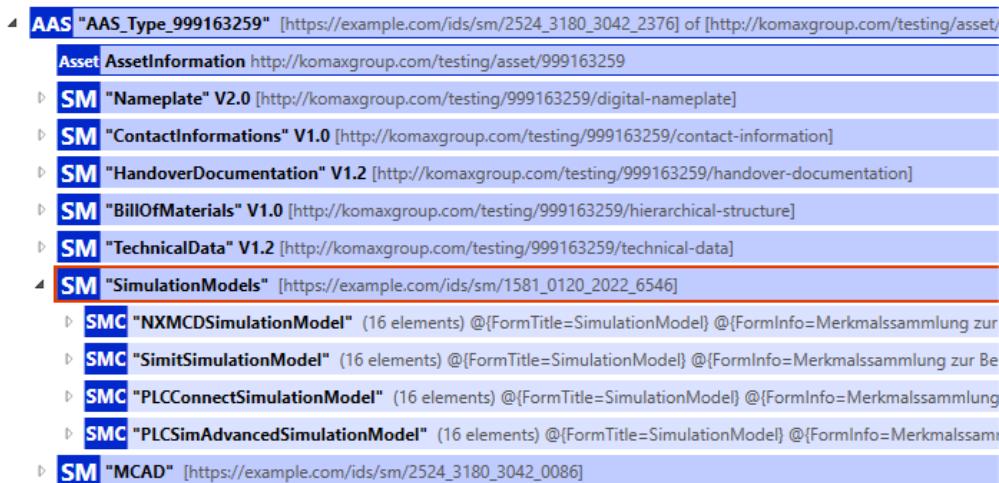


Abbildung 6-19: Simulationsmodelle in der Verwaltungsschale des Prüfmoduls

6.3.6 Ausblick

Auf Grundlage der in AP5.3 beschriebenen Lösungsansätze können Simulationsmodelle partnerübergreifend vernetzt und zu komplexen Verbundsimulationen zusammengeführt werden.

Damit Teilmodelle auch in den Simulationswerkzeugen referenziert werden können, müssen die entsprechenden Werkzeuge und Standards (SSP, FMU, etc.) angepasst werden, sodass die Simulationen in verteilten Systemen bereitgestellt werden können. Beispiel: Der SSP-Standard erfordert aktuell, dass alle Teilmodelle sich in der Simulationsdatei (.ssp) befinden – eine Referenzierung externer Simulationsmodelle bspw. innerhalb einer separaten Verwaltungsschale wäre wünschenswert und würde einer Daten-/Modelldopplung entgegenwirken.

6.4 AP 5.4 - Integration der Verwaltungsschalen

6.4.1 Zielsetzung

Zum Aufzeigen der sich aus der verknüpften Verbundstruktur von Verwaltungsschalen ergebenden Mehrwerte wurden im Rahmen von AP5.4 die folgenden vier Use Cases definiert:

1. Rückverfolgbarkeit im Fehlerfall
2. Synchronisierung von Verwaltungsschalen
3. Sachnummern-Mapping
4. Fähigkeitenabgleich

Darauf aufbauend behandelt dieses Dokument die Integration der Verwaltungsschalen über die verschiedenen Aspekte des PPR-Modells mit dem Ziel, eine durchgehende Erkundbarkeit der Verbundstruktur zu erreichen. Hierzu werden vier verschiedene Use Cases definiert und deren Umsetzbarkeit auf Basis der miteinander verknüpften Verwaltungsschalen analysiert.

In Kooperation der Teilprojekte 2, 3 und 5 sowie des Architekturteams wurden für alle betrachteten Aspekte aus den Bereichen Produkt, Prozess und Ressource exemplarische Verwaltungsschalen für das im Rahmen des PPR-Workshop definierte Beispielprodukt angelegt und gemäß der entwickelten Konzepte miteinander verknüpft. Ein Ausschnitt aus den resultierenden Verwaltungsschalen zeigt Abbildung 6-20. An dieser Stelle soll dabei nicht näher auf Spezialthemen aus den einzelnen Teilprojekten bzw. Themenbereichen eingegangen werden, da diese jeweils im Rahmen der zugehörigen Arbeitspakete beschrieben sind. Stattdessen sollen im Folgenden exemplarische Use Cases sowie deren Umsetzung diskutiert werden, die die sich aus der verknüpften Verbundstruktur ergebenden Mehrwerte aufzeigen.

| |
|--|
| ▷ AAS "DRX00014" [www.oem.com/ids/aas/3210_0010_1010_3773] of [www.oem.com/ids/asset/4010_0010_1010_639] |
| ▷ AAS "DRX00019" [www.oem.com/ids/aas/8320_0010_1010_7490] of [www.oem.com/ids/asset/7400_0010_1010_582] |
| ▷ AAS "Draexlmaier_200616545" [www.draexlmaier.com/ids/aas/6210_0010_1010_5990] of [www.draexlmaier.com/ids/asset/5100_0010_1010_1505] |
| ▷ AAS "Draexlmaier_200023941" [www.draexlmaier.com/ids/aas/5010_0010_1010_1505] of [www.draexlmaier.com/ids/asset/4500_0010_1010_5537] |
| ▷ AAS "Draexlmaier_200024015" [www.draexlmaier.com/ids/aas/7110_0010_1010_5537] of [www.draexlmaier.com/ids/asset/6210_0010_1010_1505] |
| ▷ AAS "Draexlmaier_200028148" [www.draexlmaier.com/ids/aas/4500_0010_1010_7281] of [www.draexlmaier.com/ids/asset/3400_0010_1010_9246] |
| ▷ AAS "Draexlmaier_200002382" [www.draexlmaier.com/ids/aas/3400_0010_1010_9246] of [www.draexlmaier.com/ids/asset/2310_0010_1010_5788] |
| ▷ AAS "Draexlmaier_200002479" [www.draexlmaier.com/ids/aas/2310_0010_1010_5788] of [www.draexlmaier.com/ids/asset/1110_0010_1010_9517] |
| ▷ AAS "Draexlmaier_200002428" [www.draexlmaier.com/ids/aas/0020_0010_1010_5025] of [www.draexlmaier.com/ids/asset/5500_0010_1010_9005] |
| ▷ AAS "Draexlmaier_200002439" [www.draexlmaier.com/ids/aas/5500_0010_1010_9005] of [www.draexlmaier.com/ids/asset/9093_3151_2132_7387] |
| ▷ AAS "Draexlmaier_200018565" [www.draexlmaier.com/ids/aas/9093_3151_2132_7387] of [www.draexlmaier.com/ids/asset/7140_3131_2132_6087] |
| ▷ AAS "Draexlmaier_200017542" [www.draexlmaier.com/ids/aas/7140_3131_2132_6087] of [www.draexlmaier.com/ids/asset/5130_0010_1010_9756] |
| ▷ AAS "Draexlmaier_201098090" [www.draexlmaier.com/ids/aas/1110_0010_1010_9517] of [www.draexlmaier.com/ids/asset/1314_7041_2132_0558] |
| ▷ AAS "Draexlmaier_Leitungssatz_Typ" [www.draexlmaier.com/ids/aas/5130_0010_1010_9756] of [www.draexlmaier.com/ids/asset/2135_1132_8032_2655] |
| ▷ AAS "Draexlmaier_200018565_Geschnitten" [www.draexlmaier.com/ids/sm/8094_3131_2132_0663] of [www.draexlmaier.com/ids/asset/4130_0010_1010_3671] |
| Asset AssetInformation www.draexlmaier.com/ids/asset/4130_0010_1010_3671 |
| ▷ SM "ProductSpecification" [www.draexlmaier.com/ids/submodel/1300_0010_1010_2428] |
| ▷ SM "LS_Product_BOM" [www.draexlmaier.com/ids/submodel/5310_0010_1010_6706] |
| ▷ SM "LS_Manufacturing_BOM" [www.draexlmaier.com/ids/submodel/8510_0010_1010_1387] |
| ▷ SM "BillOfProcess" [www.draexlmaier.com/ids/sm/6470_8092_6032_0890] |
| ▷ SM "RequiredCapabilities" [www.draexlmaier.com/ids/sm/2135_1132_8032_2655] |
| ▷ AAS "Komax_Sigma_688_Type" [https://example.com/ids/sm/1314_7041_2132_0558] of [www.komaxgroup.com/ids/asset/1314_7041_2132_0558] |

Abbildung 6-20: Ausschnitt der erstellten Verwaltungsschalen

6.4.2 Definition und Umsetzung der Use Cases

Im Folgenden sollen die eingangs genannte vier Use Cases, deren Umsetzung sowie der sich ergebende Mehrwert genauer beschrieben werden.

6.4.2.1 Use Case 1: Rückverfolgbarkeit im Fehlerfall

Bei der Rückverfolgbarkeit im Fehlerfall geht es darum, bei Auftreten eines Fehlers dessen Ursachen ermitteln zu können und entsprechende Auswirkungen gering zu halten bzw. gezielt zu korrigieren. Dadurch können beispielsweise Rückrufe verringert werden, wodurch dieser Use Case eine enorme wirtschaftliche Relevanz besitzt.

Eine typische Ausprägung sieht dabei wie folgt aus: Bei einem ausgelieferten Fahrzeug bzw. dem enthaltenen Leitungssatz wird ein Mangel festgestellt – bspw. eine fehlerhafte Crimpverbindung. Mögliche Fehlerursachen können hierbei eine fehlerhafte Charge einer Komponente (z.B. eines Kontaktteils) oder ein fehlerhaft ausgeführter Prozess der entsprechenden Crimp-Maschine sein. Im Folgenden geht es nun darum, die Ursache der fehlerhaften Verbindung zu ermitteln und – falls ähnliche Fehler in anderen ausgelieferten Leitungssätzen nicht ausgeschlossen werden können – die betroffenen Fahrzeuge zurückzurufen und zu kontrollieren bzw. zu reparieren. Durch eine fehlende oder mangelhafte Rückverfolgbarkeit können in diesem Fall sehr hohe Kosten entstehen, da im Zweifelsfall sehr viele Fahrzeuge zurückgerufen werden müssen. Je besser die Rückverfolgbarkeit, desto besser lässt sich der Kreis der potenziell betroffenen Fahrzeuge einschränken, die zurückgerufen werden müssen. Im Folgenden soll allgemein beschrieben werden, wie eine Rückverfolgbarkeit für das beschriebene Szenario realisiert werden kann. Eine detailliertere Beschreibung zur konkreten Umsetzung anhand zweier weiterer Use Cases wurde in der AG „Rückverfolgbarkeit“ des Architekturteams entwickelt und ist im entsprechenden Dokument zu finden.

Im oben geschilderten Beispiel ist dazu die folgende „Spur“ durch das Netzwerk verknüpfter Verwaltungsschalen zu verfolgen: Zunächst ist anhand der Serien- bzw. Fahrzeugidentifikationsnummer (FIN) die zu der Produktinstanz gehörige Verwaltungsschale des Leitungssatzes zu ermitteln. In dieser Verwaltungsschale befindet sich wie in TP 2 definiert die Produktspezifikation in Form von Zeichnungen sowie eines formalen Modells (VEC bzw. KBL). Abbildung 6-21 zeigt exemplarisch eine solche Zeichnung eines Leitungssatzes. Anhand dieser Informationen kann dann die fehlerhafte Verbindung bzw. die betroffenen Komponenten-Vorkommen eindeutig identifiziert werden. In Abbildung 6-21 könnte bspw. die Crimp-Verbindung am Kontaktteil fehlerhaft sein, welches in Kammer 1 des Gehäuses „A2*1-B“ gesteckt wird.

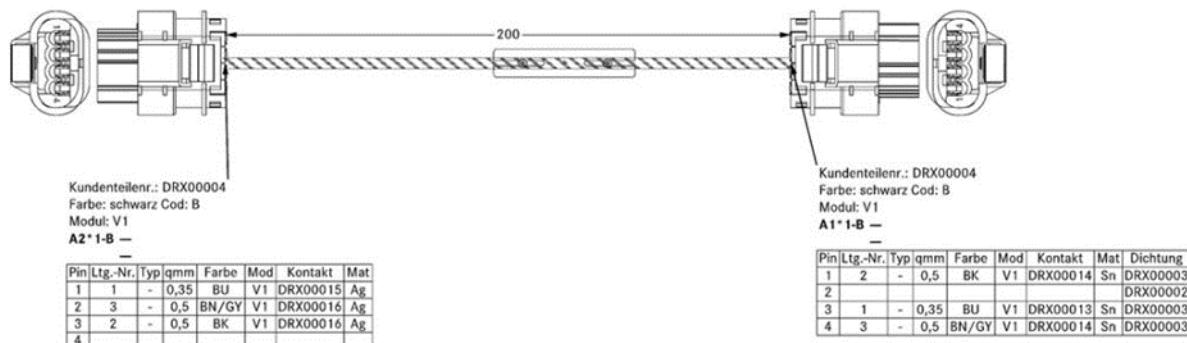


Abbildung 6-21: Exemplarische Zeichnung Leitungssatz inkl. Komponentenbezeichnungen

Durch die im Rahmen der AG „Verlinkung“ im Architekturteam geschaffenen Strukturen kann anschließend genau dieses Komponenten-Vorkommen bis hin zur Verwaltungsschale des Halbfabrikats verfolgt werden, im Rahmen dessen Herstellungsprozesses die entsprechende Verbindung hergestellt wurde. In dieser Verwaltungsschale finden sich wie in TP 3 definiert die Liste der bei der Herstellung des Halbfabrikats durchgeführten Prozesse sowie die im Rahmen der Prozesse verwendeten Materialien.

An dieser Stelle kann nun die eigentliche Fehlersuche bzw. die Ermittlung der Auswirkungen beginnen. So können bspw. die als Teil der ausgeführten Prozesse gespeicherten Parameter (bspw. Crimp-Kraft-Verläufe) analysiert werden. Durch die konsistente Verknüpfung der Verwaltungsschalen für den Produktionsprozess kann anschließend unter anderem auch ermittelt werden, welche Halbfabrikate noch als Teil der gleichen Charge (Lot bzw. Batch) produziert wurden. Da jede Verwaltungsschale für einen Produktionsprozess außerdem auch eindeutig die genutzte(n) Chargen von Ausgangsmaterialien ausweist, kann durch eine Rückwärtssuche im Folgenden ermittelt werden, welche anderen Halbfabrikate ebenfalls Vorkommen einer möglicherweise fehlerhaften Charge enthalten.

Auf Basis einer solchen Menge potenziell ebenfalls betroffener Halbfabrikate kann anschließend durch ähnliche Mechanismen wie oben beschrieben zunächst auf die betroffenen Leitungssätze sowie auf die genaue Position der verbauten möglicherweise fehlerhaften Halbfabrikate in diesen Leitungssätzen geschlossen werden. Auf Basis dieser Informationen können durch die Seriennummer der Leitungssätze zunächst die betroffenen Fahrzeuge ermittelt und zurückgerufen sowie die potenziell fehlerhaften Crimp-Verbindungen gezielt überprüft werden.

6.4.2.2 Use Case 2: Synchronisierung von Verwaltungsschalen

Im Rahmen der AG „Modularisierung“ des Architekturteams wurde bereits beschrieben, dass jeder Partner der Wertschöpfungskette die für ihn relevanten Daten normalerweise in einem eigenen Server/Repository aufbewahren will, damit eine durchgängige Datenverfügbarkeit gewährleistet ist. Dies führt zu vielfältigen Dopplungen von Daten, die für mehrere Partner der Wertschöpfungskette relevant sind.

Wie im Rahmen von TP 2 beschrieben, existiert darüber hinaus in aller Regel nicht die „eine Wahrheit“ (Single Source of Truth) für Informationen bspw. zu einer Komponente oder einer Produktspezifikation für einen Leitungssatz. Stattdessen haben verschiedene Partner der Wertschöpfungskette eine teilweise unterschiedliche – unter Umständen sogar widersprüchliche – Datenbasis.

Für Komponenten können dies bspw. unterschiedliche Grenzwerte sein, im Rahmen derer die Komponente verwendet werden darf: So schreibt bspw. der Komponentenhersteller (Tier 2) bestimmte Grenzwerte vor, die von unterschiedlichen OEMs ggf. weiter eingeschränkt werden. Ein weiteres Beispiel ist eine Produktspezifikation, für die grundsätzlich unterschiedliche Daten bei OEM und Konfektionär (Tier 1) existieren, weil der Konfektionär in aller Regel noch zusätzliche Teile ergänzt, die lediglich für die Produktion bzw. Montage relevant sind (bspw. Klebebänder).

In solch einem heterogenen Umfeld ist die (semi-)automatische Synchronisierung bzw. Änderung von Daten essenziell. Nur so kann gewährleistet werden, dass Änderungen von Datenständen anderen Partnern bekannt gemacht und von diesen – unter Umständen nach einer manuellen Überprüfung oder Anpassung – übernommen werden können. Dadurch wird die Gefahr nicht-konsistenter Daten reduziert und die gemeinsame Wertschöpfung erleichtert

Heutzutage erfolgt eine solche Synchronisierung in aller Regel manuell. So werden Änderungen bspw. per E-Mail kommuniziert. Durch den Verbund verknüpfter Verwaltungsschalen ergibt sich nun die Möglichkeit, Änderungen automatisiert zu kommunizieren und falls notwendig zu übernehmen. Dazu wurde im Rahmen der AG „Synchronisierung“ des Architekturteams ein auf dem Publish-Subscribe-Modell basierendes Konzept entwickelt, um Änderungen an bestimmten Verwaltungsschalen bzw. Teilmodellen zu „abonnieren“. Eine entsprechende Änderungsnachricht kann dann automatisiert ausgewertet werden und die entsprechende Änderung automatisch oder bspw. nach einer manuellen Prüfung übernommen werden.

6.4.2.3 Use Case 3: Sachnummern-Mapping

Für eine Komponente – bspw. einen bestimmten Crimp-Kontakt – existieren in aller Regel mehrere Teile- bzw. Sachnummern: So gibt es für eine Komponente oftmals mehrere Lieferanten (Tier 2), die jeweils eine eigene Teilenummer vergeben. Darüber hinaus hat auch jeder OEM sowie jeder Konfektionär (Tier 1) eine eigene Sachnummer, unter der er die Komponente in seinem ERP-System führt. Eine Herausforderung bei der Leitungssatz-Entwicklung ist daher das sogenannte „Sachnummern-Mapping“: So erstellt bspw. der OEM eine Produktspezifikation für einen Leitungssatz und verwendet dabei seine internen Sachnummern für die verwendeten Komponenten. Anschließend wird ein Konfektionär mit der Produktion des Leitungssatzes beauftragt. Daraufhin muss dieser zunächst für jede OEM-Sachnummer seine entsprechende eigene Sachnummer ermitteln und diese in der Produktspezifikation ersetzen. Bei der anschließenden Ermittlung möglicher Lieferanten müssen wiederum für die internen Sachnummern des Konfektionärs die entsprechenden Sachnummer möglicher Zulieferer ermittelt werden.

Derzeit sind solche „Mapping-Tabellen“ in aller Regel informell auf sog. „Zeichnungen“ der verschiedenen OEMs spezifiziert wie in Abbildung 6-22 dargestellt. Diese informellen Informationen müssen dann in einem aufwändigen Prozess häufig noch von Hand in die ERP-Systeme der verschiedenen Konfektionäre eingepflegt werden.

Im Rahmen von TP5.2 wurde daher ein Mechanismus beschrieben, wie man diese bisher informell im Rahmen von Zeichnungen ausgetauschten Informationen formal und explizit im Rahmen der entsprechenden Verwaltungsschalen abbilden kann, um somit den Prozess des Sachnummern-Mappings zu erleichtern. Dazu können in einer Verwaltungsschale bspw. einer Komponente neben der eigenen auch noch eine beliebige Anzahl „externer“ Sachnummern hinterlegt werden, wodurch der Prozess des Sachnummern-Mappings automatisiert werden kann.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | |
|---|---|------------------------------|----------------------|----------------------------|--|------------------------------|--|----------------------|----------------------------|
| Teile-Nr. | N 107 766 01 | | | | N 107 766 02 | | | | |
| mm ² | 1.0 | | | | Sn | | | | |
| Oberfläche | | | | | (12) | | | | |
| Typ | FLR | | | | ELA | | | | |
| A | Code | KOSTAL | AMP/TE | Yazaki | APTIV | KOSTAL | AMP/TE | Yazaki | APTIV |
| Werkstoff Grundkörper | 1 | 4 | 2 | 3 | | 1 | 4 | 2 | 3 |
| Werkstoff Kontaktlamelle | (14) | 1 | 3 | - | 2 | 1 | 3 | - | 2 |
| Werkstoff Überfeder | - | - | - | 1 | - | - | - | 1 | - |
| Einzelreiterabdichtung (ELA) nach TAB | | (30) | - | - | | (30) | TAB.003.656.H, TAB.003.656.F B1.1 (27) | | |
| Nr. | 3 21 40 73403 3 3 21 40 73403 0 (A) | 7-1452659-1 | (28) 7206-0020-02 | 35506842 | 3 21 40 73413 3 3 21 40 73413 0 (A) | 7-1452671-1 | 7206-0021-02 (28) | 35555156 | |
| R | Steckhäufigkeit / Wiring cycle frequency | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| Oberfläche im Bereich der Kontaktlamellen/Kontaktfeder | 1-3 µm verzintt | 1-3 µm feuerverzinnt SnAg | 1-3 µm feuerverzinnt | 1,2-3,5µm feuerverzinnt | 1-3 µm verzintt | 1-3 µm feuerverzinnt SnAg | 1-3 µm feuerverzinnt | 1-3 µm feuerverzinnt | 1,2-3,5µm feuerverzinnt |
| Oberfläche im Drahtcrimpbereich | 0,8-3 µm feuerverzinnt | 1-3 µm feuerverzinnt | | | | | | | |

Abbildung 6-22: Exemplarische OEM-Komponentenspezifikation inkl. Sachnummern

6.4.2.4 Use Case 4: Fähigkeitenabgleich

Eine entscheidende Voraussetzung für die Produktion eines Leitungssatzes bzw. der enthaltenen Halbfabrikate ist die Ermittlung möglicher Ressourcen, auf denen die für die Produktion des Leitungssatzes/Halbfabrikats notwendigen Prozesse umgesetzt werden können. Grundlage dafür ist ein Fähigkeitenabgleich, der die zur Produktion eines Halbfabrikats benötigten Fähigkeiten mit den von Ressourcen angebotenen Fähigkeiten vergleicht und auf Basis dieses Vergleiches ein bestimmtes Subset an geeigneten Maschinen ermittelt.

Ein solcher Algorithmus zum Fähigkeitenabgleich wurde im Rahmen von TP5.5 entwickelt und basiert auf Informationen aus verschiedenen Verwaltungsschalen aus dem gesamten PPR-Kontext:

Ausgangspunkt ist dabei die Produktspezifikation eines Halbfabrikats bzw. eines Leitungssatzes, in der die eingesetzten Komponenten sowie die benötigten Verbindungen beschrieben sind. Auf dieser Basis können anschließend die zur Herstellung des Produkts benötigten Prozesse ermittelt und in Form der sog. „Bill of Process“ beschrieben werden (s. TP 3). Für eine durchgängige Verknüpfung verweisen dabei die Prozesse auf die verarbeiteten Komponenten, wie bereits in Abschnitt 6.4.2.1 beschrieben. Für jeden Prozess wird anschließend eine sog. „Required Capability“ beschrieben, die die zur Durchführung des Prozesses benötigte Fähigkeit beschreibt. Die entsprechenden Parameter (bspw. die Leitungslänge bei einem Schneidprozess) ergibt sich dabei aus dem Prozessmodell bzw. indirekt aus der Produktspezifikation.

Auf der anderen Seite stehen Ressourcen, die bestimmte Fähigkeiten anbieten und dabei bestimmte Randbedingungen für Parameter beschreiben (bspw. die maximal mögliche Länge bei einem Schneid-Prozess). Ein weiterer wichtiger Punkt ist die explizite Modellierung eventuell notwendiger Werkzeuge zur Durchführung eines Prozesses (z.B. eines Crimp-Applikators zum Herstellen einer Crimp-Verbindung durch eine Maschine). Hierdurch werden Ressourcen-Verwaltungsschalen im Rahmen einer Verbundkomponente miteinander verbunden.

Im Rahmen des Fähigkeitenabgleichs werden nun diese Informationen aus den Bereichen Produkt, Prozess und Ressource genutzt und automatisiert ausgewertet. Das Netzwerk verknüpfter Verwaltungsschalen kann dabei entweder im Rahmen einer Typ- oder einer Instanz-Abfrage ausgewertet werden: Bei einer Typ-Abfrage werden mögliche Maschinen inkl. evtl. zu montierender Werkzeuge ermittelt während bei einer Instanz-Abfrage die in einer Maschine montierten Werkzeuge ausgewertet werden.

Somit bietet der automatisierte Fähigkeitenabgleich das Potenzial für eine effiziente Produktionssteuerung auf Basis der im Rahmen von Verwaltungsschalen bereitgestellten Informationen, bei der bspw. auch weitere, evtl. in zusätzlichen Teilmustern bereitgestellte Informationen wie Energieverbräuche oder Rüstzeiten mitberücksichtigt werden können.

6.5 AP 5.5 - Mapping zwischen Produktionsressource und Leitungssatz

6.5.1 Zielsetzung

Das AP 5.5 befasst sich mit dem Mapping zwischen Produktionsressource und Leitungssatz, d.h. mit der automatisierten Auswahl bestimmter Produktionsressourcen zur Fertigung eines Leitungssatzes oder eines Halbfabrikats.

Im Rahmen einer solchen Fertigung sind Prozesse auszuführen. Kern einer Bewertung, ob eine bestimmte Ressource ein bestimmtes Produkt fertigen kann, sind daher die Fähigkeiten, die zur Fertigung eines bestimmten Produktes benötigt werden, sowie die Fähigkeiten, die von einer bestimmten Produktionsressource bereitgestellt werden.

Basierend auf diesen Informationen kann ein Fähigkeitenabgleich („Capability Matching“) durchgeführt werden, im Rahmen dessen eine Bewertung der Erfüllbarkeit einer geforderten Fähigkeit durch eine Ressource durchgeführt wird. Besonders zu beachten ist, dass eine Ressource zur Bereitstellung einer Fähigkeit oftmals noch weitere Ressourcen benötigt: Ein Beispiel hierfür ist eine Crimp-Maschine, die zum Crimpen noch mit einem bestimmten Crimp-Gesenk bzw. einem entsprechenden Applikator ausgestattet werden muss. Für diese Montage kann dann evtl. noch ein entsprechendes Modul oder ein Adapter benötigt werden. Die verschiedenen Teilressourcen bieten dann gemeinsam eine bestimmte kombinierte Fähigkeit an.

Die Abbildung solcher – sowohl aktueller als auch theoretisch realisierbarer – Maschinenstrukturen wurden bereits im Rahmen von D5.3 beschrieben und können an dieser Stelle genutzt werden.

Abbildung 6-23 zeigt diese Zusammenhänge am Beispiel eines Crimp-Prozesses. Die dargestellten Prinzipien gelten aber auch für die anderen im Rahmen von TP3 abgeleiteten Prozesse, wobei für manche Prozesse in der Regel keine Werkzeuge benötigt werden (z.B. Schneiden).

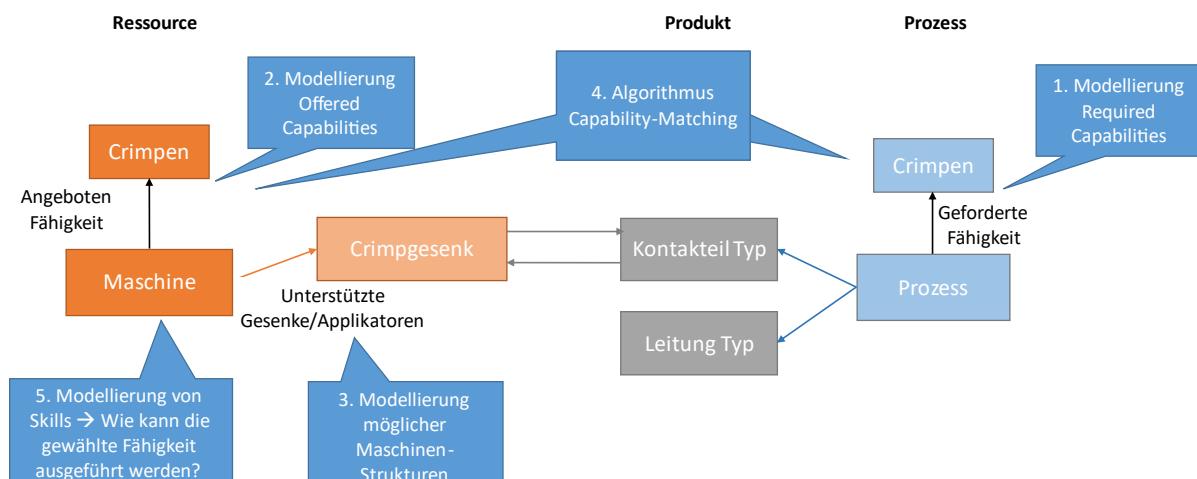


Abbildung 6-23 Capability-Matching am Beispiel eines Crimp-Prozesses

Entsprechend der Abbildung werden im Rahmen dieses Dokuments zunächst in den Abschnitten 6.5.2 und 6.5.3 die Modellierung von geforderten und angebotenen Fähigkeiten diskutiert. Darauf aufbauend wird in Abschnitt 6.5.4 anschließend das Konzept für einen Algorithmus zum Fähigkeitenabgleich entwickelt. Abschließend wird in Abschnitt 0 die Frage diskutiert, wie von einer Fähigkeit auf den ausführbaren *Skill* verwiesen werden kann.

6.5.2 Modellierung geforderter Fähigkeiten

Für die Beschreibung von Fähigkeiten ist das Teilmodell „[IDTA 02020 Capability Description](#)“ [41] bei der IDTA in Entwicklung, konnte jedoch noch nicht in VWS4LS verwendet werden. Allerdings wurde im Rahmen des Forschungsprojektes [BaSys 4.2](#)⁴³ ein erster Vorschlag für ein solches Teilmmodell erarbeitet, welches die generische Beschreibung von beliebigen *Capabilities* (z.B. „Cut“) und deren Parameter bzw. *Properties* (z.B. „Länge“) ermöglicht. Für *Properties* können darüber hinaus entweder feste Werte

⁴³ <https://www.iese.fraunhofer.de/blog/basisys-4-2-das-basisystem-fuer-industrie-4-0-geht-in-die-naechste-runde/>

vorgegeben oder Einschränkungen formuliert werden, wobei letzteres im Rahmen von BaSys 4.2 nicht bis ins letzte Detail ausgearbeitet wurde.

Im Rahmen von TP5 wurde beschlossen, das im Rahmen von BaSys 4.2 entwickelte Modell als Ausgangspunkt zu nutzen und um fehlende Aspekte zu erweitern. Ein erster im Rahmen von BaSys 4.2 nicht betrachteter Aspekt ist die Unterscheidung zwischen (zur Herstellung eines Produkts) *benötigten* und (von einer Ressource) *angebotenen* Fähigkeiten. Um klarzustellen, welche Art von Fähigkeiten ein bestimmtes Teilmodell abbildet, müsste im Rahmen einer möglichen Standardisierung bei der IDTA konkrete semantische IDs (z.B. *supplementalSemanticIds*) oder zusätzliche Eigenschaften (z.B. *Type-OfCapability*) für die Unterscheidung der Fähigkeit-Arten bereitgestellt werden. Für die Zwecke dieses Dokuments wurde beschlossen, den Namen (Eigenschaft *idShort*) des entsprechenden Teilmodells entweder auf „*RequiredCapabilities*“ oder „*OfferedCapabilities*“ zu setzen, um so bereits auf den ersten Blick eine Unterscheidung zu ermöglichen.

Innerhalb des Teilmodells kann dann zunächst die in BaSys 4.2 definierte Grundstruktur genutzt werden, wie exemplarisch in Abbildung 6-24 dargestellt. Es lassen sich folglich eine beliebige Anzahl von Fähigkeiten im Rahmen von „*CapabilityContainer*“-Elementen abbilden, wobei jeder dieser Container ein „*Capability*“-Element enthält, welches per *SemanticID* eine bestimmte Fähigkeit ausweist.

Für die Leitungssatz-Produktion relevanten Fähigkeiten, die im Rahmen der Zusammenarbeit mit TP3 ausgearbeitet wurden, sind entsprechende *SemanticIDs* nach dem Muster „<https://arena2036.de/vws4ls/capability/1/0/<capability-name>>“ festgelegt worden, also z.B. „<https://arena2036.de/vws4ls/capability/1/0/CutCapability>“. Eine vollständige Liste der definierten SemanticIDs findet sich in

Anhang: Liste von Fähigkeiten und zugehörigen Parametern.

Für eine Fähigkeit relevante Parameter lassen sich im „*PropertySet*“ beschreiben, wie ebenfalls in Abbildung 6-24 zu sehen. Die im Rahmen der Zusammenarbeit mit TP3 für jede Fähigkeit abgeleiteten relevanten Parameter sind ebenfalls aufgelistet. Der für einen bestimmten Parameter relevante Wert ergibt sich dabei in aller Regel entweder aus dem Produktmodell (bspw. die Länge beim Schneiden einer Leitung) oder aus Anforderungen an die Prozessdurchführung (bspw. ob beim Abisolieren einer Leitung eine Einschneide-Überwachung durchzuführen ist).

| | | |
|--------------|-------------------------|--|
| AAS | "Product_Type_0_5mm^2" | [https://www.arena2036.de/aas/8074_8002_2042_7117] |
| Asset | AssetInformation | https://www.arena2036.de/asset/3174_8002_2042_7736 |
| ▷ SM | "LS_Product_BOM_01" | [https://www.arena2036.de/sm/7374_8002_2042_4915] |
| ▷ SM | "BillOfProcess" | [https://www.arena2036.de/sm/3574_8002_2042_5469] |
| ◀ SM | "RequiredCapabilities" | [https://www.arena2036.de/sm/7084_8002_2042_5688] |
| ◀ SMC | "CapabilitySet" | (3 elements) |
| ▷ SMC | "CapabilityContainer01" | (2 elements) |
| ▷ SMC | "CapabilityContainer02" | (2 elements) |
| ◀ SMC | "CapabilityContainer03" | (2 elements) |
| Cap | "CrimpCapability" | |
| ◀ SMC | "PropertySet" | (4 elements) |
| ◀ SMC | "PropertyContainer01" | (1 elements) |
| Prop | "WireType" | = FLR |
| ◀ SMC | "PropertyContainer02" | (1 elements) |
| Prop | "WireCrossSectionArea" | = 0.5 |
| ◀ SMC | "PropertyContainer03" | (1 elements) |
| Prop | "TerminalPartNumber" | = 32140734113 |
| ◀ SMC | "PropertyContainer04" | (1 elements) |
| Prop | "CrimpForceMonitoring" | = false |

Abbildung 6-24: Exemplarisches Teilmodell "RequiredCapabilities"

6.5.3 Modellierung angebotener Fähigkeiten

Wie bereits in Abschnitt 6.5.2 erläutert wurde, kann für die Modellierung von durch Ressourcen angebotenen Fähigkeiten grundsätzlich die gleiche Teilmodellstruktur verwendet werden. Gegenüber der Modellierung von benötigten Fähigkeiten ergeben sich hier aber vor allem zwei Unterschiede bzw. Erweiterungen, die im Folgenden genauer diskutiert werden sollen:

- die Einschränkung von Eigenschaftswerten über sogenannte „*Constraints*“ und
- die Beschreibung von Fähigkeiten, die sich erst aus dem Zusammenschluss zweier oder mehrerer Ressourcen ergeben.

6.5.3.1 Einschränkung von Parameterwerten

Bei der Beschreibung geforderter Fähigkeiten werden für Parameter in aller Regel feste Werte vorgegeben. Ein Beispiel ist das Schneiden einer Leitung auf eine bestimmte Länge (bspw. 500 mm).

Im Gegensatz dazu kommt es bei der Beschreibung angebotener Fähigkeiten häufig vor, dass für einen Parameter kein einzelner, konkreter Wert definiert werden kann. So kann eine Schneidmaschine Leitungen in der Regel in einem bestimmten, durch die physikalischen Eigenschaften der Maschinen bestimmten, Bereich schneiden (z.B. 50 – 5000 mm).

Um solche Einschränkungen beschreiben zu können, sieht das im Rahmen von BaSys 4.2 definierte Teilmodell bereits das Konzept der *Constraints* vor, die sich auf ein bestimmtes *Property* beziehen, wie in Abbildung 6-25 dargestellt. Die konkrete Modellierung der eigentlichen *Constraints* jedoch wurde im Rahmen von BaSys 4.2 nicht bis ins letzte Detail festgelegt, sodass entsprechende Festlegungen im

Rahmen der Arbeit zu AP5.5 getroffen wurden. Die Parameter von Eigenschaften beschreiben in der Regel numerische Werte, Zeichenketten oder boolesche Werte. Dabei sind für unterschiedliche Datentypen von Parametern unterschiedliche Einschränkungen sinnvoll:

- Für boolesche Werte können bei angebotenen Fähigkeiten in der Regel konkrete Werte angegeben werden (Parameter erfüllt oder nicht erfüllt) – es müssen folglich keine Einschränkungen beschrieben werden.
- Für numerische Werte ist, wie oben beschrieben, die Vorgabe eines oberen und unteren Grenzwertes sinnvoll. Hierfür kann das AAS-Element *Range* genutzt werden (vgl. Abbildung 6-25).
- Für Zeichenketten ist oftmals die Vorgabe einer bestimmten Liste von Werten sinnvoll (z.B. die Liste von Teilenummern, für die ein Crimp-Gesenk verwendet werden kann). Hierzu kann das AAS-Element *SubmodelElementList* genutzt werden.

Weitere Arten von *Constraints* sind theoretisch denkbar (bspw. die Vorgabe einer bestimmten *Regular Expression* für Zeichenketten), wurden im Rahmen dieses Projekts aber nicht weiter betrachtet. Das beschriebene Konzept lässt sich aber entsprechend erweitern.



Abbildung 6-25: Exemplarisches Teilmodell "OfferedCapabilities"

Eine Besonderheit stellen allerdings dynamische Einschränkungen dar, die nicht pauschal bspw. durch Vorgabe eines bestimmten Wertebereichs beschrieben werden können. Beispiele hierfür sind Toleranzbereiche, die sich in Abhängigkeit des geforderten Wertes ändern. So kann eine Schneidmaschine bspw. bei kleinen Schneidlängen einen geringen Toleranzbereich aufweisen, der sich aber bei größeren Längen aufweitet.

Zur Abbildung solcher komplexen Abhängigkeiten wurde zusätzlich ein *Operation-Constraint* definiert, welches die dynamische Beurteilung der Erfüllbarkeit eines bestimmten geforderten Parameter-Wertes auf Basis eines Operationsaufrufs ermöglicht. Somit kann durch Instanziieren eines AAS-Elements *Operation* ein beliebig komplexe Berechnungsvorschrift hinterlegt werden, die theoretisch sogar weitere Parameter wie den aktuellen Maschinenzustand mit einbeziehen könnte.

6.5.3.2 Beschreibung zusammengesetzter Fähigkeiten

Individuelle Ressourcen können ihre Fähigkeiten oftmals nicht eigenständig realisieren. So benötigt bspw. eine Crimp-Maschine ein auf das Kontaktteil zugeschnittenes Crimp-Werkzeug, um einen Crimp-Prozess auszuführen. Das Crimp-Werkzeug wiederum benötigt eine entsprechende Maschine, die es verwendet. Diese Zusammenhänge sind auf der linken Seite von Abbildung 6-23 dargestellt.

Dies wirft die Frage auf, welche dieser beiden Ressourcen nun eigentlich die Fähigkeit besitzt – und folglich, in welcher Verwaltungsschale das entsprechende Fähigkeiten-Teilmodell angelegt werden sollte.

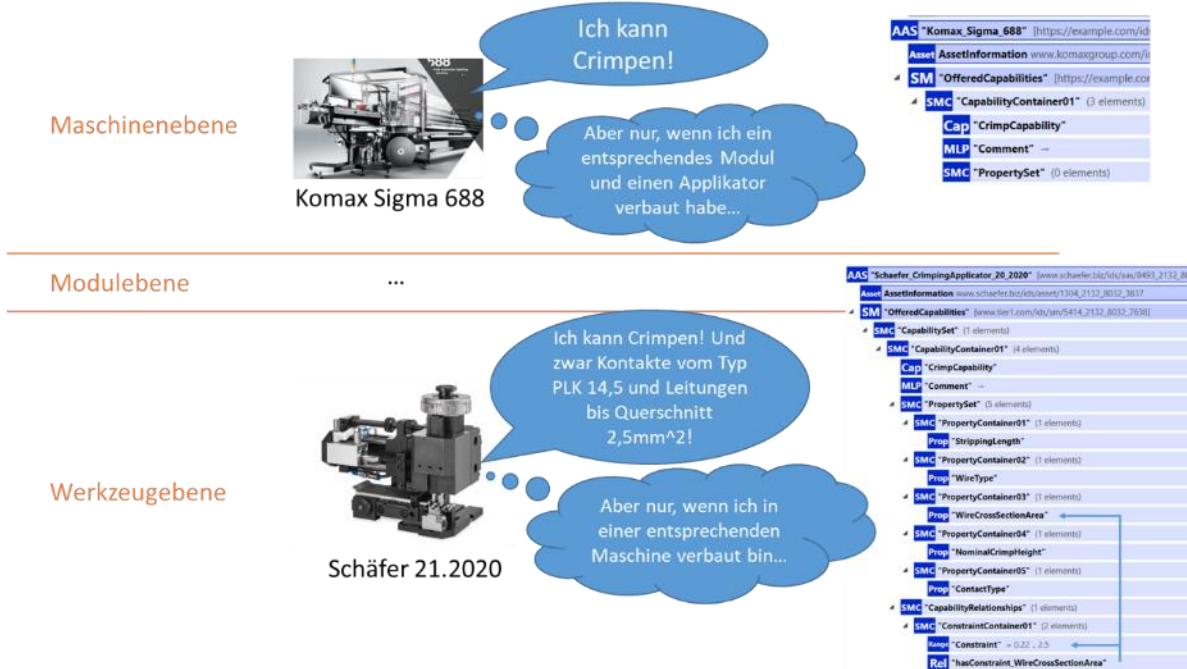


Abbildung 6-26. Konzept zur Modellierung zusammengesetzter Fähigkeiten

Sowohl eine Modellierung lediglich auf Seiten der Maschine als auch eine Modellierung lediglich auf Seiten des Werkzeugs bringt dabei gravierende Nachteile mit sich, da bestimmte Parameter oftmals sowohl von Maschinen- als auch von Werkzeugseite her eingeschränkt werden. Hierdurch wird eine kombinierte Modellierung dieser Einschränkungen sehr schnell sehr komplex, da viele Abhängigkeiten berücksichtigt werden müssen.

Aus diesem Grund bietet es sich stattdessen an, eine geteilte Modellierung vorzunehmen. Dabei können Fähigkeiten und entsprechende Einschränkungen auf allen relevanten Ebenen modelliert werden, wie auf der rechten Seite von Abbildung 6-26 dargestellt.

Auf allen Ebenen können dabei die in Abschnitt 6.5.3.1 beschriebenen Konzepte genutzt werden. Einzig zur Abbildung der Notwendigkeit der Verknüpfung von Maschine und Werkzeug wird ein zusätzliches Modellierungskonzept benötigt. Hierzu kann das auch bereits in BaSys 4.2 angedachte Konzept der *CapabilityConditions* weiterentwickelt bzw. konkretisiert und eine spezielle Bedingung *RequiresToolCondition* definiert werden.

Wie in Abbildung 6-27 dargestellt, kann mit dieser Bedingung auf Maschinenebene beschrieben werden, dass zur Ausführung einer Fähigkeit ein bestimmter Werkzeugtyp benötigt wird. Statt hier explizit – bspw. per Referenz – auf ein konkretes Werkzeug zu verweisen, was bei mehreren möglichen Werkzeugen schnell zu einer hohen Komplexität führen würde, kann dabei explizit ein Typ von Werkzeugen angegeben werden. Das in Abbildung 6-27 dargestellte Modell erfordert bspw. ein Werkzeug des Typen *CrimpingApplicator*.

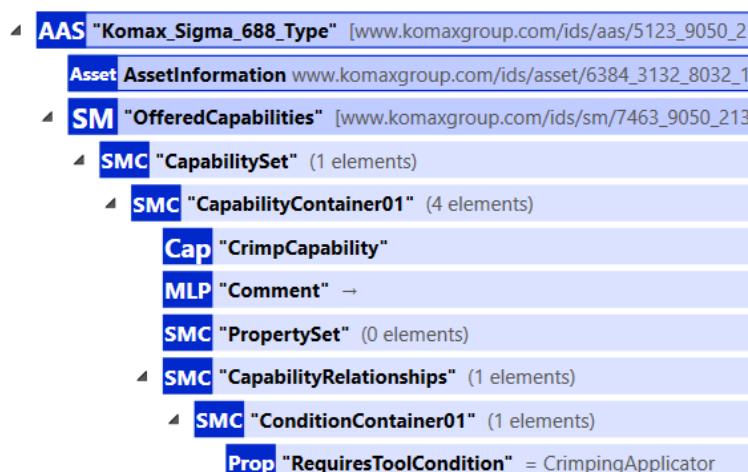


Abbildung 6-27: Bedingung "RequiresToolCondition"

Auf Seiten der Werkzeuge kann der entsprechende Werkzeugtyp einfach über ein entsprechendes *Extension*-Element beschrieben werden, wie in Abbildung 6-28 gezeigt. Diese indirekte Modellierung erlaubt die dynamische Ermittlung möglicher Maschine-Werkzeug-Kombinationen, ohne bspw. im Voraus bereits sämtliche in einer Fabrik vorhandene Werkzeuginstanzen zu kennen.

| AssetAdministrationShell (according IEC63278) | |
|---|--|
| Referable: | |
| idShort: | Schaefer_CrimpingApplicator_21_2020_Instance |
| HasExtension: | |
| Extension 1: requiredSlot | |
| name: | requiredSlot |
| semanticId: | (GlobalReference) http://arena2036.de/requiredSlot/1/0 |
| valueType: | xs:string |
| value: | KomaxApplicator |
| Extension 2: toolType | |
| name: | toolType |
| semanticId: | (GlobalReference) http://arena2036.de/toolType/1/0 |
| valueType: | xs:string |
| value: | CrimpingApplicator |

Abbildung 6-28: Beschreibung von Werkzeugtypen per "toolType"-Extension

6.5.4 Algorithmus zum Fähigkeitenabgleich

Aufbauend auf den in den Abschnitten 6.5.2 und 6.5.3 dargelegten Konzepten zur Beschreibung von geforderten und angebotenen Fähigkeiten lässt sich nun ein Algorithmus ableiten, mit dessen Hilfe ein Fähigkeitenabgleich durchgeführt werden kann. Ein solcher Algorithmus soll dabei ermitteln, ob eine bestimmte Maschine (evtl. unter der Zuhilfenahme entsprechender Werkzeuge) in der Lage ist, eine bestimmte Fähigkeit anzubieten bzw. auszuführen.

Eingangsvoraussetzung für den Algorithmus ist folglich eine bestimmte geforderte Fähigkeit – also der Verweis auf einen *CapabilityContainer*, der Teil eines *RequiredCapability*-Teilmodells ist (vgl. Abschnitt 6.5.2) – und eine (Liste von) Maschine(n), die beurteilt werden soll – also eine bestimmte (Liste von) Asset- bzw. AAS-IDs.

Auf Basis dieser Informationen kann der Algorithmus dann sequenziell die folgenden Schritte ausführen bzw. Fragen beantworten:

1. *Kann die Maschine die Fähigkeit grundsätzlich erfüllen?*

Hierzu wird das *OfferedCapabilities*-Teilmodell der Maschine ausgewertet. Es wird zunächst überprüft, ob dieses grundsätzlich die geforderte Fähigkeit beschreibt, und ob darüber hinaus auch die entsprechenden Parametereinschränkungen erfüllt sind. Hierzu werden die definierten Constraints mit den geforderten Werten verglichen.

Falls für die entsprechende Fähigkeit nicht zusätzlich noch eine *RequiredToolCondition* spezifiziert ist, kann nach diesem Schritt bereits eine Aussage über die Erfüllbarkeit der geforderten Fähigkeit getroffen werden.

2. *Welche Werkzeuge sind bekannt, die zur Realisierung der Capabilities genutzt werden können?*

Ist auf Seiten der Maschine für die geforderte Fähigkeit eine *RequiredToolCondition* spezifiziert, müssen zunächst alle bekannten Werkzeuge vom entsprechenden *toolType* ermittelt werden. Dies kann bspw. durch Abfrage eines entsprechenden AAS-Repositories erfolgen.

Für jedes gefundene Werkzeug ist anschließend analog zu Schritt 1 zu beurteilen, ob dieses Werkzeug grundsätzlich in der Lage ist, die entsprechende geforderte Fähigkeit bereitzustellen.

3. *(Wie) können die möglichen Werkzeuge in die Maschine montiert werden?*

Für jedes in Schritt 2 identifizierte mögliche Werkzeug muss anschließend geprüft werden, ob bzw. wie dieses Werkzeug in die Maschine montiert werden kann. Hierzu kann das im Rahmen von AP5.3 entwickelte Konzept der *Abbildung möglicher Maschinenstrukturen* die Grundlage bieten, indem es die Voraussetzung zur Ableitung möglicher Montagepfade bietet.

4. Welche der gefundenen Kombinationen erfüllen alle Constraints?

Für jeden der gefundenen möglichen Montagepfade müssen abschließend noch einmal die entsprechenden *Constraints* ausgewertet werden. Ein Montagepfad muss zudem nicht immer direkt sein (Maschine ↔ Werkzeug), sondern kann eventuell Zwischenebenen besitzen (z.B. Maschine ↔ Adapter ↔ Werkzeug). Gemäß des in Abschnitt 6.5.3.2 entwickelten Konzeptes können für diese Zwischenebenen theoretisch ebenfalls Einschränkungen für die Fähigkeiten-Parameter beschrieben werden. Diese sind folglich ebenfalls gegenüber den geforderten Parametern zu prüfen.

Nach durchlaufen dieser Schritte kann der Algorithmus nicht nur eine Aussage liefern, ob eine geforderte Fähigkeit von einer Maschine erfüllt werden kann oder nicht, sondern auch, welche Werkzeuge hierzu benötigt werden und wie diese Werkzeuge in die Maschine verbaut werden können. Diese Informationen können bspw. zur Planung von Rüstvorgängen durch Produktionsmitarbeiter oder zum Abgleich von vorhandenen und benötigten Maschinenstrukturen genutzt werden.

Eine prototypische Implementierung des in diesem Abschnitt beschriebenen Algorithmus steht unter <https://github.com/VWS4LS/vws4ls-capability-matching> bereit.

6.5.5 Anbindung von Skills

Die bisher in diesem Dokument betrachteten angebotenen *Fähigkeiten* bzw. *Capabilities* ermöglichen eine Bewertung, ob eine bestimmte Fähigkeit von einer Ressource umgesetzt werden kann – sie stellen folglich den Einstiegspunkt bei der Suche nach Maschinen zur Umsetzung bestimmter Prozesse dar.

Nachdem bspw. durch ein MES auf Basis eines Fähigkeitenabgleichs eine passende Maschine gefunden wurde, stellt sich als nächstes allerdings die Frage, wie diese Fähigkeit nun abgerufen werden kann. Es stellt sich folglich die Frage nach einer konkret ausführbaren *Fertigkeit* bzw. *Skill*.

Im Rahmen von TP1 wurde zur Implementierung solcher konkret ausführbaren Fähigkeiten eine *OPC UA Companion Specification* für die Prozesse der Leitungssatz-Produktion entwickelt (OPC 40570) [18]. Wie in Abbildung 6-29 dargestellt, muss daher ein automatischer Übergang von der im Rahmen einer Verwaltungsschale beschriebenen Fähigkeit zu einer durch einen OPC UA-Server auf einer Maschine bereitgestellten Fertigkeit ermöglicht werden.

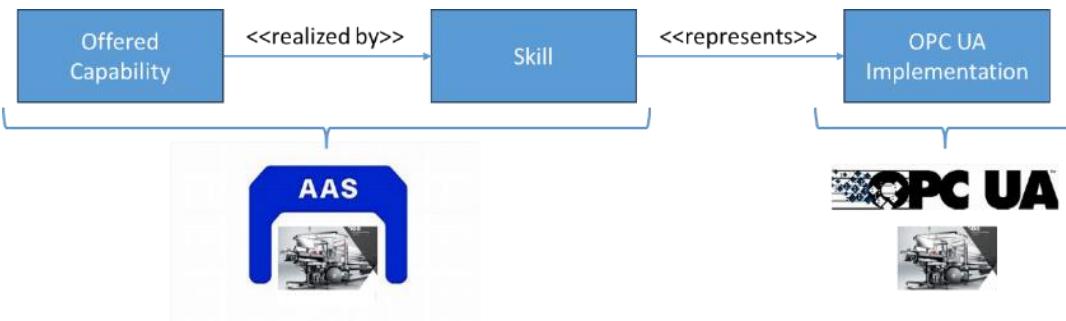


Abbildung 6-29: Zusammenhang zwischen Fähigkeit (Capability) und Fertigkeit (Skill)

Um eine solche automatische Durchgängigkeit realisieren zu können, bietet das im Rahmen von BaSys 4.2 entwickelte Teilmodell bereits die Möglichkeit, mit Hilfe einer *realizedBy*-Beziehung auf einen entsprechenden *Skill* zu verweisen (vgl. Abbildung 6-29). Im Rahmen des Teilmodells ist allerdings nicht festgelegt, wie solch ein Skill konkret aussieht.

Daher wurden im Rahmen von AP5.5 verschiedene bestehende und in Entwicklung befindliche Teilmodelle daraufhin analysiert, inwiefern sie eine adäquate Beschreibung des OPC UA Servers einer Maschine ermöglichen. Hierfür wurde das Teilmodell „IDTA 02017: Asset Interface Description“, [42] ausgewählt, welches die Beschreibung von Datenquellen ermöglicht, die von einem Asset angeboten werden. Da in der aktuellen Version 1.0 allerdings noch keine Unterstützung von OPC UA enthalten ist, wurde diese prototypisch ergänzt (vgl. Abbildung 6-30) – bei einer offiziellen Ergänzung in einer zukünftigen Version kann diese prototypische Umsetzung dann entsprechend ausgetauscht werden.

Relevante Punkte für den Übergang aus der Verwaltungsschale zum OPC UA-Server sind dabei vor allem:

- der Endpunkt des OPC UA-Servers (siehe Element *EndpointMetadata* in Abbildung 6-30),
- dass der OPC UA-Server die oben erwähnte *Companion Specification* für die Leitungssatzproduktion umsetzt (siehe *supplementalSemanticId*, „<http://www.w3.org/2011/opc40570>“ in Abbildung 6-30), sowie
- Identifizierbarkeit des für die Ausführung des Skill relevanten *Nodes* (siehe z.B. *Node „CutToLengthInstance01“* in Abbildung 6-30) – dieser ist auch gleichzeitig das Ziel der *realizedBy*-Beziehung.

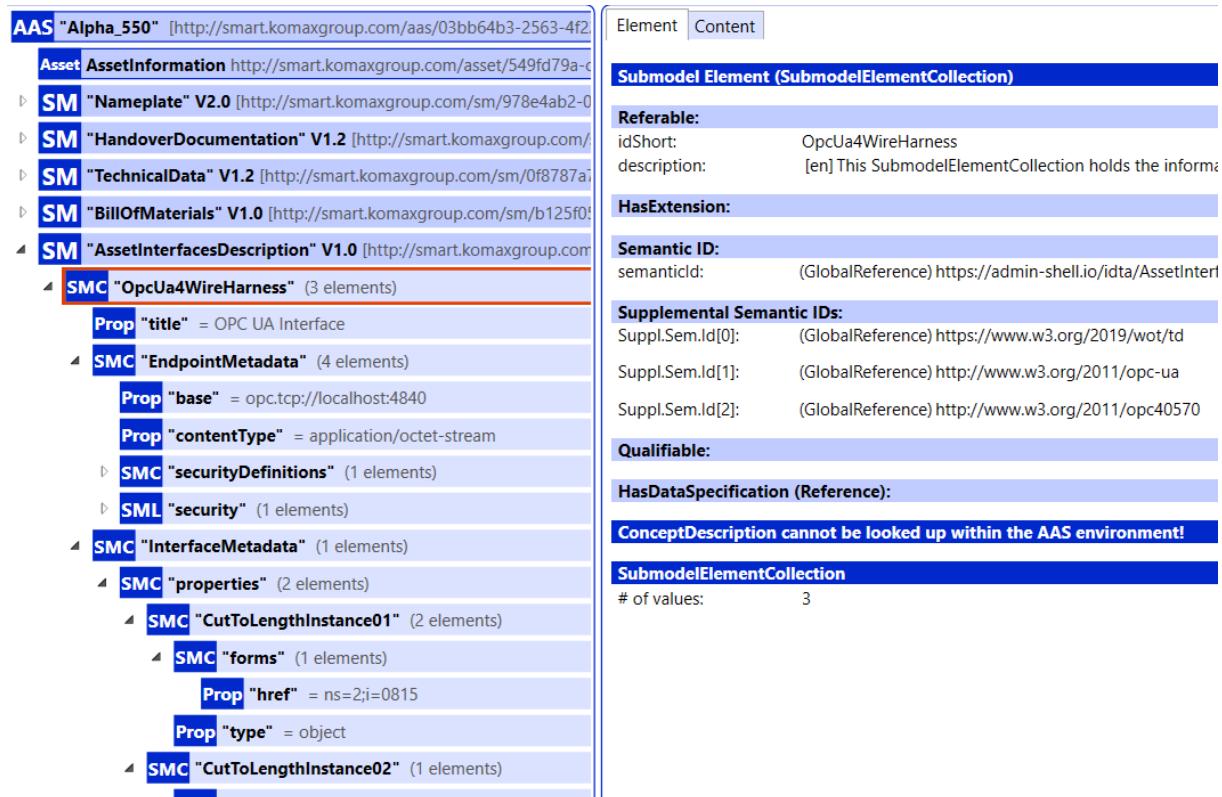


Abbildung 6-30: Prototypische Ergänzung des AID-Teilmodells [42] für OPC UA

6.5.6 Anhang: Liste von Fähigkeiten und zugehörigen Parametern

In den nachfolgenden Kapiteln sind die Parameter der Fähigkeiten der jeweiligen Produktionsprozesse des Leitungssatzes definiert. Dieser Abschnitt ist nicht normativ und soll als Vorschlag für eine mögliche Standardisierung dienen. Siehe auch <https://github.com/VWS4LS/vws4ls-aaspe-plugin/blob/main/src/AasxPluginVws4ls/Utils/Vws4lsCapabilitySMUtils.cs> [24].

6.5.6.1 Cut

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/CutCapability>

Parameter:

- NominalLength (xs:double)
- LengthUpperLimit (xs:double)
- LengthLowerLimit (xs:double)

6.5.6.2 Cut Wire

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/CutWireCapability>

Parameter:

- NominalLength (xs:double)
- LengthUpperLimit (xs:double)
- LengthLowerLimit (xs:double)
- WireType (xs:string)
- WireCrossSectionArea (xs:double)

6.5.6.3 Cut Tube

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/CutTubeCapability>

Parameter:

- NominalLength (xs:double)
- LengthUpperLimit (xs:double)
- LengthLowerLimit (xs:double)
- TubeType (xs:string)
- TubeDiameter (xs:double)

6.5.6.4 Strip

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/StripCapability>

Parameter:

- NominalStrippingLength (xs:double)
- StrippingLengthUpperLimit (xs:double)
- StrippingLengthLowerLimit (xs:double)
- CenterStripping (xs:boolean)
- Layer (xs:boolean)
- IncisionMonitoring (xs:boolean)
- WireEnd (xs:string)

6.5.6.5 Slit

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/SlitCapability>

Parameter:

- NominalSlittingLength (xs:double)
- Layer (xs:boolean)
- WireEnd (xs:string)

6.5.6.6 Crimp

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/CrimpCapability>

Parameter:

- WireType (xs:string)
- WireCrossSectionArea (xs:double)
- TerminalPartNumber (xs:string)
- CrimpForceMonitoring (xs:boolean)
- CrimpHeightUpperLimit (xs:double)
- CrimpHeightLowerLimit (xs:double)
- CrimpWidthUpperLimit (xs:double)
- CrimpWidthLowerLimit (xs:double)

6.5.6.7 Mark Wire

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/MarkWireCapability>

Parameter:

- MarkingType (xs:string)
- ContentType (xs:string)
- CharHeight (xs:double)
- Color (xs:string)

6.5.6.8 Tinning

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/TinningCapability>

Parameter:

- TemperatureAccuracyUpperLimit (xs:double)
- TemperatureAccuracyLowerLimit (xs:double)

6.5.6.9 Refinement Of Cable Lugs

SemanticId:

<https://arena2036.de/vws4ls/capability/1/0/RefinementOfCableLugsCapability>

Parameter:

- SolderDurationUpperLimit (xs:double)
- SolderDurationLowerLimit (xs:double)

6.5.6.10 Seal

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/SealCapability>

Parameter:

- SealPartNumber (xs:string)
- SealPositionUpperLimit (xs:double)
- SealPositionLowerLimit (xs:double)

6.5.6.11 Sleeve

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/SleeveCapability>

Parameter:

- SleevePartNumber (xs:string)
- PullOutCheck (xs:boolean)

6.5.6.12 Strand Twist

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/StrandTwistCapability>

Parameter:

- WireCrossSectionArea (xs:double)

6.5.6.13 Shield Twist

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/ShieldTwistCapability>

Parameter:

- ---

6.5.6.14 Blockload

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/BlockloadCapability>

Parameter:

- ---

6.5.6.15 Connector And Housing

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/ConnectorAndHousingCapability>

Parameter:

- ConnectorPartNumber (xs:string)
- InsertionForceCurveMonitoring (xs:boolean)

6.5.6.16 Ultra Sonic Weld

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/UltraSonicWeldCapability>

Parameter:

- SumCrossSectionArea (xs:double)

6.5.6.17 Terminal

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/TerminalCapability>

Parameter:

- TerminalPartNumber (xs:string)

6.5.6.18 Cover

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/CoverCapability>

Parameter:

- BundleDiameter (xs:double)
- StartNode (xs:string)
- EndNode (xs:string)

6.5.6.19 Shrink

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/ShrinkCapability>

Parameter:

- NominalTemperature (xs:double)

6.5.6.20 Spot Tape

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/SpotTapeCapability>

Parameter:

- ---

6.5.6.21 Tape

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/TapeCapability>

Parameter:

- TapingMethod (xs:string)

6.5.6.22 Tube

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/TubeCapability>

Parameter:

- TubeType (xs:string)

6.5.6.23 Wire Twist

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/WireTwistCapability>

Parameter:

- NominalWireEndLength (xs:double)
- NominalTensionForce (xs:double)
- NominalOpenEndSide1 (xs:double)
- NominalTrimmedOpenEndLengthSide1 (xs:double)
- SpotTapeSide1 (xs:boolean)
- NominalOpenEndSide2 (xs:double)
- NominalTrimmedOpenEndLengthSide2 (xs:double)
- SpotTapeSide2 (xs:boolean)
- WireEndLengthUpperLimit (xs:double)
- WireEndLengthLowerLimit (xs:double)
- LayLengthUpperLimit (xs:double)
- LayLengthLowerLimit (xs:double)
- OpenEndLowerLimitSide1 (xs:double)
- OpenEndUpperLimitSide1 (xs:double)
- OpenEndLowerLimitSide2 (xs:double)
- OpenEndUpperLimitSide2 (xs:double)

6.5.6.24 Simple Twist

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/SimpleTwistCapability>

Parameter:

- NominalElasticity (xs:double)

6.5.6.25 Route

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/RouteCapability>

Parameter:

- ---

6.5.6.26 Fuse Box Assembly

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/FuseBoxAssemblyCapability>

Parameter:

- FuseBoxPartNumber (xs:string)
- MountForceAccuracyLowerLimit (xs:double)
- MountForceAccuracyUpperLimit (xs:double)

6.5.6.27 Foam

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/FoamCapability>

Parameter:

- FoamObjectPartNumbers (xs:string)

6.5.6.28 Screw

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/ScrewCapability>

Parameter:

- FuseBoxPartNumber (xs:string)
- PositionName (xs:string)
- WrenchSize (xs:double)
- NominalTorque (xs:double)
- TighteningCurve (xs:boolean)
- TorqueLowerLimit (xs:double)
- TorqueUpperLimit (xs:double)
- AngleLowerLimit (xs:double)
- AngleUpperLimit (xs:double)

6.5.6.29 Scan

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/ScanCapability>

Parameter:

- ComponentPartNumber (xs:string)
- BarcodePosition (xs:string)
- BarcodeType (xs:string)
- Grading (xs:boolean)

6.5.6.30 Test

SemanticId: <https://arena2036.de/vws4ls/capability/1/0/TestCapability>

Parameter:

- TestSpecificationId (xs:string)

6.6 AP 5.6 - Erfassung von Qualitätsdaten

6.6.1 Zielsetzung

Das AP 5.6 befasst sich mit der Erfassung und Auswertung von Qualitätsdaten von bei der Leitungssatz-Produktion ausgeführten Prozessen. Besonderes Augenmerk wurde daraufgelegt, wie eine solche Erfassung und Auswertung vor dem Hintergrund von Verbundkomponenten erfolgen kann: Wie bereits im Rahmen der Beschreibung angebotener Fähigkeiten erläutert (s. AP5.5), führen Fertigungsmaschinen bestimmte Prozesse häufig nicht allein aus, sondern benötigen hierzu spezielle Werkzeuge. Eine Ermittlung von Qualitätsdaten vor diesem Hintergrund ist besonders herausfordernd, da (1) Maschine und Werkzeug oftmals nicht vom gleichen Hersteller stammen und (2) Werkzeuge oftmals passive Komponenten sind, welche über keine eigene Kommunikationsschnittstelle verfügen.

Um zu zeigen, wie diesen Herausforderungen mit Hilfe der Verwaltungsschale begegnet werden kann wurde im Rahmen von AP5.6 folgender Use Case umgesetzt: Betrachtet wird ein Crimp-Prozess, der durch eine Maschine unter Nutzung eines speziellen Werkzeugs (Applikator) ausgeführt wird. Da der Applikator über die Zeit verschleißt und irgendwann ausfällt, sollen Qualitätsdaten ermittelt werden. Auf Basis dieser Qualitätsdaten kann der Hersteller des Applikators dann durch einen speziellen Algorithmus ermitteln, wann ein Ausfall zu befürchten ist und entsprechend frühzeitig eine Warnung auslösen, woraufhin der Applikator rechtzeitig gewartet/ausgetauscht werden kann („Predictive Maintenance“).

Da der Applikator eine passive Komponente ohne Kommunikationsschnittstelle und ohne Sensorik ist, kann er nicht selbstständig feststellen, ob er „benutzt“ wurde. Die Auswertungsalgorithmen wiederum sind geistiges Eigentum des Herstellers des Applikators und können somit nicht einfach durch den Hersteller/Betreiber der Maschine implementiert/bereitgestellt werden. Aus diesem Grund soll der Applikator über eine aktive Verwaltungsschale verfügen, in der durch die (Verwaltungsschale der) Maschine relevante Qualitäts- bzw. Nutzungsdaten hinterlegt werden und die daraufhin selbstständig die entsprechenden Algorithmen zur Auswertung dieser Daten ausführen kann.

Als plakatives Beispiel wurde im Rahmen von AP5.6 davon ausgegangen, dass der Applikator unabhängig von anderen Faktoren typischerweise eine bestimmte Anzahl von Crimp-Vorgängen aushält und danach getauscht werden muss. Die entsprechenden Qualitätsdaten lassen sich somit in Form einer einfachen „Zählervariable“ in der Verwaltungsschale des Applikators hinterlegen, die durch die Maschine bei Ausführung eines Crimp-Prozesses erhöht wird. Die Verwaltungsschale des Applikators kann diese „Zählervariable“ überwachen und bei Überschreiten eines vorher festgelegten Grenzwertes ein entsprechendes Wartungs-Event auslösen. Die folgende Abbildung zeigt das Grundprinzip des beschriebenen Use Cases:

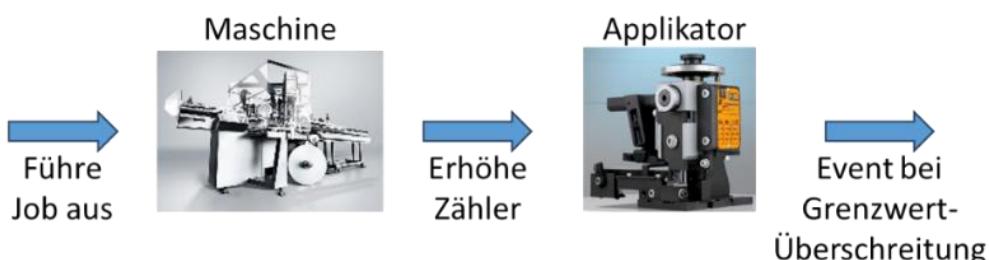


Abbildung 6-31: Konzept Use Case "Predictive Maintenance"

6.6.2 Relevante Teilmodelle

Für die Umsetzung des Use Cases „Predictive Maintenance“ wurde zunächst untersucht, welche bestehenden IDTA-Teilmodelle hierfür genutzt werden können. Dabei wurden die folgenden drei Teilmodelle als relevant eingestuft:

- **IDTA 02008 Time Series Data [43]**

Mit Hilfe dieses Teilmodells können beliebige Zeitreihendaten abgebildet werden. Für den oben beschriebenen Use Case kann es genutzt werden, um den zeitlichen Verlauf der „Zählervariable“ abzubilden. Es wird folglich für jeden ausgeführten Crimp-Prozess ein entsprechender Datenpunkt mit dem neuen Zählerstand (und dem zugehörigen Zeitstempel) angelegt.

- **IDTA 02048 Predictive Maintenance [44]**

Dieses im Rahmen des Forschungsprojektes [InterOpera](#)⁴⁴ erstellte Teilmodell deckt verschiedene Aspekte rund um den Bereich „Predictive Maintenance“ ab. Für den oben beschriebenen Use Case sind vor allem die folgenden beiden Bereiche relevant:

„**Condition Indicators**“ mit entsprechenden „**Condition Levels**“. Im Rahmen des betrachteten Use Cases ist der im Rahmen des Zeitreihenmodells abgebildete „Zählerstand“ ein solcher Indikator – im Rahmen des Teilmodells lässt sich ein Verweis auf das entsprechende *Time Series Data*-Teilmodell hinterlegen. Mit Hilfe von „Condition Levels“ lassen sich dann mit Bezug auf den referenzierte Wert bestimmte Intervalle definieren – bspw. ein Intervall „Normalbetrieb“ und ein Intervall, welches besagt, dass der Applikator ausgetauscht werden soll.

„**Remaining Useful Life**“: Für den betrachteten Use Case stellt dieses die voraussichtlich verbleibende Anzahl an möglichen Crimp-Vorgängen bis zum Ausfall des Werkzeugs dar.

- **IDTA 02010 Service Request Notification [45]**

Das Teilmodell „Service Request Notification“ erlaubt die Beschreibung von Aufforderungen zu Wartungstätigkeiten. Für den betrachteten Use Case kann dieses Teilmodell dazu genutzt werden, den durchzuführenden Werkzeugwechsel zu kommunizieren und dann bspw. in einem MES-System anzuseigen.

6.6.3 Zusammenspiel der verschiedenen Teilmodelle

Für den betrachteten Use Case spielen die in Abschnitt 6.6.2 beschriebenen Teilmodelle wie folgt zusammen: Die (Verwaltungsschale der) Maschine legt bei Ausführung eines Crimp-Prozesses im Teilmodell „Time Series Data“ des aktuell montierten Werkzeugs einen neuen Eintrag an. Der neue Eintrag enthält dabei einen gegenüber dem letzten Wert inkrementierten Zählerstand.

Die Verwaltungsschale des Werkzeugs überwacht Änderungen in ihrem Teilmodell „IDTA 02008: Time Series Data“ [43] und aktualisiert die „Remaining Useful Live“ im Teilmodell IDTA 02048: Predictive Maintenance“ [44], welche sich aus der erwarteten Lebensdauer abzüglich des aktuellen Zählerstandes ergibt.

Zusätzlich prüft die Verwaltungsschale des Werkzeugs den Eintritt in das „Condition Level“, welches die vorsorgliche Wartung bzw. den Austausch des Applikators vorsieht. Bei Erreichen dieses Levels wird im Teilmodell „Service Request Notification“ ein neuer Service-Antrag angelegt, welcher die Wartung bzw. den Austausch des Applikators fordert.

Die folgende Abbildung fasst die beschriebenen Zusammenhänge zusammen und zeigt die Strukturen der unterschiedlichen Teilmodelle:

⁴⁴ <https://interopera.de/>

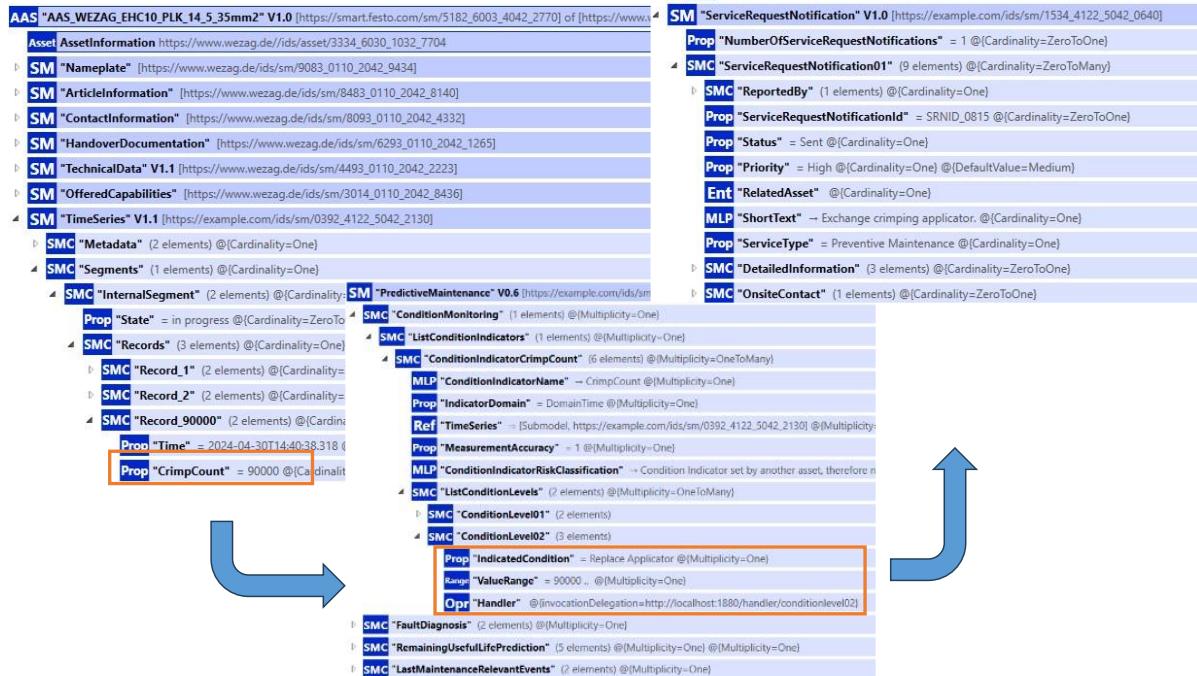


Abbildung 6-32: Zusammenhang der verschiedenen Teilmodelle

6.6.4 Prototypische Implementierung

Der in den bisherigen Abschnitten dieses Dokuments aufgezeigte Use Case wurde im Rahmen von AP 5.6 prototypisch umgesetzt. Hierzu wurden einerseits die entsprechenden Verwaltungsschalen und Teilmodelle aufgesetzt (vgl. Abbildung 6-32) und per Basyx⁴⁵-Server bereitgestellt. Darüber hinaus wurden aktive Verwaltungsschalen sowohl für eine Maschine als auch für einen Applikator aufgesetzt, um die in Abbildung 6-31 dargestellte Interaktion umsetzen zu können. Die aktiven Verwaltungsschalen wurden dabei durch zwei Node-RED⁴⁶-Instanzen umgesetzt, die die in Abschnitt 6.6.3 Algorithmen umsetzen. Abbildung 6-33 zeigt exemplarisch die Node-RED-Umsetzung zur Auswertung des Teilmodells „Predictive Maintenance“ inklusive der Abonnierung von Änderungen am Teilmodell „Predictive Maintenance“ per MQTT sowie der Reaktion auf entsprechende Änderungen durch Anpassen der „Remaining Useful Life“ sowie des Aufrufs eines entsprechenden Handlers bei Eintritt in ein neues „Condition Level“.

Die Konfigurationsdateien hierfür finden sich in „[basyx-predictive-maintenance-multiple-node-red-instances.zip](#)“

⁴⁵ <https://basyx.org/>

⁴⁶ <https://nodered.org/>

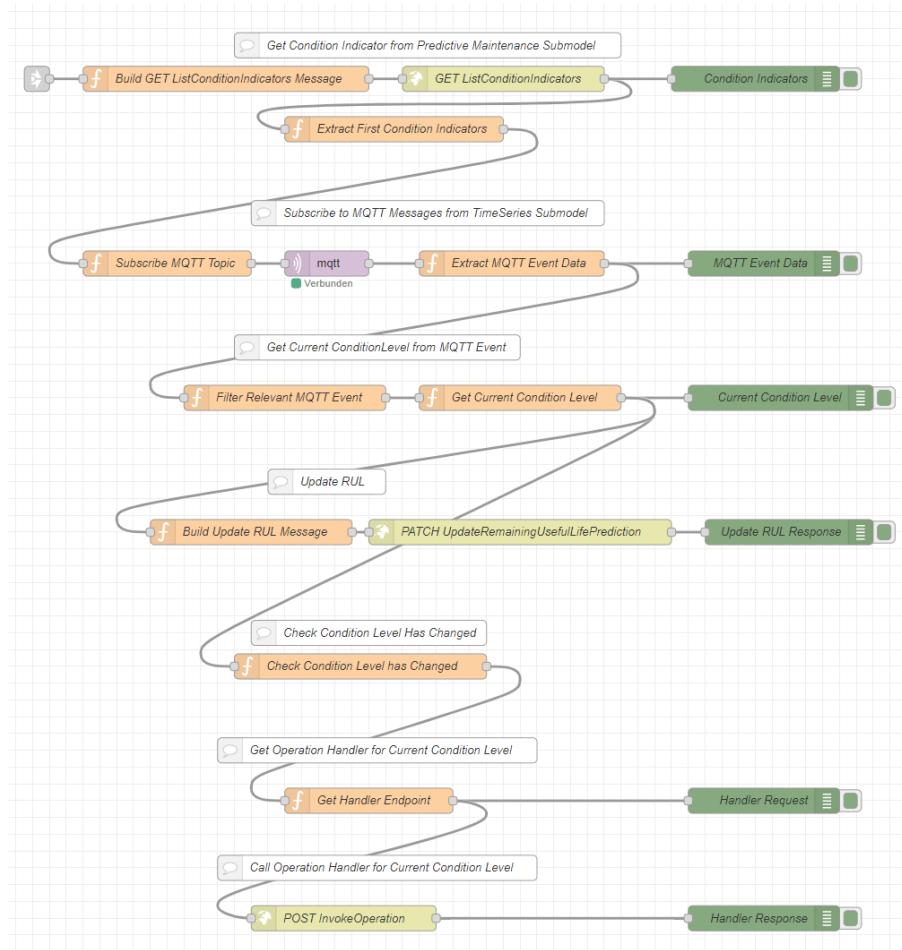


Abbildung 6-33: NodeRed-Flow zur Auswertung des Teilmodells "Predictive Maintenance"

6.7 Fazit

Im Rahmen von TP5 wurde das Konzept der *Verbundkomponente* genutzt, um vernetzte Verwaltungsschalenstrukturen zu realisieren. Hierzu wurden zunächst sowohl für Produkte als auch Ressourcen relevante Beziehungen zwischen den unterschiedlichen Typen von Verwaltungsschalen analysiert sowie für jede gefundene Beziehungsart entsprechende Modellierungsmöglichkeiten ermittelt bzw. entworfen. Mithilfe dieser Modellierungsmöglichkeiten kann dann ein Netzwerk von verknüpften Verwaltungsschalen erstellt werden, welches vor allem auch die im Rahmen der anderen Teilprojekte entworfenen Verwaltungsschalen und Teilmodelle miteinander in Beziehung setzt. Ein solches Netzwerk wurde exemplarisch unter anderem für das im Rahmen des „PPR-Workshops“ definierte Beispielprodukt erstellt.

Der zweite Teil von TP5 hat sich dann mit der Schaffung bzw. dem Aufzeigen von Mehrwerten befasst, die auf Basis aus einer entsprechenden vernetzten Struktur von Verwaltungsschalen realisieren lassen. Hierzu wurden die folgenden vier Use Cases definiert sowie deren Umsetzung analysiert:

1. Rückverfolgbarkeit im Fehlerfall
2. Synchronisierung von Verwaltungsschalen
3. Sachnummern-Mapping
4. Fähigkeitenabgleich

Besonderes Augenmerk wurde auf das Thema „Fähigkeitenabgleich“ gelegt. Entsprechende Modellierungsmöglichkeiten und Algorithmen wurden im Rahmen von AP5.5 entworfen, wobei ein besonderer Fokus auf der „gemeinsamen“ Realisierung von Fähigkeiten aus einer Kombination aus Maschine und Werkzeug lag. Abschließend wurde im Rahmen eines „Predictive Maintenance“-Use Cases gezeigt, wie sich Mehrwerte durch eine Firmen- und VWS-übergreifende Zusammenarbeit generieren lassen.

7 TP6 - Automatisierte Verhandlungsprozesse

Im TP 6 "Automatisierte Verhandlungsprozesse" wurde ein Konzept mit Randbedingungen für automatisierte Entscheidungen definiert und Wege für die Kommunikation und Interaktion zwischen autonomen I4.0-Komponenten (z.B. Maschine zu Maschine, Leitungssatz zu Maschine oder Leitstand (MES) zu Maschine) entwickelt. Die Verhandlung zwischen den Systemen soll autonom durch die sog. "I4.0-Komponenten" geführt werden, die anhand von Daten und Informationen Entscheidungen treffen.

Hierzu wurde als Mechanismus zum Austausch von I4.0-Nachrichten das Konzept der "Operation" des Verwaltungsschalen-Metamodells vorgeschlagen.

Das TP 6 ist wie folgt gegliedert:

- AP 6.1 - Konzeptentwicklung für die technische und automatisierte Verhandlung
- AP 6.2 - Verhandlungsszenarien und -strategien
- AP 6.3 - Capabilities
- AP 6.4 - ECLASS und Industrie 4.0-Sprache

7.1 AP 6.1 - Konzeptentwicklung für die technische und automatisierte Verhandlung

Im AP 6.1 "Konzeptentwicklung für die technische und automatisierte Verhandlung" wurde ein Konzept mit Randbedingungen für das eigenständige Entscheiden definiert und Wege für die Kommunikation und Interaktion zwischen autonomen I4.0-Komponenten aufgezeigt.

Als ein erster Ansatz der aktiven Verwaltungsschale wird die Verbindung mit dem sog. Komponentenmanager betrachtet. Der Komponentenmanager soll anhand von Algorithmen ein zielgerichtetes Verhalten einer I4.0-Komponente (VWS und dazugehöriges Asset) beschreiben. Es wird ein Vorschlag für die Umsetzung unter Verwendung von Low-Code-Plattformen skizziert.

7.1.1 Definition von Anwendungsfällen

Für die Arbeitspakete AP 6.1 bis AP 6.4 werden konkrete Anwendungsfälle definiert, anhand derer die Verhandlungskonzepte entwickelt und soweit möglich auch mit Demonstratoren validiert werden. Die Anwendungsfälle des TP 6 weisen einen Bezug zu den Use Cases des Gesamtprojektes VWS4LS auf und greifen sowohl Anforderungen als auch Ergebnisse anderer Teilprojekte auf.

Im Folgenden werden die als Anwendungsfälle ausgewählten Verhandlungsszenarien beschrieben.

7.1.1.1 Szenario „Order Driven Production“

7.1.1.1.1 Kontext

Angenommen wird ein Produktionsauftrag für ein definiertes Halbfabrikat „verdrilltes Leitungspaar“, für dessen Herstellung verschiedene Maschinen zum Einsatz kommen können, darunter auch Maschinen verschiedenen Typs. Die Herstellung kann dabei vollständig automatisiert auf einer einzigen Maschine erfolgen, die mehrere Fähigkeiten anbietet, oder alternativ über verkettete Arbeitsschritte an mehreren unterschiedlichen Maschinen, wobei auch manuelle Tätigkeiten enthalten sein können.

Im Sinne einer auftragsgetriebenen Produktion verhandelt die VWS des Produkts mit den in Frage kommenden Produktionsressourcen, welche konkreten Ressourcen zu seiner Herstellung eingesetzt werden. D.h., bei diesem Ansatz „sucht“ sich das Produkt seinen Weg durch die Fabrik. Inwieweit mit dieser Vorgehensweise Optimierungsziele auf Ebene einer Fertigungsinsel mit mehreren Maschinen oder einer ganzen Fabrik erreicht werden können, also den gesamten Auftragsbestand betreffend, wurde im Projektteam kritisch diskutiert. Es wurde entschieden, das Szenario dennoch zu betrachten. Zum einen kann das Verhandlungsfenster so eingeschränkt werden, dass die Entscheidung sich im Rahmen einer vorab festgelegten Grobplanung bewegt. Die Durchführung der automatisierten Verhandlungsprozesse hat dann zum Ziel eine Feinplanung zu realisieren, mit der auf a priori nicht bekannte Einflussfaktoren im kurzfristigen Bereich reagiert werden kann. Außerdem kann das Konzept auf einen automatisierten Verhandlungsprozess mit Beteiligung einer VWS, die einen Fabrikleitstand (MES) darstellt und die eine Sicht auf alle Produktionsressourcen und den gesamten Auftragsbestand hat, übertragen werden. Mit dieser Sicht kann im Verhandlungsprozess wieder ein „fabrikweites“ Optimum erreicht werden.

7.1.1.1.2 Technische Rahmenbedingungen

Für die konkrete Instanz des herzustellenden Halbfabrikats existiert eine zugehörige VWS „Produkt“ ab dem Zeitpunkt der Erstellung des Produktionsauftrags, d.h. zeitlich bevor das Asset physisch existiert. Die VWS „Produkt“ ist als proaktive VWS (Typ 3) ausgeführt, die über die Fähigkeit verfügt, im Interaktionsprotokoll Ausschreibungsverfahren nach VDI/VDE 2193-2 [46] in der Rolle des Auftraggebers (Service Requester) teilzunehmen.

Die Produktionsressourcen zur Herstellung des Halbfabrikats sind ebenfalls durch proaktive VWS (Typ 3) repräsentiert. Diese VWS verfügen über die Fähigkeit, im Interaktionsprotokoll Ausschreibungsverfahren in der Rolle als Auftragnehmer (Service Provider) teilzunehmen. Es wird in Betracht gezogen, dass stellvertretend für eine oder mehrere Produktionsressourcen ein MES oder ähnliches System die Rolle als Auftragnehmer einnimmt. Dieses System verfügt dann über eine VWS-Schnittstelle.

Das Auffinden der Verwaltungsschalen der in Frage kommenden Service Provider erfolgt über die Abfrage von vorab definierten VWS-Registries. An dieser Stelle ist eine Abfrage- bzw. Suchfunktion in der VWS-Registry erforderlich. Die Suchfunktion soll eine Liste von Verwaltungsschalen zurückliefern, die als Typ 3 VWS das entsprechende semantische Interaktionsprotokoll implementiert haben.

Die von der gemeinnützigen [Object Management Group \(OMG\)](#)⁴⁷ verwaltete und in ISO/IEC 19510:2013 [37] standardisierte „[Business Process Model and Notation](#)“ (BPMN) soll für die Modellierung und Digitalisierung der Prozesse verwendet werden. Dieser Standard wird international in vielen Organisationen eingesetzt und ermöglicht durch die visuelle Natur von BPMN ein besseres Verständnis komplexer Abläufe.

7.1.1.1.3 Erwartete Ergebnisse

Mit einem ersten Demonstrator-Aufbau wird geklärt werden, ob das Konzept der Umsetzung des proaktiven Anteils der VWS mittels einer BPMN-Workflow Engine trägt, wie in AP 3.3 beschrieben.

Es ist geklärt, wie die Nachrichten zwischen Verwaltungsschalen technisch ausgetauscht werden und welche Voraussetzungen an Teilmodelle, Modellierung der Beziehungen zwischen Verwaltungsschalen und Anforderungen an die VWS-Infrastruktur (VWS-Registry, AAS-Server) sich daraus ergeben.

Es ist validiert, dass die in den einzelnen Teilprojekten erarbeiteten Teilmodelle zu Fähigkeiten dafür geeignet sind, eine Entscheidung zu treffen, ob das Beispielprodukt mit einer bestimmten Ressource gefertigt werden kann oder nicht.

Es ist validiert, dass die in TP 5 erarbeitete Funktion zum Fähigkeitenabgleich dafür geeignet ist, eine Entscheidung zu treffen, ob das Beispielprodukt mit einer bestimmten Ressource gefertigt werden kann oder nicht.

Es ist definiert, welche Inhalte die Nachricht zur Anfrage beinhalten muss, damit der Auftragnehmer die notwendigen Informationen zur Entscheidung über die Machbarkeit (Fähigkeitenabgleich) erhält oder weiß, wo er diese abrufen kann.

Es ist definiert, welche Inhalte die Nachricht zum Angebot beinhalten soll, damit der Auftraggeber die Angebotsauswahl treffen kann.

7.1.1.1.4 Ablauf der Verhandlung

1. Ausschreibung

Die Initiative zur Verhandlung geht von der VWS des Produktes aus. Über die Abfrage von vorab definierten VWS-Registries werden die Verwaltungsschalen der Produktionsressourcen gefunden. Die VWS des Produkts sendet eine Ausschreibung an alle gefundenen Verwaltungsschalen der Produktionsressourcen. Neben der Beschreibung des Produktes beinhaltet die Ausschreibung die angefragte Gesamtmenge des Produkts, eine Bündelgröße sowie den gewünschten Fertigstellungstermin.

2. Prüfung der Machbarkeit

⁴⁷ <https://www.omg.org/spec/BPMN>

Durch einen Abgleich der vom Produkt geforderten Fähigkeiten mit den in der Produktionsressource verfügbaren Fähigkeiten wird in der VWS der Produktionsressource eine Entscheidung zur Machbarkeit getroffen. Dabei ist explizit vorgesehen, dass nicht alle erforderlichen Produktionsschritte durch die Ressource ausgeführt werden können, sondern dass nur ein Teil der Produktionsschritte mit den verfügbaren Fähigkeiten machbar sind. Ein Angebot soll in diesem Fall trotzdem abgegeben werden. Die Angebotsnachricht enthält eine Kennzeichnung, welche mit der Ausschreibung angefragten Produktionsschritte Bestandteil des Angebots sind und welche Produktionsschritte (aufgrund des Fähigkeitenabgleichs) nicht Bestandteil des Angebots sind.

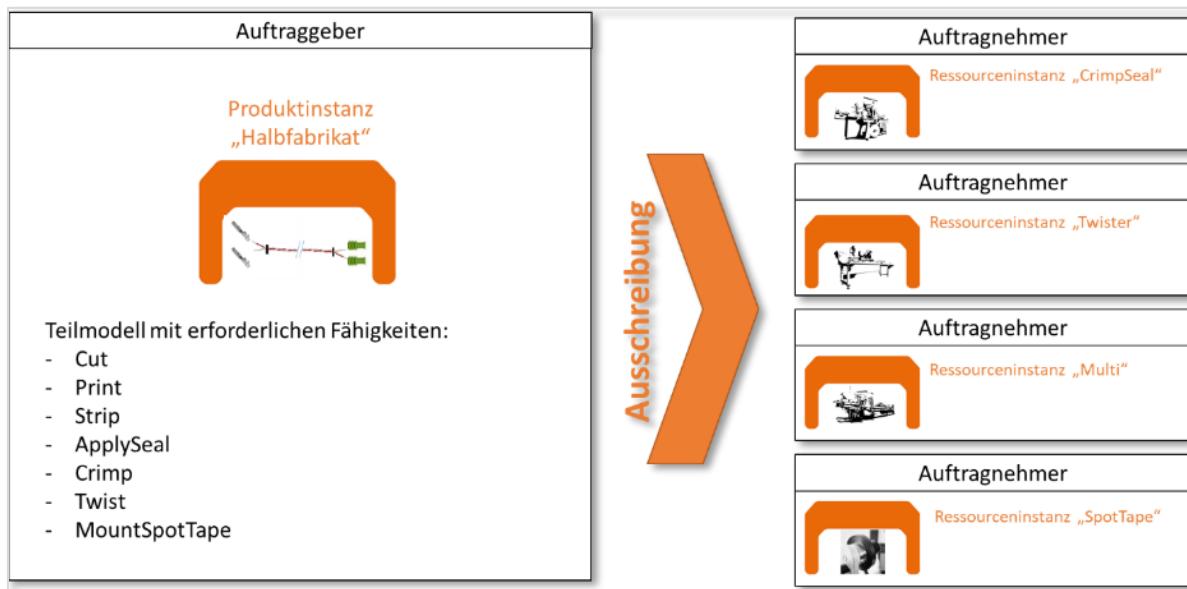


Abbildung 7-1: Verhandlungsszenario 1 „Order Driven Production“

7.1.1.5 Bezug zu VWS4LS Use Cases

Das Verhandlungsszenario „Order Driven Production“ adressiert in erster Linie den Use Case 4 „Automatisierung von flexiblen modularen Produktionsabläufen“. Mit dem Verhandlungsprozess wird die Grundlage für die Flexibilität innerhalb der Produktionsabläufe ohne zentrale Steuerungsinstanz geschaffen. So könnte z.B. nach dem Hinzufügen von zusätzlichen Produktionsmitteln mit neuen Fähigkeiten eine Zusteuerung von Produktionsaufträgen auf diese Produktionsmittel erfolgen, ohne dass ein zentraler Eingriff notwendig ist. Die Verhandlungsabläufe zwischen Produkt und Maschine und Maschine zu Maschine ermöglichen die Flexibilität in der Produktion, da sie kurzfristige Anpassungen erlauben und die Notwendigkeit für vorab stattfindende Planungsprozesse reduzieren.

Die im Szenario zu entwickelnden Möglichkeiten zum automatisierten Fähigkeitenabgleich sind im Use Case 2 „Berücksichtigen der automatisierten Produktionsfähigkeiten im Engineering“ relevant. Durch einen Verhandlungsprozess zwischen einer Typ-VWS zum Leitungssatz während des Engineering-Prozesses mit VWS von möglichen Produktionsressourcen können direkt Erkenntnisse zur automatisierten Produktionsfähigkeit gewonnen werden, indem man die Angebote der Produktionsressourcen aus dem Verhandlungsprozess entsprechend auswertet.

7.1.1.6 Bezug zu VWS4LS Teilprojekten

Die grundlegende Modellierung von geforderten und verfügbaren Fähigkeiten in der VWS sowie ein allgemeingültiger Algorithmus zum Fähigkeitenabgleich wurden in TP 5 entwickelt. Das Datenmodell zum Produkt Leitungssatz wurde in TP 2 definiert und die Grundlage zum Abbilden der Produktionsprozesse wurde in TP 3 erarbeitet. In der automatisierten Verhandlung wird auf Daten zu Produkt, Prozess und Ressource zugegriffen, damit Algorithmen Machbarkeit und Kosten bewerten und daraus Entscheidungen ableiten können.

7.1.1.2 Szenario „Zukaufteil“

7.1.1.2.1 Kontext

Angenommen wird ein Produktionsauftrag für das in Verhandlungsszenario “Order Driven Production” definiertes Halbfabrikat „verdrilltes Leitungspaar“. Dieses Halbfabrikat soll als Unterauftrag an ein weiteres Unternehmen vergeben werden. Gefordert ist die komplette Herstellung des Halbfabrikats, wobei Freiheitsgrade bezgl. der eingesetzten Produktionsressourcen und der einzelnen Produktionsschritte bestehen.

Die Verwaltungsschalen auf Seite der Auftragnehmer repräsentieren in diesem Kontext nicht einzelne Produktionsmaschinen, sondern sind hierarchisch höher auf Ebene Fabrik oder Unternehmen angesiedelt und agieren als Softwareagent, der die Gesamtheit der in der Organisationseinheit zur Verfügung stehenden Produktionsressourcen vertritt.

Welche Unternehmen als Auftragnehmer in Frage kommen, ist im Vorfeld geklärt worden. Die rechtlich notwendigen Voraussetzungen, wie z.B. Abschluss eines Rahmenvertrags, der den automatisierten Verhandlungsprozess mit Delegation von Entscheidungen an die Softwareagenten regelt, werden als gegeben betrachtet. Die technisch notwendigen Kommunikationswege wurden im Vorfeld eingerichtet.

Auf Seite des Auftragnehmers erfolgt jeweils eine Prüfung der technischen Machbarkeit zur Herstellung des ausgeschriebenen Produktes. Ist eine Machbarkeit nicht gegeben, wird die Ausschreibung abgelehnt und eine Absage gesendet. Ansonsten wird zur Angebotserstellung ein Preis und ein Liefertermin ermittelt. Zur Umsetzung verschiedener Auswahlstrategien auf der Auftraggeberseite könnten weitere Informationen im Angebot notwendig sein, z.B. PCF-Daten. Es ist zu prüfen, welche Lösungsvorschläge oder Spezifikationen für den Nachrichteninhalt “Angebot” bereits vorhanden sind und welche Ansätze es gibt, um geforderte Angebotsinhalte zu kommunizieren. Gegebenenfalls sind entsprechende Vorschläge in AP 6.2 zu erarbeiten.

In einer Erweiterung des Verhandlungsszenarios “Zukaufteil” sind zusätzlich für jedes Produktionslos bei Produktionsstart Schliffbilder der Crimpterminals mit Messung und Beurteilung bezüglich Einhaltung der OEM- und Herstellervorgaben gefordert. Die Übermittlung der Bilder und Messergebnisse wird innerhalb der Instanz-Verwaltungsschalen der gelieferten Produkte erwartet. Die Anforderung wird in Form einer “Required Capability” bzw. eines “Required Service” mit der Ausschreibung kommuniziert und ist bei der Überprüfung der Machbarkeit auf Auftragnehmerseite mit den eigenen Fähigkeiten abzugleichen.

Neben Anforderungen an das Produkt kann es auch grundlegende Anforderungen an das Unternehmen bzw. die Unternehmensprozesse des Auftraggebers geben. Als Beispiel dient die Anforderung, dass auf Auftragnehmerseite ein Informationssicherheitsmanagementsystem nach ISO 27001 etabliert ist. Zum Nachweis dient ein Security Certification Certificate (SCC), wie es im Diskussionspapier [“Vertrauensinfrastrukturen im Kontext von Industrie 4.0”](#) [47] der [Plattform Industrie 4.0](#) beschrieben ist. Es soll überprüft werden, ob die zu erarbeitenden Mechanismen zum Fähigkeitenabgleich auch dazu geeignet sind, Anforderungen zu SCC zu überprüfen.

7.1.1.2.2 Technische Rahmenbedingungen

Im Lebenszyklus Planungsstadium fungiert auf Auftraggeberseite eine Instanz-VWS des Halbfabrikats als Softwareagent, welcher ein Ausschreibungsverfahren initiiert. Auf Auftraggeberseite gibt es eine Registry für Verwaltungsschalen von potenziellen Auftragnehmern oder eine Liste mit Verweisen auf entsprechende Registries als Verzeichnis von Verzeichnissen.

Das Befüllen der Registry passiert während vorher stattfindender “Onboarding”-Prozesse neuer potenzieller Zulieferer. Ein IDS wie [Catena-X](#) kann hier über ein Unternehmensverzeichnis oder eine zentrale [“Digital Twin Registry”](#) die Lösung zum Auffinden der “Verhandlungspartner”-Verwaltungsschalen liefern (siehe dazu TP 8).

Zur Durchführung dieser Machbarkeitsprüfung kann die VWS auf Auftragnehmerseite wiederum ein weiteres Ausschreibungsverfahren an die internen Produktionsressourcen zur Herstellung des ausgeschriebenen Produktes starten, analog dem Ausschreibungsverfahren aus dem Verhandlungsszenario “Order Driven Production”.

Ist die Machbarkeit gegeben wird der Prozess der Angebotserstellung angestoßen. Es ist eine automatisierte Bewertung des angefragten Produktes nach Herstellkosten notwendig. Zusätzlich zu den Herstellkosten werden in der Regel verschiedene Gemeinkostenzuschläge berücksichtigt, z.B. für Entwicklung oder Verwaltung. Die zu berücksichtigende Marge kann von verschiedenen Faktoren abhängen, z.B. vom anfragenden Kunden, vom angefragten Volumen, von der aktuellen Auslastung der Produktionsressourcen, etc. Für die technische Umsetzung der Angebotserstellung sind zwei Strategien denkbar:

1. Angebotserstellung komplett innerhalb der VWS-Infrastruktur mit einem Preisfindungsalgorithmus, der innerhalb der verhandelnden VWS implementiert ist und Daten verwendet, die ausschließlich über VWS von Produkt, verwendeten Komponenten und verwendeten Ressourcen bereitgestellt werden
2. Angebotserstellung außerhalb der VWS-Infrastruktur, delegiert an ein System (z.B. ERP oder Vertriebssystem) dass definierte Daten über API-Aufruf erhält (Stückliste, Prozessliste) und als Rückgabewert den Angebotspreis und Liefertermin zurückgibt.

Auf Auftraggeberseite müssen mehrere eingegangene Angebote miteinander verglichen werden. Dazu ist zunächst einmal ein Speicher für die eingegangenen Angebote erforderlich. Aus Gründen der Nachvollziehbarkeit ist eine Anforderung denkbar, die eingegangenen Angebote auch nach der Entscheidung aufzubewahren. Es soll versucht werden, die Speicherung der Angebotsdaten über eine entsprechende Teilmodell der VWS abzubilden.

7.1.1.2.3 Erwartete Ergebnisse

Im Verhandlungsszenario 2 werden die Nachrichten zwischen Typ-3 VWS über Unternehmensgrenzen hinweg ausgetauscht. Es ist zu prüfen, ob das Dataspace Konzept (Catena X / Tractus X) zu dem gewählten technischen Weg für den Nachrichtenaustausch kompatibel ist.

In diesem Zusammenhang sollen Anforderungen an die VWS-Infrastruktur in Bezug auf Sicherheit, Datenschutz sowie Benutzer- und Rechteverwaltung (als Input für TP 8) ermittelt werden.

Die für Verhandlungsszenario 1 definierten Interaktionsprotokolle und Spezifikationen zum Nachrichtenaufbau sollen für das unternehmensübergreifende Szenario erweitert werden.

Für die automatisierte Erstellung eines Angebots im kaufmännischen Sinne wird in der Praxis die Integration eines ERP-System notwendig sein, in dem z.B. Einkaufspreise für die verwendeten Rohmaterialen oder Kalkulationsschemata für die Ermittlung einer Preisuntergrenze abgebildet sind. Hierzu soll exemplarisch aufgezeigt werden, wie eine Integration von weiteren, nicht AAS-basierten Systemen darstellbar ist.

7.1.1.2.4 Bezug zu VWS4LS

Mit unternehmensübergreifenden automatisierten Verhandlungsprozessen wird insbesondere die Automatisierung des Änderungsmanagements entlang der gesamten Wertkette unterstützt, da die Entscheidungen über Machbarkeit und Preisfindung sowie die Prozesse zur Beauftragung automatisiert werden.

In [TP 8](#) werden die Grundlagen für einen sicheren Datenaustausch über Unternehmensgrenzen hinweg erarbeitet. Die dort erarbeiteten Grundlagen zur Nutzung von International Dataspaces (IDS) zum verteilten Zugriff auf Verwaltungsschalen werden als Voraussetzungen für den Zugriff auf Produktdaten und geforderte Fähigkeiten im Rahmen des Ausschreibungsverfahren angesehen. Konzepte wie Unternehmensverzeichnisse ermöglichen das Auffinden von möglichen Auftragnehmern.

7.1.1.3 Szenario „Synchronisation von Werten“

Im TP 2 wurde im Rahmen des Konzeptes “Single Point of Truth” eine Verlinkung von verschiedenen VWS zum selben Typ einer Leitungssatzkomponente definiert. Die verschiedenen VWS bilden die Sichten der verschiedenen Partner in der Wertschöpfungskette des Leitungssatzes auf dasselbe Asset, also dieselbe Komponente, ab. In dem Zusammenhang wurde auch eine technische Lösung zur Synchronisierung von Änderungen an den Teilmodellen und Teilelementen zwischen den verlinkten Verwaltungsschalen gefordert.

Ziel ist es, für diesen Synchronisierungsvorgang ein semantisches Protokoll zwischen zwei Typ 3 VWS zu entwickeln, angelehnt an das Ausschreibungsverfahren. Dabei initiiert die VWS, in der die Wertänderung vorgenommen wurde, den Dialog zu den verlinkten VWS, indem eine Nachricht mit der Wertänderung gesendet wird. Zum Senden wird die in den vorhergehenden Szenarien eingeführte Operation für I4.0-Nachrichten verwendet.

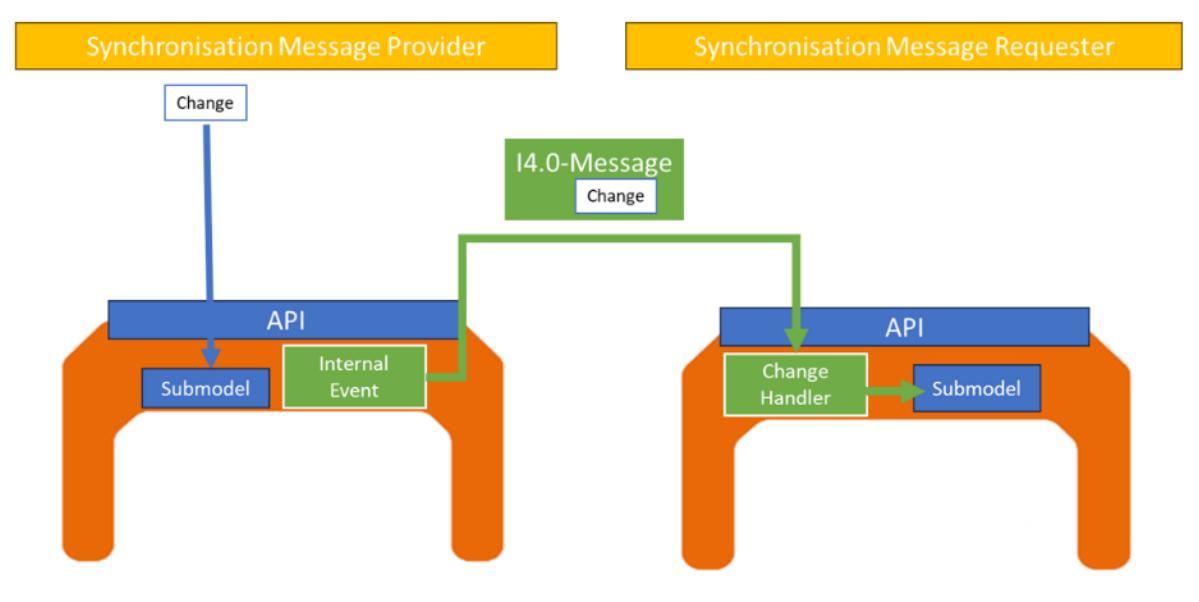


Abbildung 7-2: Nutzung der I4.0-Nachricht für Synchronisation von Änderungen an Teilmodellen

Die empfangende VWS führt auf ihrer Seite eine Machbarkeits- oder Plausibilitätsprüfung bzgl. des neuen Zielwertes durch (siehe Komponente „Change Handler“ in Abbildung 7-2). Gegebenenfalls kann vor Übernahme der Wertänderung auf der Empfängerseite ein Freigabeprozess integriert werden. Erst nach erfolgreicher Machbarkeitsprüfung und ggf. erfolgter Freigabe wird die Wertänderung in der empfangenden VWS umgesetzt und es erfolgt eine Rückmeldung an die VWS, welche die Wertänderung propagiert hat. Auch im Falle, dass die Änderung in der empfangenden VWS nicht umgesetzt wird, erfolgt eine entsprechende Rückmeldung. So kann auch die aufrufende VWS eine Übersicht über den Synchronisierungszustand führen.

7.1.1.4 Szenario „Änderungsmanagement“

Die in den beiden Verhandlungsszenarien zum Ausschreibungsverfahren und zur Synchronisierung erarbeiteten Mechanismen sollen verwendet werden, um die Informationsverteilung und Abstimmungsprozesse in der Erarbeitung und in der Umsetzung einer technischen Änderung am Leitungssatz zu unterstützen. Gegebenenfalls werden die Spezifikationen zu den semantischen Protokollen und zu den Nachrichten anhand dieses Szenarios noch erweitert oder verfeinert.

7.1.2 Technische Umsetzung der Verhandlungskonzepte

Um die Interoperabilität von I4.0-Komponenten zu realisieren, ist auch die Technologie für die Kommunikation zwischen den VWS des Typs 3 zu spezifizieren.

Das Diskussionspapier der Plattform Industrie 4.0 “Verwaltungsschale in der Praxis” [38] benennt als mögliche Technologien für die Interaktion zwischen aktiven Verwaltungsschalen OPC UA sowie die Protokolle MQTT, AMQP und HTTP. Als einfachste Form der Interaktion wird der API-Zugriff einer proaktiven VWS (Typ 3) auf eine reaktive VWS (Typ 2) beschrieben.

Da für den Nachrichtenaustausch im Rahmen des semantischen Protokolls "Ausschreibungsverfahren" weder in OPC UA eine Spezifikation vorliegt noch eine auf den anderen genannten technischen Protokollen eine Spezifikation bekannt ist, wird vorgeschlagen, die [REST-API der VWS](#)⁴⁸ auch für die horizontale Kommunikation zwischen zwei VWS vom Typ 3 zu nutzen (siehe Abbildung 7-3).

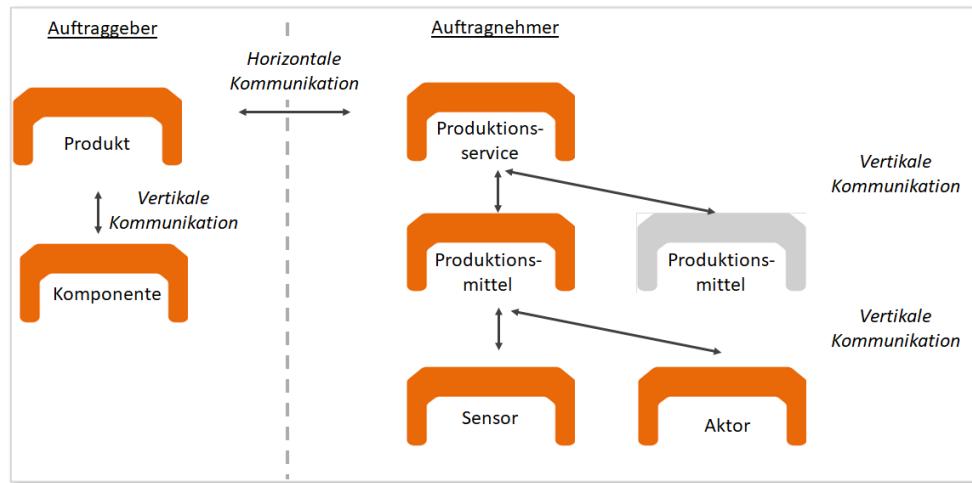


Abbildung 7-3: Horizontale Kommunikation zwischen zwei Verwaltungsschalen vom Typ 3

[IDTA 01002-3 Verwaltungsschale im Detail – Teil 2](#) [48] beschreibt das Konzept einer Operation, die über das Interface "Submodel" mit den Methoden "[InvokeOperationSync](#)" oder "[InvokeOperationAsync](#)" aufgerufen werden kann.

In der REST-API wird für den Aufruf der "[invoke](#)"-Endpunkt zur Verfügung gestellt und zum Abholen der Ergebnisse eines asynchronen Aufrufs der Endpunkt "[operation-results](#)".

Das in AP 6.1 erarbeitete Konzept sieht vor, für die Fähigkeit zur Teilnahme am semantischen Protokoll "Ausschreibungsverfahren" in der Rolle "Auftraggeber" oder "Auftragnehmer" ein Teilmodell zu erstellen, welches eine Operation vorsieht, mit der die Nachrichten des Interaktionsprotokolls ausgetauscht werden können.

Die Operation erhält eine Implementierung in der VWS-Middleware, welche die Nachricht an eine interne Workflowengine weiterleitet, welche dann die Verarbeitung der Nachricht und daraus abgeleitete Reaktionen orchestriert.

⁴⁸<https://github.com/admin-shell-io/aas-specs-api>



Abbildung 7-4: Ausschreibungsverfahren zwischen Auftraggeber und Auftragnehmer

Während der Verhandlung laufen sowohl auf Seite des Auftraggebers als auch auf Seite der Auftragnehmer Entscheidungsalgorithmen ab. So ist z.B. der Inhalt der Ausschreibung zu analysieren, eine technische Machbarkeit zur Umsetzung der Ausschreibung zu prüfen, ein Angebot zu erstellen, die eingegangenen Angebote zu analysieren und eine Entscheidung zur Angebotsannahme bzw. -ablehnung zu treffen. Statt diese Algorithmen in Form von Programmcode in der VWS zu hinterlegen oder in einem VWS-Repository zu implementieren, wurde im [TP 6](#) entschieden, einen sog. „[Low-Code](#)“-Ansatz umzusetzen. Dazu ist geplant in AP 6.2 einen Demonstrator aufzubauen, der ein VWS-Repository mit einer BPMN-Workflowengine kombiniert. Die Entscheidungsalgorithmen werden in BPMN modelliert und innerhalb der Workflowengine ausgeführt. In [Abbildung 7-5](#) ist eine Beispielimplementierung für einen Algorithmus auf der Auftragnehmerseite des Ausschreibungsverfahrens dargestellt. Aus dem Workflow heraus können Teilmodelle bzw. -elemente der VWS abgefragt werden. Operationen, die auf die VWS aufgerufen werden, sowie Wertänderungen in der VWS können Ereignisse in der Workflowengine auslösen.

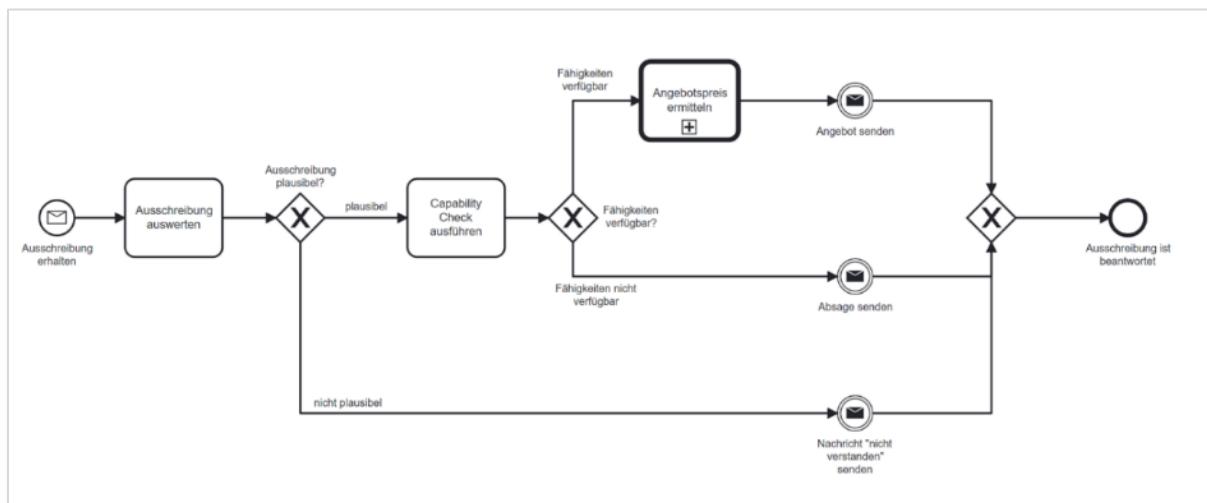


Abbildung 7-5: Beispielimplementierung für einen Algorithmus

7.1.3 Proof-of-Concept

Ein erster Proof-of-Concept wurde mit der Open Source Workflow-Engine “[Flowable](#)”⁴⁹ unter Verwendung des [BaSyx-SDK](#)⁵⁰ umgesetzt. Die Verwendung von BPMN zur Implementierung der Typ 3 VWS sollte dabei transparent für die anderen Systeme sein, die mit der VWS interagieren. Dies wird dadurch erreicht, indem die Interaktion ausschließlich mit der API der VWS erfolgt und niemals direkt mit der API der Workflow Engine.

Durch den Einsatz der Low-Code Technologie soll eine einfache Anpassbarkeit des Verhaltensmodells der Typ 3 VWS über den gesamten Lebenszyklus des Assets hinweg gewährleistet werden. Die BPMN mit ihrer graphischen Darstellung liefert den Mehrwert einer Übersicht des Verhaltensmodells und ist insbesondere auch geeignet den aktuellen Zustand eines laufenden Verhandlungsprozesses für den Menschen zu visualisieren. Dies hat Relevanz für die Untersuchungen in Rahmen des Forschungsprojektes und es wird erwartet, dass auch im praktischen Einsatz Monitoring- und Analyseaufgaben anfallen werden, bei denen die graphische Notation unterstützt.

⁴⁹ <https://www.flowable.com/open-source/docs/>

⁵⁰ <https://github.com/eclipse-basyx/basyx-java-server-sdk>

7.2 AP 6.2 - Verhandlungsszenarien und -strategien

Im AP 6.2 „**Verhandlungsszenarien und -strategien**“ wurde im Wesentlichen an der technischen Umsetzung der in AP 6.1 erarbeiteten Verhandlungskonzepte und deren Erprobung an Demonstratoren gearbeitet. Ziel des Arbeitspaktes war es ursprünglich Strategien und Regeln zu definieren, die zum einen automatisierten Verhandlungsprozess verschiedener VWS zulassen und zum anderen den Entscheidungsverlauf nachvollziehbar machen. Dies wurde im Arbeitspaket erreicht, indem die Abläufe in Form von BPMN-Modellen formalisiert und in einer BPMN-Engine ausführbar gemacht wurden. Die Bearbeitung der in AP 6.1 definierten Verhandlungsszenarien zu Synchronisation und Änderungsmanagement erfolgte hauptsächlich im Rahmen des Architekturteams und die Ergebnisse sind entsprechend auch im Kapitel „Architektur AAS“ dargestellt.

7.2.1 Start des Verhandlungsprozesses

Um den Ablauf eines Verhandlungsprozesses starten zu können, wurde in der VWS des *Service Requesters* eine Operation „*startBiddingProcess*“ zur Verfügung gestellt. Diese Operation kann zum Testen und Simulieren manuell über die Weboberfläche aufgerufen werden. In der finalen Version des Demonstrators erfolgt der Aufruf jedoch automatisiert über entsprechende Geschäftslogik in der VWS-Middleware. Realisiert wurde im finalen Demonstrator das Verhandlungsszenario „*Order Driven Production*“. Wenn der Produktionsauftrag (Service Requester) einen bestimmten Status erreicht, wird die Operation „*startBiddingProcess*“ ausgelöst. Daraufhin schickt der *Service Requester* eine Ausschreibungsnachricht betreffenden der Herstellung des entsprechenden Leitungssatzes bzw. Halbfabrikats. Ziel dieser Ausschreibung ist es von allen Produktionsmaschinen, die das angefragte Produkt herstellen können, Angebote zu bekommen und nach vorgegebenem Optimierungskriterium die Ziemaschine auszuwählen. Beim manuellen Aufruf wird das Optimierungskriterium in der Weboberfläche übergeben.

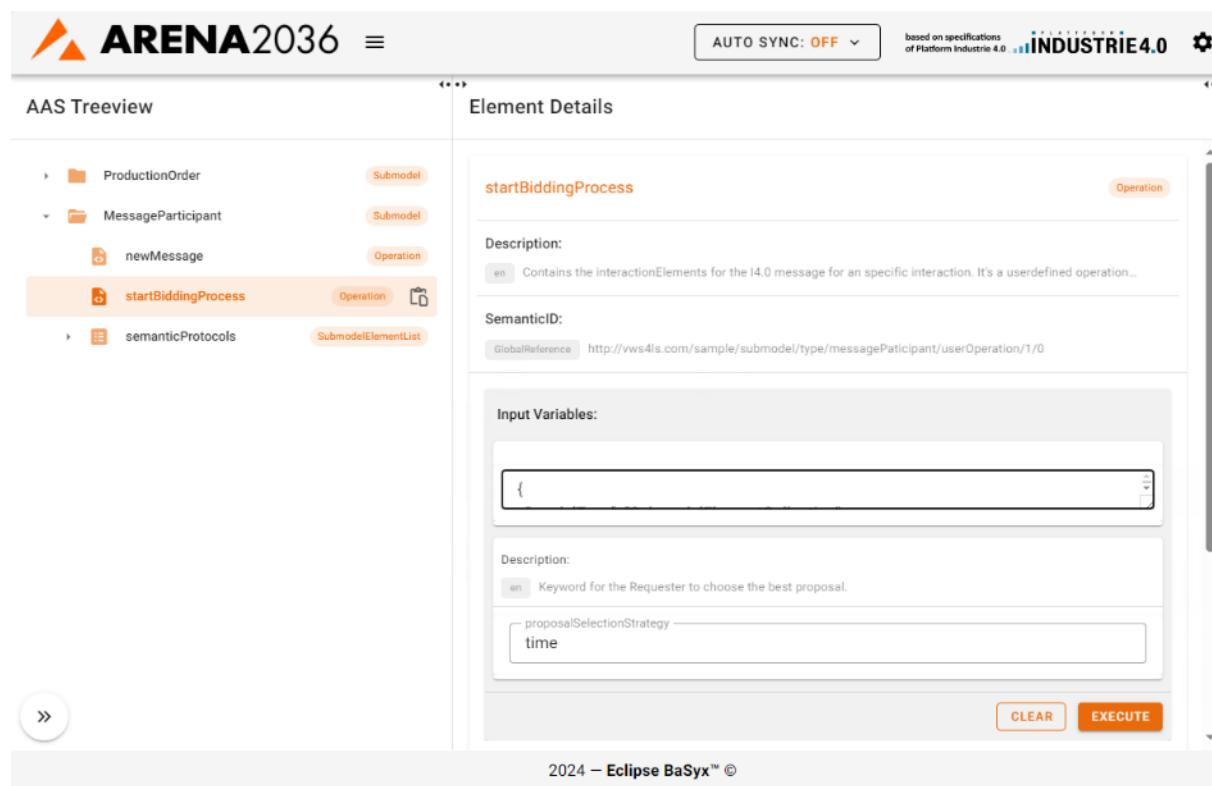


Abbildung 7-6: Start des Verhandlungsprozesses über die AAS-Weboberfläche

Für die Geschäftslogik der Ausschreibungserstellung, Senden der Ausschreibung sowie Entgegennehmen und Auswerten der eingegangenen Angebote wurde in Flowable ein Workflowmodell „*Call for proposal*“ erstellt. Die grundlegende Abfolge der Aktionen und Entscheidungen wurde mit dem in der Flowable-Weboberfläche enthaltenen BPMN-Modellierungswerkzeug definiert. Komplexere Aufgaben wie das Erstellen der I4.0-Nachrichten oder der Zugriff auf VWS über die REST-API wurden in eigenen

Java-Klassen implementiert. Die Java-Klassen müssen als JAR-Datei in einem der Installationsverzeichnisse des Flowable-Servers kopiert werden und beim Start des Flowable-Services zur Verfügung stehen. Das Flowable Framework stellt dazu ein Interface "JavaDelegate" zur Verfügung (siehe <https://www.flowable.com/open-source/docs/bpmn/ch07b-BPMN-Constructs#java-service-task>), von dem abgeleitet werden kann, um eigene Programmlogik zu implementieren. Im BPMN-Modellierungswerkzeug wird dann eine Aufgabe vom Typ "JavaServiceTask" eingefügt, bei der man den Namen der Klasse angeben kann, deren Programmlogik an dieser Stelle ausgeführt werden soll.

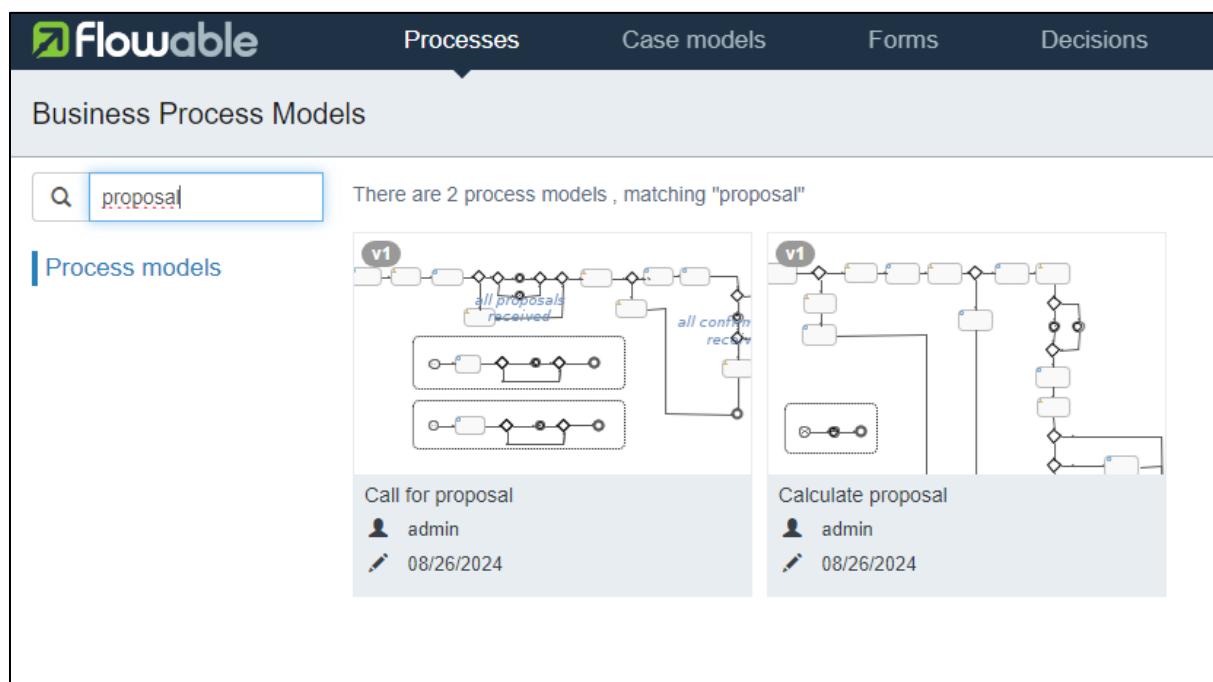


Abbildung 7-7: [Prozessmodelle](#) in der Flowable-Weboberfläche

Für die Geschäftslogik der Ausschreibungserstellung, Senden der Ausschreibung sowie Entgegennehmen und Auswerten der eingegangenen Angebote wurde in Flowable ein Workflowmodell "Call for proposal" erstellt. Die grundlegende Abfolge der Aktionen und Entscheidungen wurde mit dem in der Flowable-Weboberfläche enthaltenen BPMN-Modellierungswerkzeug definiert. Komplexere Aufgaben, wie das Erstellen der I4.0-Nachrichten oder der Zugriff auf VWS über die [REST-API](#) wurden in eigenen Java-Klassen implementiert. Die Java-Klassen müssen als JAR-Datei in einem der Installationsverzeichnisse des Flowable-Servers kopiert werden und beim Start des Flowable-Services zur Verfügung stehen. Das Flowable Framework stellt dazu ein Interface "JavaDelegate" (siehe <https://www.flowable.com/open-source/docs/bpmn/ch07b-BPMN-Constructs#java-service-task>) zur Verfügung, von dem abgeleitet werden kann, um eigene Programmlogik zu implementieren. Im BPMN-Modellierungswerkzeug wird dann eine Aufgabe vom Typ "JavaServiceTask" eingefügt bei der man den Namen der Klasse angegeben, deren Programmlogik an dieser Stelle ausgeführt werden soll.

Aufgabe der VWS-Operation "startBiddingProcess" ist es nun, in Flowable eine Instanz von diesem Workflowmodell "Call for proposal" zu starten. Flowable bietet hierzu eine REST-API an, um mit der Workflowengine zu interagieren.

Starten einer Workflowinstanz über die [Flowable REST-API](#):

```
POST runtime/process-instances
```

Request body:

```
{
  "processDefinitionKey": "callForProposal",
  "returnVariables": false,
  "tenantId": "tenant1",
  "variables": [
    {
      "name": "proposalCount",
      "value": 10
    }
  ]
}
```

```

        "name":"proposalSelectionStrategy",
        "value":"time"
    },
    {
        "name":"reqAasId",
        "value":"https://www.arena2036.de/ids/aas/5214_6011_7042_0566"
    },
    [
}
}

```

Im ersten Proof-of-Concept in AP 6.1 wurde das Feature “*Operation Delegation*” genutzt der BaSyx Version 1 Middleware genutzt, um die Operation zu implementieren. Dieses Feature erlaubt es, den Operationsaufruf an einen HTTP-Endpunkt weiterzuleiten. Für dieses Projekt wurde das Feature auch für die BaSyx Version 2 implementiert. Um die Weiterleitung in BaSyx zu konfigurieren, wird der Operation in der VWS ein Qualifier hinzugefügt mit dem Typ “*invocationDelegation*”. Der Wert des Qualifiers ist die URL, an die weitergeleitet werden soll.

Um den oben gezeigten Request-Body zum Erzeugen einer Workflowinstanz in Flowable zur erstellen, erfolgt die Weiterleitung aus BaSyx zunächst an einen HTTP-Endpunkt in NodeRed. Die Aufgabe dieses NodRed-Flows “*startBiddingProcess*” ist es in einer Javascript-Funktion den Request-Body für den Flowable-Endpunkt “*runtime/process-instances*” aufzubereiten und an diesen weiterzuleiten. Im Demonstrator hat der NodeRed-Flow darüber hinaus mit den zusätzlichen grün markierten Debug-Knoten die Aufgabe, den Ablauf durch zusätzliche Ausgaben nachvollziehbar zu machen.

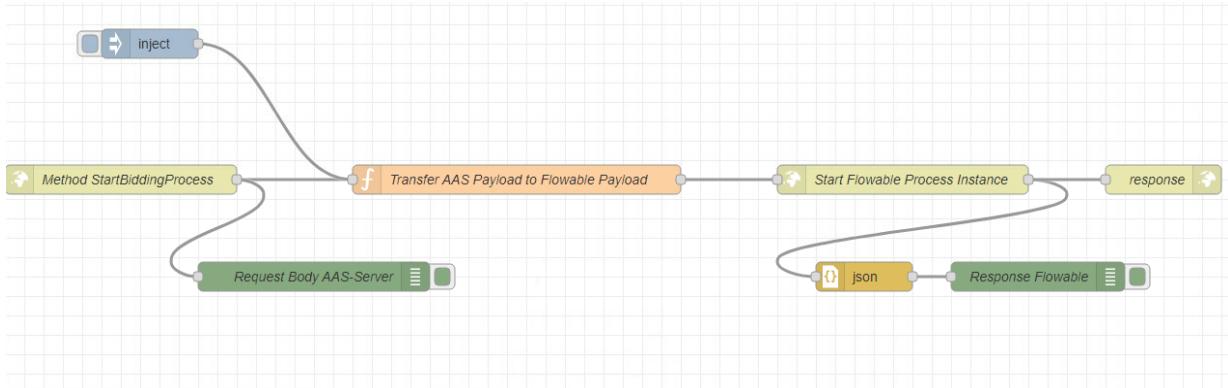


Abbildung 7-8: NodeRed-Flow “[startBiddingProcess](#)”

In dem Aufruf an Flowable werden die Strategie für die Auswahl des Angebots und die AAS-Id der Verwaltungsschale, die die Rolle des *Service Requester* einnimmt, übergeben. Prozessinstanzvariablen dienen der Steuerung der Logik innerhalb des Workflows und können auch von den Java Service Tasks gelesen und geschrieben werden. Die Übergabe der AAS-Id des *Service Requester* ist notwendig, da dieselben Instanzen von NodeRed und Flowable für verschiedene *Service Requester* eingesetzt werden. Somit ist eine Zuordnung der Workflowinstanz zur jeweiligen Verwaltungsschale möglich und die AAS-Id wird als Sender in den I4.0-Nachrichten benötigt.

Um die AAS-Id von der BaSyx aus an den NodeRed-Flow zu übermitteln, wird sie als Query-Parameter in der URL der *OperationDelegation* hinterlegt.

01baaf91-710d-11ef-8810-0242c0a81008 [← Return to list](#)

| | | | |
|----------------------|--------------------------------------|----------------------------|---|
| ID: | 01baaf91-710d-11ef-8810-0242c0a81008 | Business key: | (None) |
| Name: | (None) | Description: | (None) |
| Status: | Completed | Process definition: | Call for proposal |
| Activity ID started: | startEvent | Started by: | admin |
| Tenant: | (None) | Super process instance ID: | |
| End time: | Sep 13, 2024 10:32 AM | Duration: | 0 years 0 days 18 hours 48 minutes 35 seconds |
| Delete reason: | (None) | Ended in activity: | approveEnd |

Tasks 4 Variables 29 Subprocesses 0 Jobs 0 Decisions 0 Form instances 4

This process instance has 29 variables.

| Name | Type | Value |
|---------------------------|---------|---|
| proposalCounter | integer | 0 |
| proposalSelectionStrategy | string | time |
| reqAasId | string | https://www.arena2036.de/ids/aas/5214_6011_70 |
| requestedAASID | null | |

Abbildung 7-9: Darstellung der Workflow-Instanz mit Variablen in Flowable

Die Workflowengine von Flowable generiert für jede Workflow-Instanz eine eindeutige Nummer als "Processinstance-ID". Diese Nummer wird im folgenden Verhandlungsprozess in allen I4.0-Nachrichten als Konversations-Id verwendet. Damit ist eine eindeutige Zuordnung der eingegangenen Nachricht zur richtigen Workflowinstanz und damit auch zur richtigen VWS möglich.

7.2.2 Ermittlung der möglichen Service Provider

Eine Fragestellung, die es grundsätzlich zu lösen gilt, ist welche Typ-3 VWS die Ausschreibungsnachricht erhalten sollen. Damit die VWS grundsätzlich an dem Verhandlungsprozess teilnehmen kann, muss das Teilmodell *MessageParticipant* mit der Operation *newMessage* implementieren.

Für den Demonstrator wird diese Fragestellung so beantwortet, dass alle in Frage kommenden VWS in einer zentralen AAS-Registry bzw. deren *MessageParticipant*-Teilmodelle in einer zentralen Submodel Registry registriert sind. Im Teilmodell *MessageParticipant* ist in einer Submodel Element List hinterlegt welche Interaktionsprotokolle und welche Rollen im Interaktionsprotokoll die jeweilige VWS unterstützt. Gesucht wird nach VWS die das Interaktionsprotokoll VDI/VDE 2193-2 [46] in der Rolle "ServiceProvider" unterstützen. Jede VWS auf die das zutrifft erhält die Ausschreibungsnachricht, d.h. der Workflow des *Service Requester* ruft auf jede dieser VWS die Operation "newMessage" mit einer Ausschreibungsnachricht auf. Eine Filterung auf die Fähigkeiten des *Service Provider* erfolgt an dieser Stelle bewusst nicht. Die Entscheidung, ob der *Service Provider* über notwendige Fähigkeiten verfügt, sollte auch auf der Seite des *Service Provider* getroffen werden. Kann der *Service Provider* das angefragte Produkt nicht herstellen bzw. keinen der angefragten Prozesse ausführen, so wird gem. Interaktionsprotokoll eine Ablehnung der Ausschreibung erwartet.

7.2.3 Übergabe der I4.0-Nachrichten an BPMN-Workflows

Beim Durchlauf des Interaktionsprotokolls kommt es abwechselnd zu der Situation, dass einer der Verhandlungspartner auf die Antwortnachricht des anderen warten muss. Im Fall der TP6-Implementierung des Verhandlungsprozesses gibt es eine Workflowinstanz für den *Service Requester* und eine Workflowinstanz je *Service Provider*, in denen diese Wartezustände abgebildet werden müssen. In BPMN kann

dieser Wartezustand über das Notationselement vom Typ Ereignis modelliert werden. Im Demonstrator kommen dabei die Arten Zeitereignis, Nachrichtenereignis und Signalereignis zum Einsatz. Mit Ereignis wird ermöglicht, dass nach Ablauf eines vordefinierten Zeitraums die Bearbeitung der eingegangenen Angebote erfolgt, auch wenn noch nicht alle Service Provider auf die Ausschreibung geantwortet haben. Sind jedoch alle Angebote eingegangen, so soll nicht auf den Ablauf des Zeitintervalls gewertet werden, sondern unmittelbar mit dem weiteren Ablauf fortgefahrene werden. Dies wird durch zwei parallele Ereignisse erreicht, die durch ein XOR-Gateway zusammengeführt werden.

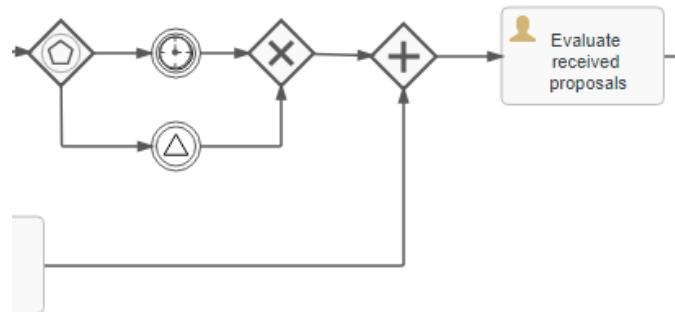


Abbildung 7-10: Zeit- und Signalereignis im [Workflow des Service Requesters](#)

Das Signal, dass alle erwarteten Antworten eingegangen sind, wird dabei aus einem Unterworkflow heraus ausgelöst wird, welcher für jedes eingegangene Angebot durchlaufen wird.

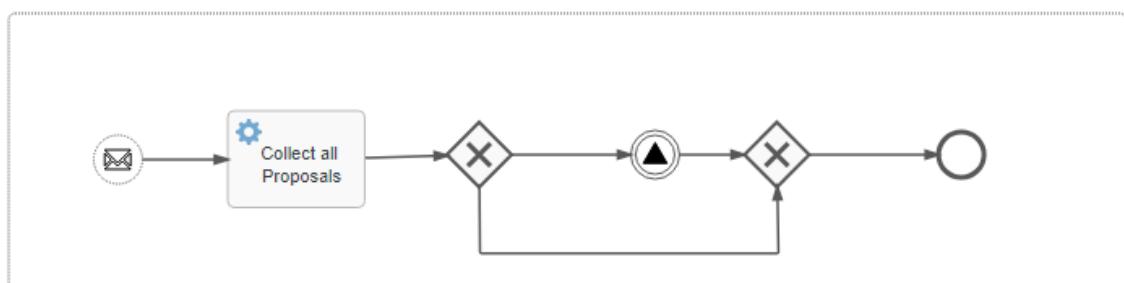


Abbildung 7-11: [Unter-Workflow](#) für eingehende Angebots-Nachricht

Um von außen ein Ereignis in der Flowable Workflowengine zu generieren, gibt es einen Rest-Endpunkt, der für eine "Execution" aufgerufen werden kann. Eine "Execution" ist ein Bestandteil einer Workflowinstanz und bei parallelen Ausführungspfaden innerhalb einer Workflowinstanz werden ggf. mehrere "Executions" angelegt.

Erzeugen eines Nachrichtenereignisses in Flowable über die REST-API (<https://www.flowable.com/open-source/docs/bpmn/ch14-REST#execute-an-action-on-an-execution>):

PUT runtime/executions/{executionId}

Request body:

```
{
  "action": "messageEventReceived",
  "messageName": "proposalReceived",
  "variables": [...]
}
```

Der Rest-Endpunkt `POST query/executions` in Flowable ermöglicht die Ermittlung der korrekten Execution-Id. Als Query-Parameter dienen der Name des relevanten Prozessmodells, z.B. CallForProposal, und die Prozessinstanz-Id, die als Konversations-Id im Rahmen der I4.0-Nachrichten - wie oben beschrieben - mitgesendet wird.

Das Element `messageName` im Request Body identifiziert das jeweilige Nachrichtenereignis im BPMN-Modell, das angesprochen werden soll und wird aus dem Typ der I4.0-Nachricht abgeleitet. Im Element `variables` werden die Nachrichteninhalte transportiert.

Die Implementierung der Operation `newMessage`, welche die in I4.0-Nachricht in der VWS entgegennimmt wurde zunächst wieder durch eine Weiterleitung an einen NodeRed-Flow implementiert. In NodeRed erfolgt zunächst eine Anfrage an Flowable mit der Konversations-Id, um die korrespondierende Execution-Id zu ermitteln, und es wird das Nachrichtenereignis in Flowable generiert. Ein Sonderfall stellt der Eingang einer Ausschreibungsnachricht auf der Seite des Service Providers dar, denn in diesem Fall muss erst eine Workflowinstanz vom entsprechenden Workflow-Modell "Calculate Proposal" gestartet werden. In dem Workflowmodell "Calculate Proposal" erfolgt dann die Überprüfung der Machbarkeit, d.h. die Prüfung, ob der Service Provider über die geforderten Fähigkeiten verfügt, sowie die notwendige Bewertung des ausgeschriebenen Produktes. Die Details hierzu sind im folgenden AP 6.3 beschrieben.

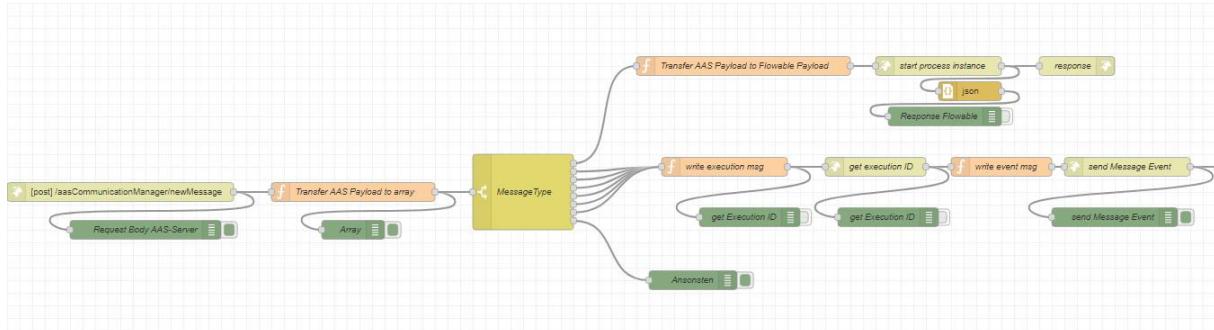


Abbildung 7-12: Mapping von I4.0-Nachricht zu BPMN-Ereignissen in [NodeRed](#)

7.3 AP 6.3 - Capabilities

7.3.1 Anforderungen an den Fähigkeitenabgleich in automatisierten Verhandlungsprozessen

Das AP 6.3 „**Capabilities**“ wurde ursprünglich definiert, um die ein Konzept für einen automatisierten Abgleich der verfügbaren Ressourcen mit den geforderten Fähigkeiten im Rahmen des Verhandlungsprozesses zu entwickeln. Im Kontext TP 6 ist dies z.B. notwendig, um einen Produktionsauftrag auf die richtige Maschine zuzuweisen oder um im Rahmen eines Ausschreibungsverfahrens zu entscheiden, ob für eine Ausschreibung ein Angebot abgegeben werden kann oder nicht.

In TP5 wurde ein generischer Fähigkeitenabgleich entwickelt, der es erlaubt, aus Sicht eines einzelnen Produktionsprozesses diesen Abgleich durchzuführen. Dabei werden in der VWS des Produktes modellierte, geforderte Fähigkeiten mit den in der VWS der Produktionsressource modellierteren verfügbaren Fähigkeiten abgeglichen. Der entwickelte Algorithmus liefert eine Aussage, ob für den angefragten Prozess alle Fähigkeiten abgedeckt sind und im Sonderfall des Crimpverfahrens darüber hinaus eine zusätzliche Information, ob für das jeweilige zu crimpende Kontaktteil ein geeignetes Crimpwerkzeug verfügbar bzw. in der Maschine montiert ist. Das in AP 6.3 entwickelte Konzept basiert direkt auf diesem Algorithmus und den dazugehörigen Teilmustern.

Zunächst wurde in AP 6.3 analysiert welche Anforderungen an einen Fähigkeitenabgleich, die sich aus den automatisierten Verhandlungsprozessen ergeben, durch den oben beschriebenen Algorithmus nicht abgedeckt sind. Die in AP 6.1 und AP 6.2 beschriebenen Verhandlungsszenarien umfassen nicht nur die Ausführung eines einzelnen Produktionsprozesses, sondern umfassen die Herstellung eines komplexen Halbfabrikats zu dessen Herstellung mehrere Produktionsprozesse nacheinander, in einer bestimmten Reihenfolge auszuführen sind. Es ist also notwendig, den Algorithmus aus TP5 für jeden einzelnen Produktionsprozess auszuführen und die Einzelergebnisse zu einem Gesamtergebnis zusammenzuführen.

Ein entworfenes Verhandlungsszenario sieht vor, dass von Auftraggeberseite das gesamte Produkt mit allen notwendigen Prozessen ausgeschrieben wird, aber von Auftragnehmerseite explizit Angebote erwartet werden, die nur einen Teilumfang der Prozesse beinhalten. Der Auftraggeber, kann dann Teilaufträge an verschiedene Auftragnehmer vergeben. Die Teilaufträge aneinandergekettet ergeben dann zusammen den kompletten Produktionsprozess. Aus Sicht des Fähigkeitenabgleichs bedeutet das, dass es nicht ausreichend ist, alle geforderten Produktionsprozesse zu prüfen und sobald ein Prozess nicht erfüllt werden kann, ein negatives Ergebnis zurückzumelden. Es ist vielmehr erforderlich, mögliche Teilumfänge aus der Gesamtprozesskette zu ermitteln, mit der Angabe eines Startprozesses, evtl. Zwischenschritten und einem Endprozess. Es wurde entschieden, die Datenstruktur des Angebots so zu gestalten, dass ein Auftraggeber mehrere Teilumfänge in einer Angebotsnachricht anbieten kann.

Stehen diese Möglichkeiten zur Verfügung, kann auf Auftragnehmerseite entschieden werden, ob ein Angebot abgeben werden kann oder ob die Ausschreibung abzulehnen ist. Es kann auch der Umfang festgelegt werden, also welche Teilaufträge angeboten werden. Zur Erstellung des Angebotes fehlt jetzt noch die quantitative Bewertung der Teilaufträge. Je nach Kontext ist eine Ermittlung der notwendigen Fertigungszeit, der Herstellkosten, eines Angebotspreises oder der zu berücksichtigenden CO2-Emissionen notwendig.

Um im Verhandlungsprozess optimale Entscheidungen treffen zu können und wettbewerbsfähige Angebote erstellen zu können, ist es wichtig, dass diese Bewertungsfunktion Ergebnisse liefern kann, die möglichst nahe an der Realität liegen. Dabei sollen sowohl individuelle Eigenschaften des angefragten Produktes als auch die konkreten Gegebenheiten bzgl. der vom Auftragnehmer eingesetzten Ressourcen und der Produktionsumgebung berücksichtigt werden.

Die Berechnungsfunktion kann am besten vom Maschinenhersteller, der alle internen Details der Ressource kennt, erstellt werden. Dabei könnten durch die verwendete Berechnungsformeln oder einzelne Parameterwerte Rückschlüsse auf interne Details der Maschine möglich werden, die zum schützenswerten geistigen Eigentum des Maschinenherstellers zählen. Deshalb sollte es möglich sein, die Bewertungsfunktionen zur Verfügung zu stellen, ohne diese offen zu legen.

Auf der anderen Seite sollten die Bewertungsfunktionen vom Maschinenbetreiber angepasst oder übersteuert werden können. In Abhängigkeit von der individuellen Produktionsumgebung beim Maschinenbetreiber können sich Abweichungen ergeben, die aber im Verhandlungsprozess berücksichtigt werden sollen.

Bestimmte Parameter, die bei den Bewertungsfunktionen berücksichtigt werden müssen, wie z.B. Materialeinkaufpreise oder Maschinenstundensätze sind heute in klassischen ERP- oder MES-System abgelegt. Deshalb ist es sinnvoll, wenn diese Systeme direkt in die Bewertungsfunktion einbezogen werden können.

7.3.2 Implementierung des erweiterten Fähigkeitenabgleichs

Der in TP 5 entwickelte Algorithmus arbeitet generisch aufgrund der Informationen, die in standardisierten Teilmodellen abgelegt sind. Der Vorteil ist, dass dieser Algorithmus direkt in Verwaltungsschalen-Middleware implementiert werden kann und dann für alle Verwaltungsschalen, die in der Middleware gehostet werden, zur Verfügung steht.

Die oben beschriebenen Anforderungen lassen sich jedoch mit diesem generischen Ansatz nicht oder nur schwer umsetzen. Deshalb wurde vorgeschlagen, den erweiterten Fähigkeitenabgleich individuell je Asset auf Instanzebene zu implementieren. Um Interoperabilität zu gewährleisten, wird der Fähigkeitenabgleich als Verwaltungsschalen-Operation zur Verfügung gestellt. Der Algorithmus, der in der Auftragnehmer-Verwaltungsschale die Angebotserstellung implementiert, ruft also wiederum eine Operation der eigenen Verwaltungsschale auf, um den erweiterten Fähigkeitenabgleich mit Bewertungsfunktion auszuführen.

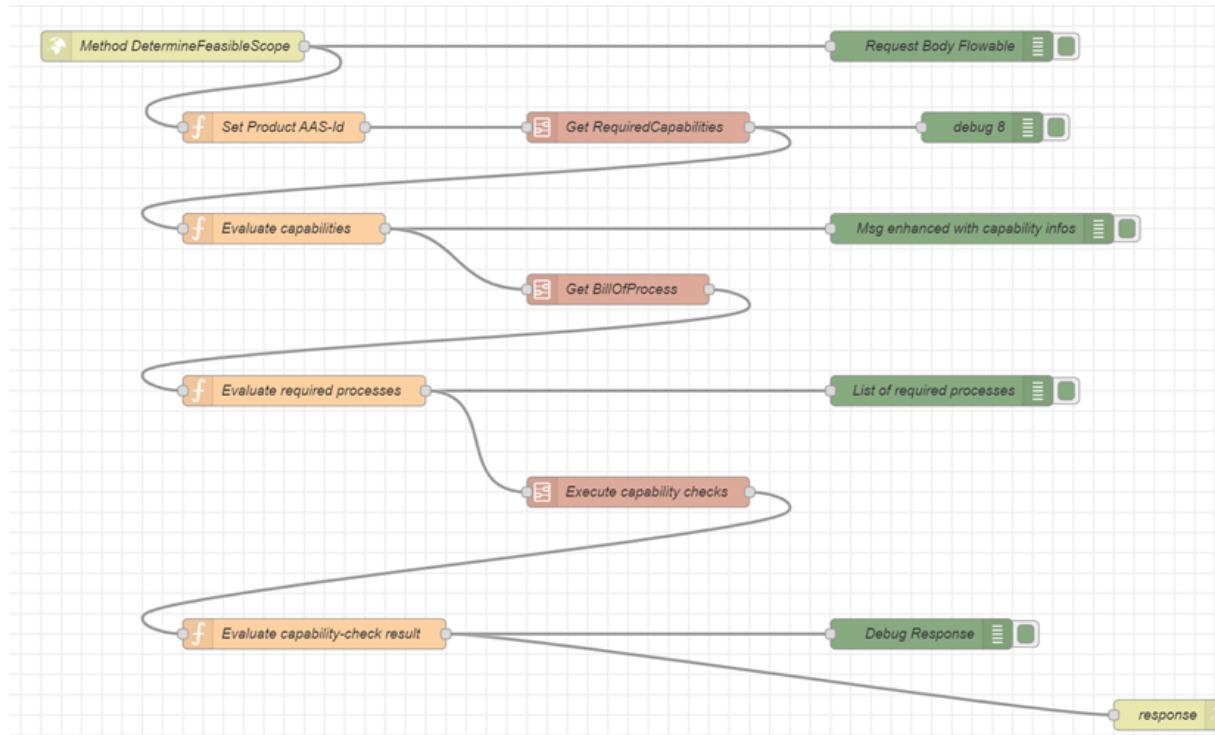


Abbildung 7-13: Erweiterter Fähigkeitenabgleich

Für die Entwicklung des Demonstrators wurde die Operation „*determineFeasibleScope*“ in der Verwaltungsschale angelegt (siehe Abbildung 7-13) und mit dem Feature der Basyx Middleware an einen HTTP-REST Endpunkt in NodeRED weitergeleitet.

Die Funktion wird im Verhandlungsprozess aus einem komplexeren Workflow zur Angebotserstellung heraus aufgerufen, der in der BPMN-Engine Flowable umgesetzt wurde. Da die Workflow-Engine ein Implementierungsdetail der Service Provider Verwaltungsschalen ist, genauso wie die Implementierung der Funktion *determineFeasibleScope*, wurde entschieden den Aufruf aus dem BPMN-Workflow heraus direkt als HTTP-Aufruf des NodeRed-Endpunktes umzusetzen. Die Nutzung der VWS-Operation als Schicht, die Interoperabilität gewährleistet, ist nicht notwendig, da der Bewegungsrahmen hier innerhalb

der Systemgrenzen einer einzelnen VWS-Implementierung ist. Der direkte HTTP-Aufruf hat den Vorteil, dass das Ergebnis direkt als Antwort des HTTP-Aufrufs gesendet werden kann.

Die Rückgabe einer Nachricht im HTTP-Response ist im Basyx-Feature “*OperationDelegation*” derzeit nicht vorgesehen.

Im Demonstrator ist ein NodeRed-Endpunkt für alle "Service Provider"-Verwaltungsschalen zuständig. Deshalb wird als Übergabeparameter neben der AAS-Id der Produkttyp-VWS auch die AAS-Id der Maschinen-VWS übergeben.

Von der Produkttyp-VWS werden die beiden Teilmodelle *RequiredCapabilities* und *BillOfProcess* geladen. Dann wird die Liste der Prozesse im Teilmodell *BillOfProcess* der Reihenfolge nach abgearbeitet, in dem je Prozess die relevanten geforderten Fähigkeiten aus dem Teilmodell *RequiredCapabilities* ermittelt werden und für diese wird dann der in AP 5.5. entwickelte Algorithmus zum Fähigkeitenabgleich aufgerufen (siehe TP5).

Aufgabe der Funktion “*determineFeasibleScope*” ist es, eine Liste mit Angeboten zu erstellen, die mögliche, d.h. machbaren Teilumfänge der gesamten Prozessliste des *BillOfProcess*-Teilmodells beinhalten. Dabei können sich bei einer Maschine, die mehrere Prozesse ausführen kann, mehrere Teilumfänge ergeben. Je Teilumfang wird angegeben, welches der Start- und End-Prozess ist, sowie welche Prozesse dazwischen im jeweiligen Angebot beinhaltet sind. Bei der Implementierung der Funktion für die Schneid-Crimp-Vollautomaten wird berücksichtigt, dass nur Leitungen bearbeitet werden können, die als ungeschnittene Leitung von der Leitungsspule zugeführt werden. Die machbaren Teilumfänge in den Angeboten müssen dementsprechend immer mit einem Cut-Prozess beginnen.

Die gesamte Ergebnisnachricht des Funktionsaufrufs besteht aus einer Liste von Angeboten mit einem Angebot je ermitteltem, machbaren Teilumfang. Jedes Angebot beinhaltet einen ermittelten Preis, ermittelte Ausführungsduer und einen Wert zum CO2-Fußabdruck (PCF, Carbon Footprint).

Im folgenden Kapitel wird die Ermittlung dieser Werte für das Angebot beschrieben.

7.3.3 Bewertung der Herstellungsprozesse aus Sicht der Produktionsressource

Das Konzept zur Implementierung der Operation “*determineFeasibleScope*” sieht vor, mit den Informationen der Produkttyp-VWS eine Bewertung aus Sicht jeweiligen Produktionsressource in Hinblick auf notwendige Produktionszeit sowie Herstellkosten und CO2-Fußabdruck durchzuführen.

Die Ermittlung der Herstellkosten und des CO2-Fußabdrucks basiert auf der Produktionszeit und wird mit den zugehörigen Faktoren bewertet. Zur Ermittlung der Materialkosten wird die Stückliste (BOM) des Produktes mit den Preisen der einzelnen Materialien bewertet. Es ist davon auszugehen, dass die entsprechenden Werte, wie Maschinenstundensätze oder Einkaufspreise, heute in ERP- und MES-Systemen zu finden sind und diese Systeme über entsprechende Programmierschnittstellen verfügen, um diese Daten abzufragen. In der Umsetzung des VWS4LS-Demonstrators könnte solche eine Abfrage z.B. mittels HTTP-REST erfolgen.

Mit voranschreitender Standardisierung ist zu erwarten, dass auch entsprechende Teilmodelle veröffentlicht werden, welche diese Informationen dann direkt über die VWS von Produktionsressource bzw. Verbundkomponente “Produkt” zugänglich machen.

Um die Machbarkeit dieses Konzept zu überprüfen, wurden im AP 6.3 für alle notwendigen Produktionsprozesse Formeln erarbeitet, die eine realistische Ermittlung der notwendigen Laufzeit an der Maschine erlauben. Die Parameter, die sich auf die Produkteigenschaften beziehen, ließen sich dabei alle in dem von TP 3 erarbeitenden Teilmodell *BillOfProcess* finden. Die Parameter, die Eigenschaften der Ressource darstellen, sind Implementierungsdetails der Operation “*determineFeasibleScope*”.

Beim Aufstellen der Formeln wurden die in AP 6.1 beschriebenen Maschinen zu Grunde gelegt. Dabei ist davon auszugehen, dass eine Maschine über die Fähigkeit verfügt, mehrere Prozesse unmittelbar nacheinander auszuführen. Es wird deshalb im Rahmen des Verhandlungsprozesses ein Angebot erwartet, mit der Angabe auf welche Prozesse des Teilmodells *BillOfProcess* sich das Angebot bezieht, d.h. mit Angabe des Startprozesses, des Endprozesses und der dazwischen ausgeführten Prozesse. Dabei wird eine Gesamtbewertung aller angebotenen Prozesse erwartet, nicht aber wie sich die Gesamtzeit oder die Gesamtkosten auf die einzelnen Prozesse verteilen.

7.3.3.1 Zeitermittlung für den Prozess Cut

Bei der Ausführung des Prozesses Cut sind das Einziehen der Leitung von der Leitungsspule und die Bewegung des Messers zu bewerten. Diese beiden Schritte werden nacheinander ausgeführt. Beim Einziehen ist in der Realität mit einer Beschleunigung am Anfang und mit einem Abbremsen am Ende des Einzugsvorgang zu rechnen, aber für die Zwecke des Verhandlungsprozesses ist die Annäherung mir konstanter Einzugsgeschwindigkeit hinreichend genau.

Formel für den Prozess Cut:

$$t_{cut} = \text{Cut.NominalLength} : v_{Feeding} + t_{CutterMovement}$$

Beschreibung der Werte:

Cut.NominalLength:

VWS-Teilmodellelement NominalLength des Prozesses Cut in Teilmodell *BillOfProcess* des angefragten Produktes.

$v_{Feeding}$:

Einzugsgeschwindigkeit als Wert, der innerhalb der Implementierung der Operation *determineFeasability* festgelegt wird.

$t_{CutterMovement}$:

Zeitdauer für die Bewegung des Messers als Wert, der innerhalb der Implementierung der Operation *determineFeasability* festgelegt wird.

7.3.3.2 Zeitermittlung für den Prozess MarkWire

Es wurden zwei verschiedene Arten von Markierungen erörtert: Zum einen Markierungsverfahren bei denen die Markierung mit einer Art Stempel in einem Zug aufgebracht wird (z.B. Hotstamp-Verfahren) und Tintenstrahl- oder Laserbedruckungsverfahren, bei denen kontinuierlich einzelne Reihen von Bildpunkten aufgebracht wird. Beide Verfahren werden während des Leitungseinzugs durchgeführt, was zu einer Unterbrechung oder Verlangsamung des Leitungseinzugs führt. Die betrachteten Maschinen können den Prozess Mark nur in Verbindung mit dem Prozess Cut ausführen, weshalb die Bewertung als Zuschlag zur in Cut Ermittelten Zeit angelegt wurden.

Welches Markierungsverfahren für das Produkt anzuwenden ist, geht aus dem Teilmodellelement *MarkType* des Prozesses MarkWire im Teilmodell *BillOfProcess* hervor.

Formel für das Hotstamp-Verfahren:

$$t_{mark} = \text{MarkWire.NumberOfRepetitions} * t_{stamp}$$

Beschreibung der Werte:

MarkWire.NumberOfRepetitions:

VWS-Teilmodellelement NumberOfRepetitions des Prozesses MarkWire in TeilModell *BillOfProcess* des angefragten Produktes.

t_{stamp} :

Zusätzliche Zeitdauer je Markierungsvorgang durch das Abbremsen oder Anhalten des Leitungseinzugs zur Ausführung des Markierungsvorgangs als Wert, der innerhalb der Implementierung der Operation *determineFeasibleScope* festgelegt wird.

Formel für das Tintenstrahl- und Laserdruckverfahren:

$$t_{mark} = \text{MarkWire.NumberOfRepetitions} * t_{printWidthUnit} * \text{lengthOf(} \text{MarkWire.Content} \text{)}$$

Beschreibung der Werte und Funktionen:

MarkWire.NumberOfRepetitions:

VWS-Teilmodellelement NumberOfRepetitions des Prozesses MarkWire in TeilModell *BillOfProcess* des angefragten Produktes.

$t_{printWidthUnit}$:

Zusätzliche Zeitdauer je Ausdruck einer Druckeinheit (Bildpunktzeile, Buchstabe, Millimeter) durch das Abbremsen des Leitungseinzugs zur Ausführung des Druckvorgangs als Wert, der innerhalb

der Implementierung der Operation *determineFeasibleScope* festgelegt wird. Die verwendete Einheit für den Zeitwert ist dabei auf die Funktion *lengthOf* abgestimmt.

lengthOf:

Die Funktion *lengthOf* ermittelt die Anzahl Druckeinheiten (bspw. in Bildpunktzeilen, Buchstaben, Millimetern) die zur Darstellung des als Inputparameter gegebenen Inhalts notwendig sind.

MarkWire.Content:

VWS-Teilmodellelement Content des Prozesses MarkWire in Teilmödell *BillOfProcess* des angefragten Produktes.

7.3.3.3 Zeitermittlung für den Prozess Strip

Bei der Ausführung des Prozesses Strip ist die Bewegung des Leitungsende hin zur Werkzeugstation und die Zeitdauer für das Einschneiden und Abziehen zu berücksichtigen. Es wurde angenommen, dass bezogen auf einen Abisolationsvorgang der Zeitunterschied bei unterschiedlichen Abisolationslängen vernachlässigt werden kann. Es gibt allerdings eine Obergrenze der Abisolationslänge, die die Maschine in einem Abisolationsvorgang bearbeiten kann. Das bedeutet, wenn die gewünschte Abisolationslänge diese Obergrenze übersteigt, so wird die Maschine mehrere Abisolationsvorgänge hintereinander ausführen.

Formel für den Prozess Strip:

$$t_{\text{Strip}} = \text{ceil}((\text{Strip.StopPosition} - \text{Strip.StartPosition}) / L_{\max}) * t_{\text{StripStep}}$$

Beschreibung der Werte und Funktionen:

ceil

Funktion, die eine übergebene Dezimalzahl auf den nächsthöheren Ganzahlwert aufrundet.

Strip.StopPosition

VWS-Teilmodellelement StopPosition des Prozesses Strip in Teilmödell *BillOfProcess* des angefragten Produktes gibt die Endposition des abzusolierenden Bereichs in Bezug auf das Leitungsende an.

Strip.StartPosition

VWS-Teilmodellelement StartPosition des Prozesses Strip in Teilmödell *BillOfProcess* des angefragten Produktes gibt die Startposition des abzusolierenden Bereichs in Bezug auf das Leitungsende an

L_{\max}

Maximal mögliche Länge in einem Abisolationsvorgang als Wert, der innerhalb der Implementierung der Operation *determineFeasibleScope* festgelegt wird.

$t_{\text{StripStep}}$

Zeitdauer für einen Abisolationsvorgang als Wert, der innerhalb der Implementierung der Operation *determineFeasibleScope* festgelegt wird.

7.3.3.4 Zeitermittlung für den Prozess Crimp

Bei der Ausführung des Prozesses Crimp ist die Bewegung des Leitungsende hin zur Werkzeugstation und die Zeitdauer für das Ausführen des Crimps, was im Wesentlichen aus einem Werkzeughub besteht, zu berücksichtigen. Die Geschwindigkeit für die Bewegung des Schwenkarms hin zur Werkzeugstation kann von Leitungstyp und Leitungsquerschnitt abhängig sein. Die Zeitdauer für den Werkzeughub kann von Leitungsquerschnitt und verwendetem Terminal abhängig sein.

Es wurde deshalb vorgeschlagen diese beiden einzelnen Zeitwerte über Funktionen zu ermitteln. Die Implementierung dieser Funktionen könnte zum Beispiel durch Auswerten einer Tabelle mit Vorgabewerten erfolgen.

Die beiden Funktionen benötigen als Eingabeparameter aus dem Teilmödell *BillOfProcess* das Teilmödelement *Crimp.WireCrossSectionArea* und aus dem allgemeinen Bereich die Referenzen auf die verwendeten Materialien, die in *MaterialOccurrence* angegeben sind.

Für die Implementierung des Demonstrators wurde entschieden, einen fixen Zeitwert je Crimpvorgang in der maschinenabhängigen Implementierung zu hinterlegen.

7.3.3.5 Zeitermittlung für den Prozess Seal

Bei der Ausführung des Prozesses Seal ist die Bewegung des Leitungsende hin zur Werkzeugstation und die Zeitdauer für das Aufschieben der Einzelleiterabdichtung zu berücksichtigen. Die Geschwindigkeit für die Bewegung des Schwenkarms hin zur Werkzeugstation kann von Leitungstyp und Leitungsquerschnitt abhängig sein. Die Zeitdauer für das Aufschieben kann von Leitungsquerschnitt und verwendetem Dichtungselement abhängig sein.

Es wurde deshalb vorgeschlagen, diese beiden einzelnen Zeitwerte über Funktionen zu ermitteln. Die Implementierung dieser Funktionen könnte zum Beispiel durch Auswerten einer Tabelle mit Vorgabewerten erfolgen.

Die beiden Funktionen benötigen als Eingabeparameter aus dem Teilmodell *BillOfProcess* das Teilmolelement *Crimp.WireCrossSectionArea* und aus dem allgemeinen Bereich die Referenzen auf die verwendeten Materialien, die in *MaterialOccurrence* angegeben sind.

Für die Implementierung des Demonstrators wurde entschieden, einen fixen Zeitwert je Vorgangsausführung in der maschinenabhängigen Implementierung zu hinterlegen.

7.3.3.6 Zeitermittlung für den Prozess Twist

Für die Bewertung des Prozesses Twist ist die Anzahl der Umdrehungen zu berücksichtigen, welche die Maschine mit einer bestimmten Drehzahl ausführen muss. Die Anzahl der Umdrehungen ergibt sich aus der verdrillten Länge und der Schlaglänge. Zusätzlich kann es einen längenunabhängigen Wert geben, der für das Herstellen einer Ausgangs- oder Endposition notwendig ist, wie z.B. für das Greifen der Leitungsenden.

Formel für den Prozess Twist:

$$t_{\text{Twist}} = t_{\text{Fix}} + (\text{WireTwist.NominalWireEndLength} - \text{WireTwist.Side1.NominalOpenEnd} - \text{WireTwist.Side1.NominalOpenEnd}) / \text{WireTwist.NominalLayLength} / v_{\text{Rotation}}$$

Beschreibung der Werte:

t_{Fix}

Fester Zeitwert für Herstellen der Eingangs- und Ausgangskonfiguration, z.B. Greifen der Leitungen, der innerhalb der Implementierung der Operation *determineFeasibleScope* festgelegt wird.

WireTwist.NominalWireEndLength

VWS-Teilmolelement NominalWireEndLength des Prozesses WireTwist in Teilmodell *BillOfProcess* des angefragten Produktes gibt die Soll-Leitungslänge im verdrillten Zustand einschließlich der unverdrillten Leitungsenden an.

WireTwist.SideX.NominalOpenEnd

VWS-Teilmolelement NominalOpenEnd des Prozesses WireTwist in Teilmodell *BillOfProcess* des angefragten Produktes gibt die offene, d.h. nicht verdrillte Länge am Leitungsende Side1 oder Side2 an.

WireTwist.NominalLayLength

VWS-Teilmolelement NominalOpenEnd des Prozesses WireTwist in Teilmodell *BillOfProcess* des angefragten Produktes gibt die Schlaglänge an.

v_{Rotation}

Anzahl Umdrehungen je Zeiteinheit bezogen auf die Leitungen, d.h. Umdrehung die genau einen Verdrillschlag bewirkt. Dieser Wert wird innerhalb der Implementierung der Operation *determineFeasibleScope* festgelegt oder ermittelt.

7.3.3.7 Zeitermittlung für den Prozess SpotTape

Bei der Ausführung des Prozesses SpotTape ist das Positionieren des Wickelkopfes und die Umdrehungen des Wickelkopfes zu bewerten. Für das Teilmodell *BillOfProcess* wurde in TP3 festgelegt, dass die Informationen im Prozess WireTwist bereitgestellt wird.

Formel zur Zeitberechnung:

$$t_{\text{Twist}} = t_{\text{Fix}} + \text{WireTwist.NumberOfLayersSpotTape} * v_{\text{Rotation}}$$

Beschreibung der Werte:

t_{Fix}

Fester Zeitwert für Herstellen der Eingangs- und Ausgangskonfiguration, z.B. Greifen der Leitungen, der innerhalb der Implementierung der Operation *determineFeasability* festgelegt wird.

WireTwist.NumberOfLayersSpotTape

VWS-Teilmodellelement *NumberOfLayersSpotTape* des Prozesses *WireTwist* in Teilmodell *BillOfProcess* des angefragten Produktes gibt die Anzahl der notwendigen Lagen vor.

v_{Rotation}

Anzahl Umdrehungen des Wickelkopfes je Zeiteinheit. Dieser Wert wird innerhalb innerhalb der Implementierung der Operation *determineFeasibleScope* festgelegt oder ermittelt.

7.4 AP 6.4 - ECLASS und Industrie 4.0-Sprache

Das AP 6.4 „**ECLASS und Industrie 4.0-Sprache**“ wurde ursprünglich definiert, um die vorhandenen Spezifikationen zu Vokabular, Struktur der Nachrichten und Interaktionsprotokollen auf Anwendbarkeit im Kontext des Projektes zu überprüfen und aufzuzeigen, welche Erweiterungen notwendig sind, um die Anwendungsfälle aus der Domäne Leitungssatz abilden zu können.

Das in VDI/VDE 2193-2 [46] spezifizierte Interaktionsprotokoll zum Ausschreibungsverfahren konnte für die in AP 6.1 entwickelten Anwendungsfälle eingesetzt werden. Eine Anpassung oder Erweiterung des Interaktionsprotokolls war dafür nicht notwendig.

Der Aufbau der Synchronisationsnachrichten folgt VDI/VDE 2193-1 [49] und teilt sich in Nachrichtenrahmen und Interaktionselementen mit den Nutzdaten zur Übermittlung der Wertänderungen. Die Interaktionselemente des Nachrichtenrahmens wurden wie folgt festgelegt:

| Nachrichten-element | Beschreibung | Verwendung |
|------------------------|--|---------------|
| Typ | Zweck der Nachricht: Synchronisation, Heartbeat, Canceled | Verpflichtend |
| Sender | Absender einer Nachricht (VWS-ID) | Verpflichtend |
| Empfänger | Empfänger einer Nachricht (VWS-ID) | Optional |
| Konversations-ID | ID der Synchronisationsbeziehung, festgelegt während der Konfiguration (Verhandlungsprozess) | Verpflichtend |
| Nachrichten-ID | ID einer Nachricht, um z.B. eine doppelte Verarbeitung auf Empfängerseite zu verhindern. | Verpflichtend |
| Als-Antwort-Auf | Ausdruck, der die Ursprungsnachricht referenziert, wird in diesem Interaktionsprotokoll bisher nicht benötigt. | Nicht genutzt |
| Antworten-bis | Zeitpunkt bis zu dem die Antwort vorliegen muss, wird in diesem Interaktionsprotokoll bisher nicht benötigt. | Nicht genutzt |
| Semantisches Protokoll | Identifikation des semantischen Protokolls auf das sich der Zweck bezieht, hier Synchronisation | Verpflichtend |
| Rolle | Aufgabe des Senders der Nachricht im semantischen Protokoll, hier Synchronisation Message Provider | Optional |

Tabelle 7-1: Aufbau der Synchronisationsnachrichten

Da zur technischen Umsetzung des Nachrichtenaustauschs die VWS-Operation gewählt wurde, wurde ein Mapping der I-4.0-Nachrichtenelemente auf Aufrufparameter der VWS-Operation erarbeitet. Die Operation „newMessage“ ist generell dafür vorgesehen I4.0-Nachrichten auszutauschen, unabhängig vom Interaktionsprotokoll. Für diese Operation wurde ein Aufrufparameter mit idShort „frame“ für die Attribute des Nachrichtenrahmens vorgesehen und ein zweiter Aufrufparameter mit idShort „interactionElements“ für die anwendungsfallspezifischen Erweiterungen. Diese Operation ist über ein Teilmodell „MessageParticipant“ definiert. Die Operation „NewMessage“ ist im Interaktionsprotokoll Ausschreibungsverfahren sowohl vom *Service Requester* als auch vom *Service Provider* verpflichtend zu implementieren. Für *Service Requester* ist darüber hinaus eine Operation *startBiddingProcess* vorgesehen, mit der ein Durchlauf des Ausschreibungsverfahrens gestartet werden kann.

Damit man erkennen kann, welche VWS welches Interaktionsprotokoll implementiert, ist im Teilmodell „MessageParticipant“ eine Liste „semanticProtocols“ enthalten, in der angegeben wird, welches Interaktionsprotokoll implementiert ist und in welchen Rollen die VWS in diesem Interaktionsprotokoll einnehmen kann-.

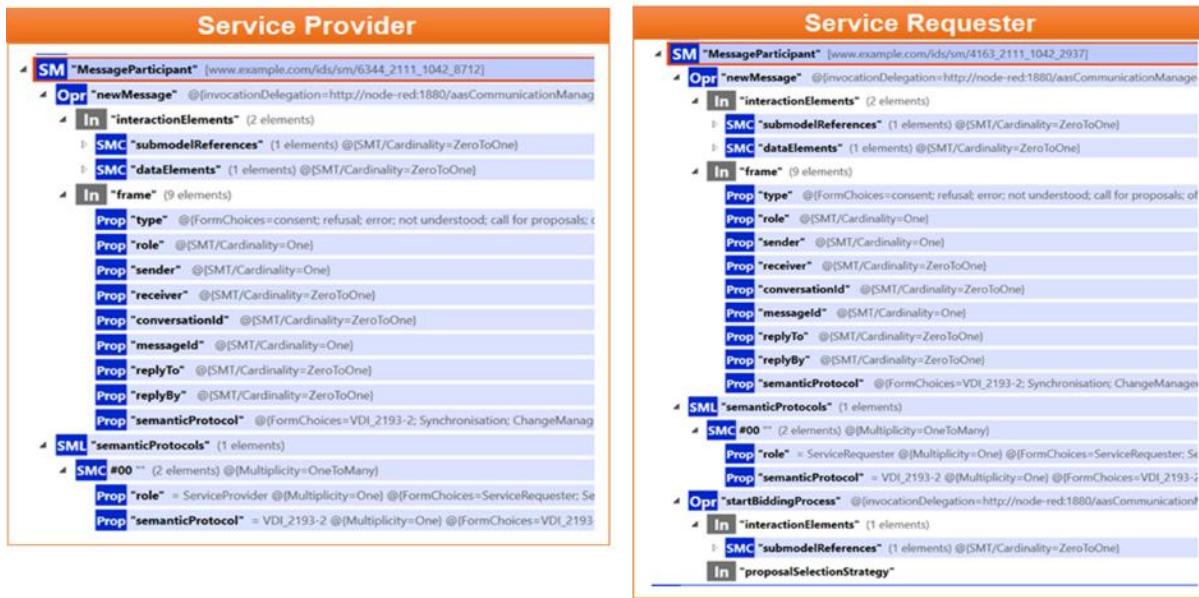


Abbildung 7-14 VWS-Teilmodelle für [Service-Requester](#) und [Service-Provider](#)

Als dritte Ebene neben Interaktionsprotokoll und Nachrichtenstruktur wurde in AP 6.4 das Vokabular der I4.0-Sprache betrachtet. Ein Abgleich von geforderten und in der Ressource verfügbaren Fähigkeiten konnte so generisch implementiert werden, dass auch ohne zusätzliche Semantik eine automatisierte Entscheidung möglich ist. Wie in AP 6.3 gezeigt ist aber eine gesamte Bewertung des Produktes bis hin zu einem Angebot in dieser generischen Form nicht mehr möglich. An dieser Stelle ist eine standardisierte Produktbeschreibung wichtig. Die semantisch eindeutig beschriebenen Teilmodellelemente sind durch das in TP 3 entwickelte Teilmodell "Bill Of Process" [26] gegeben. Allen Arbeiten an Informationsmodellen mit Bezug zum Produkt Leitungssatz lag das Modell des VEC zugrunde, das als umfassendes Datenmodell in der Branche immer noch weiterentwickelt wird. Was die Beschreibung der technischen Merkmale des Produktes Leitungssatz angeht, wurde keine Notwendigkeit erkannt, zusätzliche Semantik zu definieren.

Im Kontext des unternehmensübergreifenden Verhandlungsprozesses sind jedoch zusätzliche, nicht technische Merkmale für Ausschreibung und Angebot notwendig. Hier wurde so vorgegangen, dass zunächst eine Sammlung dieser Merkmale erfolgt ist und im Anschluss im ECLASS-Katalog entsprechende Merkmale und deren eindeutige IRDI gesucht wurden. Das Ergebnis ist in nachfolgender Tabelle dargestellt.

| Nachrichtentyp | Datum | Abbildung in I4.0-Nachricht | Semantik | IRDI |
|----------------|-------------------------------------|-----------------------------|----------|----------------------|
| Ausschreibung | Zieltermin | Interaktions-Element | ECLASS | 0173-1#02-ABH165#002 |
| Ausschreibung | Mengeneinheit | Interaktions-Element | ECLASS | 0173-1#02-ABC475#003 |
| Ausschreibung | Menge | Interaktions-Element | ECLASS | 0173-1#02-AAR706#002 |
| Ausschreibung | Lieferbedingung (Ausführung) | Interaktions-Element | ECLASS | 0173-1#02-BAE397#006 |
| Ausschreibung | Lieferbedingung (Gütegrad) | Interaktions-Element | ECLASS | 0173-1#02-BAE398#006 |
| Ausschreibung | Lieferbedingung/ Lieferzustand | Interaktions-Element | ECLASS | 0173-1#02-BAE399#006 |
| Ausschreibung | Lieferbedingung (Prüfumfang) | Interaktions-Element | ECLASS | 0173-1#02-BAE400#006 |
| Ausschreibung | Lieferbedingung | Interaktions-Element | ECLASS | 0173-1#02-BAE401#004 |
| Ausschreibung | Norm Technische Lieferbedingungen | Interaktions-Element | ECLASS | 0173-1#02-ABL353#001 |
| Ausschreibung | Erwartete Verpackung | Interaktions-Element | ECLASS | 0173-1#02-AAK660#006 |
| Ausschreibung | Menge je Verpackungseinheit | Interaktions-Element | ECLASS | 0173-1#02-BAA306#005 |
| Ausschreibung | Mengeneinheit je Verpackungseinheit | Interaktions-Element | ECLASS | 0173-1#02-ABC475#003 |
| Ausschreibung | Ersteller der Anfrage | Nachrichten-Rahmen | Sender | n.a. |

| | | | | |
|---------------|--------------------------------------|----------------------|-------------------------------------|---|
| Ausschreibung | Anfragenummer | Nachrichten-Rahmen | Konversations-Id | n.a. |
| Ausschreibung | Zeitpunkt Rückmeldung | Nachrichten-Rahmen | Antworten bis | n.a. |
| Ausschreibung | Verweis auf Anforderungsdokumente | Interaktions-Element | ECLASS / Aspekt Zusatzdokumentation | 0173-1#01-ADN464#011 |
| Angebot | Zugesicherter spätester Liefertermin | Interaktions-Element | ECLASS | 0173-1#02-ABH165#002 |
| Angebot | Angebotspreis Brutto | Interaktions-Element | ECLASS | 0173-1#02-AAQ214#002 |
| Angebot | Angebotspreis Netto | Interaktions-Element | ECLASS | 0173-1#02-BAF749#003 / 0173-1#02-AAO822#002 |
| Angebot | Angebotswährung | Interaktions-Element | ECLASS | 0173-1#02-AAO288#006 |
| Angebot | Lieferkondition | Interaktions-Element | ECLASS | 0173-1#02-AAX092#003 |

Tabelle 7-2: Zusätzliche Merkmale für VWS4LS-I4.0-Nachrichten zu Ausschreibung und Angebot

7.5 Fazit

Im Rahmen von TP 6 wurden zunächst die Vorarbeiten zur I4.0-Sprache analysiert. Obwohl die I4.0-Sprache im Hinblick auf die aktive VWS entwickelt wurde, gibt es bisher nur wenige Umsetzungen, die diese direkt in die VWS integrieren. Bisher wurden häufig agentenbasierte Ansätze gewählt, bei denen Software außerhalb der Systemgrenzen der Verwaltungsschale die Kommunikation übernimmt und Entscheidungen automatisiert. Als Kommunikationskanal kommen dabei häufig Nachrichtenprotokolle wie MQTT zum Einsatz, deren Anwendung im Zusammenhang mit der VWS derzeit nicht standardisiert ist.

In TP 6 wurde zum Nachrichtenaustausch bewusst die VWS-Operation gewählt, weil diese vollständig in "Specification of the Asset Administration Shell Part 2" [48] spezifiziert ist und somit die Interoperabilität im Hinblick auf die technische Übertragung von Nachrichten zwischen verschiedenen VWS-Implementierungen gewährleistet werden kann. Der mit VDI/VDE 2193-1 [49] standardisierte Nachrichtenaufbau konnte auch in den Aufrufparametern der VWS-Operation implementiert werden.

Die Nutzung der standardisierten VWS-API für den Nachrichtenaustausch bringt insbesondere im unternehmensübergreifenden Wertschöpfungsnetzwerk Vorteile. Dort, wo die Anbindung zum Lesen und Schreiben von Teilmustern erfolgreich über Unternehmensgrenzen umgesetzt wird, können damit auch I4.0-Nachrichten ausgetauscht werden ohne zusätzliche Kommunikationskanäle einzurichten und absichern zu müssen.

Durch die in VWS4LS entwickelten Teilmustere steht das notwendige Vokabular für die automatisierten Verhandlungen im Kontext der Leitungssatzproduktion zur Verfügung.

Mit dem Einsatz der "Low Code"-Systeme NodeRed und Flowable sowie deren Erweiterung mit JavaScript-Modulen und Java Spring Beans wurde aufgezeigt, wie die Implementierung einer VWS über den Lebenszyklus des Assets hinweg anpassbar und flexibel gelöst werden kann. Mit der Implementierung (<https://github.com/VWS4LS/vws4ls-bidding-process>) wurden die Konzepte Komponentenmanager und Interaktionsmanager "nahe" an der Verwaltungsschale umgesetzt und eine Blaupause für die Umsetzung in VWS-Middleware geschaffen, siehe Abbildung 7-15.

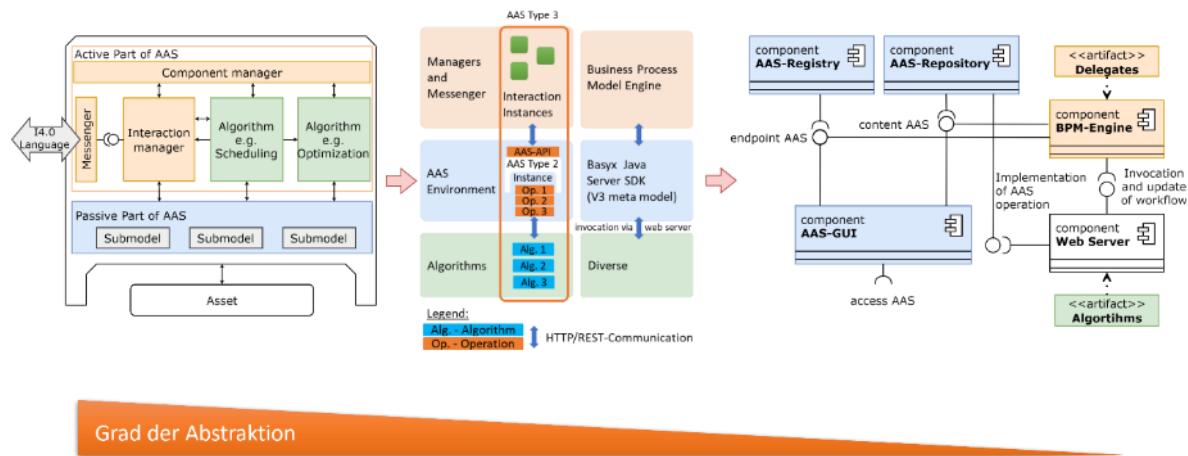


Abbildung 7-15: Architektur für die VWS Typ 3 ("Proaktive VWS")

8 TP7 - Data Business Policy, Data Governance und Monetarisierungskonzepte

Das TP 7 "Data Business Policy, Data Governance, Monetarisierung" fokussierte auf die effiziente Nutzung von Daten in der Wertschöpfungskette, der Gewährleistung der Datensicherheit und der Schaffung von Mehrwert für Unternehmen, d.h. damit Unternehmen, die innovative Geschäftsmodelle entwickeln auch neue Einkommensquellen generieren können, basierend auf dem Zugang zu Informationen und Diensten über die VWS.

Die drei Arbeitspakete konzentrierten sich auf die Anforderungen und Rahmenbedingungen auf organisatorischer Ebene für den herstellerübergreifenden Datenaustausch und den Umgang mit Daten mithilfe der VWS. Im ersten Schritt wurden Anforderungen gesammelt und analysiert, danach wurde ein Konzept entwickelt, wie eine Data Governance aussehen soll. Im nächsten Schritt wurde ein repräsentativer Anwendungsfall im Bereich der Leitungssatzherstellung definiert, der den Austausch von Informationen zwischen Teilnehmern der Wertschöpfungskette darstellt. Dieser Anwendungsfall wurde unter der Perspektive der Verwendung eines Internationalen Datenraums (IDS) beschrieben, um auf die Analyse der Informationsübertragung zu fokussieren. Zuletzt wurden in einem Piloten die Ansätze geprüft.

8.1 AP 7.1 - Anforderungserhebung

Im AP 7.1 "Anforderungserhebung" wurden, ausgehend von den aktuellen Datenhaltungssituationen der Partner, Anforderungen für eine Datenhaltung im Hinblick auf die gemeinsame Nutzung entlang der Wertkette „Leitungssatz“ erarbeitet. Dabei wurde für jede Anforderungskategorie die zu betrachtenden Aspekte, sowie die zu bedienenden potenziellen Schnittstellen aufgelistet.

8.1.1 Datenrichtlinien (Data Business Policy)

Eine Datenrichtlinie dient als Leitfaden und Orientierung für die Verwendung und den Umgang mit vertraulichen Daten. Sie ist eng mit der Data Governance (DG) verknüpft und bildet einen Teilbereich davon [50] [51]. Eine Datenrichtlinie setzt sich aus einer formellen Sammlung von Regeln, Prinzipien und Verfahren zusammen, die einen kontinuierlichen und korrekten Umgang mit Daten sicherstellen sollen. Ein grundlegendes legislatives Regelwerk hierzu ist die [Datenschutz-Grundverordnung \(DSGVO\)](#)⁵¹. Zudem behandelt eine Datenrichtlinie verschiedene Aspekte, wie bspw. Anforderungen an die Datenqualität (DQ), den Datenzugriff, die Datensicherheit sowie die Verwendung und Nutzung der Daten [50] [51]. Auf diese Weise soll eine Datenrichtlinie bedeutenden Teil der organisatorischen Anforderungen an den herstellerübergreifenden Datenaustausch abdecken. Im Folgenden werden stark angelehnt an [52] und [53] die Aspekte hergeleitet und beschrieben, die die Einhaltung einer Datenrichtlinie betreffen.

8.1.1.1 Datensicherheit

Der Datenaustausch zwischen den Herstellern muss gemäß den von beiden Parteien vereinbarten und/oder gesetzlich vorgeschriebenen Datenrichtlinien erfolgen. Dazu gehören die Gewährleistung des gegenseitigen Vertrauens und/oder der Authentifizierung der beteiligten Parteien, ein sicheres Informationsmanagement und die Datenintegrität, wie z. B. die Datenverschlüsselung [50] [52]. Zu diesem Zweck wurde bereits die Norm ISO 27001 [53] definiert und diverse [BSI-Standards](#)⁵² veröffentlicht. Diese Norm zielt darauf ab, ein Grundvertrauen auf der Basis eines Zertifikats zu ermöglichen, das von einer Zertifizierungsstelle nach erfolgreicher Authentifizierung ausgestellt wird. ISO 27001 vergibt dieses Zertifikat an Organisationen, die ein sicheres Informationsmanagementsystem gewährleisten können. Damit können kooperierende Organisationen, die Daten austauschen wollen, sicher sein, dass die Daten in der anderen Organisation mit der notwendigen und erforderlichen Sicherheit behandelt werden. Für den herstellerübergreifenden Datenaustausch von personenbezogenen Daten besteht die zusätzliche Anforderung der [Datenschutz-Grundverordnung \(DSGVO\)](#), deren strikte Einhaltung zwingend erforderlich ist. Aufgrund der zahlreichen Voraussetzungen, Regelungen und strengen Kontrollen wird im Kontext der VWS die Annahme getroffen, dass personenbezogene Daten sich nicht für den automatisierten Datenaustausch über die VWS eignen und daher nicht für diesen Zweck freigegeben werden.

⁵¹ <https://dsrgvo-gesetz.de/>

⁵² <https://www.bsi.bund.de/dok/6604686>

Folglich unterliegt der herstellerübergreifende Datenaustausch von personenbezogenen Daten weiterhin einer manuellen Prüfung und Freigabe. Um dies zu gewährleisten, werden die Daten zunächst klassifiziert. Diese Klassifikation der Daten wird in der Spezifikation des Teilespekts Rollen- und Rechtemanagement erläutert.

Definition of Done: Die Anforderung gilt als erfüllt, wenn der Datenaustausch zwischen den Herstellern nach vereinbarten Datenrichtlinien erfolgt, bei denen die Anforderungen an die Datensicherheit, einschließlich Authentifizierung und Autorisierung, erfüllt sind. Darüber hinaus müssen die Bestimmungen der Allgemeinen Datenschutzverordnung strikt eingehalten werden, und personenbezogene Daten können nicht automatisch ausgetauscht werden. Die manuelle Überprüfung und Freigabe personenbezogener Daten muss nach festgelegten Rollen- und Rechteverwaltungsverfahren erfolgen und umgesetzt werden.

8.1.1.2 Interoperabilität

Die Interoperabilität der VWS bezieht sich auf die Fähigkeit, Daten zwischen verschiedenen Systemen und Organisationen zu übertragen. Dies bedeutet, dass Organisationen mit unterschiedlichen Organisationsstrukturen oder Organisationssystemen dennoch Daten miteinander austauschen können [54]. Um Interoperabilität zu gewährleisten, sind einheitliche Datenformate, Strukturen und Kommunikationsprotokolle erforderlich [54] [55]. Hierzu wurden im Rahmen der VWS standardisierte Datenaustauschformate wie XML, JSON und AASX definiert [55] [56] [54]. Einen weiteren wichtigen Aspekt stellt die semantische Interoperabilität dar. Diese bezieht sich auf die Fähigkeit, eine gemeinsame und eindeutige Bedeutung von Daten sicherzustellen. Das bedeutet, dass Daten in Form von Properties in der Struktur der VWS mit global standardisierten Lexika und den darin enthaltenen IDs, Nomenklaturen, Konzeptbeschreibungen oder Metadaten versehen werden.

Definition of Done: Die Anforderung gilt als erfüllt, wenn die VWS die Übertragung von Daten zwischen verschiedenen Systemen und Organisationen ermöglicht und dabei standardisierte Datenformate, Strukturen und/oder Kommunikationsprotokolle verwendet. Zudem muss sichergestellt werden, dass Daten in der VWS eine gemeinsame und eindeutige Bedeutung haben und von allen beteiligten Akteuren interpretiert werden können.

8.1.1.3 Datenintegrität

Datenintegrität bezeichnet die DQ über den gesamten Datenlebenszyklus hinweg [50] [54]. Im Rahmen des herstellerübergreifenden Datenaustauschs, sowohl während der Übertragung als auch der Speicherung, wird die Vollständigkeit, Richtigkeit, Unveränderlichkeit, Verfügbarkeit und Aktualität dieser Daten gewährleistet [54][55].

Definition of Done: Diese Anforderung gilt als erfüllt, wenn die Daten in der VWS während des gesamten Lebenszyklus, einschließlich Übertragung und Speicherung, die genannten Kriterien hinsichtlich Vollständigkeit, Richtigkeit, Unveränderlichkeit, Verfügbarkeit und Aktualität erfüllen. Hierfür können Authentifizierungsmethoden und/oder Rollen- sowie Rechtemodelle in der Konzeption berücksichtigt und umgesetzt werden.

8.1.1.4 Datensouveränität

Datensouveränität bezeichnet im Kontext der VWS die Sicherstellung, dass die jeweiligen Organisationen die Kontrolle, Datenhoheit und das Recht über ihre Daten während des herstellerübergreifenden Datenaustauschs beibehalten [54]. Das soll unabhängig von der Anzahl teilnehmender Akteure oder Systeme sichergestellt werden. Demnach sollen Organisationen im Zuge der Datensouveränität eigenständig entscheiden können, welche Daten, in welchem Umfang und mit wem geteilt werden.

Definition of Done: Die Anforderung gilt als erfüllt, wenn die Datensouveränität basierend auf den Datenrichtlinien berücksichtigt und gewährleistet ist. Somit müssen die Datenrichtlinien vor dem Datenaustausch von dem Datenbenutzer akzeptiert werden.

8.1.2 Zugriffskontrolle

Eine Zugriffskontrolle umfasst die Sicherstellung der Authentifizierung, Autorisierung und Vertrauenswürdigkeit aller beteiligten Akteure, welche Datenzugriff auf die VWS erhalten möchten. Die Zugriffskontrolle stellt somit eine fundamentale Grundlage für die Einhaltung der Datensicherheit, Vertraulichkeit und Datenintegrität dar. Die Authentifizierung soll die Identität und Echtheit des Benutzers sicherstellen, bspw. Soll mittels Zertifikate, Passwörter oder Benutzernamen überprüft werden, ob es sich bei dem zugreifenden Benutzer tatsächlich um diesen handelt [54]. Die Autorisierung legt nach erfolgreicher

Authentifizierung fest, welcher Benutzer auf welche Daten und Inhalte zugreifen darf [54]. Für die Festlegung der Granularität des Datenzugriffs können bspw. Rollen und Rechte den entsprechenden Benutzern zugewiesen werden. Die Zugriffsrechte regeln und beschränken die Aktionen, welche die Rollen und Benutzer ausführen können.

Definition of Done: Die Anforderung gilt als erfüllt, wenn der Zugriff auf die Daten der VWS durch eine Zugriffskontrolle beschränkt wird. Durch die Authentifizierung muss erfolgreich die Identität der Benutzer überprüft werden. Durch die Autorisierung müssen erfolgreich die Zugriffsrollen und Zugriffsrechte festgelegt werden, die die Aktionen der Benutzer regulieren und beschränken.

8.1.3 Rollen und Rechtekonzept

Die geteilte Nutzung der Daten einer VWS durch andere Akteure der Wertkette setzt Lösungen für das Rollen- und Rechtemanagement voraus, mit denen Zugriffe verwaltet werden. Das Rollen- und Rechtemanagements stellt die letzte Restriktion vor dem Datenzugriff dar und soll die Granularität für den Datenzugriff durch verschiedene Benutzer, Rollen und/oder Attribute sicherstellen. Dabei sollen die jeweiligen verwendeten Rollen, Rechte sowie Attribute zwischen den beiden Akteuren in der Datenrichtlinie definiert werden. Ebenso sollte den Rollen ein Zugriffszeitraum zugewiesen werden, welcher regelt, wie lange der Datenzugriff auf die entsprechende VWS gewährt wird. Nach dem Ablauf des Zugriffszeitraums soll der Prozess der Zugriffskontrolle erneut durchgeführt werden [54]. Für die Zugriffskontrolle haben sich bisher zwei Konzepte etabliert: die rollenbasierte Zugriffskontrolle (RBAC = *role-based access control*) und die attributbasierte Zugriffskontrolle (ABAC = *attribute-based access control*). Für RBAC müssen Rollen definiert werden, die sowohl eine unterschiedliche Tiefe des Zugriffs, also wie weitgreifend der Zugriff auf die Daten sein darf, als auch welche operativen Rechte die jeweiligen Rollen erhalten, also ob Daten lediglich gelesen oder auch verändert oder kopiert werden dürfen. Diese Rollen müssen für beide Parteien gültig sein. Dieselben Unterscheidungen gelten auch für das ABAC, hierbei müssen jedoch Attribute definiert werden, anhand derer die Zugriffsrechte unterschieden werden können. Diese Attribute können unterschiedlicher Art sein, wie die Position innerhalb eines Unternehmens oder der Nachweis einer bestimmten Fähigkeit. Dieser Katalog an Attributen muss bei beiden Parteien, also dem Datenumheber und der Partei, die Zugriff auf die Daten erhalten möchte, gültig sein. Der Vorteil von ABAC gegenüber RBAC ist, dass jeder Person individuell anhand eigens definierter Attribute ein präzises Zugriffsprofil erstellt werden kann. Bei RBAC können den verschiedenen Personen lediglich voreingestellte, statische Rollen zugewiesen werden. Das ABAC-Modell erfordert jedoch weit größere Verwaltungsaufwände als das RBAC-Modell.

Definition of Done:

- Die Anforderung gilt als erfüllt, wenn das Rollen- und Rechtemanagement die Granularität für den Datenzugriff sicherstellt, Rollen, Rechte und Attribute in der Datenrichtlinie definiert und festhält
- Beschreibung des Konzepts zu Rollen und Rechtenmanagement von VWS liegt vor. Der Schwerpunkt liegt hier auf der Konzeption einer Lösung, die sich gut in die Struktur der VWS integriert
- Definition der möglichen Rollen und Rechte für einen bestimmten Use Case
- Erprobung des Konzeptes. BaSyx bietet bereits einige Möglichkeiten, Rollen und Rechte zu definieren

8.1.4 Datensicherheit beim Datenaustausch

Beim Austausch von Daten zwischen Unternehmen spielt die Sicherheit eine fundamentale Rolle, um die Vertraulichkeit, Integrität und Verfügbarkeit der Informationen zu gewährleisten. In diesem Zusammenhang zielt die Datensicherheit darauf ab, die Informationen vor unbefugtem Zugriff durch Benutzer zu schützen. Zu diesem Zweck können Technologien zur Datenverschlüsselung sowohl während der Übertragung als auch bei der Speicherung eingesetzt werden. Ein exemplarisches Beispiel hierfür ist die Anwendung bereits bekannter Konzepte wie dem Transport Layer Security (TLS) für die Datenübertragung und der Datenverschlüsselung. Zusätzlich ermöglicht die Aufzeichnung der Prozesse durch Traceability-Tools die Analyse von Anomalien im Verhalten der Daten. In diesem Kontext spielen die Speicherung der Daten und deren Sicherungskopien eine sehr wichtige Rolle. Im Folgenden werden

die grundlegenden Aspekte im Zusammenhang mit dem Thema Datensicherheit und ihre jeweiligen Anforderungen vorgestellt:

8.1.4.1 Vertraulichkeit:

Vertraulichkeit bezieht sich nach ISO/IEC 27002 [57] auf den Aspekt, nach dem ausschließlich autorisierte Benutzer Zugriff auf relevante Informationen haben. Zur Gewährleistung der Vertraulichkeit von Informationen in der VWS-Umgebung können verschiedene Techniken und Konzepte in Betracht gezogen werden.

Aspekte:

- Einschränkung des Zugriffs durch Benutzerklassifizierung: Begrenzung des Zugriffs von Benutzern auf Repositories und Datensätze von Unternehmen, die am Datenaustausch beteiligt sind, durch die Erstellung eines Modells zur Klassifizierung potenzieller Benutzer mit Interesse an den in der VWS vorhandenen Informationen und Diensten
- Konzeption eines Verschlüsselungsmodells: Verschlüsselung von Informationen in der VWS, um die Daten vor nicht autorisiertem Zugriff zu schützen
- Klassifizierung vor der Übertragung: Klassifizierung von Informationen nach ihrer Sensibilität vor der Übertragung in die VWS-Repositories. Hierzu wird die Umsetzung eines Konzepts zur Klassifizierung von Daten anhand verschiedener Kategorien unter Verwendung der etablierten Struktur der VWS vorgeschlagen
- Verwendung von Geheimhaltungsvereinbarungen (NDA): Geheimhaltungsvereinbarungen (NDA) stellen ein rechtliches Instrument dar, um die Vertraulichkeit zu gewährleisten

Schnittstellen:

- Rollen- und Rechte-Management
- Authentifizierung
- Autorisierung
- Aspekte des Architekturteams, wie bereits umgesetzte Konzepte zu Modularisierung, Versionierung und Verlinkung

8.1.4.2 Integrität

Der Begriff "Integrität" in diesem Kontext bezieht sich auf die Originalität der Daten, d.h. dass die betreffenden Daten vollständig und unverändert bleiben. Geeignete Integrations- und Schutzmaßnahmen sollen verhindern, dass die Daten beschädigt oder unerwartet verändert werden. Dabei spielt auch der Speicherort der Daten eine entscheidende Rolle. Daher ist ein Konzept unerlässlich, das die Integrität der Daten innerhalb einer VWS oder einer untergeordneten Informationsstruktur, wie einem Teilmodell, einer Sammlung von Teilmödellen oder einer Eigenschaft, gewährleisten kann. In diesem Zusammenhang sollte es jederzeit möglich sein festzustellen, ob die Daten in ihrem Originalzustand verfügbar sind, also frei von unbefugten Änderungen.

Aspekte

- Konzept zur Datenintegrität muss vorhanden sein
- Die entwickelte Lösung muss sich nahtlos in die Struktur der VWS integrieren lassen

Schnittstellen:

- Rollen und Rechtekonzepte
- Verschlüsselungstechnologien
- Digitale Signatur
- Aspekte des Architekturteams, wie bereits umgesetzte Konzepte zu Rückverfolgbarkeit und Verlinkung

8.1.4.3 Verfügbarkeit

Verfügbarkeit in diesem Kontext beschreibt die Fähigkeit auf Daten zuzugreifen, wann immer sie benötigt werden. Eine hohe Datenverfügbarkeit bedeutet, dass Daten zuverlässig und ohne Unterbrechung zugänglich sind. Dies erfordert oft redundante Systeme, Datensicherung und Notfallwiederherstellungspläne, um sicherzustellen, dass Daten auch bei Hardwarefehlern, Naturkatastrophen oder anderen unerwarteten Ereignissen verfügbar bleiben. Die Wahl der Mittel folgt dem Status Quo der geeigneten Technologien.

Aspekte:

- Funktionalität: Die VWS kann ihre Aufgaben erfüllen ohne unerwartete Ausfälle und Probleme zu haben
- Erreichbarkeit: Bewertung erfolgt durch Netzwerkverfügbarkeit und Konnektivität.
- Das Konzept kann durch beteiligte Systeme unterstützt/umgesetzt werden (BaSx, Catena-X, EDC etc.)

Schnittstellen

- Cloud und Synchronisierungsdienste
- Datensicherheit
- Datenintegrität
- Aspekte des Architekturteams, wie bereits umgesetzte Konzepte zu Modularisierung, Versionierung

8.1.5 Datensouveränität

Datensouveränität beschreibt die rechtlichen Rahmenbedingungen, die den Umgang mit Daten betreffen. In diesem Zusammenhang spielt der Datenschutz eine unverzichtbare Rolle. Die sogenannten Schutzziele sind die Eckpfeiler der Informationsschutzes. Die drei wichtigsten sind auch als CIA-Dreiklang bekannt: Vertraulichkeit (Confidentiality), Integrität (Integrity) und Verfügbarkeit (Availability).

Darüber hinaus impliziert Datensouveränität den selbstbestimmten Umgang mit den eigenen Daten und damit die volle Kontrolle über die Erhebung, Speicherung, Nutzung und Verarbeitung dieser Daten. Souveränität bezieht sich auf die Definition, wem die Daten gehören, wo die Daten gespeichert werden, wie sie genutzt werden und welche Regeln für den Missbrauch oder die Manipulation von Daten gelten. Maßgebliche Regelungen im Bereich der Datensouveränität sind dabei nationale und internationale Datenschutzbestimmungen wie die Datenschutz-Grundverordnung (DSGVO) und das Datenschutzgesetz.

Aspekte

- Konzeption eines Modells für die Speicherung von Daten auf der Grundlage des Zustands der Daten, wobei zwischen gespeicherten und verwendeten Daten unterschieden wird. In der VWS-Umgebung ist es wichtig zu definieren, wo sich das Repository des VWS befinden wird
- Definition von Regeln, d. h. vertragliche Regelung des Umgangs mit Daten, die gemeinsam genutzt oder gespeichert werden
- Konzeption eines Modells zur Pseudonymisierung und Verschlüsselung der Daten, das die Sicherheit der Daten gewährleistet

Schnittstellen

- Datensicherheit
- Dezentral Repository
- Aspekte des Architekturteams, wie bereits umgesetzte Konzepte zu Versionierung und Rückverfolgbarkeit

8.1.6 Datenhaltung (Data Storage Policy) im Hinblick auf die gemeinsame Nutzung

Die Datenhaltung in der VWS ist ein zentraler Aspekt der Industrie 4.0 und bezieht sich auf die systematische Erfassung, Speicherung und Verwaltung von Daten. Sie ermöglicht die Integration von Daten

aus verschiedenen Quellen, gewährleistet deren Konsistenz und stellt sicher, dass sie für die relevanten Akteure zugänglich sind.

In der VWS dient die Datenhaltung als Brücke zwischen der physischen und der digitalen Welt. Sie ermöglicht die digitale Repräsentation von physischen Objekten und Systemen, was für die Interoperabilität und Zusammenarbeit von Systemen entscheidend ist. Die umfangreichen Möglichkeiten der Datenhaltung in der VWS führen zu einer Reihe von technischen und technologischen Anforderungen.

Das Ziel der Datenspeicherung ist die Speicherung, der Abruf und die Nutzung von Informationen in einer sicheren Art und Weise entsprechend dem Bedarf des Nutzers. In diesem Zusammenhang ist es notwendig, die Speicherung von VWS-Informationen vor Ort oder in der Cloud zu vergleichen.

Zum einen erfordert die effektive Speicherung und Verwaltung von Daten robuste Datenbanktechnologien und -infrastrukturen. Diese müssen in der Lage sein, große Mengen an Daten zu verarbeiten und gleichzeitig hohe Leistung, Sicherheit und Zuverlässigkeit zu gewährleisten. Gleichzeitig stellen Richtlinien zur Datenspeicherung einen wichtigen Bestandteil der Data Governance dar.

Zum anderen erfordert die Integration von Daten aus verschiedenen Quellen und deren Bereitstellung für verschiedene Akteure fortschrittliche Technologien zur Datenintegration und -transformation. Hier kommen beispielsweise Technologien wie APIs, Microservices und Middleware zum Einsatz.

Aspekte:

- **Datenbanktechnologien und -infrastrukturen:** Die effektive Speicherung und Verwaltung von Daten erfordert robuste Datenbanktechnologien und -infrastrukturen. Diese Systeme müssen in der Lage sein, große Mengen an Daten zu verarbeiten und gleichzeitig hohe Leistung, Sicherheit und Zuverlässigkeit zu gewährleisten. Darüber hinaus müssen sie skalierbar sein, um mit dem Wachstum der Datenmengen Schritt zu halten, und sie müssen flexible Datenmodelle unterstützen, um eine Vielzahl von Datentypen und -strukturen zu verarbeiten
- **Datenintegrations- und Transformationstechnologien:** Die Integration von Daten aus verschiedenen Quellen und deren Bereitstellung für verschiedene Akteure erfordert fortschrittliche Technologien zur Datenintegration und -transformation. Diese Technologien müssen in der Lage sein, Daten aus verschiedenen Formaten und Strukturen zu extrahieren, zu transformieren und zu laden (ETL-Prozesse), um eine einheitliche und konsistente Sicht auf die Daten zu gewährleisten. Beispiele für solche Technologien sind APIs, Microservices und Middleware
- **Datenstrukturierung und -standardisierung:** Die Daten in der VWS sollten in einer strukturierten und standardisierten Form vorliegen. Dies bedeutet, dass die Daten in einer Weise organisiert sein sollten, die eine konsistente Interpretation und Nutzung über verschiedene Systeme und Plattformen hinweg ermöglicht. Die Verwendung internationaler Standards wie OPC UA kann dabei helfen, ein hohes Maß an Interoperabilität zu gewährleisten
- **Datenqualität:** Die Daten müssen genau, aktuell und relevant sein. Unvollständige oder veraltete Daten können zu falschen Entscheidungen führen und die Effizienz der Prozesse beeinträchtigen. Daher sollte ein effektives Datenqualitätsmanagement implementiert werden, das die Genauigkeit, Vollständigkeit und Aktualität der Daten sicherstellt
- **Datensicherheit:** Die Sicherheit der Daten ist ein wichtiger Aspekt. Die Daten in der Verwaltungsschale sollten vor unbefugtem Zugriff geschützt sein. Dies kann durch verschiedene Mechanismen wie Verschlüsselung, Zugriffskontrollen und sichere Übertragungsprotokolle erreicht werden. Darüber hinaus sollte die Verwaltungsschale in der Lage sein, auf Sicherheitsvorfälle zu reagieren und geeignete Maßnahmen zur Behebung von Sicherheitslücken zu ergreifen. Die Daten in der VWS sollten sicher gespeichert und übertragen werden. Dies umfasst sowohl physische Sicherheitsmaßnahmen (z.B. sichere Server) als auch digitale Sicherheitsmaßnahmen (z.B. Verschlüsselung)
- **Datenschutz:** Im Einklang mit den Datenschutzgesetzen und -richtlinien müssen personenbezogene Daten geschützt und vertraulich behandelt werden. Die VWS sollte Mechanismen zur Einhaltung dieser Anforderungen implementieren, einschließlich der Gewährleistung der Einwilligung der betroffenen Personen, der Begrenzung der Datennutzung auf den angegebenen

Zweck und der Gewährleistung des Rechts auf Zugang, Berichtigung und Löschung. Datenschutz und Compliance: Die Dateninfrastruktur sollte die Einhaltung von Datenschutzgesetzen und -richtlinien gewährleisten. Dies umfasst Aspekte wie die sichere Speicherung und Übertragung von Daten, die Einhaltung von Zugriffskontrollen und die Implementierung von Datenschutzmaßnahmen wie Datenverschlüsselung und Anonymisierung. Standards und Frameworks wie ISO 27001 und GDPR bieten Richtlinien und Best Practices für den Datenschutz und die Compliance.

- **Datenlebenszyklusmanagement:** Die Verwaltungsschale sollte in der Lage sein, den gesamten Lebenszyklus der Daten zu verwalten, von der Erstellung und Nutzung bis zur Archivierung oder Löschung. Dies beinhaltet auch die Nachverfolgbarkeit und Historisierung der Daten, um Änderungen im Laufe der Zeit nachvollziehen zu können und die Einhaltung von Vorschriften zu gewährleisten
- **Datenzugänglichkeit und -nutzbarkeit:** Die Daten in der Verwaltungsschale sollten leicht zugänglich und nutzbar sein. Dies bedeutet, dass die Daten in einer für Menschen lesbaren und verständlichen Form präsentiert werden sollten. Gleichzeitig sollte die Verwaltungsschale APIs oder andere Schnittstellen bereitstellen, um den maschinellen Zugriff auf die Daten zu ermöglichen. Darüber hinaus sollte die Verwaltungsschale in der Lage sein, die Daten in einer Weise zu präsentieren, die die Entscheidungsfindung unterstützt, beispielsweise durch Visualisierungen oder aggregierte Berichte
- **Interoperabilität:** Die Dateninfrastruktur sollte den Austausch von Daten mit anderen Unternehmen ermöglichen. Hierfür könnten Technologien wie APIs, Web-Services und standardisierte Datenformate wie JSON oder XML verwendet werden. Darüber hinaus sollte die Dateninfrastruktur den Standard OPC UA [58] [18] unterstützen, der eine sichere und zuverlässige Kommunikation zwischen verschiedenen industriellen Geräten ermöglicht.
- **Data Governance:** Unter dem Aspekt einer Data Governance sind folgende Aspekte zu berücksichtigen und zu definieren:
 - Speicherort der Daten (ist vorgegeben beim jeweiligen Asset-Hersteller)
 - Dauer der Speicherung
 - Mechanismen zur Identifizierung der Versionen der Daten
 - Wann wird die Datenspeicherung in eine Datenarchivierung umgewandelt.

Definition of Done:

- Konzept unter Berücksichtigung der Data Storage Policy Anforderungen wurde erstellt
- Definition einer anwendbaren Dateninfrastruktur die Datenintegration und -transformation unterstützt

Schnittstellen:

- Architektur Team
- Single point of truth (SPoT)
- Integration von Daten mit VWS

8.1.7 Datenhaltungsformate und -infrastrukturen

Datenhaltungsformate und -infrastrukturen in der VWS beziehen sich auf die Art und Weise, wie Daten strukturiert, gespeichert und verwaltet werden. Sie sind entscheidend für die Effizienz und Effektivität der Datenverarbeitung. Datenhaltungsformate bestimmen, wie Daten repräsentiert und interpretiert werden. Sie können von einfachen Textdateien bis hin zu komplexen relationalen oder nicht-relationalen Datenbankstrukturen reichen. Datenformate wie KBL [5] und VEC [6] werden in der Leitungssatz-Branche verwendet. Im Kontext VWS4LS wurde das AASX-Format [59] eingeführt. Bei allen diesen Dateitypen handelt es sich um strukturierte XML-Formate. Im IDS-Kontext werden vor allem JSON-basierte Datenobjekte verwendet. Datenhaltungsinfrastrukturen beziehen sich auf die physischen und logischen Ressourcen, die zur Speicherung und Verwaltung von Daten verwendet werden. Sie umfassen Hardware-Ressourcen wie Server und Speichersysteme sowie Software-Ressourcen wie Datenbankma-

gementsysteme und Datenverarbeitungstools. So ist z.B. die Auswahl der geeigneten Datenbanktechnologie entscheidend für eine effiziente Speicherung der gemeinsam ausgetauschten Daten. Hier kommen sowohl [relationale Datenbanken](#) (z.B. MySQL, PostgreSQL) als auch [NoSQL-Datenbanken](#) (z.B. MongoDB, Cassandra) in Frage, je nach den spezifischen Anforderungen an die Datenstruktur und -abfrage. Beispielsweise eignen sich relationale Datenbanken für eine strukturierte, tabellenbasierte Datenhaltung, während für unstrukturierte oder semi-strukturierte Daten NoSQL-Datenbanken besser geeignet sind [60]. Zur Datenverarbeitung von Echtzeitdaten oder großen Datenmengen können Technologien wie [Apache Kafka](#) oder [Apache Hadoop](#) zum Einsatz kommen. Im Folgenden werden die relevanten Aspekte der Datenspeicherformate und -infrastrukturen erläutert:

- **Skalierbarkeit:** Die Dateninfrastruktur muss skalierbar sein, um den Anforderungen des Datenwachstums und der Nutzung gerecht zu werden. Skalierbare Cluster- oder Cloud-Datenbanken, wie z.B. [Amazon DynamoDB](#) oder [Google Cloud Spanner](#), bieten hier Lösungen, da sie eine nahezu unbegrenzte Skalierbarkeit und hohe Leistung bieten
- **Hochverfügbarkeit:** Die Dateninfrastruktur sollte hochverfügbar und ausfallsicher sein, um einen kontinuierlichen Betrieb zu gewährleisten. Technologien wie Redundanz, Replikation und automatisches Failover können hier eingesetzt werden, um die Verfügbarkeit und Zuverlässigkeit der Dateninfrastruktur zu erhöhen
- **Datensicherung und Wiederherstellung:** Es ist wichtig, regelmäßige Datensicherungen durchzuführen und Mechanismen für die schnelle Wiederherstellung von Daten im Falle eines Datenverlusts zu implementieren. Technologien wie Snapshots, Point-in-Time-Recovery und Backup-Lösungen können hier eingesetzt werden
- **Datenschutz und Compliance:** Die Dateninfrastruktur sollte die Einhaltung von Datenschutzgesetzen und -richtlinien gewährleisten. Dies umfasst Aspekte wie die sichere Speicherung und Übertragung von Daten, die Einhaltung von Zugriffskontrollen und die Implementierung von Datenschutzmaßnahmen wie Datenverschlüsselung und Anonymisierung

Diese Aspekte sind entscheidend für die effektive Implementierung einer Datenhaltungsinfrastruktur im Kontext der VWS. Es ist wichtig, dass diese Anforderungen bei der Auswahl und Implementierung der Dateninfrastruktur berücksichtigt werden. Es ist auch wichtig zu beachten, dass die spezifischen Anforderungen je nach Anwendungsfeld und spezifischen Bedürfnissen des Unternehmens variieren können.

Aspekte:

- Definition der Haltung von Daten im Unternehmen
- Erweiterung der Datenformate
- Definition der optimalen Infrastrukturlösungen für die Datenhaltung

Schnittstellen:

- Architekturteam, Modularisierung

8.1.8 Anbindung an Catena-X

[Catena-X](#) als führender IDS im Automotive-Bereich soll für VWS4LS nutzbar gemacht werden. Daher muss ein systemübergreifende Gesamtkonzept definiert werden und die entsprechenden System-Schnittstellen bedient werden können.

Aspekte:

- Definition der Vorteile und Herausforderungen der von Catena-X angebotenen Schnittstellen für Ihren VW-Datenanschluss
- Kompatibilität der im Projekt verwendeten semantischen Modelle und der von Catena verwendeten Konzepte
- Definition der notwendigen Infrastruktur zur Bereitstellung der erforderlichen Dienste im Rahmen des Datenaustausches auf der Ebene der Wertschöpfungskette

Schnittstellen:

- AAS-Discovery

- DT-Registry

8.1.9 Daten-Monetarisierung

Ziel ist die Entwicklung eines umfassenden Rahmens zur Konzeptualisierung, Implementierung und Validierung eines Modells zur Monetarisierung von Daten im Kontext der Struktur der VWS und seiner Anwendung in der Fertigungsindustrie. Das umfasst einen klaren und strukturierten Ansatz für die digitale Darstellung von Produkten, Prozessen und Ressourcen, den bidirektionalen Austausch von Daten sowie die damit verbundenen Herausforderungen im Bereich Datenmanagement und -sicherheit. Bestehende Monetarisierungsmodelle in anderen Bereichen sollen auf Ihre Anwendbarkeit auf das Konzept des AAS untersucht werden, bspw. Pay-per-Use oder datengestützte Dienstleistungen. Ein weiterer Schwerpunkt liegt auf der Identifizierung der Wertschöpfungskategorien, die durch die Monetarisierung von Daten erreicht werden können, wobei zwischen internem (innerhalb des Unternehmens) und externem (kundenorientierten) Mehrwert unterschieden wird. Dazu müssen klare und messbare wirtschaftliche Indikatoren festgelegt werden, um die direkten wirtschaftlichen Vorteile zu bewerten, wie bspw. Kostenreduzierung oder Produktivitätssteigerung. Ein System zur Festlegung von Zugriffsrechten bei gleichzeitiger Gewährleistung der Souveränität des Datenbesitzers muss definiert werden. Im Rahmen der Implementierungsstrategien sind konkrete Strategien zur Validierung des Monetarisierungsmodells auf Basis eines spezifischen Anwendungsfalls notwendig, um sicherzustellen, dass die Bedingungen des industriellen Umfelds repräsentativ sind. Dazu sollte in einem Pilotprojekt die Sammlung von Daten und die Analyse der Leistung des Modells unter realen Bedingungen stattfinden. Abschließend sollte eine umfassende Analyse der während des Piloten erzielten Ergebnisse durchgeführt werden, indem die erreichten wirtschaftlichen Vorteile mit den während der Konzeptionsphase festgelegten Indikatoren verglichen werden. Das Modell sollte basierend auf den in der Bewertung gewonnenen Erkenntnissen angepasst werden, um seine Robustheit und Anpassungsfähigkeit an verschiedene Kontexte innerhalb der Fertigungsindustrie sicherzustellen.

Diese Anforderung zielt darauf ab, einen systematischen Ansatz zur Konzeptualisierung, Entwicklung und Validierung eines Monetarisierungsmodells für Daten auf der Grundlage des AAS zu etablieren, das nicht nur direkte wirtschaftliche Vorteile bietet, sondern auch die digitale Transformation der Fertigungsindustrie fördert und Innovation sowie Wettbewerbsfähigkeit unterstützt.

Aspekte

- Adaption bereits etablierter Geschäftsmodellkonzepte im Rahmen der VWS
- Berücksichtigung von Datenzugriffskonzepten und deren Granularität im Rahmen der Monetarisierung von Informationen und Diensten

Schnittstellen

- Aspekte der Daten-Governance
- Datensicherheit

8.2 AP 7.2 - Fachliche Konzeption der Daten-Policy entlang der Wertkette

Im Rahmen des Arbeitspakets 7.2 wurden die relevanten Aspekte der DG, die für das VWS-Ökosystem benötigt werden, durch eine Literaturanalyse ermittelt. Ausgehend von diesen identifizierten Aspekten werden Zugangskontrolle, Rollen- und Rechtemanagement und Richtlinien für die Datenverwaltung im Detail behandelt. Darüber hinaus werden Lösungen für die Integration dieser konzeptionellen Ansätze in das aktuelle VWS-Metamodell diskutiert. Dieser Ansatz schafft die Grundlage für die Einführung von VWS in der Industrie und fördert standardisierte Praktiken für die gemeinsame Nutzung von Daten durch die Akteure der Industrie. Dieses Konzept wurde in zwei wissenschaftlichen Veröffentlichungen vorgestellt [61] [62].

Nachdem die Aspekte, die intern in den Unternehmen zu berücksichtigen sind, bestimmt und definiert wurden, wird zusätzlich eine Architektur für den externen Austausch von Informationen und Diensten rund um die Wertschöpfungskette auf der Grundlage des International Data Space (IDS) Ökosystems vorgestellt. Für den Automobilsektor wird [Catena-X](#) als IDS für den sicheren Austausch von Informatio-

nen und Diensten betrachtet und deshalb in diesem Arbeitspaket als Lösungsansatz genannt. Ergänzend wird eine Unterscheidung zwischen operativen und nicht-operativen Daten vorgeschlagen, um zum einen die Sicherheit besser zu gewährleisten und zum anderen eine bessere architektonische Trennung der beteiligten Systeme (VWS und IDS) zu realisieren.

8.2.1 Integration von Data-Governance-Aspekten in das Metamodell der VWS

8.2.1.1 Relevante Aspekte

Um die relevanten Aspekte der Daten-Governance für den Datenaustausch zwischen Herstellern im VWS-Ökosystem zu ermitteln, wurden eine Literaturrecherche unter Verwendung etablierter Datenbanken wie [Web of Science](#), [IEEE Explore](#), [ACM Digital Library](#) und [Semantic Scholar](#) sowie der [Plattform Industrie 4.0](#) durchgeführt. Zur Sicherstellung der Reproduzierbarkeit und Eingrenzung des Umfangs wurde eine Reihe mithilfe boolescher Operatoren verknüpfter Schlüsselwörter festgelegt:

„asset administration shell“ UND („data governance“ ODER „data policy“ ODER „data exchange“ ODER „access control“ ODER „access permission“ ODER „authorization“ ODER „authentication“ ODER „security“)

Da die frühesten Publikationen im AAS-Bereich mit diesen Schlüsselwörtern aus dem Jahr 2017 stammen, konzentriert sich diese Arbeit auf Veröffentlichungen aus diesem Zeitraum bis zum aktuellen Jahr 2023, in dem diese Untersuchung durchgeführt wird. Die systematische Überprüfung umfasste Schritte wie die Datenbanksuche, die Entfernung irrelevanter Veröffentlichungen und die Überprüfung der verbleibenden Arbeiten. Von den ursprünglich 50 identifizierten relevanten Publikationen wurden nach Verfeinerung und Vorauswahl 29 beibehalten. Nach einer detaillierten Analyse wurden weitere 17 Arbeiten ausgeschlossen, da sie das Thema der Daten-Governance nur oberflächlich behandelten.

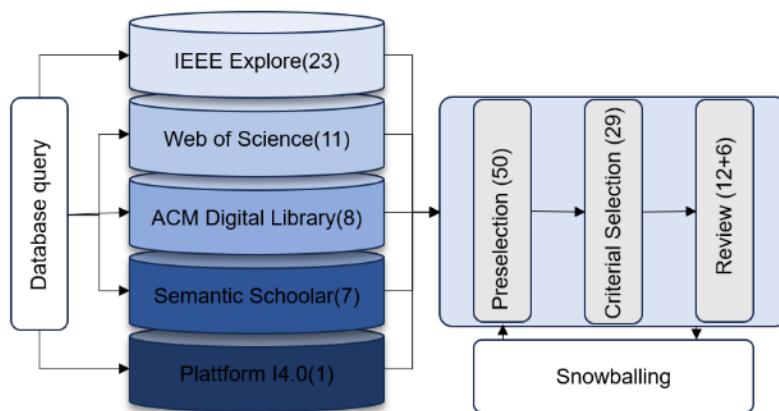


Abbildung 8-1:Überblick über den systematischen Prozess der Literaturrecherche

Abschließend wurden unter Anwendung der [Snowballing-Methode](#) sechs zusätzliche Publikationen identifiziert, was zu einem finalen Satz von 18 relevanten Arbeiten führte, die eingehend untersucht wurden, um Einblicke in die Behandlung der Daten-Governance-Aspekte im AAS-Ökosystem zu gewinnen. Abbildung 8-1 illustriert die angewandte Methodik ab, die Zahlen in Klammern repräsentieren die Anzahl der aufgefundenen Publikationen.

Die so identifizierten Publikationen wurden anschließend anhand von zwei Kategorien analysiert:

- Die erste Kategorie klassifiziert die Publikationen nach ihrem Ziel: Stand der Technik, Konzeption und Vorschlag. Die Publikationen zum Stand der Technik untersuchen theoretische Fortschritte in VWS und Daten-Governance (DG). Die Konzeptionen präsentieren konzeptionelle Rahmenwerke, während die Vorschläge praktische Lösungen auf der Grundlage der Literatur bieten.
- Die zweite Kategorie hebt die Aspekte der DG hervor, die in den Publikationen behandelt werden, wie Datensicherheit (15), Authentifizierung (15), Autorisierung (10), Vertraulichkeit (8), Rollen- und Rechteverwaltung (10), Datensouveränität (5) und Datenintegrität (9). Diese Publikationen vertiefen

die Themen Sicherheit beim Datenaustausch zwischen Herstellern, Authentifizierung, Zugriffskontrolle und Vertraulichkeit, unter anderem. Es werden verschiedene Methoden der Authentifizierung und Autorisierung, wie X.509 und OAuth2.0, diskutiert. Die Rollen- und Rechteverwaltung stellt sicher, dass Benutzer nur auf die relevanten Informationen zugreifen, während die Datensouveränität gewährleistet, dass die Teilnehmer die Kontrolle über ihre Daten behalten. Die Datenintegrität ist entscheidend, um die Qualität und Sicherheit der Daten während ihres gesamten Lebenszyklus zu gewährleisten. Für weitere Details wird auf die genannten Publikationen verwiesen [61] [62].

Die Analyse der Literatur zeigte eine erhebliche Lücke bei der mit DG verbundenen Aspekte im VWS-Ökosystem. Obwohl einige spezifische Aspekte der DG diskutiert wurden, oft auf theoretische Weise, fehlt es deutlich an konkreten Implementierungen. Daher wird auf Basis grundlegender Prinzipien vorgeschlagen, das DG-Konzept durch drei zentrale Aspekte für den Informationsaustausch in der Wertschöpfungskette zu behandeln:

Im Hinblick auf die **Datenrichtlinien** wird die Bedeutung der Wahrung von Vertraulichkeit, Datensouveränität, Sicherheit und Integrität betont. Derzeit gibt es jedoch kein spezifisches Konzept, das beschreibt, wie diese Prinzipien im bestehenden VWS-Metamodell strukturiert werden sollten. Daher wird vorgeschlagen, dass die Datenrichtlinien Anforderungen wie Datensicherheit, Interoperabilität, Datenintegrität, Datensouveränität, Zugangskontrolle und Rollen- und Rechteverwaltung umfassen. Die **Zugangskontrolle** wurde in den analysierten Publikationen ebenfalls nicht ausreichend behandelt. Um die Authentizität des Benutzers oder Systems, das Zugriff auf die VWS anfordert, sicherzustellen, wird empfohlen, die Datenrichtlinien durch Zugangskontrolle zu ergänzen und diese mit den Aspekten Authentifizierung und Autorisierung zu verbinden. Schließlich wird das **Rollen- und Rechtemanagement** als ein entscheidender Aspekt diskutiert, der den Zugriff auf die VWS-Informationen gemäß den definierten Rollen und Rechten garantiert und autorisiert.

8.2.1.2 Data-Governance-Aspekte im Metamodell der VWS

Um die identifizierten Aspekte einer Daten-Governance (DG), d.h. **Datenrichtlinien**, **Zugangskontrolle** sowie **Rollen- und Rechtemanagement** in das Ökosystem der VWS zu integrieren, werden zwei Optionen betrachtet.

Die erste Option integriert die DG-Aspekte in Form eines SM, basierend auf einem SM-Template der IDTA. Dies erfordert, solch ein DG-SM zu jeder Instanz des Daten-SM einzufügen, bevor mit dem Datenaustausch fortgefahren wird. Da DG ein Konzept ist, das alle SM umfasst, ist diese Idee anfällig für Fehler und unnötige Komplexität, da die DG-Aspekte, wie das Rollen- und Rechtemanagement, manuell auf jedes SM referenziert werden müsste.

Die zweite Option integriert die DG-Aspekte direkt in die Struktur der VWS, d.h. das Metamodell der VWS wird erweitert. Auf diese Weise würde die AAS die identifizierten DG-Aspekte inhärent enthalten und die Notwendigkeit einer zusätzlichen Integration beseitigen. Daher schlagen wir vor, das Metamodell der VWS mit den identifizierten DG-Aspekten zu erweitern. Dieser Ansatz bietet eine kohärentere und effizientere Lösung, welche Redundanzen vermeidet und den Integrationsprozess der DG-Aspekte in die VWS-Umgebung vereinfacht. Für die Integration wird das vorhandene Metamodell der VWS (DIN EN IEC 63278-1, Version 3) erweitert um die vorgeschlagenen Aspekte der DG.

Um ein Verständnis für die Erweiterung des bestehenden Metamodells im Hinblick auf die neuen Klassen und Attribute zu schaffen, werden zunächst die integrierten Superklassen (SC) **UserInformation** und **Authorizable** erläutert.

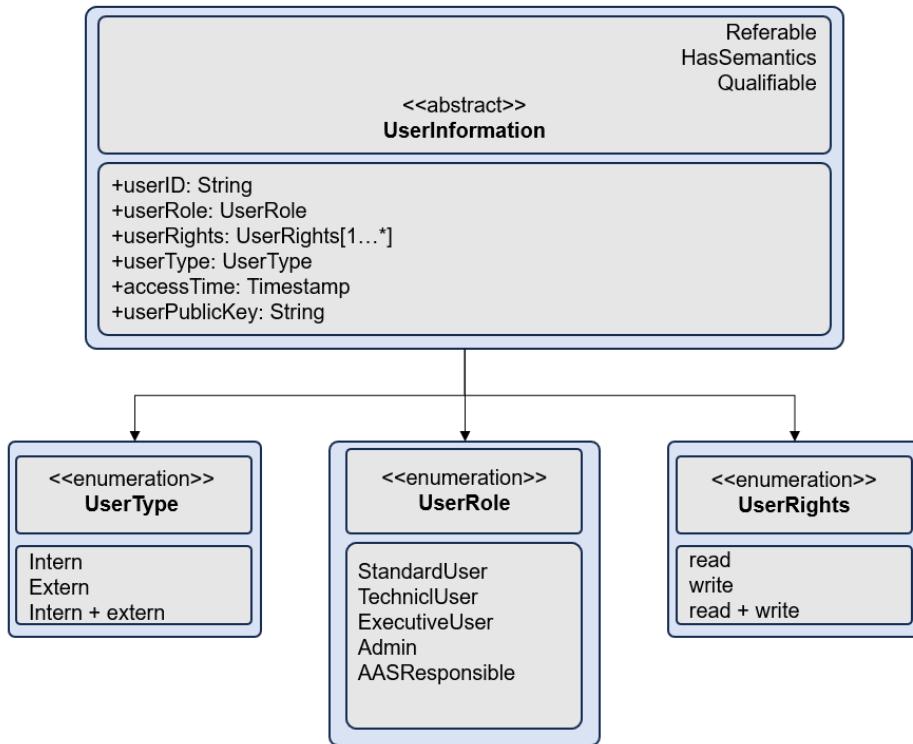


Abbildung 8-2: Überblick über die Superklasse „UserInformation“

Die SC *UserInformation* (Abbildung 8-2) kann verwendet werden, um auf die Benutzerinformationen in einem Element des Metamodells der VWS zu verweisen, sodass sie Informationen über den Benutzer enthalten kann, wie Identifikation, Rollen oder Rechte, wenn dies für den weiteren Prozess erforderlich ist. Die SC *UserInformation* erbt alle Attribute der Klassen *Referable*, *HasSemantics* und *Qualifiable* für eine konkrete Spezifikation, Referenz und weitere Informationen über den Ersteller des Benutzers. Die Informationen über den Benutzer setzen sich aus den Attributen *userID*, *userRole*, *userRights*, *userType*, *accessTime* und *userPublicKey* zusammen. Die *userID* enthält eine lokal eindeutige ID des Benutzers in Form einer Zeichenkette (*string*), so dass der Benutzer eindeutig identifiziert und im jeweiligen System referenziert werden kann. Das Attribut *userRole* referenziert die Rolle des Benutzers aus der Menge der definierten Rollen der Klasse *UserRole*, welche eine Aufzählung der definierten Benutzerrollen enthält. Damit kann der Benutzer als *StandardUser*, *TechnicalUser*, *ExecutiveUser*, *Administrator* oder *AASResponsible* ausgewiesen werden. Um die Kohärenz des Konzepts zu wahren, wird jedem Benutzer eine Rolle gemäß den Spezifikationen zugewiesen, da andernfalls die Zuordnung der mit der jeweiligen Rolle verbundenen Rechte zu Inkonsistenzen und Mehrdeutigkeiten führen kann. Analog repräsentiert das Attribut *userRights* die Rechte des Benutzers aus den definierten Rechten. *UserRights* enthält eine Aufzählung der definierten Benutzerrechte, die in Benutzerrollen verankert sind. Damit wird spezifiziert, ob der Benutzer *Lese-*, *Schreib-* oder *vollen Zugriff* auf das SM hat. Diese Rechte sind inkrementell strukturiert und bauen aufeinander auf, was die Rollen abgrenzt und sicherstellt, dass die Rechte mit der dem Benutzer zugewiesenen Rolle übereinstimmen. Das Attribut *userType* assoziiert jeden Benutzer mit einem Benutzertyp aus den vordefinierten Benutzertypen der Klasse *UserType*, die angibt, ob der Benutzer *intern*, *extern* oder sowohl *intern als auch extern* ist. Das Attribut *accessTime* enthält einen Zeitstempel mit dem Zeitpunkt des Zugriffs des Benutzers auf die Informationen oder Dienste des jeweiligen SM. Diese Zugriffszeit wird gespeichert und jedes Mal überschrieben, wenn der Benutzer authentifiziert wird und soll dazu dienen, den Zugriffszeitraum in einer spezifischen Implementierung zu verwalten und einzuschränken. Das letzte Attribut *userPublicKey* enthält den öffentlichen Schlüssel des Benutzers in Form eines *string*, der zur Authentifizierung mit einem Zertifikat verglichen wird.

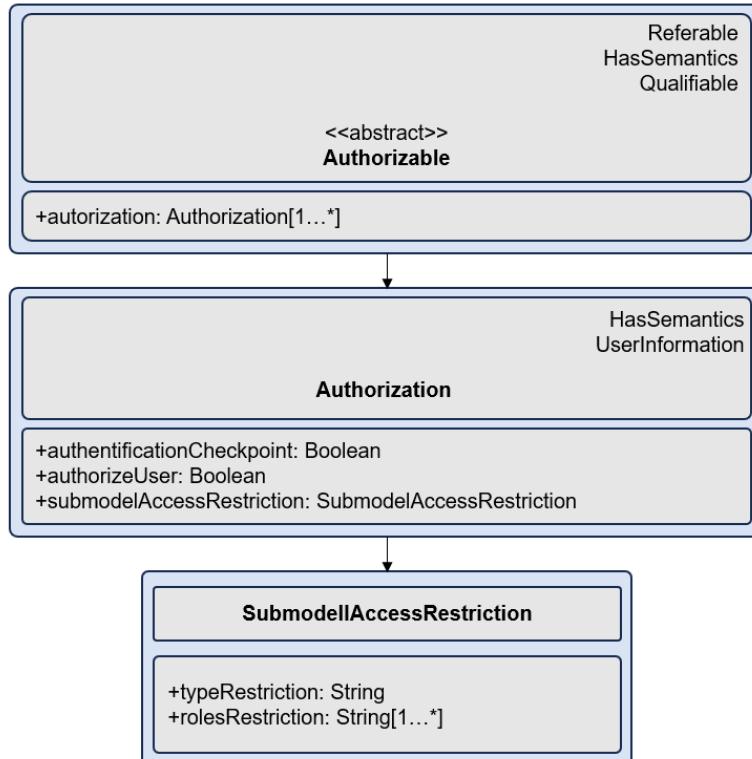


Abbildung 8-3: Überblick über die Superklasse „Authorizable“

Die Abbildung 8-3 zeigt einen Überblick über die erweiterte Superklasse *Authorizable*, welche von den SC *HasSemantics*, *Referable* und *Qualifiable* erbt. Insbesondere repräsentiert die Klasse *Authorization* die Autorisierung des Benutzers mit den Attributen *authenticationCheckpoint*, *authorizeUser* und *submodelAccessRestriction*. Das Attribut *authenticationCheckpoint* vom Typ *Boolean* repräsentiert, ob der Benutzer erfolgreich authentifiziert wurde. Ähnlich repräsentiert das Attribut *authorizeUser* ob die Eigenschaften des Benutzer mit den Autorisierungsbeschränkungen übereinstimmen. Die Klasse *Authorization* enthält das Attribut *submodelAccessRestriction*, um mögliche Beschränkungen der Autorisierung für den Datenzugriff zu definieren. Die Klasse *SubmodelAccessRestriction* definiert die Beschränkungen für die Autorisierung des Benutzers. Zu diesem Zweck enthält die Klasse die Attribute *typeRestriction* und *rolesRestriction*. Die Rechte sind, wie bereits erwähnt, mit der jeweiligen Rolle des Benutzers verbunden. Das Attribut *typeRestriction* vom Typ *string* repräsentiert Sichtbarkeit für das SM, die zugewiesenen Benutzertypen aus der zulässigen Menge von Benutzertypen sind in dieser *string* definiert. Analog zur Sichtbarkeit werden die zulässigen Rollen für den Datenzugriff über das Attribut *rolesRestriction* definiert.

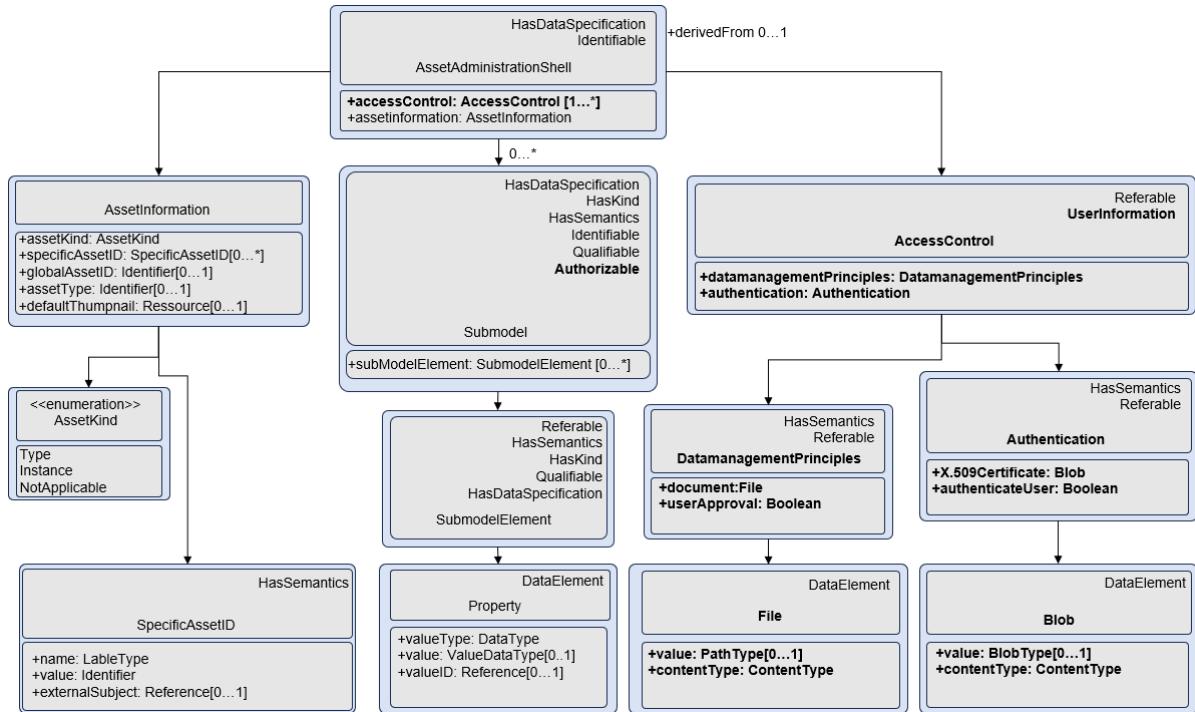


Abbildung 8-4: Übersicht über die Integration der identifizierten Data-Governance-Aspekte in das bestehende Metamodell der Verwaltungsschale (Erweiterungen fettgedruckt)

Die Abbildung 8-4 veranschaulicht die Integration einer DG in die VWS-Struktur. Das erweiterte Metamodell umfasst das Attribut `accessControl`, das Sicherheitsaspekte wie Authentifizierung und Datenrichtlinien für die VWS-Zugangskontrolle anspricht. Das Attribut `accessControl` gehört zur Klasse `AccessControl`, die `Authentifizierungsmethoden` und `Datenrichtlinien` verwaltet. Die Klasse `AccessControl` erbt von der SC `UserInformation`, die Informationen für die Benutzerauthentifizierung und die Authentifizierung von Kontrollpunkten bereitstellt. Darüber hinaus erbt `AccessControl` von der Klasse `Referable` für lokale Referenzen. Das Attribut `datamanagementPrinciples` bedeutet die Anbindung an erforderliche Datenrichtlinien und die Benutzerbestätigung und gehört zur Klasse `DatamanagementPrinciples`. Das Attribut `authentication` bezeichnet die Authentifizierungsmethode für den Datenzugriff und die Benutzerauthentifizierung und gehört zur Klasse `Authentication`. Die Klasse `DatamanagementPrinciples` umfasst das Attribut `document` vom Typ `File` für Datenrichtlinien und das Attribut `userApproval` für die Benutzerbestätigung. Die Attribute `value` und `contentType` der `File`-Klasse repräsentieren den Dokumentpfad und den Inhalt. Die `Authentication`-Klasse enthält `X.509Certificate` zur Benutzerauthentifizierung mit Zertifikaten und `authenticateUser` für die Benutzerauthentifizierung als `Boolean`. Die `Blob`-Klasse, die die Zugangskontrolle der VWS abschließt, umfasst die Attribute `value` und `contentType` für Blob-Instanzen. Schließlich wird die Klasse `SM` zur Benutzerautorisierung durch die Superklasse `Authorizable` erweitert. Die Klasse `SM` erbt von der Superklasse `Authorizable`, um Zugriffsbeschränkungen durchzusetzen, die eine individuelle Zuweisung basierend auf den Spezifikationen für Rollen- und Rechtemanagement ermöglichen. Es ist bemerkenswert, dass die Annahme der Datenklassifizierung nach Sensibilität integraler Bestandteil bleibt. `SM` kann mehrfach repliziert und erstellt werden, jeweils mit unterschiedlichen Zugriffsbeschränkungen und zulässiger Sichtbarkeit gemäß den Spezifikationen der SC `Authorizable`.

8.2.2 IDS-Architektur

Nachdem die Governance der Daten beim internen Datenaustausch sichergestellt wurde, wird eine High-Level-Architektur vorgestellt, die einen Weg zum Austausch von Informationen und Diensten über ein interoperables System aufzeigt.

Konzeptionell ist die entwickelte High-Level-Architektur an den Datenaustausch über ein IDS wie Catena-X angelehnt (Abbildung 8-5). Der EDC dient in diesem Kontext als Schnittstelle für den Datenaustausch über Catena-X, auch zwischen Unternehmen.

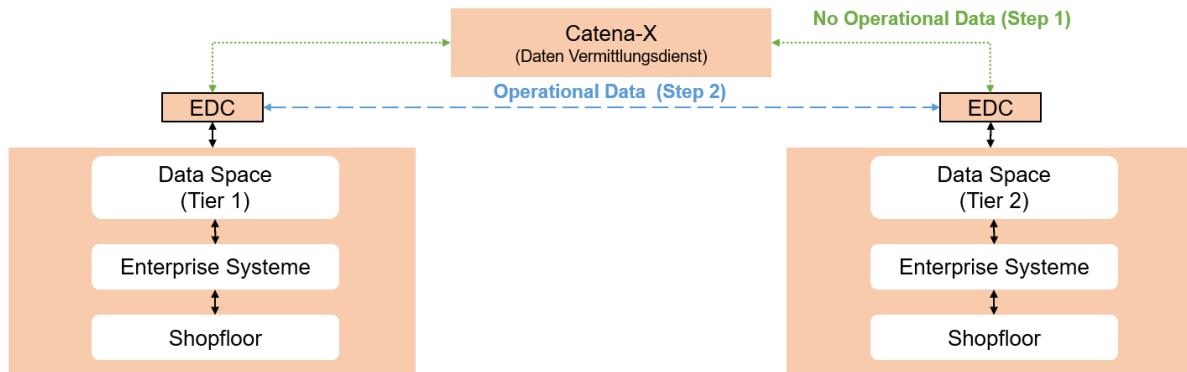


Abbildung 8-5: High-Level Architektur unternehmensübergreifender Datenaustauschs mittels IDS

Bei dem Datenaustausch über Catena-X handelt es sich ausschließlich um sog. „non-operational“ Daten, wie bspw. Kontaktdaten der jeweiligen Unternehmen, die über Catena-X an dem Datenaustausch partizipieren wollen, sowie Statusinformationen des Business-Workflows (siehe Abbildung 8-6).

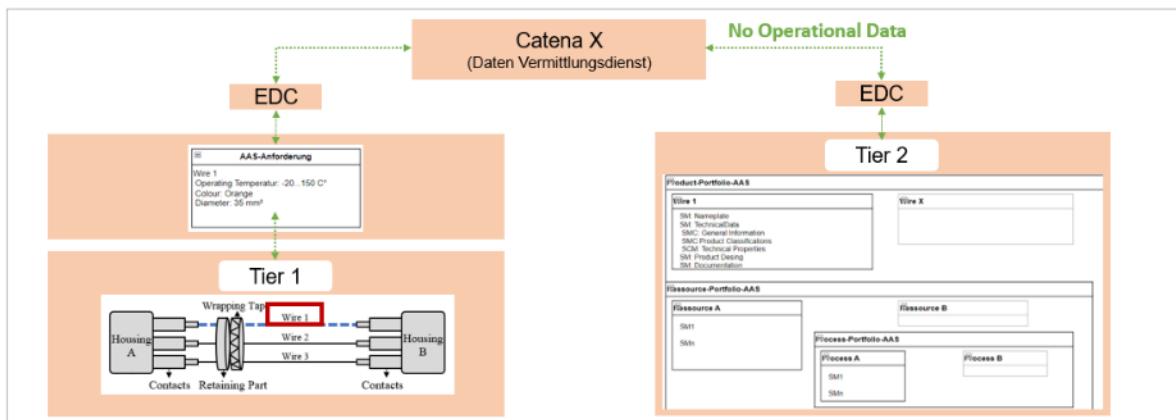


Abbildung 8-6: Übermittlung von non-operational data

Demgegenüber werden sog. „operational“ Daten, also die konkret produkt- oder prozessbezogenen technischen Informationen, nicht im IDS vorgehalten, sondern in den AAS-Repositories der beteiligten Stakeholder und transparent (d.h. als VWS) über den direkten Weg per EDC zwischen den Unternehmen ausgetauscht (siehe Abbildung 8-7).

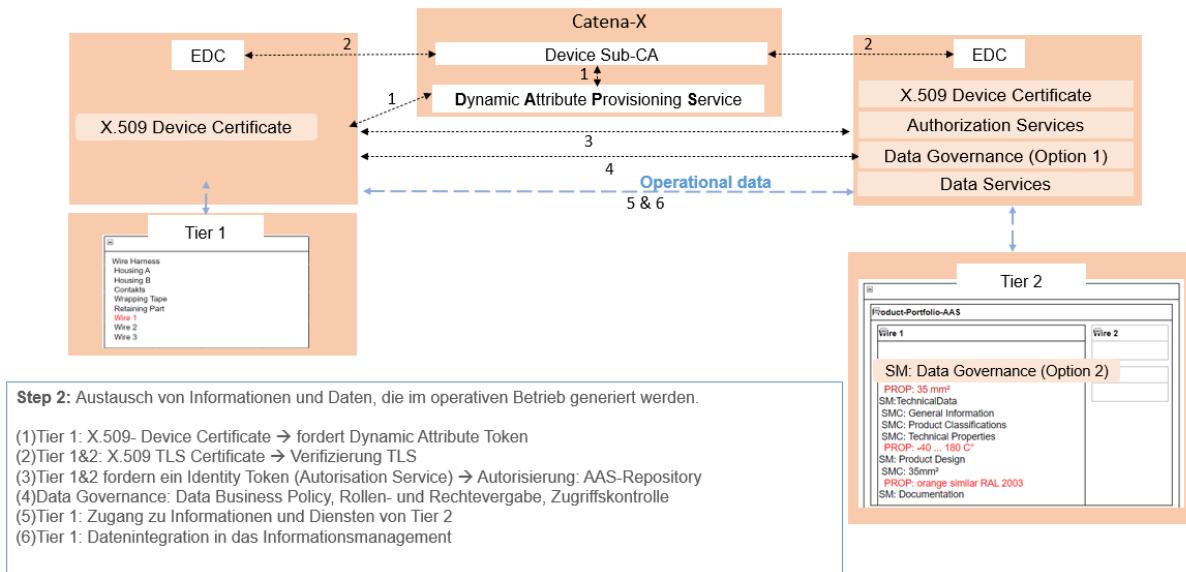


Abbildung 8-7: Übermittlung von operational data

Dafür muss sich zunächst auf einen Rahmenvertrag geeinigt werden (Contract Policy), der die Art und Weise der Datenhandhabung festlegt, wobei stets die Grundsätze der Datensicherheit und Datensouveränität zu wahren sind. Dieser Rahmenvertrag wird ebenfalls über den EDC ausgetauscht und bietet die Grundlage für den weiteren Datenaustausch.

Dieser beinhaltet ein Rollen- und Rechtekonzept, der die Zugriffsrechte auf Basis der mitgegebenen Rollen vorgibt, womit der Datenurheber zu jeder Zeit die Kontrolle darüber hat, auf welche Daten Dritte zugreifen dürfen und in welcher Form, also ob lesend oder manipulierend.

Für die Gewährleistung der Einhaltung des vereinbarten Rahmenvertrages ist das Security-Konzept des Datenurhebers verantwortlich, in dem das Rollen- und Rechtekonzept abgebildet sein muss und welches individuell auf die notwendige Granularität angepasst sein muss. So kann eine Form der Zugriffsrechte sein, dass nur der Zugriff auf einzelne Submodelle möglich ist. Um noch präziser, jedoch mit einem deutlich erhöhten Verwaltungsaufwand die Zugriffsrechte steuern zu können, muss das Rollen- und Rechtekonzept bis auf die Property-Ebene reichen, um dort einzelne Attribute der einzelnen Verwaltungsschalen verwalten zu können (Abbildung 8-8).

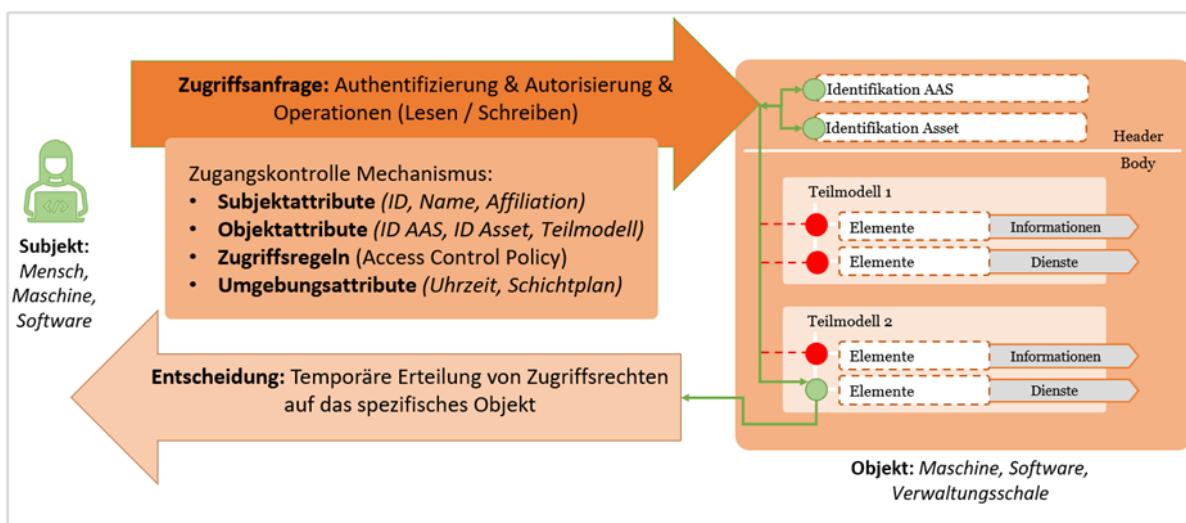


Abbildung 8-8: Konzept VWS-Security

8.2.3 Voraussetzungen für den International Data Space (IDS)

Um einen Use Case über den IDS durchführen zu können, müssen die folgenden Voraussetzungen erfüllt sein:

- Die Teilnehmer (OEM und Tier-x) müssen registrierte Mitglieder des IDS (z.B. Catena-X) sein
- Die vom IDS unterstützten Authentifizierungsmethoden müssen bedient werden können, um die Datensicherheit zu gewährleisten und Zugriff zu dem Datenraum zu bekommen. Jedes Unternehmen soll mit den von Catena X angebotenen Sicherheitsstufen einverstanden sein
- Tier-x müssen VWS-Repositories in einem mit der IDS-Architektur (z.B. Catena-X EDC) kompatiblen Format zur Verfügung haben, um die Datensouveränität jedes Stakeholders zu gewährleisten
- Submodellelementen können über die Konnektoren des IDS gefunden werden (z.B. Catena-X EDC). Tier-x müssen ein Portfolio von Produkten, Ressourcen und Prozessen (angebotene Dienste) anbieten, um am IDS-Ökosystem teilnehmen zu dürfen
- Es muss eine vom OEM als Datenurheber bereitgestellte Verwaltungsschale vorliegen, über die die jeweiligen „operational“ Daten ausgetauscht werden. Alle Teilprodukte, Prozesse und Ressourcen, die für die Erzeugung des Beispielproduktes benötigt werden, sind im Datenmodell der Verwaltungsschale abgebildet.
- Zudem müssen die Repositories, also Ablage- und Verwaltungsorte für die Verwaltungsschalen vorhanden sein, auf die der IDS im Rahmen des festgelegten Rollen- und Rechtekonzeptes zugreifen kann, um den „non operational“ B2B-Workflow abzuwickeln.

Als Basis für die nachfolgend beschriebenen Use Cases zeigt Abbildung 8-9 eine OEM-Definition des Kabelbaums, wobei zwei Arten von Variationen möglich sind. Außerdem werden die für die Herstellung dieses Bauteils erforderlichen Entwicklungs- und Produktionsprozesse beschrieben.

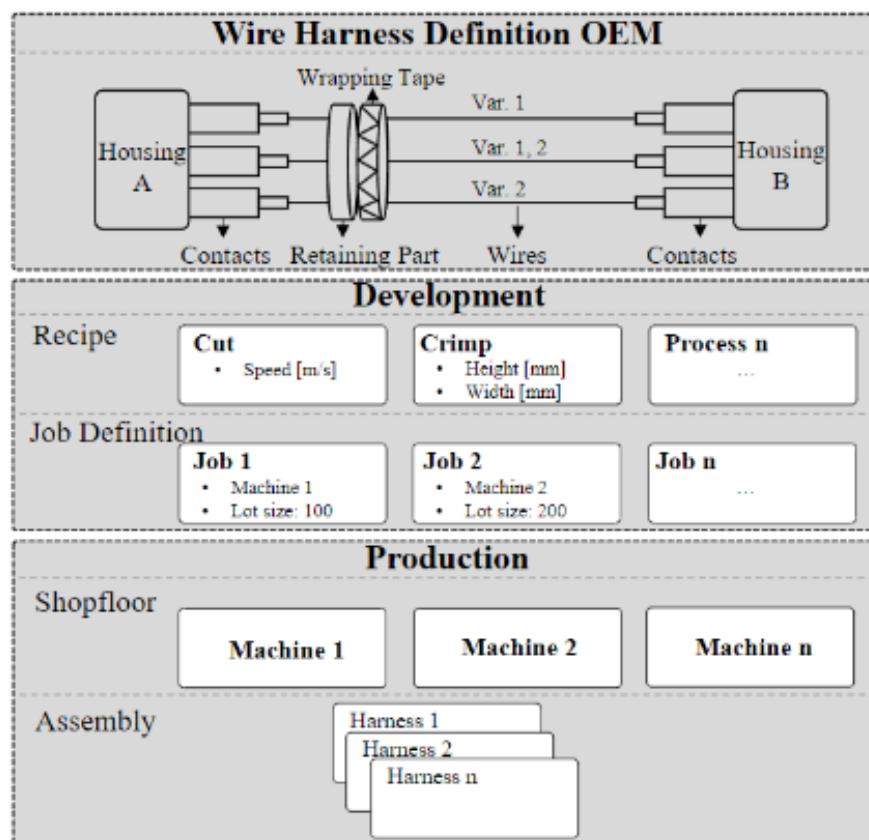


Abbildung 8-9: Definition des Kabelbaums seitens des OEM

8.2.4 Datenmonetarisierung

Das Geschäftsmodell eines Unternehmens stellt die Gesamtstrategie dar, die verschiedene Aspekte wie den Zielmarkt, das Wertversprechen und den Ansatz zur Geschäftseffizienz umfasst. Als wesentlicher Bestandteil beschreibt das Monetarisierungsmodell, wie Einnahmen generiert werden.

Im Kontext der Datenmonetarisierung geht es um mehr als nur den direkten Austausch von Daten gegen Geld. Vielmehr umfasst es die Verwertung von Erkenntnissen aus Daten, um monetäre Vorteile zu erzielen. Beispiele hierfür sind die Reduktion von Herstellungskosten und die Steigerung der Produktivität. Die Monetarisierung von Daten im erweiterten Sinne zielt darauf ab, messbare ökonomische Vorteile zu erzeugen, basierend auf der Verwendung von aggregierten Daten oder Daten-Services [63].

Es gibt zwei Hauptkategorien der Mehrwerte, die durch Datenmonetarisierung erzielt werden können: interne Mehrwerte, die innerhalb des Unternehmens generiert werden, und externe Mehrwerte, die über Unternehmensgrenzen hinausgehen und kundenzentrierte Strategien umfassen.

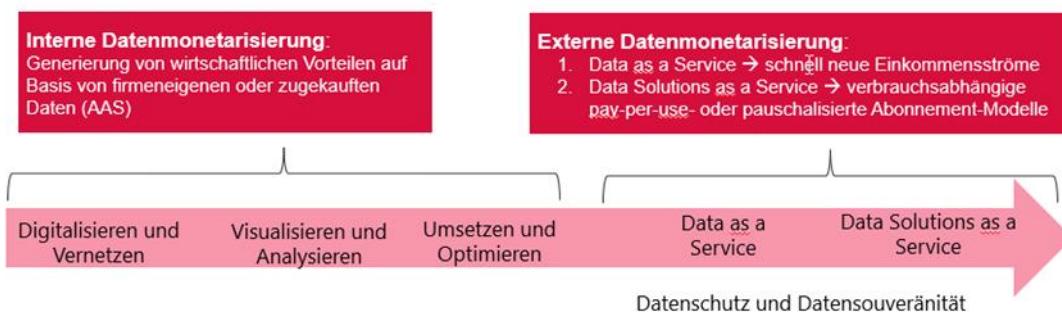


Abbildung 8-10: Framework zur Datenmonetarisierung

Die Abbildung 8-10 zur Datenmonetarisierung zeigt zwei Hauptkategorien: die interne und die externe Datenmonetarisierung.

Interne Datenmonetarisierung umfasst die Generierung von wirtschaftlichen Vorteilen auf Basis von firmeneigenen oder zugekauften Daten. Dieser Prozess gliedert sich in drei wesentliche Schritte:

- **Digitalisieren und Vernetzen:** Hierbei werden Daten digitalisiert und miteinander vernetzt, um eine umfassende Datenbasis zu schaffen.
- **Visualisieren und Analysieren:** In diesem Schritt werden die Daten visualisiert und analysiert, um wertvolle Erkenntnisse zu gewinnen.
- **Umsetzen und Optimieren:** Schließlich werden die gewonnenen Erkenntnisse umgesetzt und Prozesse optimiert, um wirtschaftliche Vorteile zu realisieren.

Externe Datenmonetarisierung bezieht sich auf die Monetarisierung von Daten durch externe Dienstleistungen. Hierbei werden zwei Hauptstrategien unterschieden:

- **Data as a Service:** Diese Strategie ermöglicht schnelle neue Einkommensströme durch den Verkauf von Daten als Dienstleistung. Unternehmen generieren wertvolle Nebeninformationen neben den Kernprozessdaten. Diese Daten ermöglichen es, Produkte besser anzupassen, was zu erhöhtem Absatz und verbesserter Qualität führt.
- **Data Solutions as a Service:** Hierbei handelt es sich um verbrauchsabhängige pay-per-use- oder pauschalisierte Abonnement-Modelle, die auf Datenlösungen basieren. Die digitale Marktwirtschaft erhöht die Nachfrage nach bestimmten Datentypen. Das Management muss diese Nachfrage verstehen und datenbasierte Lösungen entwickeln, die Kunden zusätzlichen Mehrwert bieten. Externe Daten können über Data as a Service integriert werden, um den Service zu optimieren. Das Management muss entscheiden, wie dieser Mehrwert am besten an Kunden vermittelt wird, oft durch Abonnement-Modelle.

Ein wichtiger Aspekt bei der externen Datenmonetarisierung ist der Datenschutz und die Datensouveränität, um die Privatsphäre und die Kontrolle über die Daten zu gewährleisten.

8.2.4.1 Angebot von maßgeschneiderten Maschinen

Hierbei sollen, Produkt- und Prozessdaten aus der VWS genutzt werden, um Kunden individuelle Maschinen anzubieten. Dies hilft, Überdimensionierung zu vermeiden und die Maschinen genau auf die Bedürfnisse der Kunden abzustimmen, was zu effizienteren und kostengünstigeren Lösungen führt.

8.2.4.2 Diagnose: Benachrichtigung über Leitungssatz-Ausfälle

Hierbei wird der Digital Twin von Leitungssätzen genutzt zur detaillierten Fehlerlokalisierung und Zustandsüberwachung innerhalb der Produktionsphase. Dadurch kann der Austausch des gesamten Kabels vermieden werden, indem nur die betroffenen Teile ausgetauscht werden. Dies führt zu einer effizienteren Wartung und reduziert die Ausfallzeiten.

Folgend wurde Ansatz 1 Wartung als Dienstleistung basierend auf den Prinzipien des [Business Model Canvas](#)⁵³ betrachtet.

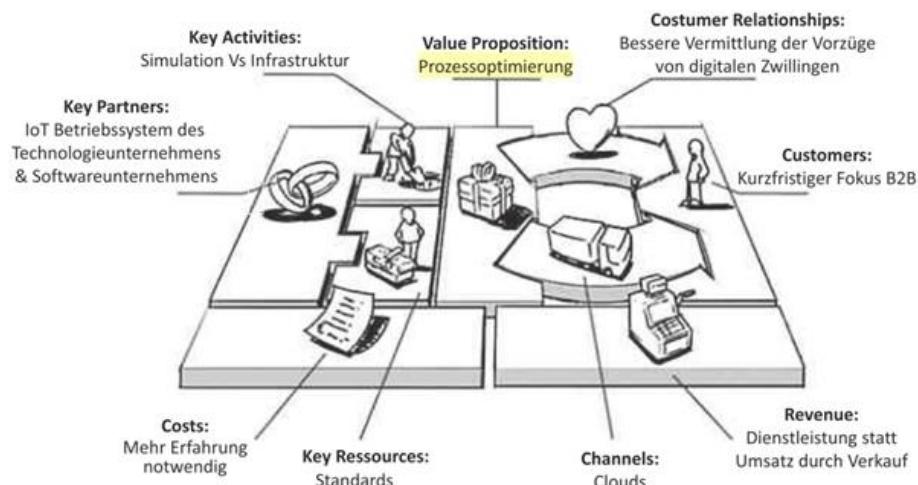


Abbildung 8-11: Einfluss digitaler Zwillinge auf den Business Model Canvas [64]

Wesentlicher Baustein dieses Konzeptes ist die Werterzeugung (value proposition), also die Erzeugung eines Nutzens durch das Produkt oder die Dienstleistung. Dabei ist entscheidend über welche Kanäle der Wert dem Kunden (customers) zu Verfügung gestellt wird. Ein gutes Kundenverhältnis (customer relationships) sichert dabei langfristige Einkommensströme (revenue) ab. Dem Einkommensstrom stehen die Kosten (costs) gegenüber, die für die Schlüsselaktivitäten (key activities), -ressourcen (key resources) und -partnerschaften (key partners) aufgebracht werden müssen.

8.2.4.3 Wartung als Dienstleistung

Diese Idee betrachtet die Wartung aus der Perspektive des Ressourcenanbieters. Es wird vorgeschlagen, individuelle Wartungspläne anzubieten, die auf Daten der VWS basieren. Dies ermöglicht eine maßgeschneiderte Wartung, die auf den spezifischen Bedarf und die Nutzung der Maschinen abgestimmt ist.

Das Monetarisierungskonzept "Wartung als Dienstleistung", basierend auf dem Konzept der «[Vorbeugenden Instandhaltung](#)»⁵⁴ (Proactive Maintenance) zielt darauf ab, durch die Bereitstellung von Wartungsservices feste Einnahmen zu generieren und gleichzeitig den Kunden einen klaren Mehrwert zu bieten. Dieses Konzept basiert auf dem Value Proposition Canvas, dass die Beziehung zwischen dem Kundensegment und dem Wertversprechen verdeutlicht.

Wertversprechen

Das zentrale Wertversprechen dieses Modells besteht in der Bereitstellung fixer Wartungskosten für Maschinenbetreiber. Dies ermöglicht den Kunden eine präzise Kalkulation ihrer Wartungsausgaben und reduziert gleichzeitig die Notwendigkeit eigener Personalkosten für Wartungsarbeiten.

Kundenbeziehungen

⁵³ https://de.wikipedia.org/wiki/Business_Model_Canvas

⁵⁴ https://de.wikipedia.org/wiki/Instandhaltung#Vorbeugende_Instandhaltung

Eine feste Bindung zwischen dem Anbieter und dem Nutzer der Dienstleistung ist essenziell. Durch den Aufbau von Vertrauen und langfristigen Beziehungen wird sichergestellt, dass die Kunden die Wartungsservices kontinuierlich in Anspruch nehmen.

Kanäle

Die Vermarktung und Bereitstellung der Wartungsdienstleistungen erfolgt über verschiedene Webschnittstellen und Verwaltungsschalen. Diese Kanäle ermöglichen einen effizienten Datenaustausch und die Verwaltung wartungsrelevanter Parameter.

Einnahmen

Die Einnahmen für den Wartungsanbieter sind verlässlich und basieren auf einer Servicegebühr pro Einheit oder Zeitperiode. Dies schafft eine stabile Einnahmequelle und ermöglicht eine langfristige Planung.

Ressourcen

Zu den notwendigen Ressourcen gehören Diagnosesysteme in den Maschinen, eine entsprechende Infrastruktur beim Serviceanbieter sowie Fachkräfte zur Auswertung der Diagnosedaten. Diese Ressourcen sind entscheidend für die Bereitstellung qualitativ hochwertiger Wartungsservices.

Aktivitäten

Die Hauptaktivitäten umfassen den Betrieb eines Servicecenters (SLA), die Schulung von Fachkräften sowie die kontinuierliche Auswertung der Verfügbarkeitsdaten. Diese Aktivitäten gewährleisten eine hohe Servicequalität und Kundenzufriedenheit.

Partner

Wichtige Partner in diesem Modell sind Wartungspartner und Maschinenbetreiber. Durch die Zusammenarbeit mit spezialisierten Wartungspartnern kann der Serviceanbieter seine Kapazitäten erweitern und eine höhere Servicequalität sicherstellen.

8.2.4.4 Use Case „Komponentensuche“

In diesem Use Case sucht Tier 1 nach einer bestimmten Komponente, in diesem Fall einer Leitung, die eine bereits festgelegte Spezifikation hat, zum Beispiel einen bestimmten Durchmesser oder eine bestimmte Farbe. Zu diesem Zweck stellt Tier 1 eine Anfrage zur Suche nach dieser Leitung und ihren Merkmalen auf einem der von verschiedenen Komponentenherstellern angebotenen Portfolios auf dem Markt des IDS.

Basierend auf diesen Annahmen wird die Suche nach einer Leitung angestoßen, die der geforderten Spezifikation entspricht. Wurde eine geeignete Leitung gefunden wird, erfolgt eine Benachrichtigung über den IDS. Der Anforderer (Tier-1) kann nun eine Angebotseinhaltung anstoßen. Nach Erfüllung der Sicherheitsanforderungen und der Angebotsannahme erhält Tier 1 das betreffende Produkt und seine digitale Repräsentation (VWS). Dazu ist es erforderlich, dass Tier 1 Zugang zu den zugehörigen VWS, ihren Teilmodellen und Merkmalen hat, um diese Informationen und Dienste, in die bereits im Ökosystem von Tier 1 vorhandenen VWS-Repositories aufnehmen und referenzieren zu können.

8.2.4.5 Use Case „Bestellung“

In diesem Use Case möchte der Leitungssatz (LS)-Entwickler (Tier 1) für die Fertigung des Leitungssatzes eine Leitung (Produkt + VWS) bei einem Komponentenhersteller (Tier 2) bestellen. Der LS-Entwickler (Tier 1) greift auf die VWS der Komponentenhersteller (Tier 2) zu, um die VWS der LS-Verbundkomponente in dem Tier 1-Umgebung zu erstellen.

Der Use Case ist durch eine aktive/parallele Arbeit der Beteiligten gekennzeichnet, d.h. beschreibt die kollaborative Arbeit (technisch den kollaborativen Zugriff auf Typ2-VWS) zwischen zwei Unternehmen während des Entwicklungsprozesses über einen IDS als Datenvermittlungsdienst. Im Folgenden werden beispielhaft die einzelnen Teilschritte des Datenaustauschs beschrieben.

8.2.4.5.1 Schritt 1: Austausch von non operational Data über einen IDS

Tier 1 erstellt eine Ausschreibung und stellt sie über den EDC in Catena-X bereit. Catena-X agiert als B2B-Vermittlungsdienst, siehe Abbildung 7-6. Über den Catena-X Marketplace Service wird nach potenziellen Kandidaten, die die Tier 1-Anforderungen erfüllen, gesucht und rückgemeldet. Von diesem Zeitpunkt an findet der in der Abbildung 7-6 dargestellte Verhandlungsprozess zwischen den potenziellen Unternehmen statt. Dadurch kann die Zahl der Kandidaten, die die Anforderungen erfüllen, verringert werden.

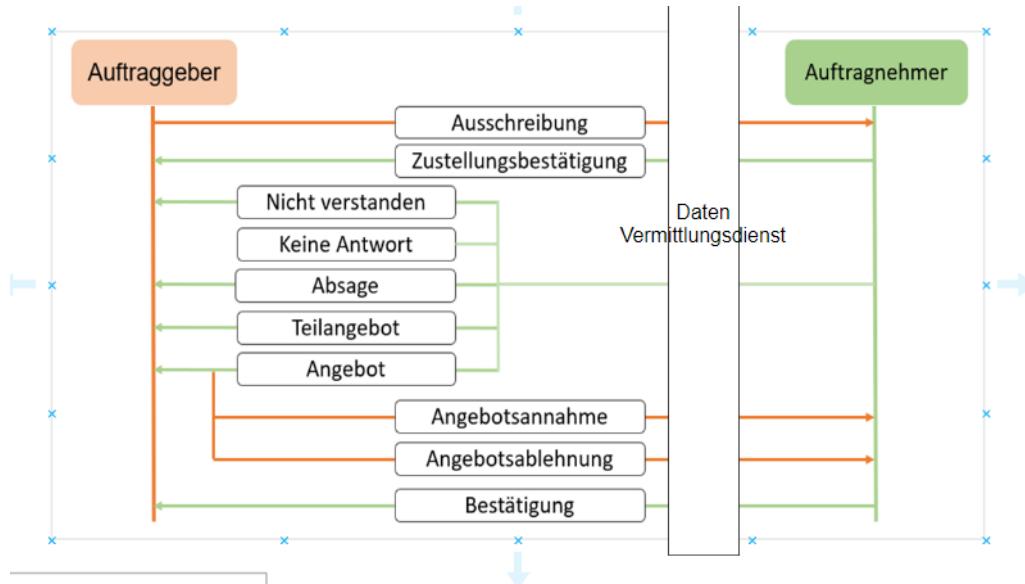


Abbildung 8-12: Ablauf des automatisierten Angebotsverhandlungsprozesses

Denkbar wäre hierfür die Anwendung des Submodell-Templates [IDTA 02051 „Purchase Request Notification“](#) [65], welches jedoch zum Zeitpunkt der Bearbeitung dieses APs noch nicht zur Verfügung stand.

8.2.4.5.2 Schritt 2: Austausch von operational Data

Nachdem der Prozess der Suche nach einem potenziellen Kandidaten, der die Anforderungen der Tier-1 erfüllt, abgeschlossen ist, erfolgt der Austausch von „operational data“, also die Informationen und Dienstleistungen, die in den internen AAS-Repositories der einzelnen Unternehmen vorhanden sind. In Bezug auf die VWS-Struktur wären dies die Informationen und Dienste, die in den Teilmodellen zu finden sind, sowie die Merkmale, die sie abbilden.

Identitätsmanagement mit geringen organisatorischen Anforderungen

In der Praxis stehen Unternehmen bislang vor Herausforderungen bzgl. eines übergreifenden (»federated«) Identity Managements, das oft an organisatorischen und technischen Hindernissen scheitert. International Data Spaces (IDS) bieten daher ein unternehmensübergreifendes Identitätsmanagement nach modernen Standards und mit geringen organisatorischen Anforderungen. Jeder Konnektor in den IDS verfügt über einen privaten Schlüssel mit einem dazugehörigen X.509-Zertifikat (»Gerätezertifikat«). Im Gegensatz zu herkömmlichen PKI-basierten Enterprise-IDM-Systemen, dienen diese statischen Zertifikate jedoch lediglich der Authentifizierung und werden nicht zum Austausch von Identitätsattributen verwendet. Letztere werden durch dynamische Tokens übertragen, die die Konnektoren von einem Attribut-Server beziehen. Dieser verwaltet Selbstbeschreibungen und attestierte (zertifizierte) Eigenschaften der Konnektoren und stellt je nach Bedarf Tokens über die erforderlichen Eigenschaften eines Konnektors aus.

Damit ist die Ausstellung der statischen X.509-Zertifikate von Identitätseigenschaften entkoppelt, die sich möglicherweise im Laufe der Zeit ändern können (bspw. durch eine Zertifizierung).

Daher wird die Verwendung des „Dynamic Attribute Provisioning Service“ (DAPS) -Konzepts für den operativen Datenaustausch in unserem Anwendungsfall vorgeschlagen. DAPS kann als Attributserver Auth2-Access-Tokens für International Data Spaces-Konnektoren ausstellen, die für den Zugriff auf die Dienste und Daten anderer Konnektoren benötigt werden. Der DAPS von Fraunhofer ist unter

<https://daps.aisec.fraunhofer.de> erreichbar und implementiert [RFC7523](#) JWT bearer client authentication für [OAuth2](#). Dieses Protokoll ermöglicht es den Konnektoren, sich mit ihrem [X.509](#)-Zertifikat am DAPS zu authentifizieren und im Gegenzug ein Access Token zu erhalten, mit dem sie auf andere Konnektoren zugreifen können.

Die Entscheidung über zulässige Zugriffe trifft hierbei nicht der DAPS, sondern immer der angefragte Konnektor selbst. Der DAPS verwaltet lediglich die IDS-Attribute der registrierten Konnektoren.

Wie in der Abbildung 8-7 dargestellt, legt Tier-1 zunächst sein Gerätezertifikat dem DAPS über einen Konnektor vor, um ein „Dynamisches Token-Attribut“ (DAT) zu erhalten. Mithilfe eines DAT kann Tier-1 auf Informationen zugreifen, die von anderen Konnektoren bereitgestellt werden. In diesem Fall sprechen wir vom Tier-2-Konnektor. In Bezug auf die Sicherheit wird ein Transport Layer Security (TLS) verwendet, das ein sicheres Protokoll für den Austausch von Informationen über das Internet bietet, siehe Abbildung Schritt 2. In Schritt vier wird schließlich der Zugang zu Informationen und Diensten gewährt, in unserem Anwendungsfall greift Tier-1 auf das AAS-Repository zu.

Der allgemeine Arbeitsablauf ist wie folgt:

- Ein anfordernder Konnektor legt dem DAPS sein Device Certificate vor, um ein dynamisches Attribut-Token (DAT) zu erhalten.
- Dieses DAT kann dann für den direkten Zugriff auf eine von einem anderen Konnektor bereitgestellte Ressource verwendet werden. Eine alternative Konfigurationsmöglichkeit wäre die Einführung eines lokalen oder anwendungsfallspezifischen Authentifizierungsdienstes, der ein benutzerdefiniertes Zugriffstoken bereitstellt. Dies ist völlig optional und daher grau markiert.
- Das entsprechende Token wird dann bei jeder Ressourcenzugriffsanfrage an einen Konnektor übergeben

8.3 AP 7.3 - Pilotierung und Erprobung: Erstellung Implementierungsguideline

Das AP 7.3 „Pilotierung und Erprobung“ wurde ursprünglich definiert, um die Architektur der Catena-X-Plattform zu untersuchen in Bezug auf Interoperabilität mit der Verwaltungsschale (Abbildung 8-13). Darüber hinaus sollte eine Implementierungsguideline bereitgestellt werden. Diese Aktivitäten wurden dann im Wesentlichen in TP8 abgewickelt und sind dort dokumentiert.

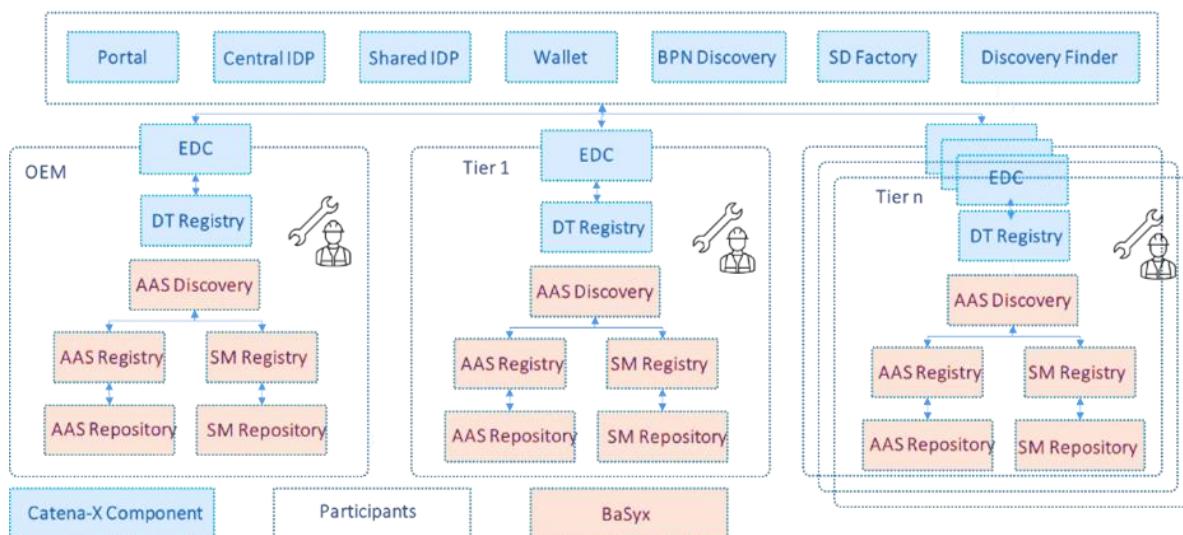


Abbildung 8-13: Demonstrator-Architektur mit Catena-X und BaSyx

8.4 Fazit

Im Rahmen von VWS4LS-TP7 wurden im AP 7.1 die grundlegenden Anforderungen an die Datengovernance erfasst und in den technischen Gesamtkontext verortet. Zudem wurden Anforderungen und Lösungsansätze zum Thema „Datenmonetarisierung“ zusammengestellt, die jedoch aus Ressourcengründen in der fachlichen Konzeption (AP 7.2) und Pilotierung (AP 7.3) nicht weiterverfolgt wurden.

Das Hauptziel von AP 7.2 war die Spezifizierung und Entwicklung eines Konzepts zur Integration der identifizierten Aspekte eines DG in die bestehende VWS-Struktur, um einen sicheren Datenaustausch zwischen den Herstellern zu gewährleisten. Um die Relevanz des Themas und das Hauptziel basierend auf bestehenden Erkenntnissen und Forschungen zu skizzieren, führten wir zunächst eine umfassende Literaturrecherche durch. Die Literaturrecherche ergab, dass die Anzahl der Publikationen zum Thema AAS in Kombination mit einem DG aufgrund der Neuheit des Themas sehr begrenzt ist. Als Ergebnis konnte keine Veröffentlichung identifiziert werden, die einen Vorschlag oder ein anfängliches Konzept zur Implementierung eines DG für den Datenaustausch zwischen Herstellern, die VWS verwenden, bot.

Es wurden jedoch drei potenziell relevante Aspekte für die Einhaltung des DG während des Datenaustauschs zwischen Herstellern identifiziert, die **Datenrichtlinien**, die **Zugangskontrolle** sowie das **Rollen- und Rechtemanagement**, welche die Grundlage für die Spezifikation der Integration in die Struktur der VWS bilden.

Auf diese Weise wird ein erster Vorschlag für zukünftige Forschungen in diesem Bereich unterbreitet. So sollte die Integration eines DG in die bestehende Struktur der VWS den Zugang zur VWS durch die Zugangskontrolle einschränken, bei der die Benutzer zunächst ein Authentifizierungs- und Autorisierungsverfahren durchlaufen müssen, bevor sie auf die in den Repositories der VWS gespeicherten Informationen und Dienste zugreifen können.

Es wurde aufgezeigt, dass das aktuelle Metamodell der VWS die Aspekte des DG noch nicht berücksichtigt. Dennoch bietet es Spezifikationen, Definitionen und geeignete Elemente zur Integration potenzieller Aspekte des DG. Das bestehende Metamodell umfasst Klassen, SC, Attribute, Datenelemente und Beziehungen, die die Hinzufügung neuer Elemente ermöglichen, um die potenzielle Aspekte des DG integrieren. Das umfasst auch die Einführung von neuen SC wie *UserInformation* und *Authorizable*, sowie die Erstellung der Klasse *AccessControl*.

Zudem wurde eine High-Level-Architektur entwickelt, um die gemeinsame Nutzung von Daten im Bereich der Leitungssatzherstellung über einen IDS zu ermöglichen. Es wurde analysiert, welche Anforderungen und Rahmenbedingungen der Integration einer Data Governance in das Konzept der VWS berücksichtigt werden müssen, sodass ein sicherer herstellerübergreifender Datenaustausch auf Basis der VWS gewährleistet werden soll. Dabei erfolgt eine eingehende Untersuchung des aktuellen herstellerübergreifenden Datenaustauschs entlang der Wertschöpfungskette.

Im AP 7.3 (bzw. AP 8.3) wurden dies dann konkret in einem Tractus-X basierten Demonstrator verprobt.

9 TP8 - Data Storage Policy, Sicherheit und Anbindung an Catena-X

9.1 AP 8.1 - Technische Anforderungsanalyse

Im AP 8.1 "Technische Anforderungsanalyse" sollten, ausgehend von den aktuellen Datenhaltungssituationen der Partner, Anforderungen für eine Datenhaltung im Hinblick auf die gemeinsame Nutzung entlang der Wertkette „Leitungssatz“ erarbeitet werden. Die Zielstellung des Arbeitspakets umfasste die Definition der notwendigen Anforderungen in Bezug zu Data Governance, Data Business Policy sowie Data Storage Policy. Da eine starke inhaltliche Überschneidung gegeben war, wurde das AP 8.1 bereits im Zuge des [AP 7.1 "Anforderungserhebung"](#) abgearbeitet und die Ergebnisse dort detailliert dokumentiert.

9.2 AP 8.2 - Technische Umsetzungskonzeption und Zielarchitektur

Im AP 8.2 „Technische Umsetzungskonzeption und Zielarchitektur“ sollten Lösungsansätze erforscht werden für die interoperable Nutzung von Daten der unterschiedlichen Akteure der Wertkette „Leitungssatz“ und über unterschiedliche Ablagestrukturen hinweg (On-Premises, Edge, Cloud). Als Leitbild dieser Lösungsbeschreibung wurde die übergeordnete Vision der Verwaltungsschale als Paradigma der Interoperabilität zwischen Produkt, Komponenten, Prozess und Produktion zugrunde gelegt und eine technische Anbindung mit Catena-X angestrebt. Inhaltlich wurde das AP 8.2 bereits in [AP 7.2 "Fachliche Konzeption der Daten-Policy entlang der Wertkette"](#) erarbeitet und die Ergebnisse dort detailliert dokumentiert.

9.3 AP 8.3 - Vorbereitung der Anbindung an Catena-X

Im AP 8.3 „Vorbereitung der Anbindung an Catena-X“ wurde die Architektur der Catena-X-Plattform untersucht in Bezug auf Interoperabilität mit der Verwaltungsschale und eine exemplarische Implementierung eines kollaborativen Datenaustauschs anhand des nachfolgend beschriebenen Use Case über Catena-X Komponenten umgesetzt, welche den bidirektionalen Datenaustausch zwischen den Akteuren anhand eines konkreten Szenarios prototypisch demonstriert. Mit diesen nachfolgend detailliert beschriebenen Aktivitäten des AP 8.3 wurde auch das [AP 7.3 "Pilotierung und Erprobung"](#) inhaltlich mit abgearbeitet.

9.3.1 Beispielprodukt: Teilleitungssatz

Für das Demonstrator-Anwendungsbeispiel wurde das vereinfachte Beispiel eines Leitungssatzes verwendet. Dieser besteht aus Stecker, Terminals und Leitungen. Zusätzlich soll die Möglichkeit gegeben sein, eine Auswahl zwischen zwei Varianten auszuwählen. Hierfür ist Variante 1, 2 Stecker, die Terminals und die beiden oberen Leitung in *Abbildung 9-1*, und Variante 2 enthält die beiden Stecker, die Terminals und die beiden unteren Leitungen.

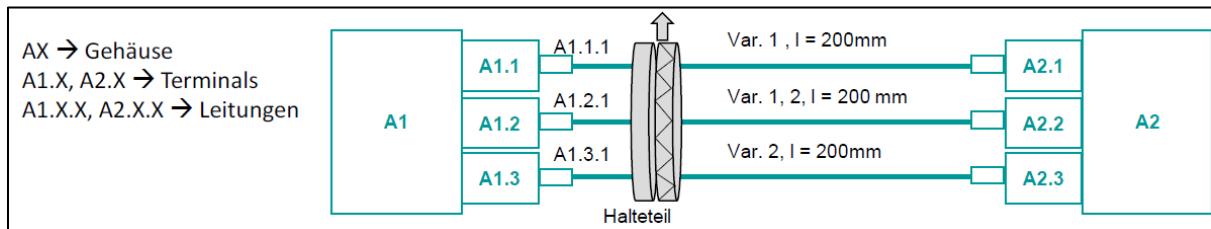


Abbildung 9-1: Vereinfachte Darstellung des Leitungssatzes

9.3.2 Das Anwendungsszenario

Ein OEM sendet einen Entwicklungsauftrag für eine 150%-Leitungssatzvariante an den Tier-1. In dem Entwicklungsauftrag ist eine Liste mit zugelassenen Suppliern mit den jeweiligen Komponenten enthalten. Der Tier-1 hat anschließend die Aufgabe alle notwendigen Informationen zu den einzelnen Komponenten von den Product-Ownern (Tier-2) zu aggregieren. Hierfür müssen die richtigen Endpunkte der Tier-2 identifiziert werden. Dazu werden die im Entwicklungsauftrag seitens des OEM hinterlegten Informationen verwendet. Ziel ist es, einen vollständigen aggregierten Digitalen Zwilling beim Tier-1 zu erzeugen, der alle Produktinformationen enthält, die von den einzelnen Tier-2 für Ihr Komponenten zu Verfügung gestellt werden.

In diesem Szenario wird eine Wertschöpfungskette des Leitungssatzes über drei Stufen abgebildet. Insgesamt werden beispielhaft fünf Teilnehmer betrachtet:

1. Ein OEM als Auftraggeber
2. Ein Tier 1 als Konfektionär und Auftragsempfänger und Data Consumer von Tier 2
3. Drei Tier 2 als Lieferanten von Tier 1 und Data Provider

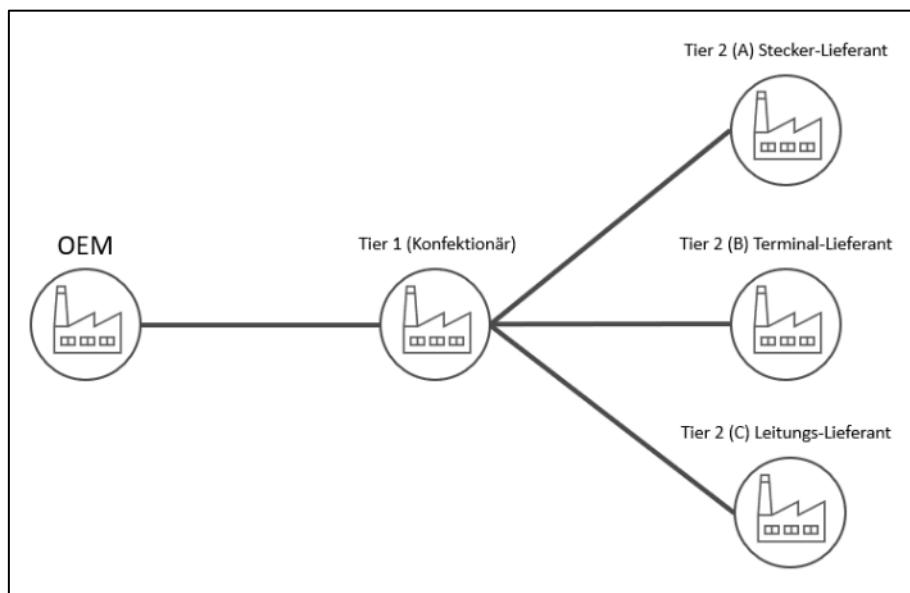


Abbildung 9-2: Beispielhafte Darstellung der Wertschöpfungskette eines Leitungssatzes

9.3.3 Ablauf des Szenarios

Die Aktionen der Akteure lassen sich grundlegend in zwei Kategorien aufteilen:

1. **Data Provisioning**
2. **Data Consumption**

Der Use Case wird in diesem Kapitel auf Business Ebene beschrieben. Die detaillierten Aktionen auf technischer Ebene zur Realisierung des Use Case-Demonstrators mit Catena-X sind im Kapitel „Umsetzung des Use Cases“ beschrieben.

9.3.3.1 OEM: Data Provisioning

Trigger für den Use Case ist die Anforderung des OEMs für einen Leitungssatz. Für unser vereinfachtes Szenario gehen wir davon aus, dass der OEM definiert, von welchem Zulieferer die Komponenten kommen sollen. Diese Anforderungen legt er in Form eines Submodels an und registriert einen Digitalen Zwilling in seiner DTR. Anschließend informiert er seinen Tier 1 (Konfektionär) über den Auftrag.

9.3.3.2 Tier 2: Data Provisioning

Die Tier 2 stellen die Informationen zu ihren Produkten in Form von Digitalen Zwillingen (AAS) bereit. Dazu erstellen sie die notwendigen Submodels. Diese werden dann zunächst in der Asset-Administration-Shell-Environment (AAS-Environment) hochgeladen. Danach werden in Catena-X entsprechende Digitale Zwillinge erstellt und in der DTR registriert.

9.3.3.3 Tier 1: Data Consumption (vom OEM)

Tier 1 greift auf die Submodelle des OEMs zu. In diesem Kontext wird davon ausgegangen, dass der Tier1 vom OEM eine Mitteilung erhält, dass ein Entwicklungsauftrag vorliegt. Er sichtet die Informationen innerhalb des Auftrags und kann anhand der enthaltenen Informationen (BPNs) die erforderlichen Endpunkte der EDCs von den entsprechenden Tier 2 herausfinden.

9.3.3.4 Tier 1: Data Consumption (von den Tier 2)

Mit den vom OEM definierten Zulieferern kann der Tier 1 die entsprechenden Endpunkte der Tier 2 herausfinden und auf die Daten zugreifen. Diese kann er im Anschluss sichten und entsprechende Komponenten für die Aggregation aussuchen.

9.3.3.5 Tier 1: Data Provisioning

Mit den Informationen der Tier 2 (welche Komponenten sie anbieten) kann der Tier 1 nun einen Leitungssatz mit den entsprechenden Komponenten darstellen. Er legt hierfür entsprechende Submodelle an und stellt die Verbindungen zu den ausgewählten Komponenten her. Daraufhin registriert er einen Digitalen Zwilling in seiner DTR mit den entsprechenden Submodels (detaillierte Beschreibung in Abschnitt 0 & 0).

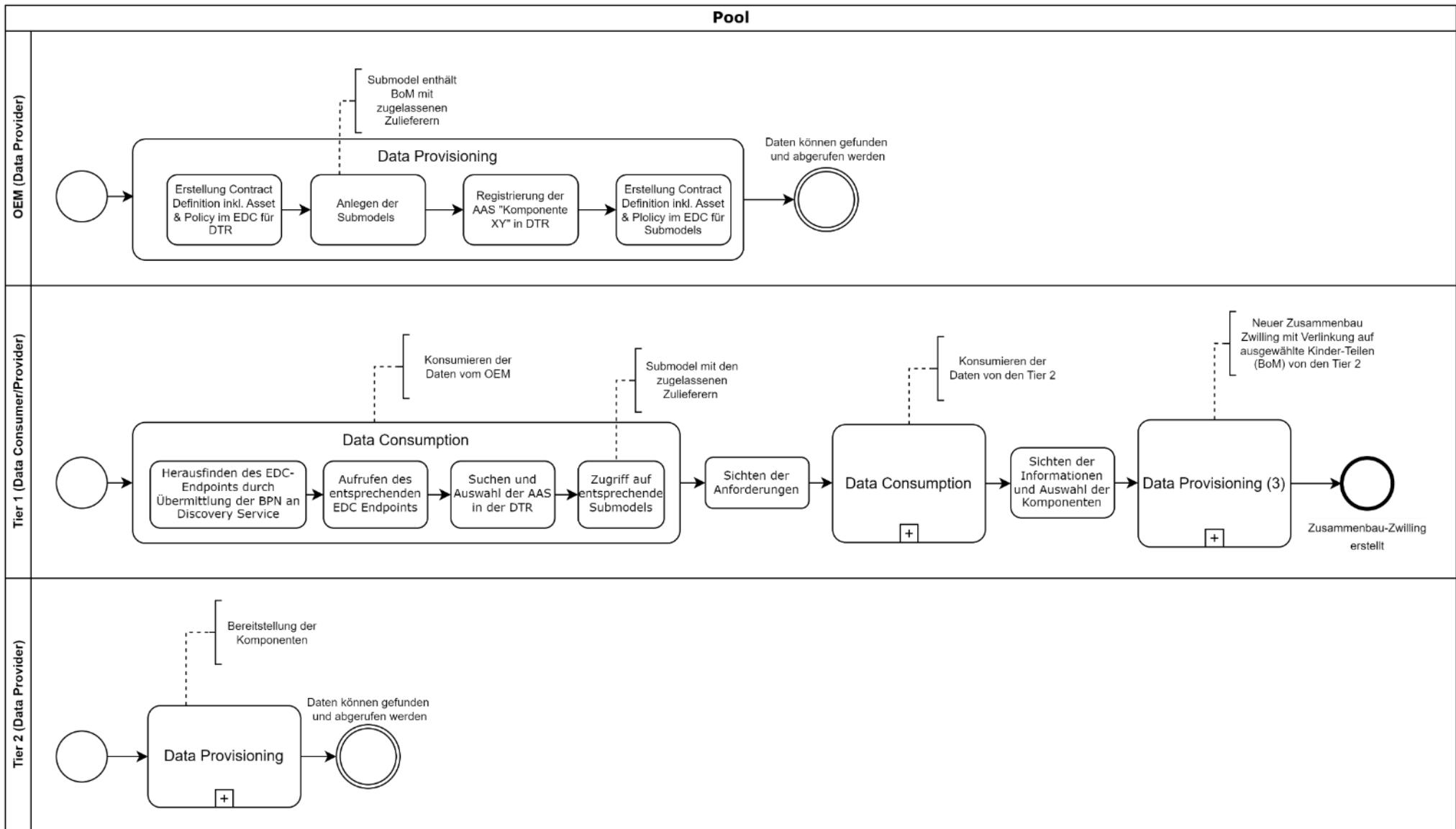


Abbildung 9-3: Prozessdiagramm Use Case

9.3.4 Aufbau der technischen Infrastruktur

Bitte beachten Sie, dass die beschriebenen Links lediglich intern (innerhalb des Netzwerks, der ARENA2036) nutzbar sind.

9.3.4.1 Benötigte Komponenten/Services für Data Provisioning

Wie in Abschnitt 9.3.3 beschrieben sind alle Akteure (OEM, Tier 1, Tier 2) Data Provider. Dementsprechend benötigen sie folgende Komponenten⁵⁵ die im Catena-X Operating Model⁵⁶ als Enablement Services klassifiziert werden:

| Komponente | Beschreibung | Referenzimplementierung | Standardisiert in |
|------------------------------|--|--|--|
| Konnektor | Ein Konnektor ist eine dezentrale Softwarekomponente für den Peer-to-Peer Datenaustausch zwischen zwei Datenökosystem-Teilnehmern. Er stellt die Datenendpunkte bereit und kann Nutzungsbedingungen mitgeben, interpretieren und durchsetzen. | Tractus-X Eclipse Dataspace Connector (EDC) ⁵⁷ | CX-0018 Dataspace Connectivity v3.0.0 |
| Digital Twin Registry | Mit einer Registry für Digitale Zwillinge veröffentlicht ein Anbieter digitale Darstellungen der Assets. Diese Daten sind in Teilmodellen gekapselt, die über Anweisungen zugänglich sind, die ein Verbraucher im Register für digitale Zwillinge (DTR) findet. Die DTR hält sich an die AssetAdministrationShellRegistryServiceSpecification ⁵⁸ und die DiscoveryServiceSpecification ⁵⁹ und enthält daher nur Metadaten. | Tractus-X Digital Twin Registry (DTR) ⁶⁰ | CX-0002 Digital Twins in Catena-X v2.2.0 |
| Submodel Server | Die Datenquelle, die einer Teilmenge der Submodel-API entspricht, wie sie in „AAS Teil-2: Application Programming Interfaces“ [48] definiert ist. Sie liefert die eigentlichen Daten, nach denen der Verbraucher sucht, über einen strukturierten Endpunkt, der durch die SubmodelServiceSpecification ⁶¹ definiert ist, und gemäß den im Repository für semantische Modelle ⁶² definierten strukturierten Datenformaten. | FA ³ ST-Framework, Eclipse BaSyx, AASX Server | CX-0002 Digital Twins in Catena-X v2.2.0 |

Tabelle 9-1: Benötigte Komponenten für Data Provisioning

Als Anforderung aus dem Projekt soll Eclipse BaSyx⁶³ als Quelle der Submodels (Submodel Server) verwendet werden. Die von jedem Teilnehmer deployten Komponenten werden in Tabelle 9-1 abgebildet.

⁵⁵ <https://eclipse-tractusx.github.io/docs-kits/kits/Digital%20Twin%20Kit/Operation%20View%20Digital%20Twin%20Kit#deployment>

⁵⁶ <https://catenax-ev.github.io/docs/next/operating-model/what-service-map>

⁵⁷ <https://github.com/eclipse-tractusx/tractusx-edc>

⁵⁸ https://app.swaggerhub.com/apis/Plattform_i40/AssetAdministrationShellRegistryServiceSpecification/V3.0_SSP-002#/

⁵⁹ https://app.swaggerhub.com/apis/Plattform_i40/DiscoveryServiceSpecification/V3.0.1_SSP-001

⁶⁰ <https://github.com/eclipse-tractusx/sldt-digital-twin-registry>⁶⁰

⁶¹ https://app.swaggerhub.com/apis/Plattform_i40/SubmodelServiceSpecification/V3.0_SSP-003

⁶² <https://github.com/eclipse-tractusx/sldt-semantic-models>

⁶³ <https://eclipse.dev/basyx>/⁶³ <https://eclipse.dev/basyx/> <https://eclipse.dev/basyx/>

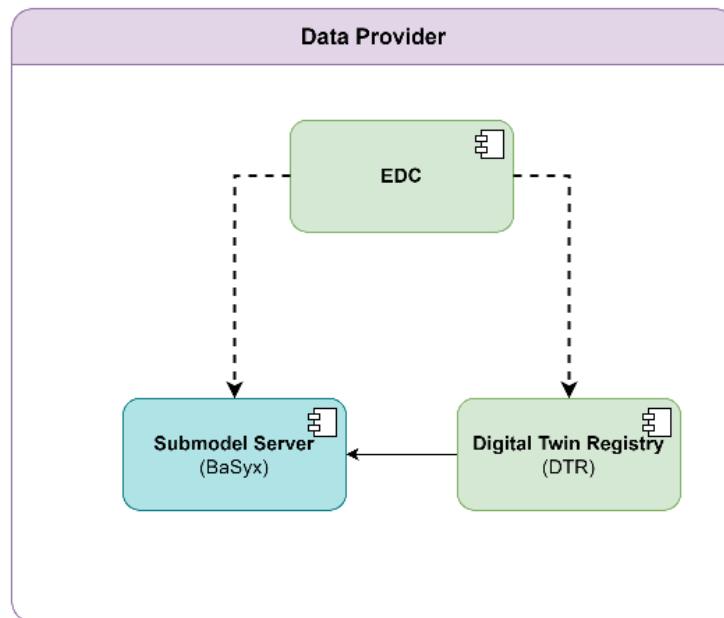


Abbildung 9-4: Benötigte Services eines Data Providers

9.3.4.2 Core Services und Onboarding Services

Für den beschriebenen Use Case muss ein komplettes Catena-X Datenökosystem bereitgestellt werden. Neben den in Abschnitt 9.3.4.1 beschriebenen Enablement Services müssen daher noch Core Services und Onboarding Services bereitgestellt werden (siehe Catena-X Operating Model⁶⁴). Diese werden üblicherweise von einer Catena-X Operating Company bereitgestellt (z.B. Cofinity-X). Für dieses Projekt wird die Operating Company in der ARENA-X-Umgebung simuliert und die entsprechenden Services eigens aufgesetzt.

Eine komplette Auflistung der für diesen Use Case genutzten Datenökosystem-Services sind in der folgenden Tabelle dargestellt:

| Service | Version |
|--|-------------------------------|
| <u>Tractus-X-Umbrella</u> | <u>0.17.0</u> |
| • <u>Portal</u> | 1.8.1 |
| • <u>Central-IDP</u> (Identity Provider) | 2.1.0 |
| • Shared-IDP (Identity Provider) | 2.1.0 |
| • <u>Discovery Finder</u> | 0.2.5 |
| • <u>Self Description (SD) - Factory</u> | 2.1.19 |
| • <u>BPN Discovery</u> | 0.2.6 |
| • <u>Managed Identity Wallet (MIW)</u> | 0.4.0 |
| • <u>Semantic-Hub</u> | 0.2.3 |
| • <u>TX-Data-Provider</u> (OEM) | 0.0.5 |
| • TX-Data-Provider (EDC Tier One) | 0.0.5 |
| • TX-Data-Provider (EDC Tier Two #1) | 0.0.5 |
| • TX-Data-Provider (EDC Tier Two #2) | 0.0.5 |
| • TX-Data-Provider (EDC Tier Two #3) | 0.0.5 |

⁶⁴ <https://catenax-ev.github.io/docs/next/operating-model/what-service-map>

| BaSyx-Services | - |
|-----------------------|----------------|
| • AAS-Environment | 2.0.0-SNAPSHOT |
| • AAS-Registry | 2.0.0-SNAPSHOT |
| • SM-Registry | 2.0.0-SNAPSHOT |
| • AAS-Discovery | 2.0.0-SNAPSHOT |
| • AAS-Web-UI | V2-240515 |

Tabelle 9-2: Zusätzliche Services (Core- und Onboarding Services)

9.3.4.3 Weitere Services

Neben den Datenökosystem-Services von Catena-X oder BaSyx wurden folgende weitere nennenswerte Services für den Betrieb benutzt. Diese befinden sich teilweise im Cluster und teilweise installiert auf der Ubuntu VM:

- Ein [nginx](#) (HTTP/S Webserver) in Version **1.18.0**
 - Ermöglicht den Externen Zugriff auf die Applikationen und leitet Web-Anfragen weiter
- Ein [MQTT](#): (Message-queue Service) in Version **2.0.15**
 - Wird von den BaSyx Komponenten für die Kommunikation zwischen Services genutzt
- Mehrer [MongoDBs](#) (Dokument-orientierte Datenbanken) in Version **5.0.10**
 - Wird in Catena-X und BaSyx für die Speicherung von Dokument-Daten verwendet
- Mehrere [HashiCorp Vaults](#) (Secret Management Services) in Version **1.10.3**
 - Speichert Secrets der einzelnen Services, um sich z.B. bei den Identity Provider Keycloaks zu identifizieren
- Mehrere [PostgreSQLs](#) (Relationale Datenbanken) in Version **15.1.0**
 - Wird von EDCs, Keycloaks, Catena-X Services, konkret z.B. Portal-Services, BPN-Discovery, Discovery-Finder und anderen benutzt, um relationale Daten zu speichern
- Mehrere [Keycloaks](#) (Identity and Access Management) in Version **22.0.3**
 - Sind für die Identity Provider Tätigkeiten im Catena-X Ökosystem nötig, um sowohl Participants als auch einzelne Services selbst zu Authentifizieren und Authorisieren

9.3.4.4 Architektur

9.3.4.4.1 Übersicht

Ausgehend von Abbildung 9-5 und mit den in den Kapiteln 9.3.4.1, 9.3.4.2 & 9.3.4.3 beschriebenen benötigten Komponenten ergibt sich folgendes Gesamtarchitekturbild:

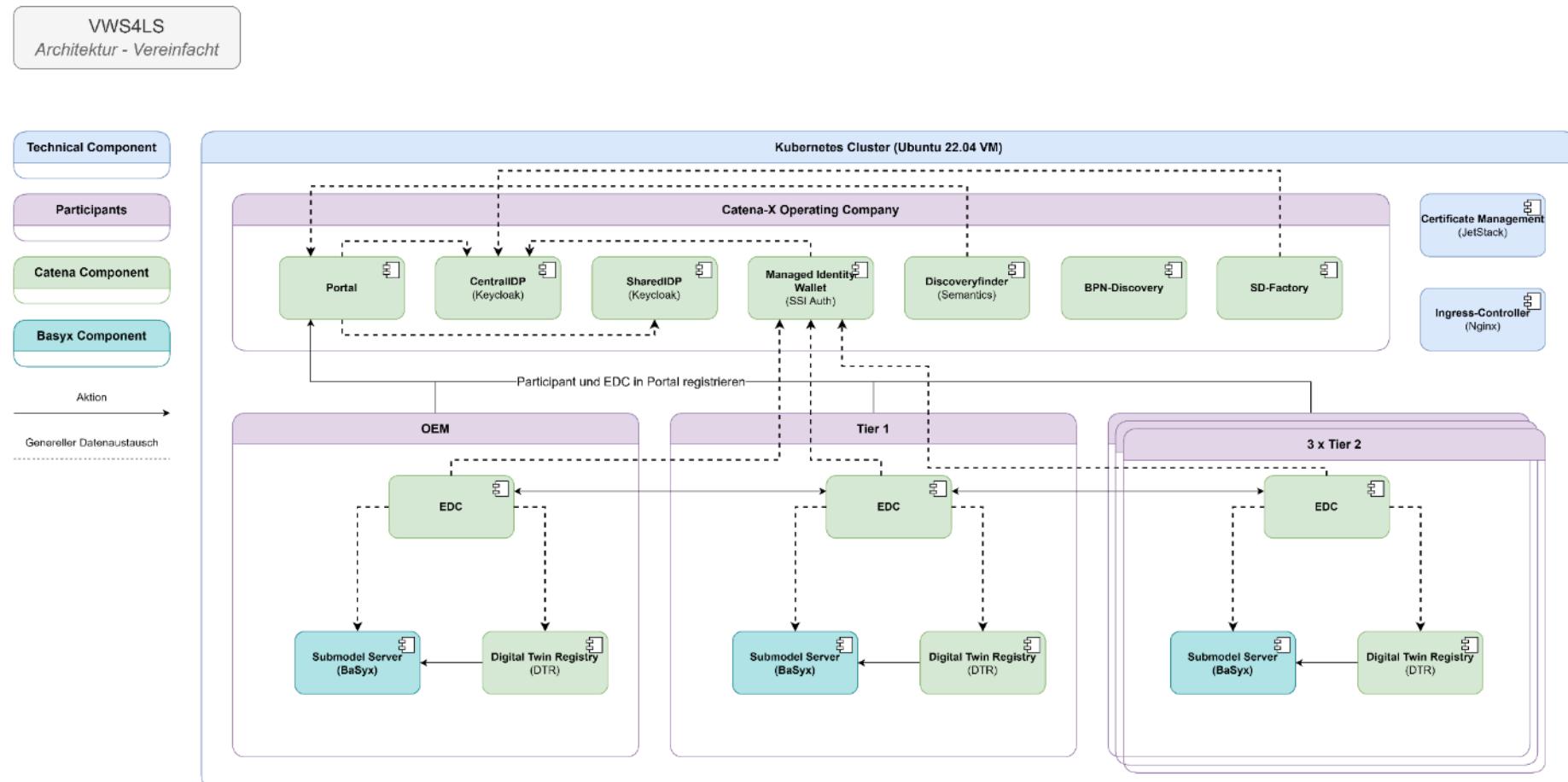


Abbildung 9-5: ARENA-X-VWS4LS-Architektur

9.3.4.4.2 Detail-Architektur

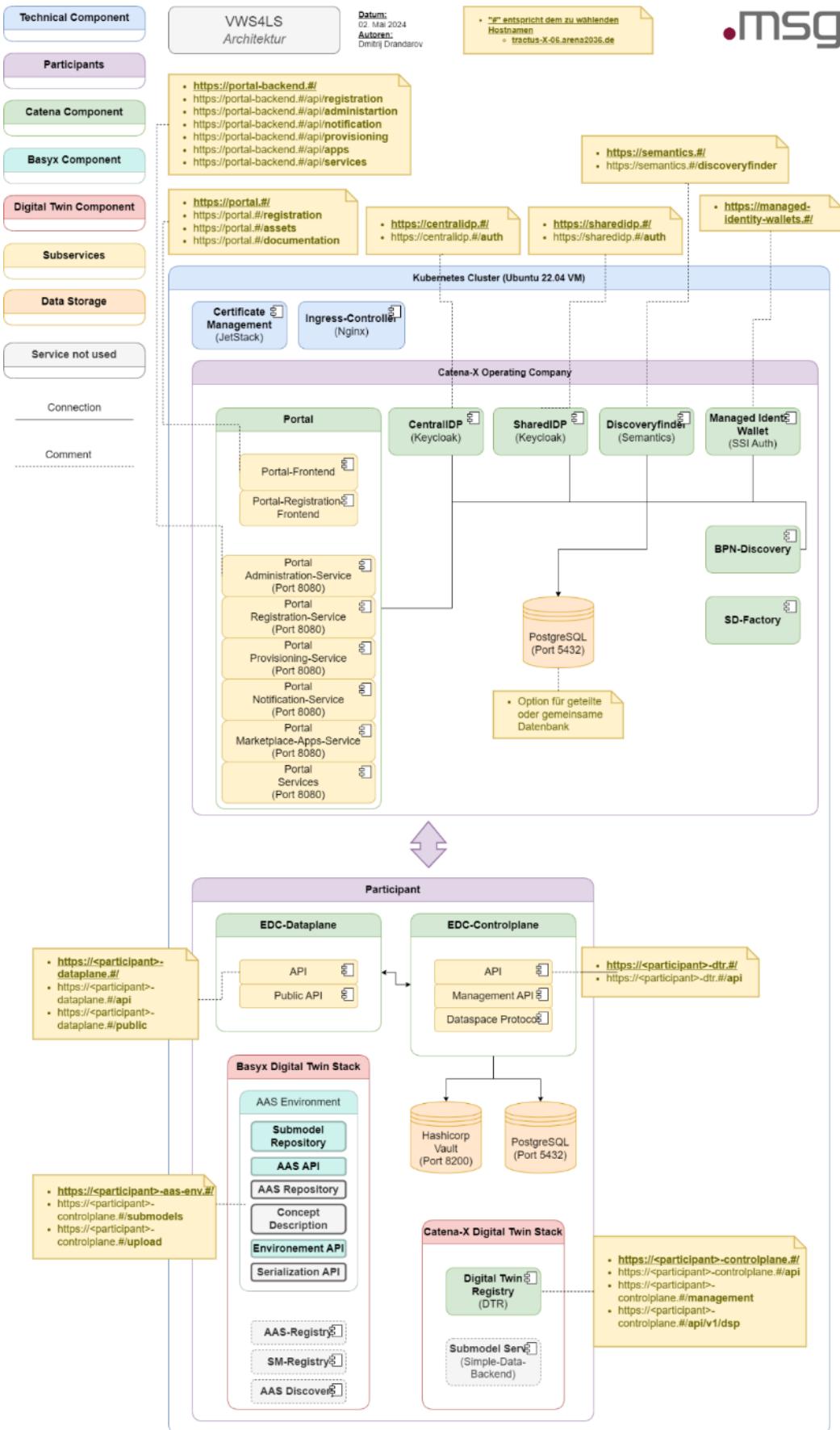


Abbildung 9-6: Detail-Architektur

9.3.4.5 Installation Server

Für die Installation wurde ein Ubuntu-Server mit der Version 20.04 unter dem Hostnamen tractus-x-06.arena2036.de mit Administrator-Rechten zur Verfügung gestellt. Die Software-Komponenten werden auf einem Minikube Kubernetes Cluster deployt. Das konkrete Deployment der Komponenten, sowie durchzuführende Aufgaben werden in den Abschnitten 0 und 0 beschrieben.

9.3.4.5.1 Entwickler-Setup

Es gibt mehrere Möglichkeiten für das Entwickler-Setup auf den VMs. Für das von uns genutzte, sind mehrere Voraussetzungen zu erfüllen:

- Auf dem Entwicklerrechner muss eine X-Server Umgebung installiert sein
- Auf dem Entwicklerrechner muss eine DISPLAY Environment-Variable definiert sein
- Man muss sich mit aktiviertem X-Server Support über ssh auf die VM verbinden. Bspw. mit folgender Konfiguration in der `~/.ssh/config`:

```
Host arena2036-06
  HostName tractus-X-06.arena2036.de
  User <user>
  ForwardX11 yes
  ForwardX11Trusted yes
```

- Eine .Xauthority Datei muss vorhanden sein. Diese kann mit einem Befehl wie `xauth generate 0.0` generiert werden und muss als Environment-Variable gesetzt sein:


```
export XAUTHORITY=$HOME/.Xauthority
```
- Wenn die Konfiguration passt, müsste es möglich sein den Browser über den Befehl `chromium` zu starten
- Dies kann in den folgenden Schritten nützlich sein, um bspw. von lokal auf Services zuzugreifen oder das Minikube Dashboard anzusprechen. Diese kann über den Befehl `minikube dashboard --port 12345` gestartet werden

9.3.4.5.2 Nginx

Die Installation der Webserver-Lösung „Nginx“ erfolgt mit dem Standard Ubuntu Package Manager „apt“:

```
sudo apt install nginx
```

Nach der Installation befindet sich die Nginx Installation inkl. der Binärdateien unter `/usr/sbin/nginx`. Die Konfigurationsdateien können hingegen unter `/etc/nginx` gefunden werden.

Für die Konfiguration des Nginx müssen `nginx.conf`-Dateien angelegt werden. Dies ist auf unterschiedliche Weise möglich. Für unser Deployment wurde die Konfiguration für die Services unter der Datei `/etc/nginx/conf.d/tractus-x` angelegt: Die entsprechende Konfiguration sieht folgendermaßen aus:

```
server {
    # register certificates
    ssl_certificate /etc/nginx/tractus-x-06.arena2036.de.crt;
    ssl_certificate_key /etc/nginx/tractus-x-06.arena2036.de.key;

    # login with password
    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/conf.d/.htpasswd;

    # forward incoming to minikube
    location / {
        proxy_set_header Host $http_host;
        proxy_pass http://192.168.49.2;
    }
}
```

9.3.4.5.2.1 Passwort-Schutz mit „Basic Authentication“ auf nginx

Um einen einfachen Schutz der Infrastruktur zu haben, wurde eine „Basic Authentication“ für Nginx erstellt.

Um verschlüsselte Passwörter zu definieren, muss zunächst eine Hilf-Applikation installiert werden. Dafür wird `htpasswd` verwendet. Für `htpasswd` ist eine Installation von `apache2-utils` erforderlich. Dies kann durch folgenden Befehle durchgeführt werden:

```
sudo apt-get install apache2-utils
sudo apt get install htpasswd
```

Für die Erstellung der initialen Passwort-Datei für einen Benutzer kann folgender Befehl genutzt werden:

```
sudo touch /etc/nginx/conf.d/.htpasswd
```

Nach Befehlseingabe kann das Passwort eingegeben werden:

```
sudo htpasswd -c .htpasswd <USER>
```

Weitere verschlüsselte Passwörter können generiert werden. Nach Befehlseingabe kann das Passwort eingegeben werden.

```
openssl passwd -apr1
Enter password: ...
```

Die Passwörter sind in der Datei `/etc/nginx/conf.d/.htpasswd` hinterlegt. Hier sind pro Zeile die Einträge hinterlegt, im Format `<USER>:<PASSWORD>`. Ein Beispiel für einen solchen Eintrag ist:

```
<USERNAME>:$apr1$itGWR13M$bIzdT/9XJNq7QEmnt4IyD0
```

9.3.4.5.2.2 Zertifikat

Das Zertifikat wird erstellt und in Nginx hinterlegt. Ein self-signed Zertifikat erfüllt nicht alle Security-Anforderungen, aber es ermöglicht den Zugriff über https für die Subdomains mit wenig Aufwand.

Das Zertifikat wird für die Domain `tractus-x-06.arena2036.de`, dessen Subdomains und dessen IP erstellt:

```
openssl req -x509 -newkey rsa:4096 -sha256 -days 3650 \
-nodes -keyout tractus-x-06.arena2036.de.key -out tractus-x-06.arena2036.de.crt
-subj "/CN=tractus-x-06.arena2036.de" \
-addext "subjectAltName=DNS:tractus-x-06.arena2036.de,DNS:*.tractus-x-06.arena2036.de,IP:141.70.214.186"
```

Es resultieren eine Private-Key-Datei `/etc/nginx/all.tractus-x-06.arena2036.de.key` und eine Zertifikatsdatei `/etc/nginx/all.tractus-x-06.arena2036.de.crt`. Diese werden in nginx registriert.

Zuletzt sollten die Zertifikate ebenfalls im Betriebssystem installiert werden. Dies passiert in der VM folgendermaßen:

```
sudo cp /etc/nginx/all.tractus-x-06.arena2036.de.crt /usr/local/share/ca-certificates/
sudo chmod 644 /usr/local/share/ca-certificates/all.tractus-x-06.arena2036.de.crt
sudo update-ca-certificates
```

Unter Windows können die Zertifikate per „Rechtsklick > „Install Certificate“ unter den „Trusted Root Certification Authorities Store“ aufgenommen werden.

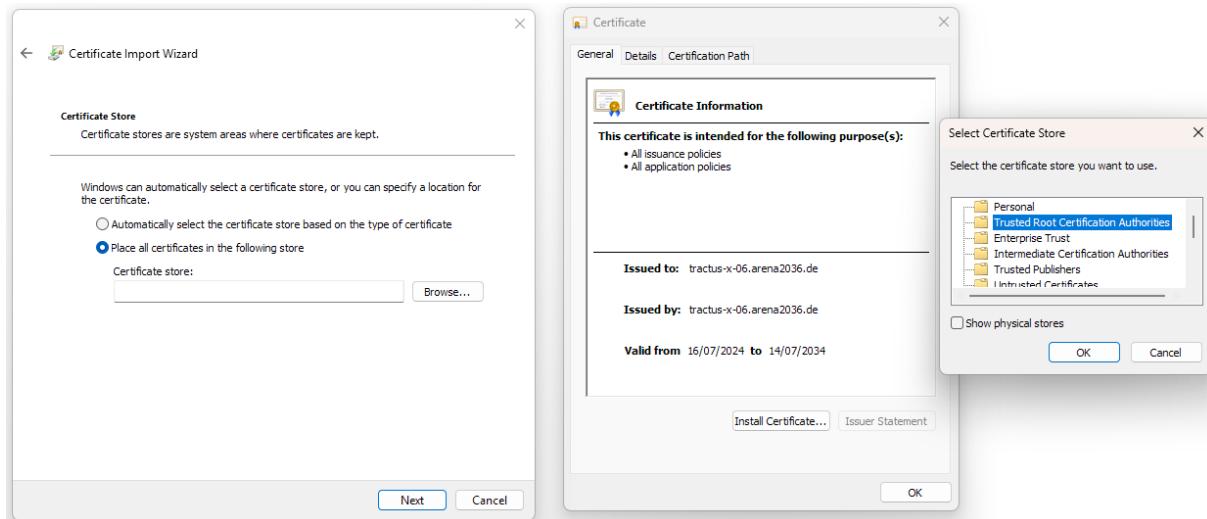


Abbildung 9-7: Erzeugte Security-Dateien zur Registrierung in nginx

9.3.4.5.2.3 Firewall-Freischaltung

Für den Zugriff auf dem Server mit http (Port 80) und https (Port 443) ist zusätzlich eine Ubuntu-Firewall-Freischaltung erforderlich. Diese kann mit folgendem Befehl durchgeführt werden:

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

Zusätzlich kann auch eine Regel für den Nginx erstellt werden, welche mit folgendem Befehl erlaubt werden kann:

```
sudo ufw allow "Nginx Full"
```

Den Status der Freischaltung kann man mit folgendem Befehl überprüfen:

```
sudo ufw status
```

Das Ergebnis kann folgendermaßen aussehen:

Das Ergebnis kann folgendermaßen aussehen:

```
Status: active
```

| To | Action | From |
|-----------------|--------|---------------|
| -- | ----- | ----- |
| 22/tcp | ALLOW | Anywhere |
| Nginx Full | ALLOW | Anywhere |
| 22/tcp (v6) | ALLOW | Anywhere (v6) |
| Nginx Full (v6) | ALLOW | Anywhere (v6) |

9.3.4.5.2.4 Hosts Dateien

Für den Zugriff auf die Komponenten im Minikube Cluster ist die Einrichtung von Redirects in der hosts-Datei erforderlich. Dies erfolgt in der /etc/hosts-Datei, nach dem Beispiel:

```
192.168.49.2    centralidp.tractus-x-06.arena2036.de
192.168.49.2    sharedidp.tractus-x-06.arena2036.de
192.168.49.2    portal.tractus-x-06.arena2036.de
...
```

Standartmäßig entspricht die Minikube IP 192.168.49.2. Diese kann über den folgenden Befehl überprüft werden:

```
minikube ip
```

Für den Zugriff vom externen Rechner auf den tractus-x-06.arena2036.de Server ist auch die Erweiterung der lokalen Host Datei erforderlich. Diese Datei ist unter Windows hier hinterlegt:

```
C:\Windows\System32\drivers\etc\hosts
```

Die externe IP kann über den folgenden Befehl überprüft werden:

```
ping -4 tractus-x-06.arena2036.de
```

Die vollständige Liste der Hosts-Einträge für einen externen Zugriff auf tractus-x-06.arena2036.de mit der IP 141.70.214.186 ist die folgende:

| | |
|----------------|--|
| 141.70.214.186 | centralidp.tractus-x-06.arena2036.de |
| 141.70.214.186 | sharedidp.tractus-x-06.arena2036.de |
| 141.70.214.186 | portal.tractus-x-06.arena2036.de |
| 141.70.214.186 | portal-backend.tractus-x-06.arena2036.de |
| 141.70.214.186 | managed-identity-wallets.tractus-x-06.arena2036.de |
| 141.70.214.186 | semantics.tractus-x-06.arena2036.de |
| 141.70.214.186 | sdfactory.tractus-x-06.arena2036.de |
| 141.70.214.186 | bpdnDiscovery.tractus-x-06.arena2036.de |
| 141.70.214.186 | discoveryfinder.tractus-x-06.arena2036.de |
| 141.70.214.186 | oem-controlplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | oem-dataplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | oem-dtr.tractus-x-06.arena2036.de |
| 141.70.214.186 | oem-sdb.tractus-x-06.arena2036.de |
| 141.70.214.186 | tierone-controlplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | tierone-dataplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | tierone-dtr.tractus-x-06.arena2036.de |
| 141.70.214.186 | tierone-sdb.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-1-controlplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-1-dataplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-1-dtr.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-1-sdb.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-2-controlplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-2-dataplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-2-dtr.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-2-sdb.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-3-controlplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-3-dataplane.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-3-dtr.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-3-sdb.tractus-x-06.arena2036.de |
| 141.70.214.186 | aas-webui.tractus-x-06.arena2036.de |
| 141.70.214.186 | aas-discovery.tractus-x-06.arena2036.de |
| 141.70.214.186 | oem-aas-env.tractus-x-06.arena2036.de |
| 141.70.214.186 | tierone-aas-env.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-1-aas-env.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-2-aas-env.tractus-x-06.arena2036.de |
| 141.70.214.186 | tiertwo-3-aas-env.tractus-x-06.arena2036.de |
| 141.70.214.186 | aas-registry.tractus-x-06.arena2036.de |
| 141.70.214.186 | sm-registry.tractus-x-06.arena2036.de |
| 141.70.214.186 | pgadmin4.tractus-x-06.arena2036.de |

9.3.4.6 Aufsetzen der Catena-X-Komponenten

Als Basis für das Catena-X Deployment werden die Umbrella-Helm Charts von Tractus-X genutzt. Diese werden unter dem folgenden Repository bereitgestellt:

- <https://github.com/eclipse-tractusx/tractus-x-umbrella>

Die genutzte Version 0.17.0 entspricht dabei dem, zu dem Zeitpunkt für die 2. Tractus-X-Community-Days vorbereiteten, Stand und Tractus-X Release 2403.

Die fertigen Konfigurationen werden auf einem separaten Fork unter folgendem Repository bereitgestellt:

- <https://github.com/Arena2036-Umbrella-Charts/tractus-x-umbrella-arena2036>

9.3.4.6.1 Konfiguration

Für die Installation der Komponenten wurden die folgenden Änderungen an den Umbrella-Helm-Charts von Tractus-X durchgeführt.

Zunächst wurde der bestehende Chart-Release unter [charts/umbrella](#) im Repository kopiert und ein neuer Chart-Release unser [charts/umbrella-arena2036](#) angelegt. Die ursprüngliche Datei [values.yaml](#) wurde als Grundlage genutzt.

Die folgenden Änderungen fanden, soweit nicht anders beschrieben, in der Datei [charts/umbrella-arena2036/values.yaml](#) statt. Im Allgemeinen gilt, dass die Änderungen allgemein gefasst werden und im Detail im entsprechenden Repository über die Git-History überprüft werden können, unter Revision Nummer ba502b927bc0e15fc395cccd2ce9de008508d68d4. Die entsprechende [values.yaml](#) ist zusätzlich als [values reference.yaml](#) im Ordner charts/umbrella-arena2036 abgelegt.

Wenn von einer Allgemeinen Hostnamen-Anpassung die Rede ist, bezieht sich das auf das Pattern **Fehler! Linkreferenz ungültig..**

9.3.4.6.1.1 Zertifikate

Zunächst muss das Zertifikatsmanagement angepasst werden. Dafür wurde eine [tractus-x-06.cluster-issuear.yaml](#) erstellt, die eine Kubernetes-Konfiguration beinhaltet und die auf das Cluster applied werden muss. Außerdem muss das Zertifikat und Key als Secret im Cluster angelegt werden. Die Befehle dafür sind:

```
cd /etc/nginx
kubectl create secret tls tractus-x-06-tls --cert=./all.tractus-x-06.arena2036.de.crt --key=~/all.tractus-x-06.arena2036.de.key

cd /<repository-pfad>/charts/umbrella-arena2036
kubectl apply -f tractus-x-06.cluster-issuer.yaml
```

9.3.4.6.1.2 Portal

- Aktivierung des Portals
- Anpassung der Host-Namen für
 - Host, Mailing, Provisioning, Frontend und Backend
- TLS-Konfiguration für
 - Frontend, Backend
- Anpassung der Service-Adressen Environment-Variablen für die Operating-Company-Services auf <https://<service>.tractus-x-06.arena2036.de>
- Anpassung der Nginx Ingress Regeln für `cors-allow-origin` und `cors-allow-credentials`
- Konfiguration der PostgreSQL Authentifizierung

9.3.4.6.1.3 Central-IDP

- Aktivierung von [Central-IDP](#)
- Anpassung des Host-Namen für Central-IDP
- [TLS Konfiguration für Central-IDP](#)
- [Keycloak Truststore Konfiguration](#)
- [TLS Zertifikats-Store Definition](#)
- Richtigstellung der Volume-Mounts `/realms`
- Konfiguration der PostgreSQL Authentifizierung
- Konfiguration eines Admin-Users

9.3.4.6.1.4 Shared-IDP

- Aktivierung von Shared-IDP

- Anpassung des Host-Namens für Shared-IDP
- TLS Konfiguration für Shared -IDP
- Keycloak Truststore Konfiguration
- TLS Zertifikats-Store Definition
- Richtigstellung der Volume-Mounts /realms
- Konfiguration der PostgreSQL Authentifizierung
- Konfiguration eines Admin-Users

9.3.4.6.1.5 BPN-Discovery

- Aktivierung der BPN-Discovery
- Anpassung des Host-Namens für BPN-Discovery
- Ingress Konfiguration erstellen, da nicht vorhanden
 - Ingress-Regeln definieren
- Authentifizierung für den Central-IDP mit Client C122-CX-BPND erstellen
- Konfiguration der PostgreSQL Authentifizierung

9.3.4.6.1.6 Discoveryfinder

- Aktivierung des Discoveryfinders
- Anpassung des Host-Namens für Discoveryfinder
- Ingress Konfiguration anpassen
 - Ingress-Regeln definieren
- Initial-Endpoint Konfiguration anpassen
- Authentifizierung für den Central-IDP mit Client C121-CX-DF erstellen
- Konfiguration der PostgreSQL Authentifizierung

9.3.4.6.1.7 Self-Description

- Aktivierung des Self-Description Services
- Anpassung des Hostnamens für die Self-Description
- Ingress Konfiguration erstellen, da nicht vorhanden
 - Ingress-Regeln definieren
- Erstellung der Central-IDP Konfiguration
 - Authentifizierung für den Central-IDP mit Client sa-c15-custodian-1 erstellen
 - Setzen von Client-Secret, Realm und Resource
 - Anpassung des Host-Namens

9.3.4.6.1.8 Managed Identity Wallet

- Aktivierung des Managed-Identity-Wallets
- Anpassung des Hostnamens für den Managed-Identity-Wallet
- Anpassung der Keycloak Konfiguration
- Ingress Konfiguration anpassen
 - Ingress-Regeln definieren
- Seeding Konfiguration erstellen und für unseren Use-Case anpassen
- Konfiguration der PostgreSQL Authentifizierung

9.3.4.6.1.9 Data Seeding

Das Data-Seeding besteht primär aus zwei Komponenten:

- Keycloak initialisierung
- Management-Identity-Wallet initialisierung für die Participants

Keycloak

Die Tractus-X Umbrella-Charts haben ein gewisses Set an BPNs und Clients, die für verschiedene Use-Cases gedacht sind und tief in die Umbrella-Charts definiert sind. Daher ist die einfachste Möglichkeit für unseren Use-Case die Template-User zu nutzen, anstatt komplett neue BPNs und Clients zu erstellen, da der Mehrwert keinen Aufwand hätte.

Für den Use-Case sind die angebotenen BPNs und Clients folgendermaßen den Participants zugeordnet:

| BPN | Alias | ClientId | Zuordnung |
|------------------|---------------------|----------|--------------------------------|
| BNPL00000003AZQP | BNP_OEM_C | satest01 | - |
| BNPL00000003AYRE | BNP_OEM_A | satest02 | OEM |
| BNPL00000003AVTH | BNP_OEM_B | satest03 | - |
| BNPL00000003AWSS | BNP_IRS_TEST | satest04 | - |
| BNPL00000003B0Q0 | BNP_N_TIER_A | satest05 | Tier One |
| BPNS0000000008ZZ | BNP_TRACEX_A_SITE_A | satest06 | - |
| BNPL00000003CNKC | BNP_TRACEX_B | satest07 | - |
| BNPL00000003B6LU | BNP_DISMANTLER | satest08 | - |
| BNPL00000003CML1 | BNP_TRACEX_A | satest09 | - |
| BPNS00000008BDFH | BNP_TRACEX_B_SITE_A | satest10 | - |
| BNPL00000003B2OM | BNP_TIER_A | satest11 | Tier Two #1 (Kostal) |
| BNPL00000003CSGV | BNP_TIER_C | satest12 | Tier Two #2 (Coroplast) |
| BNPL00000003B5MJ | BNP_TIER_B | satest13 | Tier Two #3 (Arena2036) |
| BNPL00000003AXS3 | BNP_SUB_TIER_B | satest14 | - |
| BNPL00000003B3NX | BNP_SUB_TIER_A | satest15 | - |
| BNPL00000000BJTL | BNP_SUB_TIER_C | satest16 | - |

Tabelle 9-3: Liste der BPNs

Neben der Participants gibt es eine Reihe an Service-Identities. Eine vollständige Auflistung der geseedeten Authentifizierungen befindet sich in der folgenden Tabelle:

| Beschreibung | ClientId |
|--|--------------------|
| Confidential client for BPDM | Cl7-CX-BPDM |
| Confidential client for BPDM Portal Gate | Cl16-CX-BPDMGate |
| Confidential client for Managed Identity Wallet | Cl5-CX-Custodian |
| Service account for Portal-Backend to call Keycloak | sa-cl1-reg-2 |
| Service account Clearinghouse update application | sa-cl2-01 |
| Service account SelfDescription (SD) update application | sa-cl2-02 |
| Service account AutoSetup trigger - Portal to Vendor Autosecure | sa-cl2-03 |
| Service account Discovery Finder | sa-cl21-01 |
| Service account BPN Discovery | sa-cl22-01 |
| Service account for SD Hub Call to Custodian for SD signature | sa-cl5-custodian-1 |
| Service account for Portal to call Custodian Wallet | sa-cl5-custodian-2 |
| Service account for Portal to access BPDM for Company Address publishing into the BPDM | sa-cl7-cx-5 |

| | |
|---|--------------|
| Service account for Portal to SD | sa-cl8-cx-1 |
| Service account in sharedidp master realm for portal backend | sa-cl1-reg-1 |

Tabelle 9-4: Seeding Data Authentifizierung

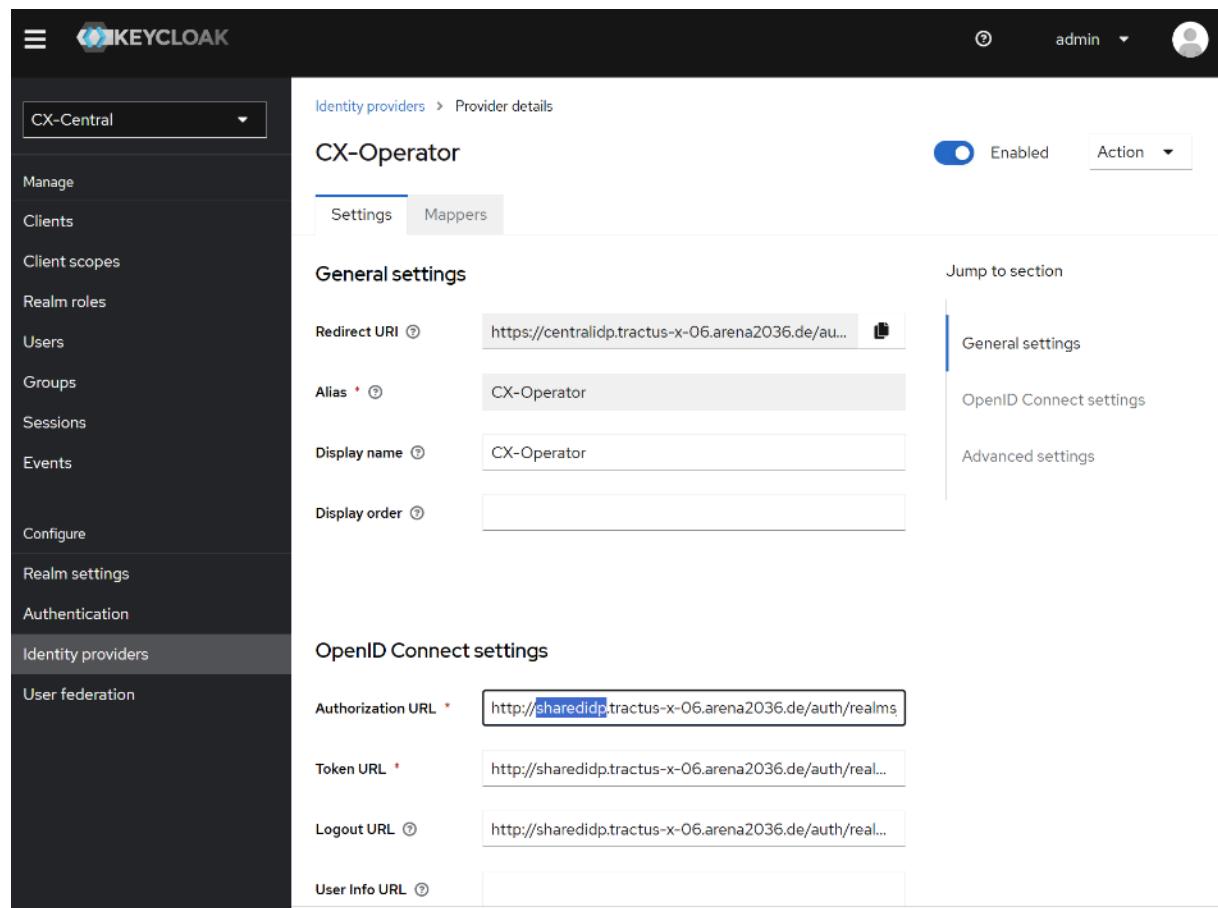
Eine kurze Beschreibung der original geseedeten Daten befindet sich u.A. in der Dokumentationsdatei concepts/seeds-overall-data.md im Repository.

Die eigentlichen Daten befinden sich an den Stellen:

- Für den Zentralen Identity Provider
 - `init-container/iam/centralidp/R2403/CX-Central-realm.JSON`
- Für den Shared Identity Provider
 - `init-container/iam/sharedidp/master-realm.JSON`
 - `init-container/iam/sharedidp/CX-Operator-realm.JSON`
 - `init-container/iam/sharedidp/CX-Operator-users-0.JSON`

Die Daten werden über einen Docker-Container eingespielt. Dieser basiert auf dem Dockerfile `init-container/iam/Dockerfile`.

Der Shared-IDP ist dabei der „Identity Provider“ für User für den Central-IDP. Dies ist z.B. auf dem folgenden Screenshot sichtbar:

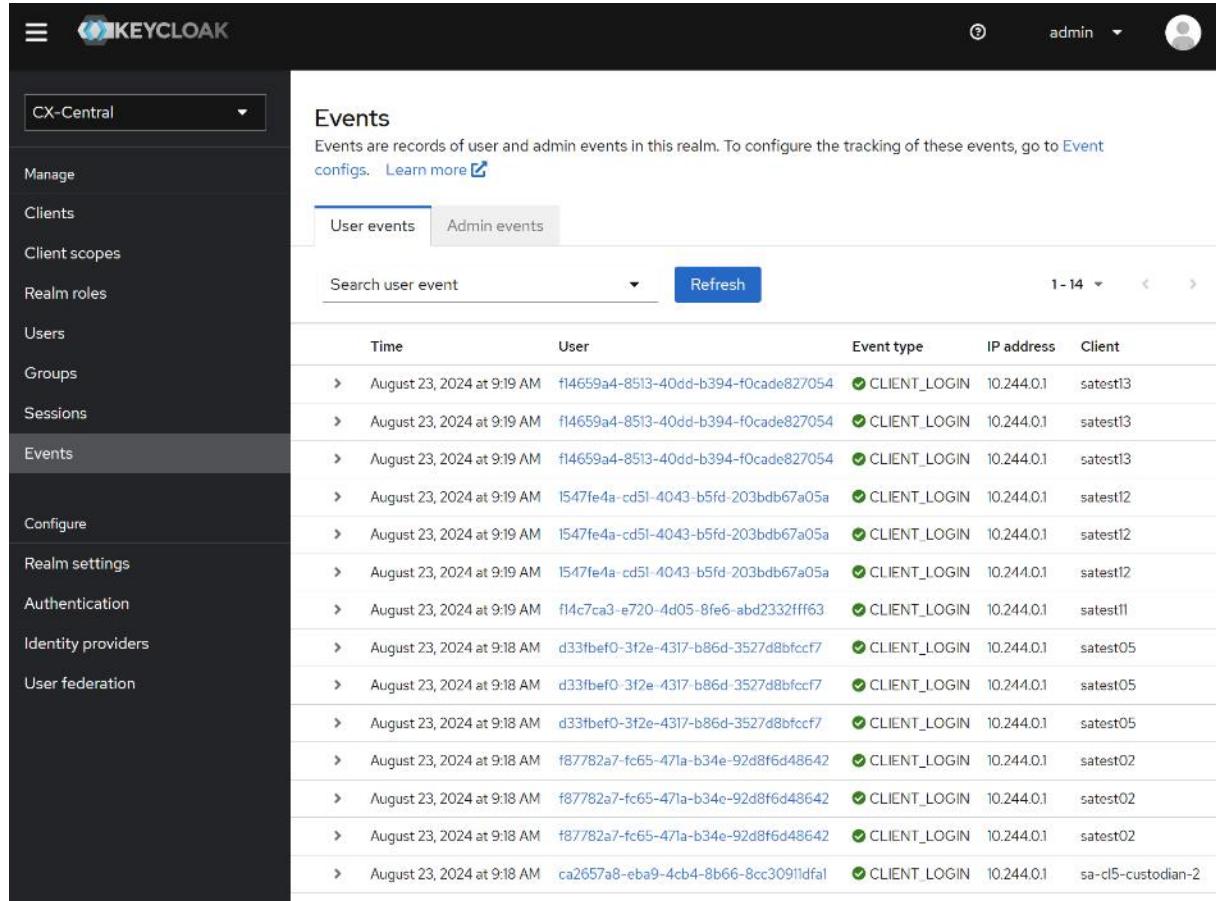


The screenshot shows the Keycloak management interface for the 'CX-Central' realm. The left sidebar is dark-themed and includes links for Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers (which is the active tab), and User federation. The main content area has a light background. It displays the 'Provider details' for the 'CX-Operator' identity provider. The 'General settings' section contains fields for 'Redirect URI' (set to `https://centralidp.tractus-x-06.arena2036.de/auth/realms/cx-central/protocol/openid-connect/token`), 'Alias' (set to 'CX-Operator'), 'Display name' (set to 'CX-Operator'), and 'Display order'. The 'OpenID Connect settings' section contains fields for 'Authorization URL' (set to `http://sharedidp.tractus-x-06.arena2036.de/auth/realm`), 'Token URL' (set to `http://sharedidp.tractus-x-06.arena2036.de/auth/real...`), 'Logout URL' (set to `http://sharedidp.tractus-x-06.arena2036.de/auth/real...`), and 'User Info URL'. A sidebar on the right lists 'Jump to section' options: General settings, OpenID Connect settings, and Advanced settings. At the top right, there is a user dropdown for 'admin'.

Abbildung 9-8: Identity Provider

Management-Identity-Wallet

Die Management-Identity-Wallet Initialisierung sieht so aus, dass nach dem Start jedes EDC pro Participant ein Token Request an den Keycloak gesendet wird, um einen Authentifizierungstoken zu generieren. Mit diesem Token wird zum einen ein User für jeden Participant angelegt als auch ein Wallet im MIW. Dies kann man beispielsweise auf diesem Screenshot sehen:



The screenshot shows the Keycloak interface with the left sidebar menu open, displaying various management options like Clients, Client scopes, Realm roles, Users, Groups, Sessions, and Events. The 'Events' option is selected. The main content area is titled 'Events' and contains a sub-header: 'Events are records of user and admin events in this realm. To configure the tracking of these events, go to Event configs. Learn more'. Below this, there are two tabs: 'User events' (selected) and 'Admin events'. A search bar labeled 'Search user event' and a 'Refresh' button are present. The table lists 14 user events from August 23, 2024, at 9:19 AM, all of which are 'CLIENT_LOGIN' events. The columns in the table are Time, User, Event type, IP address, and Client.

| Time | User | Event type | IP address | Client |
|----------------------------|---------------------------------------|--------------|------------|--------------------|
| August 23, 2024 at 9:19 AM | f14659a4-8513-40dd-b394-f0cade827054 | CLIENT_LOGIN | 10.244.0.1 | satest13 |
| August 23, 2024 at 9:19 AM | f14659a4-8513-40dd-b394-f0cade827054 | CLIENT_LOGIN | 10.244.0.1 | satest13 |
| August 23, 2024 at 9:19 AM | f14659a4-8513-40dd-b394-f0cade827054 | CLIENT_LOGIN | 10.244.0.1 | satest13 |
| August 23, 2024 at 9:19 AM | 1547fe4a-cd51-4043-b5fd-203bdb67a05a | CLIENT_LOGIN | 10.244.0.1 | satest12 |
| August 23, 2024 at 9:19 AM | 1547fe4a-cd51-4043-b5fd-203bdb67a05a | CLIENT_LOGIN | 10.244.0.1 | satest12 |
| August 23, 2024 at 9:19 AM | 1547fe4a-cd51-4043-b5fd-203bdb67a05a | CLIENT_LOGIN | 10.244.0.1 | satest12 |
| August 23, 2024 at 9:19 AM | f14c7ca3-e720-4d05-8fe6-abd2332ffff63 | CLIENT_LOGIN | 10.244.0.1 | satest11 |
| August 23, 2024 at 9:18 AM | d33fbef0-3f2e-4317-b86d-3527d8bfccf7 | CLIENT_LOGIN | 10.244.0.1 | satest05 |
| August 23, 2024 at 9:18 AM | d33fbef0-3f2e-4317-b86d-3527d8bfccf7 | CLIENT_LOGIN | 10.244.0.1 | satest05 |
| August 23, 2024 at 9:18 AM | f87782a7-fc65-471a-b34e-92d8f6d48642 | CLIENT_LOGIN | 10.244.0.1 | satest02 |
| August 23, 2024 at 9:18 AM | f87782a7-fc65-471a-b34e-92d8f6d48642 | CLIENT_LOGIN | 10.244.0.1 | satest02 |
| August 23, 2024 at 9:18 AM | f87782a7-fc65-471a-b34e-92d8f6d48642 | CLIENT_LOGIN | 10.244.0.1 | satest02 |
| August 23, 2024 at 9:18 AM | ca2657a8-eba9-4cb4-8b66-8cc30911dfal | CLIENT_LOGIN | 10.244.0.1 | sa-cl5-custodian-2 |

Abbildung 9-9: Anlegen User

9.3.4.6.1.10 Eclipse Dataspace Connectors

Für die Eclipse-Dataspace-Connectoren und die dazu gehörigen Digital-Twin Registries, müssen die Anpassungen für die verschiedenen, im Use-Case definierten, Participants durchgeführt werden.

Der folgende Abschnitt muss für unseren Use-Case für folgende Participants wiederholt werden:

- OEM
- Tier One
- Tier Two #1
- Tier Two #2
- Tier Two #3

Für jeden der folgenden Service-Stacks für jeden oben genannten Participant müssen folgende Anpassungen an den EDCs durchgeführt werden. Die Details und unterschiedlichen Werte dieser Einstellungen pro Participant sind weiter unten beschrieben.

- Anpassung der Service-URLs auf Kubernetes-Interne-Hostnamen. Am Beispiel vom OEM:

```
backendUrl: http://{{ .Release.Name }}-oem-submodelserver:8080
registryUrl: http://{{ .Release.Name }}-oem-dtr:8080/api/v3.0
controlplanePublicUrl: http://{{ .Release.Name }}-oem-edc-controlplane:8084
controlplaneManagementUrl: http://{{ .Release.Name }}-oem-edc-controlplane:8081
dataplaneUrl: http://{{ .Release.Name }}-oem-edc-dataplane:8081
```

- Setzen der BPN
- Setzen des zur BPN passenden „edc-miw-keycloak-secret“
- Setzen des „nameOverride“
- Setzen des Management-API-Keys
- Erstellung einer Digital-Twin-Registry Konfiguration, da nicht vorhanden
 - Deaktivierung des Ingresses für den unten folgenden Patch
- Anpassung der Hostnamen für:
 - Control-Plane, Data-Plane, Digital-Twin-Registry
- Anpassung der MIW-Einstellungen
- Anpassung der Vault-Einstellungen
- Konfiguration der Keycloak-Einstellungen
- Konfiguration der PostgreSQL Authentifizierung

DTR

Neben den Konfigurationsaufgaben müssen teilweise noch größere Fixes für unseren Use-Case erfolgen. So ist die Ingress-Konfiguration der Digital-Twin-Registry nicht für unseren Use-Case nutzbar und ein Patch-Ingress muss erstellt werden.

Die Konfiguration von diesem findet in der values.yaml statt:

```
dtr:
  ingress:
    enabled: true
    urlPrefix: /semantics/registry
  service:
    port: 8080
  hosts:
    - host: oem-dtr.tractus-x-06.arena2036.de
      serviceName: umbrella-oem-dtr
    - host: tierone-dtr.tractus-x-06.arena2036.de
      serviceName: umbrella-tierone-dtr
    - host: tiertwo-1-dtr.tractus-x-06.arena2036.de
      serviceName: umbrella-tiertwo-1-dtr
    - host: tiertwo-2-dtr.tractus-x-06.arena2036.de
      serviceName: umbrella-tiertwo-2-dtr
    - host: tiertwo-3-dtr.tractus-x-06.arena2036.de
      serviceName: umbrella-tiertwo-3-dtr
  annotations:
    cert-manager.io/cluster-issuer: "my-ca-issuer"
    nginx.ingress.kubernetes.io/rewrite-target: "/$1"
    nginx.ingress.kubernetes.io/use-regex: "true"
    nginx.ingress.kubernetes.io/enable-cors: "true"
    nginx.ingress.kubernetes.io/cors-allow-credentials: "true"
    nginx.ingress.kubernetes.io/cors-allow-origin: "*"
    nginx.ingress.kubernetes.io/cors-allow-methods: "*"
```

Das dazu passende Helm-Template sieht folgendermaßen aus:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: umbrella-dtr-patch
  annotations:
    {{- with .Values.dtr.annotations -}}
    annotations:
      {{- toYaml . | nindent 4 -}}
    {{- end -}}
spec:
  rules:
    {{- range .Values.dtr.hosts -}}
```

```

    - host: {{ .host }}
    http:
      paths:
        - path: /semantics/registry
          pathType: Prefix
          backend:
            service:
              name: {{ .serviceName }}
              port:
                number: {{ $.Values.dtr.service.port }}
{{- end}}
  
```

Service-Stack: OEM

- **Kubernetes-Kürzel:** oem
- **EDC-nameOverride:** oem-edc
- **BNP:** BPNL00000003AYRE
- **clientId:** satest02
- **secretAlias:** edc-miw-keycloak-secret
- **Data-Plane:** oem-dataplane.*
- **Control-Plane:** oem-controlplane.*
- **Digital-Twin-Registry:** oem-dtr.*
- **Vault:** edc-oem-vault

Service-Stack: Tier One

- **Kubernetes-Kürzel:** tierone
- **EDC-nameOverride:** tierone-edc
- **BNP:** BPNL00000003B0Q0
- **clientId:** satest05
- **secretAlias:** edc-miw-keycloak-secret
- **Data-Plane:** tierone-dataplane.*
- **Control-Plane:** tierone-controlplane.*
- **Digital-Twin-Registry:** tierone-dtr.*
- **Vault:** edc-tierone-vault

Service-Stack: Tier Two #1

- **Kubernetes-Kürzel:** tiertwonoone
- **EDC-nameOverride:** tiertwo-1-edc
- **BNP:** BPNL00000003B2OM
- **clientId:** satest11
- **secretAlias:** edc-miw-keycloak-secret
- **Data-Plane:** tietwo-1-dataplane.*
- **Control-Plane:** tietwo-1-controlplane.*
- **Digital-Twin-Registry:** tietwo-1-dtr.*
- **Vault:** edc-tietwo-1-vault

Service-Stack: Tier Two #2

- **Kubernetes-Kürzel:** tiertwonotwo
- **EDC-nameOverride:** tiertwo-2-edc
- **BNP:** BPNL00000003CSGV
- **clientId:** satest12
- **secretAlias:** edc-miw-keycloak-secret
- **Data-Plane:** tiertwo-2-dataplane.*
- **Control-Plane:** tiertwo-2-controlplane.*
- **Digital-Twin-Registry:** tiertwo-2-dtr.*
- **Vault:** edc-tiertwo-2-vault

Service-Stack: Tier Two #3

- **Kubernetes-Kürzel:** tiertwonothree
- **EDC-nameOverride:** tiertwo-3-edc
- **BNP:** BPNL00000003B5MJ
- **clientId:** satest13
- **secretAlias:** edc-miw-keycloak-secret
- **Data-Plane:** tiertwo-3-dataplane.*
- **Control-Plane:** tiertwo-3-controlplane.*
- **Digital-Twin-Registry:** tiertwonothree-dtr.*
- **Vault:** edc-tiertwonothree-vault

9.3.4.6.1.11 Semantic-Hub

- Aktivierung des Semantic-Hubs
- Anpassung des Hostnamens für den Semantic-Hubs
- Ingress Konfiguration erstellen, da nicht vorhanden
 - Ingress-Regeln definieren
- Erstellung Keycloak-Konfiguration

9.3.4.6.1.12 Post-Install MIW Setup

Die Post-Install MIW-Setup Template-Datei wurde ergänzt um eine Ausgabe der Keycloak Authentifizierung, um ein debuggen zu erleichtern. Die Datei ist charts/umbrella-arena2036/templates/post-install-miw-setup.yaml:

```
TOKEN=`cat token | grep -o '"access_token": *"[^"]*"` | sed 's#"access_to-
ken":"\(.*\)#\1#'` && export TOKEN
echo "wget {{$keycloakPath}} --header 'Content-Type: application/x-www-form-ur-
lencoded' --post-data 'grant_type=client_credentials&client_id={{ $miwUser
 }}&client_secret={{ $miwSecret }}'"
```

9.3.4.6.2 Deployment

Für das Deployment der für den Use-Case konfigurierten Komponenten werden folgende Schritte durchgeführt:

- Zunächst muss sichergestellt werden, dass die Schritte aus dem **Abschnitt „9.3.4.5. Installation Server“** durchgeführt wurden, da sie für die folgenden Schritte die Voraussetzung sind
- Das Minikube Cluster kann mit folgendem Befehl initialisiert werden:


```
minikube start --cpus=4 --memory 6gb --driver=hyperv
```
- Folgende Addons müssen aktiviert werden:

```
minikube addons enable ingress
minikube addons enable ingress-dns
```

- Die Hosts- und Zertifikats-Einstellungen sollten in dem oben genannten Kapitel bereits durchgeführt worden sein
- Für die Installation muss das Helm-Chart-Repository von Tractus-X hinzugefügt und geupdated werden:

```
helm repo add tractusx-dev https://eclipse-tractusx.github.io/charts/dev
helm repo update
```

- Um, die Helm-Charts zu nutzen muss man zu unseren Umbrella-Charts navigieren und die Dependencies bauen

```
cd charts/umbrella-arena2036
helm dependency build
```

- Zuletzt können die Helm-Charts über folgenden Befehl in das Cluster installiert werden

```
helm install -f values.yaml umbrella . --namespace umbrella --create-namespace
```

- Bei Bedarf kann der Helm-Release aktualisiert,:;

```
helm upgrade umbrella . -n umbrella --create-namespace
```

- Oder deinstalliert werden:

```
helm uninstall umbrella -n umbrella
```

9.3.4.6.2.1 Service- & API-Auflistung

Im Folgenden eine Auflistung der deployten Services und APIs:

- Central-IDP
 - Admin Console: <https://centralidp.tractus-x-06.arena2036.de/auth/admin/master/console/#>
- Shared-IDP
 - Admin Console: <https://sharedidp.tractus-x-06.arena2036.de/auth/admin/master/console/#>
- BPN-Discovery
 - API: <https://bpndiscovery.tractus-x-06.arena2036.de/bpndiscovery/swagger-ui/index.html>
- Discoveryfinder
 - API: <https://discoveryfinder.tractus-x-06.arena2036.de/discoveryfinder/swagger-ui/index.html>
- Managed Identity Wallet
 - API: <https://managed-identity-wallets.tractus-x-06.arena2036.de/ui/swagger-ui/index.html>
- Self-Description
 - API: <https://sdfactory.tractus-x-06.arena2036.de/swagger-ui/index.html>
- Portal
 - Web: <https://portal.tractus-x-06.arena2036.de>
- Portal-Backend
 - Administration & Discovery:
 - API: <https://portal-backend.tractus-x-06.arena2036.de/api/administration/swagger/index.html>
 - Notifications:

- API: <https://portal-backend.tractus-x-06.arena2036.de/api/notification/swagger/index.html>
- Services:
 - API: <https://portal-backend.tractus-x-06.arena2036.de/api/services/swagger/index.html>
- Apps:
 - API: <https://portal-backend.tractus-x-06.arena2036.de/api/apps/swagger/index.html>
- Registration:
 - API: <https://portal-backend.tractus-x-06.arena2036.de/api/registration/swagger/index.html>
- **PG-Admin**
 - Admin Console: <https://pgadmin4.tractus-x-06.arena2036.de/>
- **OEM Service-Stack**
 - OEM EDC-Dataplane
 - API: <https://oem-dataplane.tractus-x-06.arena2036.de/api>
 - Public API: <https://oem-dataplane.tractus-x-06.arena2036.de/api/public>
 - OEM EDC-Controlplane
 - API: <https://oem-controlplane.tractus-x-06.arena2036.de/api>
 - Management API: <https://oem-controlplane.tractus-x-06.arena2036.de/management>
 - Dataspace Protocol: <https://oem-controlplane.tractus-x-06.arena2036.de/api/v1/dsp>
 - OEM DTR
 - API: <https://oem-dtr.tractus-x-06.arena2036.de/swagger-ui/index.html>
- **Tier 1 Service-Stack**
 - Tier 1 EDC-Dataplane
 - API: <https://tierone-dataplane.tractus-x-06.arena2036.de/api>
 - Public API: <https://tierone-dataplane.tractus-x-06.arena2036.de/api/public>
 - Tier 1 EDC-Controlplane
 - API: <https://tierone-controlplane.tractus-x-06.arena2036.de/api>
 - Management API: <https://tierone-controlplane.tractus-x-06.arena2036.de/management>
 - Dataspace Protocol: <https://tierone-controlplane.tractus-x-06.arena2036.de/api/v1/dsp>
 - Tier 1 DTR
 - API: <https://tierone-dtr.tractus-x-06.arena2036.de/swagger-ui/index.html>
- **Tier 2 #1 (Kostal) Service-Stack**
 - Tier 2 #1 EDC-Dataplane
 - API: <https://tiertwo-1-dataplane.tractus-x-06.arena2036.de/api>
 - Public API: <https://tiertwo-1-dataplane.tractus-x-06.arena2036.de/api/public>
 - Tier 2 #1 EDC-Controlplane
 - API: <https://tiertwo-1-controlplane.tractus-x-06.arena2036.de/api>

- Management API: <https://tiertwo-1-controlplane.tractus-x-06.arena2036.de/management>
- Dataspace Protocol: <https://tiertwo-1-controlplane.tractus-x-06.arena2036.de/api/v1/dsp>
- Tier 2 #1 DTR
 - API: <https://tiertwo-1-dtr.tractus-x-06.arena2036.de/swagger-ui/index.html>
- **Tier 2 #2 (Coroplast) Service-Stack**
 - Tier 2 #2 EDC-Dataplane
 - API: <https://tiertwo-2-dataplane.tractus-x-06.arena2036.de/api>
 - Public API: <https://tiertwo-2-dataplane.tractus-x-06.arena2036.de/api/public>
 - Tier 2 #2 EDC-Controlplane
 - API: <https://tiertwo-2-controlplane.tractus-x-06.arena2036.de/api>
 - Management API: <https://tiertwo-2-controlplane.tractus-x-06.arena2036.de/management>
 - Dataspace Protocol: <https://tiertwo-2-controlplane.tractus-x-06.arena2036.de/api/v1/dsp>
 - Tier 2 #2 DTR
 - API: <https://tiertwo-2-dtr.tractus-x-06.arena2036.de/swagger-ui/index.html>
- **Tier 2 #3 (Arena2036) Service-Stack**
 - Tier 2 #3 EDC-Dataplane
 - API: <https://tiertwo-3-dataplane.tractus-x-06.arena2036.de/api>
 - Public API: <https://tiertwo-3-dataplane.tractus-x-06.arena2036.de/api/public>
 - Tier 2 #3 EDC-Controlplane
 - API: <https://tiertwo-3-controlplane.tractus-x-06.arena2036.de/api>
 - Management API: <https://tiertwo-3-controlplane.tractus-x-06.arena2036.de/management>
 - Dataspace Protocol: <https://tiertwo-3-controlplane.tractus-x-06.arena2036.de/api/v1/dsp>
 - Tier 2 #3 DTR
 - API: <https://tiertwo-3-dtr.tractus-x-06.arena2036.de/swagger-ui/index.html>

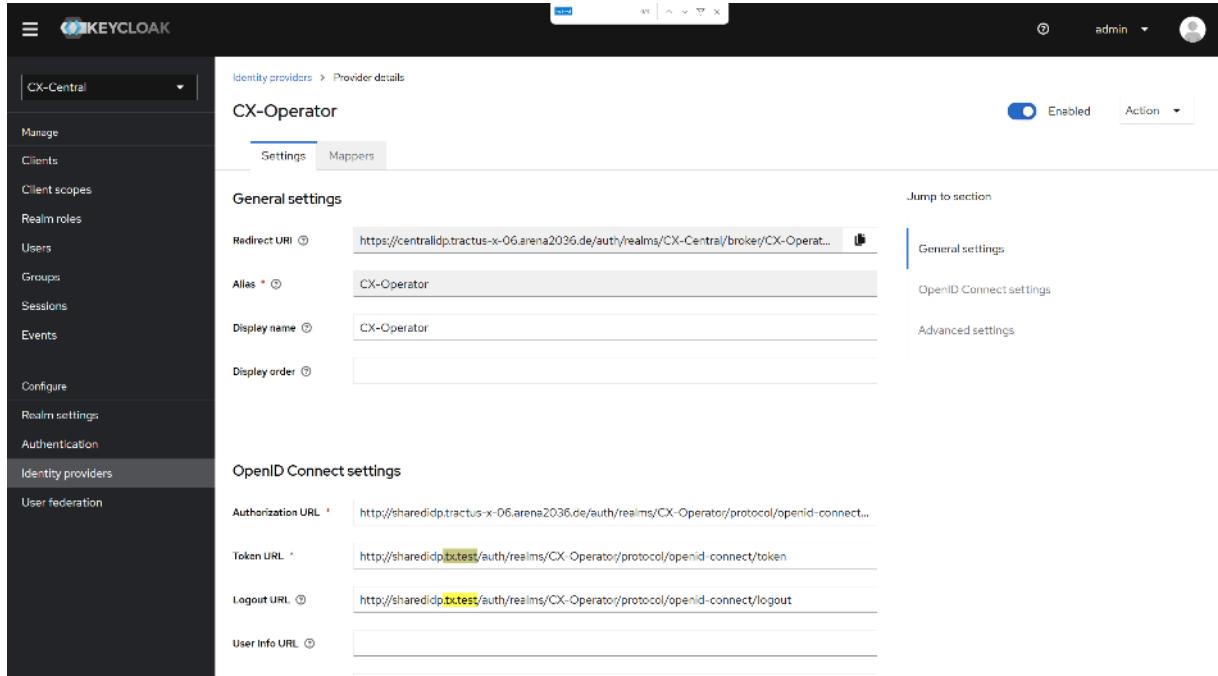
9.3.4.6.3 Live-Anpassungen und Authentifizierung

9.3.4.6.3.1 Keycloak

Da die Seeding-Daten teilweise hardcodiert sind, müssen nach dem Deployment einzelne Anpassungen durchgeführt werden.

Zunächst muss der Sub-Identity-Provider für Central-IDP angepasst werden. Dies kann in der Admin-Console durchgeführt werden:

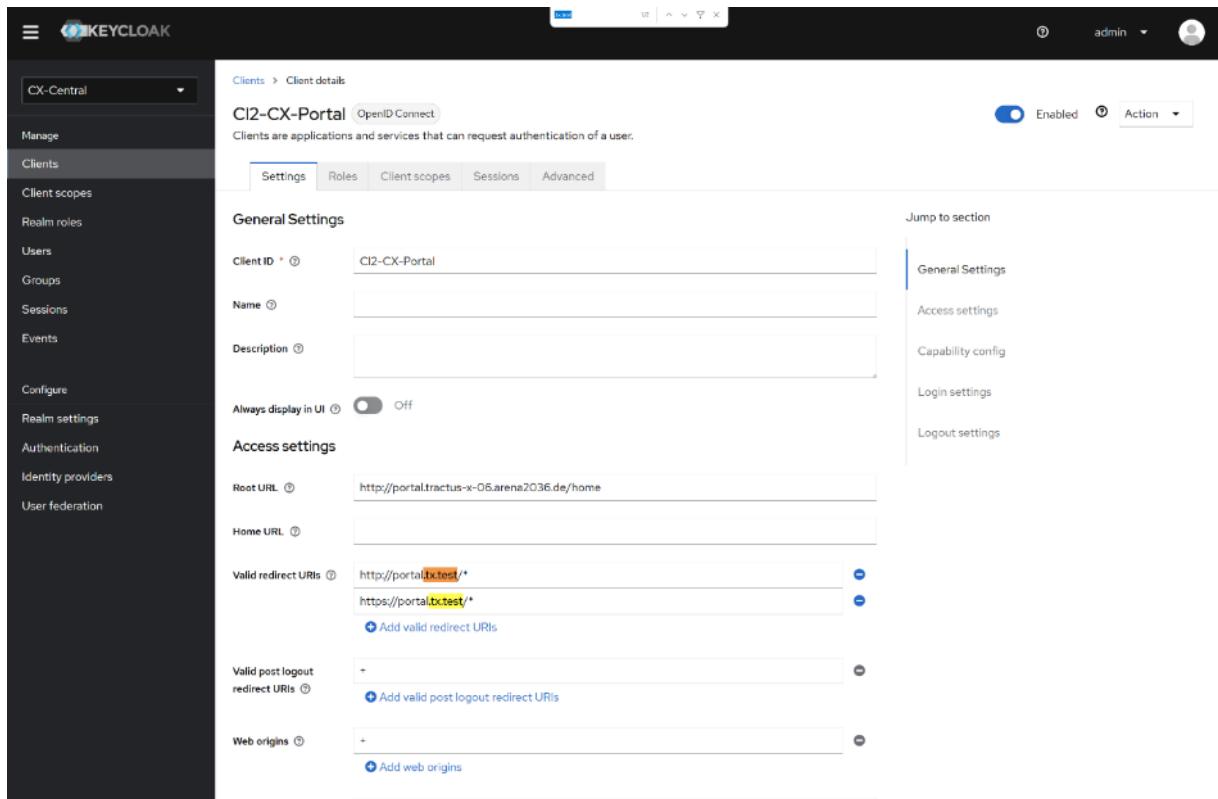
<https://centralidp.tractus-x-06.arena2036.de/auth/admin/master/console/#>



The screenshot shows the Keycloak CX-Central interface. On the left sidebar, 'Identity providers' is selected. In the main area, 'CX-Operator' is chosen under 'Provider details'. The 'Settings' tab is active. Under 'General settings', the Redirect URI is set to [https://centralidp.tractus-x-06.arena2036.de/auth/realms/CX-Central/broker/CX-Operator...](https://centralidp.tractus-x-06.arena2036.de/auth/realms/CX-Central/broker/CX-Operator). Other fields include Alias (CX-Operator), Display name (CX-Operator), and Display order (empty). To the right, there are sections for 'General settings', 'OpenID Connect settings', and 'Advanced settings'. The 'OpenID Connect settings' section contains fields for Authorization URL (<http://sharedidp.tractus-x-06.arena2036.de/auth/realms/CX-Operator/protocol/openid-connect..>), Token URL (<http://sharedidp.tractus-test/auth/realms/CX-Operator/protocol/openid-connect/token>), Logout URL (<http://sharedidp.tractus-test/auth/realms/CX-Operator/protocol/openid-connect/logout>), and User Info URL (empty).

Abbildung 9-10: Anpassung Sub-Identity-Provider

Zusätzlich müssen die Redirect-URLs des Portal-Users angepasst werden:



The screenshot shows the Keycloak CX-Central interface. On the left sidebar, 'Clients' is selected. In the main area, 'CI2-CX-Portal' is chosen under 'Client details'. The 'Settings' tab is active. Under 'General Settings', the Client ID is set to CI2-CX-Portal. Other fields include Name (empty), Description (empty), and Always display in UI (Off). To the right, there are sections for 'General Settings', 'Access settings', 'Capability config', 'Login settings', and 'Logout settings'. In the 'Access settings' section, the Root URL is set to <http://portal.tractus-x-06.arena2036.de/home>. The Valid redirect URIs field contains http://portal.tractus-test/* and https://portal.tractus-test/*. Below it, the Valid post logout redirect URIs field has an empty list. The Web origins field also has an empty list.

Abbildung 9-11: Anpassung Redirect-URLs des Portal-Users

Durch einen Fehler in den Seeding-Daten (ein Prozess, bei dem eine Datenbank mit einem anfänglichen Datensatz gefüllt werden), müssen außerdem mehrere Rollen für den Client `sa-c12-reg-2` hinzugefügt werden:

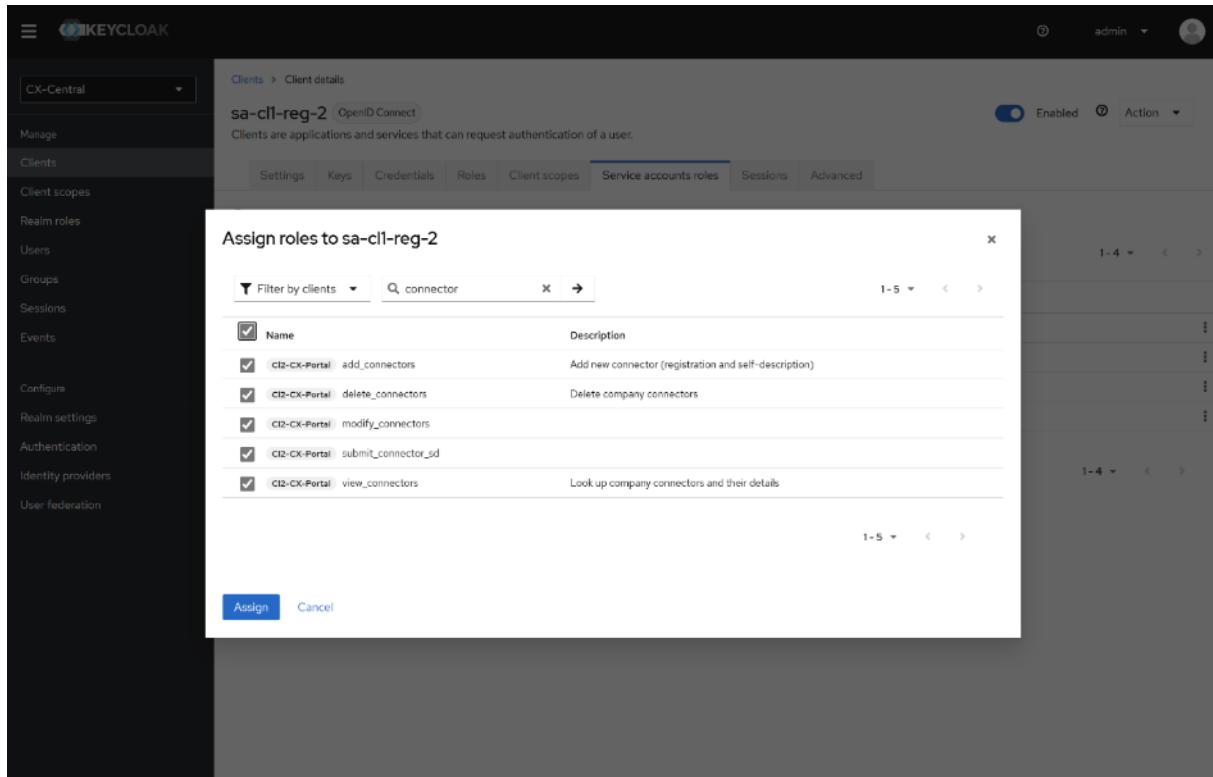


Abbildung 9-12: Hinzufügen Rollen

9.3.4.6.3.2 Wallet

- API: <https://managed-identity-wallets.tractus-x-06.arena2036.de/api>

```
wget http://centralidp.tractus-x-06.arena2036.de/auth/realms/CX-Central/protocol/openid-connect/token --header 'Content-Type: application/x-www-form-urlencoded' --post-data 'grant_type=client_credentials&client_id=sa-cl5-custodian-2&client_secret=UIqawwoohsvZ6AZ0d111LhnsUTKMWe4D'
```

9.3.4.6.3.3 BPN-Discovery

- API: <https://bpndiscovery.tractus-x-06.arena2036.de/bpndiscovery>

```
wget -q -S -O - http://centralidp.tractus-x-06.arena2036.de/auth/realms/CX-Central/protocol/openid-connect/token --header 'Content-Type: application/x-www-form-urlencoded' --post-data 'grant_type=client_credentials&client_id=sa-cl22-01&client_secret=1yDWW7BNwouRGxYRkDmzkpzqz5FG748F'
```

9.3.4.6.3.4 Discoveryfinder

- API: <https://discoveryfinder.tractus-x-06.arena2036.de/discoveryfinder>

```
wget -q -S -O - http://centralidp.tractus-x-06.arena2036.de/auth/realms/CX-Central/protocol/openid-connect/token --header 'Content-Type: application/x-www-form-urlencoded' --post-data 'grant_type=client_credentials&client_id=sa-cl21-01&client_secret=oFbXttMA7vI5MysN7AiEpobX5o3Jfbhp'
```

9.3.4.6.3.5 Portal API CentralIdp User

- API: <https://portal-backend.tractus-x-06.arena2036.de/api/administration>

```
wget -q -S -O - http://centralidp.tractus-x-06.arena2036.de/auth/realms/CX-Central/protocol/openid-connect/token --header 'Content-Type: application/x-www-form-urlencoded' --post-data 'grant_type=client_credentials&client_id=sa-cl1-reg-2&client_secret=aEoUADDw2aNPa0WAAkGAYkfC80n8sKxJ'
```

9.3.4.6.3.6 Portal API SharedIdp User

- API: <https://portal-backend.tractus-x-06.arena2036.de/api/administration>

```
wget -q -S -O - http://sharedidp.tractus-x-06.arena2036.de/auth/realms/master/protocol/openid-connect/token --header 'Content-Type: application/x-www-
```

```
form-urlencoded' --post-data 'grant_type=client_credentials&client_id=sa-cl1-reg-1&client_secret=YPA1t6BMQtPtaG3fpH8Sa8Ac6KYbPUM7'
```

Unter Windows kann alternativ zu wget, curl benutzt werden. Ein Request kann bspw. so aussehen:

```
curl `
  -S -o -
  -X POST `
  -H 'Content-Type: application/x-www-form-urlencoded' `

  --url 'http://centralidp.tractus-x-06.arena2036.de/auth/realms/CX-Central/protocol/openid-connect/token' `

  --data-urlencode 'grant_type=client_credentials' `

  --data-urlencode 'client_id=sa-cl22-01' `

  --data-urlencode 'client_secret=1yDWW7BNwouRGxYRkDmzkpzqz5FG748f'`
```

9.3.4.7 Aufsetzen der BaSyx-Komponenten

Als Basis für das BaSyx Deployment wird das Repository [eclipse-basyx/basyx-java-server-sdk](https://github.com/eclipse-basyx/basyx-java-server-sdk) (github.com) herangezogen. Dieses beinhaltet unter dem Pfad examples/BaSyxMinimal eine einfache Konfiguration, die für unseren Use-Case genutzt werden kann.

Die Minimal Konfiguration besteht aus den folgenden Komponenten:

- Eine **MongoDB** Datenbank für die Speicherung der Submodelle, Shells und weitere Komponenten
- Ein **MQTT**-Message-Queue Service, der für die Kommunikation zwischen Services genutzt wird
- Ein **Asset-Administration-Shell-Environment** Service, welcher die folgenden APIs enthält:
 - Registry and Discovery Interface
 - Submodel Repository API
 - Asset Administration Shell API
 - Asset Administration Shell Repository API
 - Concept Description Repository API
 - Environment API
 - Serialization API
- Ein **Asset-Administration-Shell-Registry** Service, welcher die folgende APIs enthält:
 - Registry and Discovery Interface
- Ein **Asset-Administration-Shell-Discovery** Service, welcher die folgenden APIs enthält:
 - Registry and Discovery Interface
 - Asset Administration Shell Basic Discovery API
- Ein **Submodel-Registry** Service, welcher die folgenden APIs enthält:
 - Submodel Registry API
- Eine **AAS-GUI**, welche ein Web-Interface bereitstellt
- Für eine detaillierte Auflistung siehe das Kapitel „9.3.4.8 BaSyx-Integration in Use-Case und Infrastruktur“.
- Zusätzlich sind folgende Konfigurationen vorhanden, die angepasst werden müssen:
 - Eine `mosquitto.conf` für MQTT
 - Eine `aas-env.properties` Datei für die AAS-Environment

- Eine `aas-discovery.properties` Datei für die AAS-Discovery
- Eine `aas-registry.yaml` Datei für die AAS-Registry
- Eine `sm-registry.yaml` Datei für die Submodel-Registry

Damit eine reibungslose Kommunikation mit der bestehenden Infrastruktur stattfinden kann benötigt man entsprechende Helm-Charts, die allerdings nicht für die BaSyx-Komponenten vorhanden sind und daher erst erstellt werden müssen.

Als Referenz wird die vorhandene Docker-Compose Konfiguration als Grundlage genommen. Daraus werden entsprechende Helm-Komponenten abgeleitet. Dies wird in den folgenden Kapiteln beschrieben.

9.3.4.7.1 Konfiguration

Zunächst werden die Konfigurationen in den Ordner `charts/umbrella-arena2036/resources` kopiert, damit sie für die Helm-Charts verfügbar sind.

In dem vorher definierten Charts-Releases `umbrella-arena2036`, wird im Ordner `charts/umbrella-2036/templates` ein neues Template namens `basyx.yaml` erstellt.

Für jeden der oben genannten Services müssen zunächst folgende Kubernetes-Komponenten (**am Beispiel von AAS-Environment**) definiert werden:

- Ein „Kubernetes-Service“

```
apiVersion: v1
kind: Service
metadata:
  name: {{ $fullName }}-aas-env
spec:
  ports:
    - port: {{ .Values.aasEnv.port }}
      targetPort: {{ .Values.aasEnv.port }}
  selector:
    app: {{ $fullName }}-aas-env
```

- Ein „Kubernetes-Deployment“

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ $fullName }}-aas-env
spec:
  selector:
    matchLabels:
      app: {{ $fullName }}-aas-env
  template:
    metadata:
      labels:
        app: {{ $fullName }}-aas-env
    spec:
      containers:
        - name: aas-env
          image: {{ .Values.aasEnv.image }}
          ports:
            - containerPort: {{ .Values.aasEnv.port }}
      volumeMounts:
        - mountPath: /application/application.properties
          subPath: application.properties
          name: aas-env-properties
        - mountPath: /application/aas
          subPath: aas
          name: aas-data
      volumes:
        - name: aas-env-properties
      configMap:
```

```

        name: aas-env-config
      - name: aas-data
        persistentVolumeClaim:
          claimName: aas-data-pvc

```

- Eine „Kubernetes-ConfigMap“

```

apiVersion: v1
kind: ConfigMap
metadata:
  name: aas-env-config
data:
  application.properties: |
    {{ .Files.Get "resources/aas-env.properties" | nindent 4 }}

```

Zusätzlich müssen bei allen BaSyx-Komponenten außer MQTT und MongoDB noch folgende Kubernetes-Komponenten definiert werden:

- Ein „Kubernetes-PersistentVolumeClaim“

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: aas-data-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

```

- Ein „Kubernetes-Ingress“

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ $fullName }}-aas-env
  annotations:
    {{- with .Values.aasEnv.ingress.annotations -}}
    annotations:
      {{- toYaml . | nindent 4 -}}
    {{- end -}}
spec:
  rules:
    - host: {{ .Values.aasEnv.ingress.hostname }}
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: {{ $fullName }}-aas-env
                port:
                  number: {{ .Values.aasEnv.port }}

```

Diese Kubernetes-Komponenten müssen außerdem mit den entsprechenden Werten in der `values.yaml` beschrieben werden. Dazu gehören z.B. Variablen für:

- Image-Pfad
- Ports
- Hostname
- Ingress Regeln

Das Beispiel anhand des AAS-Environments ist:

```

aasEnv:
  image: eclipsebasyx/aas-environment:2.0.0-SNAPSHOT
  port: 8081

```

```

ingress:
  enabled: true
  annotations:
    cert-manager.io/cluster-issuer: "my-ca-issuer"
    nginx.ingress.kubernetes.io/rewrite-target: "/$1"
    nginx.ingress.kubernetes.io/use-regex: "true"
    nginx.ingress.kubernetes.io/enable-cors: "true"
    nginx.ingress.kubernetes.io/cors-allow-origin: "https://*.tractus-x-06.arena2036.de/*, https://aas-webui.tractus-x-06.arena2036.de"
    nginx.ingress.kubernetes.io/cors-allow-credentials: "true"
  hostname: aas-env.tractus-x-06.arena2036.de

```

9.3.4.7.2 Deployment

Unter der Annahme, dass ein Deployment schon vorhanden ist und nur erweitert wird, muss lediglich das Helm-Deployment auf dem Kubernetes Cluster aktualisiert werden. Dafür werden folgende Befehle im Repository ausgeführt:

```

cd charts/umbrella-arena2036
helm upgrade umbrella . -n umbrella --create-namespace

```

Die fertig deployten Komponenten sind dann unter den folgenden URLs erreichbar:

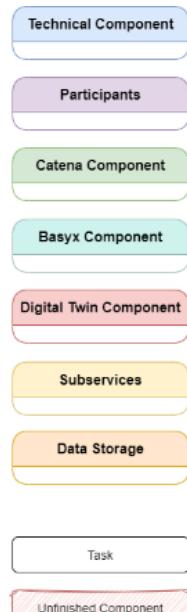
- **OEM** Service-Stack
 - OEM AAS-Environment
 - <https://oem-aas-env.tractus-x-06.arena2036.de/swagger-ui/index.html>
- **Tier 1** Service-Stack
 - Tier 1 AAS-Environment
 - <https://tierone-aas-env.tractus-x-06.arena2036.de/swagger-ui/index.html>
- **Tier 2 #1** (Kostal) Service-Stack
 - Tier 2 #1 AAS-Environment
 - <https://tiertwo-1-aas-env.tractus-x-06.arena2036.de/swagger-ui/index.html>
- **Tier 2 #2** (Coroplast) Service-Stack
 - Tier 2 #2 AAS-Environment
 - <https://tiertwo-2-aas-env.tractus-x-06.arena2036.de/swagger-ui/index.html>
- **Tier 2 #3** (Arena2036) Service-Stack
 - Tier 2 #3 AAS-Environment
 - <https://tiertwo-3-aas-env.tractus-x-06.arena2036.de/swagger-ui/index.html>
- BaSyx-AAS-Discovery
 - <https://aas-discovery.tractus-x-06.arena2036.de/swagger-ui/index.html>
- BaSyx-AAS-Registry
 - <https://aas-registry.tractus-x-06.arena2036.de/swagger-ui/index.html>
- BaSyx-Submodel-Registry
 - <https://sm-registry.tractus-x-06.arena2036.de/swagger-ui/index.html>
- BaSyx-AAS-Web-UI
 - <https://aas-webui.tractus-x-06.arena2036.de/>

9.3.4.8 BaSyx-Integration in Use-Case und Infrastruktur

Die Services, die in den Einzelnen BaSyx-Komponenten beinhalten, sind in der folgenden Grafik aufgezeichnet. Zusätzlich sind die äquivalenten Services und APIs des Catena-X Digital-Twin Stacks markiert.

Es gibt mehrere Überschneidungen an Digital-Twin-Services in Catena-X, sowie BaSyx bzw. dem IDTA-Standard. Wir haben daher für unseren Use-Case sowohl von Catena-X als auch BaSyx Software-Komponenten genutzt. Konkret nutzen wir von BaSyx primär die AAS-Environment, sowie das inkludierte Submodel-Repository, und von Catena-X die Digital Twin Registry, mit der AAS-Registry API, sowie die Integration mit dem EDC.

Die Überschneidungen an äquivalenten Services, kann man auf der folgenden Abbildung sehen:



VWS4LS
Catena-X und Basyx

Datum:
20. Juli 2024
Autoren:
Dmitrij Drandarov

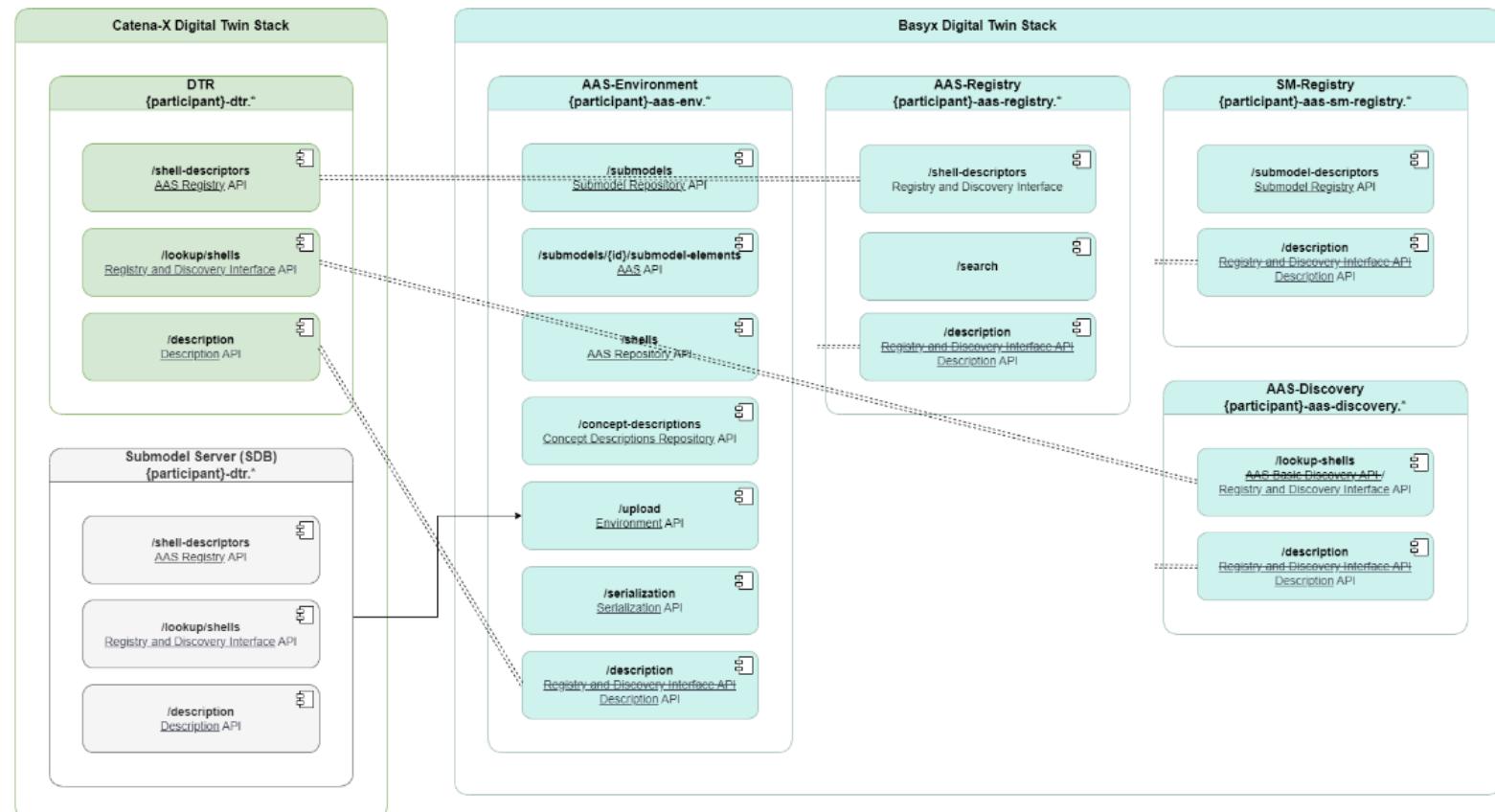


Abbildung 9-13: BaSyx-Integration in Use Case

9.3.5 Umsetzung des Use Cases

Für die Modellierung der Digitalen Zwillinge wurden im Projekt VWS4LS spezifizierte Submodelltemplates der IDTA verwendet.

Catena-X ist hauptsächlich Use-Case getrieben. Die Use Cases beschreiben ihre Anforderungen an Submodels in sog. "Aspektmodellen". Möchte man an einem Use Case von Catena-X teilnehmen, müssen die jeweiligen [Catena-X-Standards](#) eingehalten werden. Der Standardisierungsprozess wird vom Catena-X Verein vorgenommen.

Beide Modellierungskonzepte zielen darauf ab, Daten und Prozesse innerhalb eines Ökosystems zu standardisieren und damit die Interoperabilität und den Datenaustausch zwischen verschiedenen Akteuren und Systemen zu gewährleisten.

Im Folgenden gilt es daher zu klären, welche Unterschiede sich zwischen den IDTA-Submodellen und den Catena-X-Aspektmodellen ergeben und wie diese für eine Anwendung in Catena-X angepasst werden müssen.

9.3.5.1 Catena-X Aspektmodelle

Catena-X beschreibt das Konzept der "[Aspektmodelle](#)"⁶⁵, um domänen spezifische Sichten oder Aspekte eines digitalen Zwillings zu definieren, die für verschiedene Anwendungsfälle innerhalb des Catena-X-Ökosystems wesentlich sind. Aspektmodelle bieten eine strukturierte Möglichkeit zur Erfassung spezifischer Geschäftsprozesse und Datenanforderungen.

Um ein Datenmodell zu spezifizieren, wird das [Semantic Aspect Meta Model](#) (SAMM)⁶⁶ verwendet, welches unter Verwendung des [Resource Description Format \(RDF\)](#)⁶⁷ und der [Terse RDF Triple Language Syntax \(Turtle\)](#)⁶⁸ spezifiziert, zusammen mit Validierungsregeln in der [Shapes Constraint Language \(SHACL\)](#)⁶⁹. Aspektmodelle werden ebenfalls in RDF/Turtle spezifiziert, wobei die SAMM-Semantik befolgt wird.

Abbildung 9-14 zeigt beispielhaft die grafische Darstellung des Aspektmodells „PartTypeInformation“. Dafür kann der [Aspect Model Editor](#)⁷⁰ verwendet werden oder die html-Datei des semantischen Modells aus dem GitHub Repository [eclipse-tractusx](#)⁷¹ geöffnet werden. In Abbildung 9-15 sieht man das entsprechende JSON dazu.

⁶⁵ https://catena-x.net/fileadmin/user_upload/Standard-Bibliothek/Archiv/8_PC_Semantics_v2.1/SEM-002_BAMM_PlatformCapabilitySemantics_v2.1.pdf

⁶⁶ <https://eclipse-esmf.github.io/samm-specification/2.1.0/index.html>

⁶⁷ <https://www.w3.org/RDF/>

⁶⁸ <https://www.w3.org/TR/2014/REC-turtle-20140225/>

⁶⁹ <https://www.w3.org/TR/shacl/>

⁷⁰ <https://eclipse-esmf.github.io/ame-guide/introduction.html>

⁷¹ <https://github.com/eclipse-tractusx/slvt-semantic-models/tree/main>

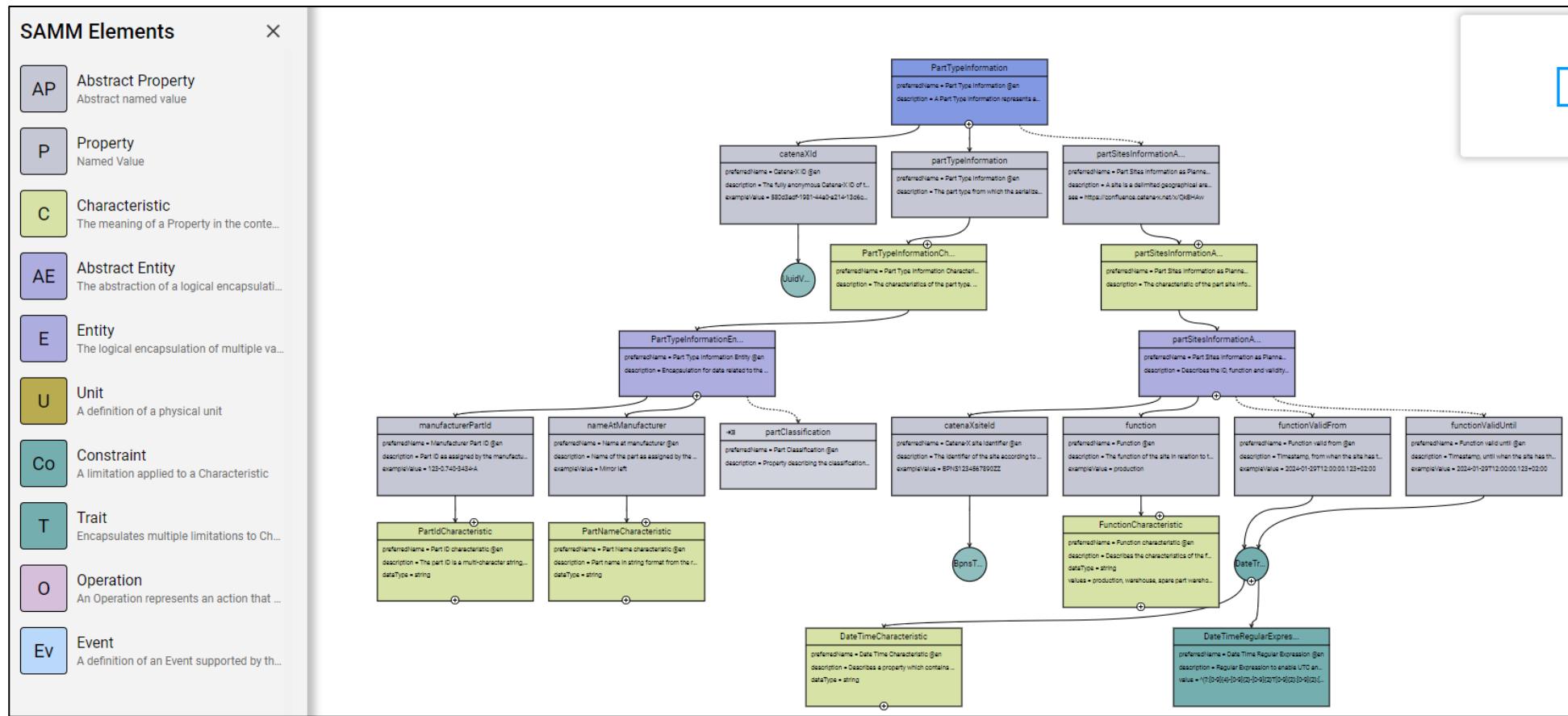


Abbildung 9-14: Grafische Darstellung des Aspektmodells `PartTypeInformation.ttl`

```

1   {
2     "partTypeInformation" : {
3       "partClassification" : [ {
4         "classificationStandard" : "IEC",
5         "classificationID" : "61360- 2:2012 ",
6         "classificationDescription" : "Standard data element types with associated classification scheme for electric components."
7       }],
8       "manufacturerPartId" : "123-0.740-3434-A",
9       "nameAtManufacturer" : "Mirror left"
10    },
11    "partSitesInformationAsPlanned" : [ {
12      "functionValidUntil" : "2024-01-29T12:00:00.123+02:00",
13      "catenaXsiteId" : "BPNS1234567890ZZ",
14      "function" : "production",
15      "functionValidFrom" : "2024-01-29T12:00:00.123+02:00"
16    }],
17    "catenaXId" : "580d3adf-1981-44a0-a214-13d6ceed9379"
18  }

```

Abbildung 9-15: Eigenschaften eines Submodells als JSON

9.3.5.2 IDTA-Submodelle

Die IDTA verwendet "**Submodel Templates**", um die Darstellung von Informationen innerhalb einer Asset Administration Shell (AAS), einer digitalen Darstellung von Assets, zu standardisieren. Das Submodel-Konzept der IDTA ist breiter angelegt, um verschiedene Industriebereiche abzudecken. Die Templates definieren spezifische Datenstrukturen und Semantiken für verschiedene Arten von Informationen (z. B. technische Daten, Wartungspläne), wodurch der Austausch und die Interpretation von Daten über verschiedene Plattformen und Branchen hinweg erleichtert wird. Die Schablonen sollen die Interoperabilität erleichtern, indem sie einen gemeinsamen Rahmen bieten, der an die verschiedenen industriellen Bedürfnisse angepasst werden kann. Alle registrierten AAS Submodel Templates werden auf der [IDTA-Website](#) gelistet und im [IDTA-Github](#)⁷² bereitgestellt.

9.3.5.3 Verwendung von Submodel Templates der IDTA in Catena-X

Catena-X schreibt dazu:

- *"In Catena-X the semantics of a Submodel is described via an Aspect Model conformant to standard CX-0003, preferable by using standardized properties conformant to standard CX-0044."*

Der Catena-X-Standard [CX-0003 „SAMM Aspect Meta Model v1.1.0“⁷³](#) definiert:

- *"Every aspect model released or standardized in Catena-X MUST be maintained in Tractus-X - SLDT Semantic Models."*⁷⁴
- *"Every aspect model in Tractus-X - SLDT Semantic Models that has the status "released" or "standardized" MUST be validated without errors against the Semantic Aspect Meta Model [...]."*
- *"Every aspect of a digital twin registered in a digital twin registry (see standard CX-0002) accessible in the Catena-X data space MUST have a semantic description (semantic ID) that is conformant to the unique identifier of the SAMM aspect model associated to it."*

Wenn das Projekt VWS4LS einen eigenen Use Case für Catena-X definieren möchte, sollten die verwendeten Submodelle in Catena-X gemäß des Standardisierungsprozesses standardisiert werden. Dann müssen diese dem Standard CX-0003 folgen (s.o.).

Für den Demonstrationszweck können die VWS4LS Submodelle für diesen Use Case technisch übertragen werden, entsprechen aber (außer dem Submodel „Nameplate“) nicht den Catena-X-Spezifikationen und können so nicht von Catena-X Lösungen interpretiert werden.

⁷² <https://industrialdigitaltwin.org/en/content-hub/submodels>

⁷³ <https://catenax-ev.github.io/docs/next/standards/CX-0003-SAMMSemanticAspectMetaModel>

⁷⁴ <https://github.com/eclipse-tractusx/sltd-semantic-models>

9.3.5.4 Digitale Zwillinge und AAS in Catena-X

In Catena-X basieren die APIs auf den Spezifikationen der Asset Administration Shell (AAS) Spezifikationen der IDTA⁷⁵. Um Catena-X konform zu sein, werden in der Demonstrator-Umsetzung die einzelnen Submodelle in der DTR registriert. Diese Registrierung zählt im Kontext von Catena-X als „Erschaffung eines Digitalen Zwilling“. Grundsätzlich wird zum aktuellen Entwicklungsstand nicht auf Asset-Ebene zugegriffen, sondern auf die einzelnen Submodelle, die in der DTR zu einem Asset „gebündelt“ werden können. Das JSON-Beispiel im [Digital Twin KIT](#)⁷⁶ verdeutlicht das. Die AAS an sich wird daher nicht angesprochen. In der DTR steht lediglich der AAS-Descriptor, der wiederum die verschiedenen Submodel-Descriptoren mit den entsprechenden Endpunkten enthält.

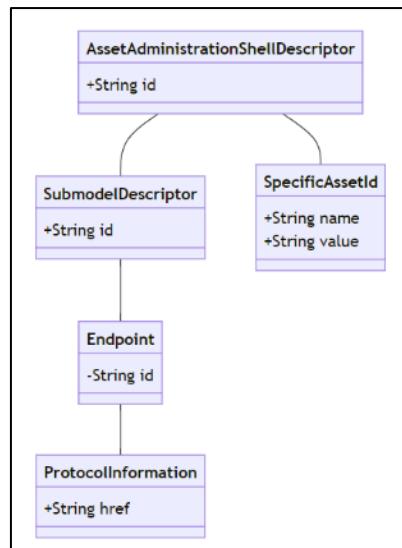


Abbildung 9-16: [Tractus-X Digital Twin Registry - Asset Administration Shell Domain Model](#)⁷⁷

Im [Digital Twin KIT](#) von Tractus-X steht zwar, dass ein Data Provider ein Daten Asset per Submodel oder Bundle erstellen kann, gleichzeitig wird aber ausgesagt:

- “There is no normative guidance on how to register multiple Submodels bundled together yet. These bundles may include all the Submodels of a specific semanticId, all Submodels of an asset or any other arbitrary quality. This may be added to [CX-0002](#) in future iterations.”⁷⁸

Als Konsequenz werden auch für den Demonstrator die Submodels als einzelne Assets angelegt.

9.3.5.5 Der Industry Core von Catena-X

Um zu verhindern, dass jeder Use-Case in Catena-X seine Digitalen Zwillinge unterschiedlich beschreibt, wurde der Industry Core ins Leben gerufen. „Core“ deshalb, weil er den Kern der Industrie, nämlich die Teile/Komponenten betrifft.

Catena-X beschreibt dazu:

- Der Industry Core ist eine gemeinsame [semantische] Basis für Use Cases, die digitale Zwillinge und Aspektmodelle in Catena-X nutzen⁷⁹.

⁷⁵ <https://eclipse-tractusx.github.io/docs-kits/kits/Digital%20Twin%20Kit/Operation%20View%20Digital%20Twin%20Kit>

⁷⁶ <https://eclipse-tractusx.github.io/docs-kits/kits/Digital%20Twin%20Kit/Software%20Development%20View/dt-kit-software-development-view#registering-a-new-twin>

⁷⁷ <https://github.com/eclipse-tractusx/sldt-digital-twin-registry/tree/main/main/docs#asset-administration-shell-domain-model>

⁷⁸ <https://eclipse-tractusx.github.io/docs-kits/kits/Digital%20Twin%20Kit/Software%20Development%20View/dt-kit-software-development-view#submodel-as-edc-data-asset>

⁷⁹ <https://catenax-ev.github.io/docs/next/standards/CX-0126-IndustryCorePartType>

- Der Industry Core beschreibt ein physisches Teil, eine Komponente oder ein Material auf Typ- und/oder Instanzebene, macht es im Netz identifizierbar und auffindbar und ermöglicht die Navigation über mehrere Ebenen hinweg⁸⁰.

Der Industry Core hat aktuell zwei Standards definiert. Darin wird einmal die Beschreibung eines Teils auf Typ-Ebene ([CX-0126](#)) und einmal auf Instanz-Ebene ([CX-0127](#)) festgelegt sowie jeweils die Verbindung zu „Kind-Teilen“, also Komponenten, die verbaut wurden.

Zusätzlich müssen folgende Punkte aus dem Standard CX-0126 beachtet werden:

- “The asset's globalAssetId MUST be equal to the unique id used in Catena-X.”
- “The following specific asset IDs⁸¹ not marked as optional MUST be available when registering a digital twin or when adding the above mentioned submodels to an existing twin for a part type in order to allow discovery.”

Da auch im Projekt VWS4LS physikalische Komponenten betrachtet werden (bspw. Leitungen und Stecker), ist es zu empfehlen, diese mittels der Industry Core Standards zu beschreiben, falls eine Teilnahme an Catena-X angestrebt wird.

Für den Demonstrator werden daher alle Digitalen Zwillinge mit dem Submodel „PartTypeInformation“ und „SingleLevelBomAsPlanned“ ergänzt.

9.3.5.6 OEM-Anforderungen als Submodel

Der OEM ist mit seinen Anforderungen der Trigger für einen Entwicklungsauftrag beim Tier 1. Für die vereinfachte Demonstration in diesem Projekt sind die Anforderungen eine Liste an autorisierten Entwicklern. Dafür wird hier eine Dummy-AAS „**OEM_SupplierRequirements**“ mit dem mit dem speziell dafür definierten Submodel „**AuthorizedSuppliers**“ erstellt. Hier sind die BPNLs der vom OEM autorisierten Business Partner hinterlegt (siehe folgende Abbildung).

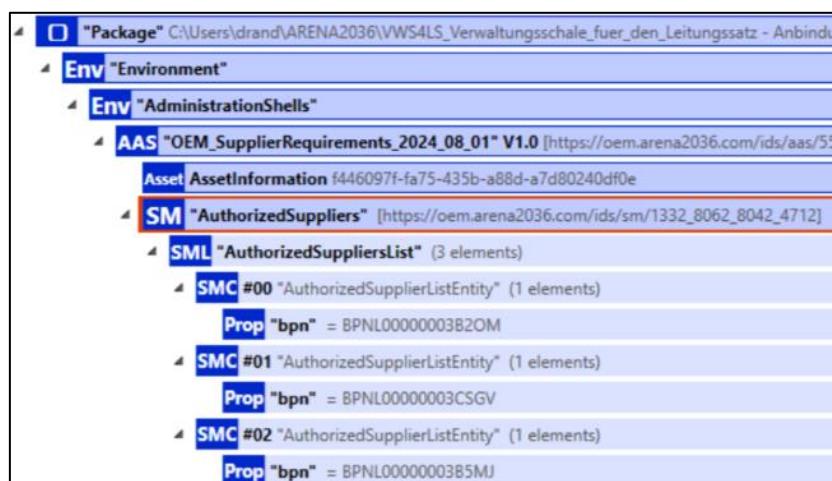


Abbildung 9-17: Die Anforderungen des OEM als AAS mit Submodell

⁸⁰ <https://eclipse-tractusx.github.io/docs-kits/category/industry-core-kit>

⁸¹ <https://catenax-ev.github.io/docs/next/standards/CX-0126-IndustryCorePartType#214-digital-twins-and-specific-asset-ids>

9.3.5.7 Aggregation des Digitalen Zwillings

Nachdem der Tier 1 sowohl Zugriff auf die Anforderungen des OEMs als auch auf die bereitgestellten Angebote der Tier 2 hat, erstellt dieser nun einen aggregierten Digitalen Zwilling. Dieser basiert auf den Submodels des Industry Cores von Catena-X (Details siehe Abschnitt zum Industry Core):

- [PartTypeInformation](#)⁸²
- [SingleLevelBomAsPlanned](#)⁸³

Das Submodel „PartTypeInformation“ enthält die Catena-X ID (entspricht der Global Asset ID) des aggregierten Digitalen Zwillings sowie die Teilenummer und den Namen.

Die eigentliche Aggregation findet über das Submodel „SingleLevelBomAsPlanned“ statt. Dieses stellt die Stückliste (BoM) dar. Hier erfolgt der Link auf alle Komponenten (Catena-X ID's der AASen), die im aggregierten Digitalen Zwilling verbaut werden können. Die folgende Abbildung zeigt dies beispielhaft (childItems):

| AAS "ZusammenbauLeitungssatz" V. [https://ter1.arena2036.com/ids/aas/7131_9062_8042_4535] of urn:uuid:d39a99cd-e6b3-4 | | Element Content |
|--|--|--|
| Asset AssetInformation urn:uuid:d39a99cd-e6b3-4ad1-8890-2f89b7046aa4 | | Submodel Element (Property) |
| SM <T> "PartTypeInformation" V. [urn:sammlio.catenaX.part_type_information:1.0.0#PartTypeInformation/submodel] | | Referable: |
| Prop "catenaXId" = urn:uuid:d39a99cd-e6b3-4ad1-8890-2f89b7046aa4 | | idShort: catenaXId displayName: [en] Catena-X Identifier |
| SMC "PartTypeInformationEntity" (2 elements) | | HasExtension: |
| Prop "manufacturerPartId" = 123-456 | | Semantic ID: semanticId: (ConceptDescription) urn:sammlio.catenaX.single_level_bom_as_planned:3.0.0#catenaXId |
| Prop "nameAtManufacturer" = Leitungssatz Variante 1 | | Supplemental Semantic IDs: |
| SM <T> "SingleLevelBomAsPlanned" V. [urn:sammlio.catenaX.single_level_bom_as_planned:3.0.0#SingleLevelBomAsPlanned] | | Qualifiable: |
| Prop "catenaXId" = urn:uuid:d39a99cd-e6b3-4ad1-8890-2f89b7046aa4 | | HasDataSpecification (Reference): |
| SML "childItems" (7 elements) | | ConceptDescription |
| SMC #00 "Kostal_Terminal_PLK_1-2_0,5mm_2024_01_25" (6 elements) | | Referable: |
| Prop "catenaXId" = urn:uuid:91d54b9c-d950-4309-85e0-2de72d4797eb | | idShort: catenaXId displayName: [en] Catena-X Identifier |
| Prop "businessPartner" = BPNI00000003820M | | HasExtension: |
| SMC "ItemQuantityEntity" (2 elements) | | Identifiable: |
| Prop "createdOn" = 2024-08-01 | | id: urn:sammlio.catenaX.single_level_bom_as_planned:3.0.0#catenaXId id (Base64): dXJuOnNbW06aW8uY2F0ZW5heC5zaW5nbGVfbGV2ZWxfym9t2FzX3BsYW5uZWQ6My4 |
| Prop "lastModifiedOn" = 2024-08-01 | | HasDataSpecification (records of embedded data specification): |
| SMC "ValidityPeriodEntity" (2 elements) | | dataSpec.[0] / Reference: dataSpec.[0]: (GlobalReference) https://admin-shell.io/DataSpecificationTemplates/DataSpecificationIec6 |
| SMC #01 "Kostal_Terminal_PLK_35mm²_HMI_2024_01_25" (6 elements) | | dataSpec.[0] / Content: Data Specification Content IEC61360: preferredName: [en] Catena-X Identifier shortName: [en] catenaXId dataType: STRING definition: [en] The Catena-X ID of the given part (e.g. the component), valid for the Catena-X dataspec |
| SMC #02 "Kostal_Terminal_PLK_50mm²_HMI_2024_01_25" (6 elements) | | Property |
| SMC #03 "Conoflex_Leitung_0,5mm²_HMI_2024_02_12" (6 elements) | | valueType: xs:string value: urn:uuid:91d54b9c-d950-4309-85e0-2de72d4797eb |
| SMC #04 "Conoflex_Leitung_35mm²_HMI_2024_01_25" (6 elements) | | |
| SMC #05 "Conoflex_Leitung_50mm²_HMI_2024_01_25" (6 elements) | | |
| SMC #06 "Kostal_GEHAEUSE_PLK_1-2_2024_01" (6 elements) | | |

Abbildung 9-18: Zusammenbau Leitungssatz mit Stückliste

⁸² https://github.com/eclipse-tractusx/sldt-semantic-models/tree/main/io.catenax.part_type_information/1.0.0/gen

⁸³ https://github.com/eclipse-tractusx/sldt-semantic-models/tree/main/io.catenax.single_level_bom_as_planned/3.0.0/gen

9.3.5.8 Zugriffskontrolle (Security)

Das Projekt VWS4LS hat die Anforderung, dass der Zugriff auf einzelne Submodels beschränkt werden kann. Ergänzend zu einer „Role-Based Access Control“ (RBAC) ist also auch eine Art „[Attribute-Based Access Control](#)“ (ABAC) [66] auf Submodell-Ebene notwendig. Hierfür werden im Folgenden mögliche Konzepte vorgestellt. Es wird dabei von folgenden Annahmen ausgegangen:

- Ein (externer) Benutzer möchte auf eine AAS zugreifen
- Das Identitätsmanagement wird von einem Identitätsanbieter (IdP) durchgeführt
 - Der IdP stellt ein Access Token zur Verfügung, das für den AAS-Server relevante Angaben enthält
 - Das Token (JWT, JSON Web Token) wird vom IdP signiert
- Der AAS-Server vertraut dem Identity Provider
- Der Zugang zur AAS wird auf der Grundlage des Access Token und der Zugangsrichtlinien gewährt
- Die Zugangsrichtlinien (Policies) werden vom AAS-Verantwortlichen eingerichtet

9.3.5.8.1 IDTA Security Spezifikation

Im Kern müssen zu dem Thema AAS-Security die folgenden Fragestellungen geklärt werden:

- 1) Wo sollen die Zugangsrichtlinien (Policies) gespeichert werden?
 - Innerhalb der AAS selbst?
 - In einem separaten Policy-Repository?
- 2) Wie können die Zugangsrichtlinien abgebildet werden?
 - Mit AAS-Elementen?
 - Mit bestehenden Sprachen wie [XACML](#)⁸⁴ oder [ODRL](#)⁸⁵?
- 3) Wie wird die Zugangsrichtlinien-Entscheidung implementiert?
 - Mit einem Interpreter, der zum Zeitpunkt der Zugriffsentscheidung aufgerufen wird?
 - Integriert in Such- und Abfragestrukturen?

Folgende Dokumente adressieren bislang das Thema „Security für Verwaltungsschalen“:

- [Security der Verwaltungsschale](#) (02/2018)
- [Zugriffssteuerung für Industrie 4.0-Komponenten zur Anwendung von Herstellern, Betreibern und Integratoren](#) (12/2018)
- [Sicherer Downloadservice](#) (12/2020)
- [Details of the Asset Administration Shell Part 1, Chapter 6, ABAC & RBAC](#) (05/2022)

In „[Details of the Administration Shell](#)“ war bis V30RC02 ein Security-Konzept für sowohl RBAC als auch ABAC enthalten. Hierfür existiert auch eine prototypische AAS-Server-Implementierung (<https://v3.admin-shell-io.com/>), siehe Abbildung 9-19, jedoch keine Benutzeroberfläche, um Zugriffsrichtlinien für AAS-Elemente zu konfigurieren.

⁸⁴ <https://en.wikipedia.org/wiki/XACML>

⁸⁵ <https://en.wikipedia.org/wiki/ODRL>



Abbildung 9-19: AAS-Security (Quelle: <https://v3.admin-shell-io.com/>)

Derzeit wird das AAS-Security-Konzept in der IDTA komplett überarbeitet und soll durch eine neue Spezifikation „Security“ ersetzt werden. Hierfür werden in der IDTA-Arbeitsgruppe vier Varianten diskutiert:

1. Security Meta Model
2. Security Submodel
3. External Model [XACL](#)
4. External Model [ODRL](#), basierend auf Mechanismen, die auch CATENA-X nutzt.

Eine erste Version der neuen Security-Spezifikation wird für Anfang 2025 erwartet.

Die generelle Zielsetzung der angestrebten Security-Spezifikation wird in der nachfolgenden Abbildung 9-20 beschrieben:

Für die technische Umsetzung wird als wesentlicher Bestandteil ein "[OpenAuth2.0 Authorization Framework](#)" gesehen. Für das Identitätsmanagement wird [OpenID Connect](#) vorgeschlagen. Die Kommunikation basiert dabei auf JSON Web Tokens, die Claims entsprechend eines Datenraums enthalten. Diese Claims können dann in ABAC-Security-Regeln verwendet werden und somit den Zugriff regeln. Das soll für Repository und Registry gleichermaßen gelten. Access Rules sollen auch für Properties und nicht nur für ganze Submodelle möglich sein.

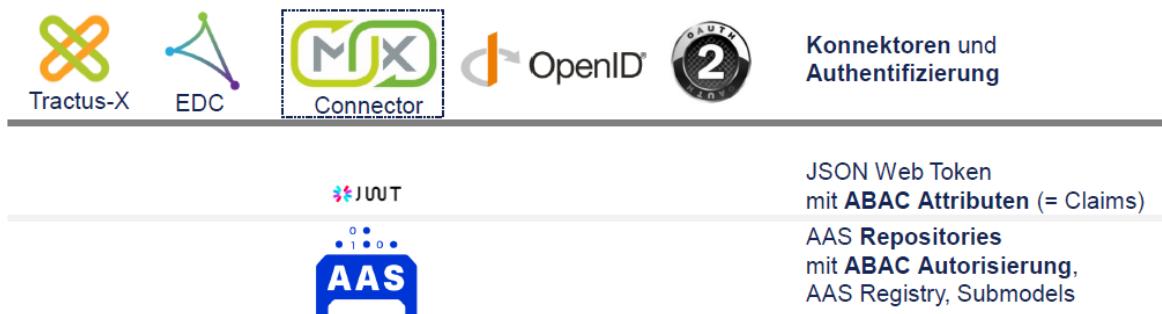


Abbildung 9-20: Grundsätzliche Interaktion zwischen Konnektoren und Verwaltungsschalen

(Quelle: A. Orzelski)

9.3.5.8.2 Security in BaSyst

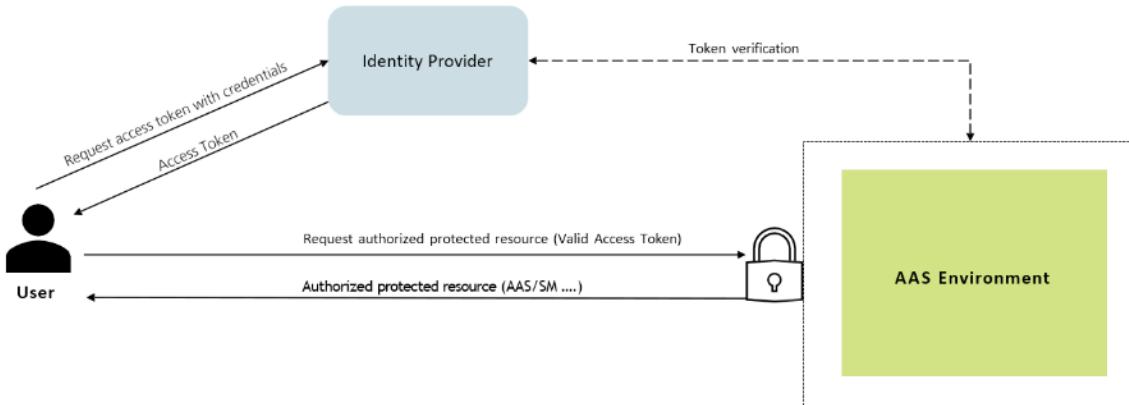


Abbildung 9-21: Dataflow RBAC in BaSyst

Definitionen für eine Attribut basierende Zugriffskontrolle (Attribute Based Access Control = ABAC) sind von der IDTA derzeit noch nicht abgeschlossen. Das bedeutet, dass auf die Standardisierung von der IDTA gewartet wird, bevor eine Implementierung in BaSyst vorgenommen wird.

Derzeit ist in BaSyst eine einfache rollenbasierte Zugriffskontrolle (Role Based Access Control = RBAC) implementiert. Jeder Service (AAS Discovery, AAS-Environment, AAS Concept Description etc.) benötigt eine eigene Konfigurationsdatei, welche in JSON-Format die Rollen und Rechte spezifisch für bestimmte AAS mit IDs oder Wildcards beschreibt. Hierfür existiert ein dediziertes [Konfigurationsdateiformat](#) sowie ein SDK-Beispiel namens „[BaSystSecured](#)“, welches den Authorisierungs-Server [Keycloak](#) verwendet.

In Abbildung 9-22 wird gezeigt, wie die Authentifizierung in BaSyst konzipiert wurde. In dem Szenario sind Admins für jeden Server (AAS Registry, AAS Repository und SM Repository) vorhanden sowie die Nutzer für AAS Registry, AAS Repository und Submodel Repository. Alle Nutzer authentifizieren sich mittels eines AccessToken von einem Authorisierungs-Server (hier [Keycloak](#)).

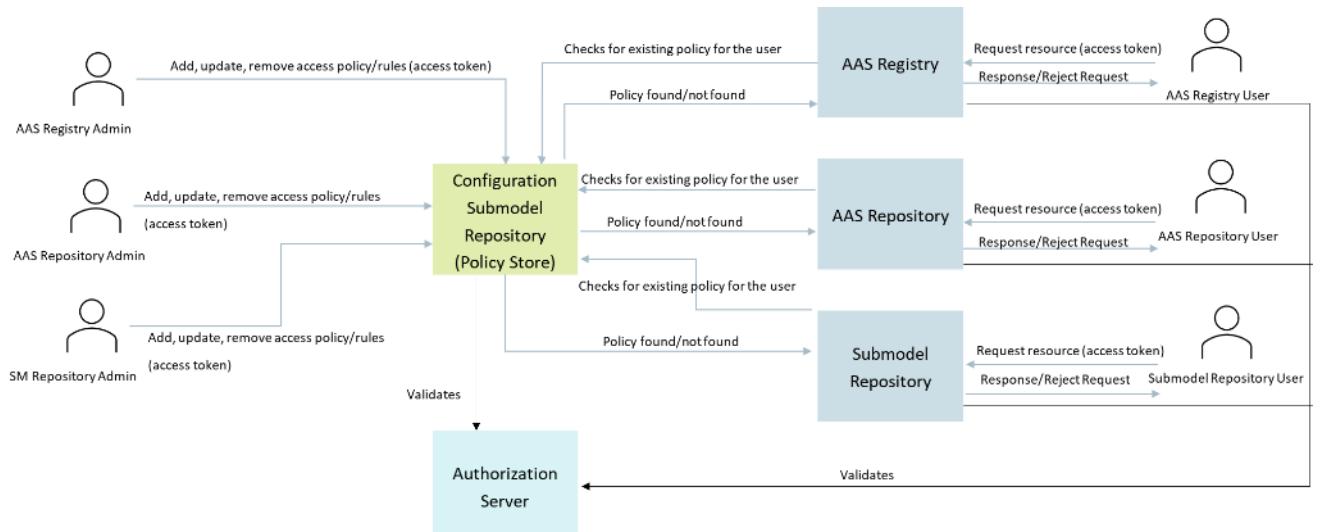


Abbildung 9-22: Zugriffskontrolle – Konzept für BaSyst

9.3.5.8.3 Security in Catena-X

Datenlieferanten in Catena-X verwenden sog. „[Policies](#)“, um den Zugang (Access-Policies) und die Nutzung (Usage-Policies) von Daten festzulegen.⁸⁶ Die Policies folgen der ODRL⁸⁷ (Open Digital Rights Language) und werden in JSON-LD beschrieben.

Weiter wird in [SC-002 ausgeführt](#):

„A Contract Definition is the connection between a set of [Assets](#) with one Access Policy and one Contract Policy.“

Die „Contract Definition“ ist dann die logische Verbindung zwischen Assets und Policy⁸⁸.

Im EDC können sogenannte Data Assets bereitgestellt werden. Ein Asset ist dabei die grundlegende Darstellung einer beliebigen Backend-Schnittstelle im EDC⁸⁹.

Ein Datenanbieter kann ein Data Asset pro Submodel erstellen⁹⁰. Diesem Data Asset kann mittels einer Contract Definition eine Policy zugeordnet werden.

Eine Zugriffsbeschränkung für einzelne Submodels im Projekt VWS4LS kann umgesetzt werden, indem den Assets (Submodels) entsprechende Access-Policies im EDC zugeordnet werden.

Ein Peer-to-Peer Datenaustausch in Catena-X erfolgt immer über Konnektoren, die das von der IDSA spezifizierte Dataspace Protocol (DSP) implementieren⁹¹. Für dieses Projekt wird die Referenzimplementierung „[Tractus-X EDC](#)“ verwendet⁹².

9.3.5.8.4 Security für dieses Projekt

Aufgrund der Datenarchitektur von Catena-X kann jedem Submodel eine individuelle Policy zugeordnet und somit der Zugriff auf Submodel-Ebene beschränkt werden, wie es auch der Anforderung an dieses Projekt entspricht (siehe Abschnitt 9.3.5.8.1).

Die Zugriffsbeschränkung für diesen demonstrativen Use Case wird mit dem Konzept der Policies nach Catena-X umgesetzt. Die anderen Konzepte (ABAC) finden sich noch in der Definition und können daher in diesem Projekt noch nicht umgesetzt werden.

⁸⁶ https://github.com/eclipse-tractusx/tractusx-edc/blob/main/docs/usage/management-api-walkthrough/02_policies.md

⁸⁷ <https://w3c.github.io/poe/model/>

⁸⁸ https://github.com/eclipse-tractusx/tractusx-edc/blob/main/docs/usage/management-api-walkthrough/03_contractdefinitions.md#creating-a-contract-definition

⁸⁹ https://github.com/eclipse-tractusx/tractusx-edc/blob/main/docs/usage/management-api-walkthrough/01_assets.md

⁹⁰ <https://eclipse-tractusx.github.io/docs-kits/kits/Digital%20Twin%20Kit/Software%20Development%20View/dt-kit-software-development-view#submodel-as-edc-data-asset>

⁹¹ <https://github.com/eclipse-tractusx/tractusx-edc/tree/main/docs/usage/management-api-walkthrough#introduction>

⁹² <https://github.com/eclipse-tractusx/tractusx-edc>

9.3.5.9 Data Provisioning & Data Consumption

Der beschriebene Use Case besteht grundsätzlich aus den Szenarien „Data Provisioning“ und „Data Consumption“ (siehe Abschnitt 9.3.3).

9.3.5.9.1 Data Provisioning

Ein Data-Provisioning besteht aus mehreren aufeinander folgenden Schritten. Sie beinhalten z.B. das Bereitstellen des Submodells, Registrieren von Digitalen Zwillingen, sowie Erstellungen von EDC Contracts, sowie den dazugehörigen Assets und Policies. Im Folgenden werden die genauen Schritte, die durchzuführen sind, detailliert beschrieben.

Für den sequenziellen Ablauf im gesamten Use-Case siehe Kapitel 9.3.5.9.4 Sequenzieller Ablauf des Use Cases.

Man beachte, dass das Data-Provisioning dann für alle Participants, die Daten providen wollen, durchgeführt werden muss.

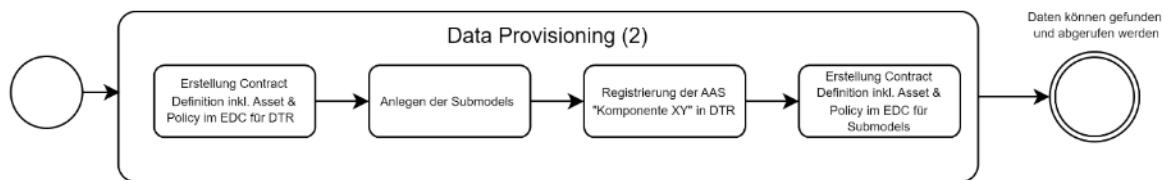


Abbildung 9-23: Data Provisioning

In den folgenden Schritten ist oft eine Authentifizierung mit dem Keycloak erforderlich. Für diese müssen in den entsprechenden API-Calls u.A. folgende Properties gesetzt werden, bspw. für den OEM:

- **clientId**: satest02
- **clientAlias**: EDC-MIW_BPN_OEM_A
- **clientSecret**: pyFUZP2L9UCSVJUSchCn3ZEgy2PgyEpg
- **clientSecretAlias**: edc-miw-keycloak-secret
- **tokenUrl**: <http://centralidp.tractus-x-06.arena2036.de/auth/realms/CX-Central/protocol/openid-connect/token>

9.3.5.9.1.1 Anlegen der Submodels

Zuerst müssen die entsprechenden Submodelle über die BaSyx Submodel Repository API hochgeladen werden. Dies muss für jedes Submodell in den oben aufgelisteten Test-Daten durchgeführt werden. Dafür erfolgt ein POST Request an die entsprechende URL. Der Request für den Upload kann bspw. folgendermaßen aussehen (der eigentliche Inhalt ist für die Lesbarkeit gekürzt):

- **POST** <https://aas-env.tractus-x-06.arena2036.de/submodels>

```
{
  "idShort": "Nameplate",
  "id": "www.kostal.com/sm/1215_8002_2042_1144",
  "kind": "Instance",
  "semanticId": {
    "type": "ExternalReference",
    "keys": [
      {
        "type": "Submodel",
        "value": "https://admin-shell.io/zvei/nameplate/1/0/Nameplate"
      }
    ]
  },
  "submodelElements": [
    {
      ...
    }
  ]
}
```

```

        <...>
        <...>
    }
],
"modelType": "Submodel"
}

```

Die Submodelle können mir folgendem Request überprüft werden:

GET <https://aas-env.tractus-x-06.arena2036.de/submodels>

Wenn man ein bestimmtes Submodell abfragen will, kann man die Submodell ID mit base64 encoden und das an die oben genannten Request-URL anhängen. Aus:

www.kostal.com/sm/1215_8002_2042_1144

wird mit base64-Encoding:

d3d3Lmtvc3RhbC5jb20vc20vMTIxNV84MDAyXzIwNDJfMTE0NA==

Der daraus entstehende Request ist:

- GET <https://aas-env.tractus-x-06.arena2036.de/submodels/d3d3Lmtvc3RhbC5jb20vc20vMTIxNV84MDAyXzIwNDJfMTE0NA==>
 - Das Ergebnis entspricht dem vorher hochgeladenen Submodel.

9.3.5.9.1.2 Registrierung der AAS in DTR

Für die Registrierung einer AAS in Catena-X bzw. der Digital Twin Registry wird das entsprechende Deployment einer DTR genutzt. Dafür wird konkret der Endpoint /shell-descriptor benutzt. Mit einem POST-Request an die entsprechende URL, wird ein AAS-Deskriptor definiert, der die oben angelegten Submodels referenziert.

- POST <https://oem-dtr.tractus-x-06.arena2036.de:443/api/v3.0/shell-descriptors>
 - Beispiel-Payload

```

{
  "id": "urn:uuid:e5c96ab5-896a-1234-8761-efd74777ca97",
  "idShort": "<AAS_ID>",
  "specificAssetIds": [
    {
      "name": "manufacturerPartId",
      "value": "123-345-567103",
      "externalSubjectId": {
        "type": "ExternalReference",
        "keys": [
          {
            "type": "GlobalReference",
            "value": "BPNL00000003AYRE"
          }
        ]
      }
    }
  ],
  "submodelDescriptors": [
    {
      "id": "e5c96ab5-896a-482c-8761-efd74777ca97",
      "semanticId": {
        "type": "ExternalReference",
        "keys": [
          {
            "type": "GlobalReference",
            "value": "urn:bamm:io.catenax.material_for_recycling:1.1.0#MaterialForRecycling"
          }
        ]
      },
      "endpoints": [
        {

```

```

        "interface": "SUBMODEL-3.0",
        "protocolInformation": {
            "href": "https://aas-env.tractus-x-06.arena2036.de/submodels/d3d3Lmtvc3RhbC5jb20vc20vMTIxNV84MDAyXzIwNDJfMTE0NA==",
            "endpointProtocol": "HTTP",
            "endpointProtocolVersion": [
                "1.1"
            ],
            "subprotocol": "DSP",
            "subprotocolBody": "id=123;dspEndpoint=http://edc.control.plane/api/v1/dsp",
            "subprotocolBodyEncoding": "plain",
            "securityAttributes": [
                {
                    "type": "NONE",
                    "key": "NONE",
                    "value": "NONE"
                }
            ]
        }
    },
    {
        ...
        ...
    }
]
}

```

Weitere Submodel Deskriptoren...

Genau, wie bei der Submodel Repository API kann über die base64-encodete ID eine Shell-Description wieder angefragt werden:

- **GET** <https://oem-dtr.tractus-x-06.arena2036.de:443/api/v3.0/shell-descriptors/api/v3.0/shell-descriptors/dXJuOnV1aWQ6YmlyMWI5ZDktZjRiMi00ZDA4LWFmY2Ut-MTE0N2Y4MTcwNWF1>
 - Das Ergebnis entspricht dem vorher hochgeladenen Shell-Descriptor.

9.3.5.9.1.3 Registrierung des DTRs im EDC als Asset

Für die Interaktion zwischen der Digital-Twin-Registry und dem EDC, muss ersteres über die Management API als Asset registriert werden.

- **POST** <https://oem-controlplane.tractus-x-06.arena2036.de/management/v3/data/assets>
 - Beispiel-Payload

```

{
    "@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": "{{_.edcAssetId}}",
    "properties": {
        "dct:type": {
            "@id": "cx-taxo:DigitalTwinRegistry"
        },
        "cx-common:version": "3.0"
    },
    "privateProperties": {},
    "dataAddress": {
        "@type": "DataAddress",
        "type": "HttpData",
        "baseUrl": "{{_.url_backend}}"
    }
}

```

```

        "proxyQueryParams": "true",
        "proxyPath": "true",
        "proxyMethod": "false",
        "oauth2:tokenUrl": "http://centralidp.tractus-x-06.arena2036.de/auth/re-
alms/CX-Central/protocol/openid-connect/token",
        "oauth2:clientId": "satest02",
        "oauth2:clientSecretKey": "edc-miw-keycloak-secret"
    }
}

```

9.3.5.9.1.4 Erstellung Contract Definition inkl. Asset & Policy im EDC

Um eine AAS in Catena-X anzubieten, müssen nach dem Anlegen der Submodelle und der Registrierung der AAS, die Contract-Pipeline im jeweiligen EDC durchgeführt werden. Dafür wird die Management API mit ihren Entsprechenden Endpoints „/assets“, „/policydefinitions“ und „/contractdefinitions“ genutzt.

Zunächst wird ein Asset z.B. unter dem folgenden Endpoint erstellt:

POST <https://oem-controlplane.tractus-x-06.arena2036.de/management/v3/data/assets>

Eine Beispiel-Payload für eine Submodel-Definition als Asset ist im Folgenden dargestellt:

```
{
  "@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": "{{_.assetId}}",
  "properties": {
    "dct:type": {
      "@id": "cx-taxo:SubmodelBundle"
    },
    "cx-common:version": "3.0"
  },
  "privateProperties": {},
  "dataAddress": {
    "@type": "DataAddress",
    "type": "HttpData",
    "baseUrl": "{{_.url_backend}}",
    "oauth2:tokenUrl": "http://centralidp.tractus-x-06.arena2036.de/auth/realm/CX-
Central/protocol/openid-connect/token",
    "oauth2:clientId": "satest02",
    "oauth2:clientSecretKey": "edc-miw-keycloak-secret",
    "proxyQueryParams": "false",
    "proxyPath": "true",
    "proxyMethod": "false"
  }
}
```

Daraufhin wird eine Policy erzeugt, z.B. unter dem folgenden Endpoint:

POST <https://oem-controlplane.tractus-x-06.arena2036.de/management/v2/data/policydefinitions>

Eine Beispiel-Payload für eine Policy-Definition zu einem Asset ist im Folgenden dargestellt:

```
{
  "@context": [
    "https://www.w3.org/ns/odrl.JSONld",
    {
      "cx-policy": "https://w3id.org/catenax/policy/"
    }
  ],
  "@type": "Policy",
  "odrl:permission": [
    {

```

```

        "odrl:action": "USE",
        "odrl:constraint": {
            "@type": "LogicalConstraint",
            "odrl:and": [
                {
                    "@type": "Constraint",
                    "odrl:leftOperand": "BusinessPartnerNumber",
                    "odrl:operator": {
                        "@id": "odrl:eq"
                    },
                    "odrl:rightOperand": "<ALLOWED CONSUMER BPN>"
                },
                {
                    "leftOperand": "cx-policy:UsagePurpose",
                    "operator": "eq",
                    "rightOperand": "cx.core.digitalTwinRegistry:1"
                }
            ]
        }
    }
}

```

Als letztes wird der Contract erstellt, der das zuvor erstellte Asset und die Policy lediglich referenziert und logisch verknüpft. Dies geschieht z.B. unter dem folgenden Endpoint:

POST <https://oem-controlplane.tractus-x-06.arena2036.de/management/v2/data/contractdefinitions>

Mit der Beispiel-Payload:

```

{
    "id": "<CONTRACT_ID>",
    "accessPolicyId": "<POLICY_ID>",
    "contractPolicyId": "<POLICY_ID>",
    "criteria": [
        {
            "operandLeft": "asset:prop:id",
            "operator": "=",
            "operandRight": "<ASSET_ID>"
        }
    ]
}

```

9.3.5.9.2 Data Consumption

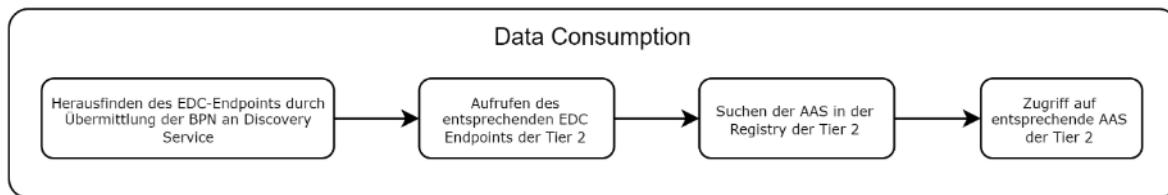


Abbildung 9-24: Data Consumption

Zum Suchen des EDC-Endpoints muss die BPN an den Discovery Service übermittelt werden. Zunächst müssen die BPNs über die BPN-Discovery, sowie Discovery-Services über den Discoveryfinder gefunden werden. Diese BPNs können dann bei der, durch den Discoveryfinder gefundene, EDC-Discovery zum Nachschlagen der EDC-Endpoints genutzt werden.

Für die folgenden Request müssen Authentifizierung-Tokens generiert werden. Diese müssen das in den HTTP-Header in der Form „Authentication: Bearer <token>“ zur Authentifizierung hinzugefügt werden.

Die Generierung dieser Tokens ist in „**9.3.4.6.3 Live-Anpassungen und Authentifizierung**“ beschrieben.

9.3.5.9.2.1 BPN-Discovery

- Authentifizierung über Token-Generierung für Client-ID `sa-c122-01`
- **POST** auf <https://bpndiscovery.tractus-x-06.arena2036.de/bpndiscovery/api/v1.0/administration/connectors/bpnDiscovery/search>

- Beispiel-Payload

```
{
  "searchFilter": [
    {
      "type": "oen",
      "keys": [
        "oen-1243",
        "oen-11"
      ]
    },
    {
      "type": "bpid",
      "keys": [
        "oen-satest05",
        "oen-satest02"
      ]
    }
  ]
}
```

- Beispiel-Response

```
{
  "bpns": [
    {
      "type": "oen",
      "key": "oen-satest05",
      "value": "BPNL00000003B0Q0",
      "resourceId": "972262d7-7e05-4578-936f-de236d7feb94"
    },
    {
      "type": "oen",
      "key": "oen-satest02",
      "value": "sBPNL00000003AYRE",
      "resourceId": "1b754aeb-c753-4adf-ae6d-52842f5a38b7"
    }
  ]
}
```

9.3.5.9.2.2 Discoveryfinder

- Authentifizierung über Token-Generierung für Client-ID `sa-c121-01`
- **POST** auf <https://discoveryfinder.tractus-x-06.arena2036.de/discoveryfinder/api/v1.0/administration/connectors/discovery/search>

- Beispiel-Payload

```
{
  "types": [
    "oen",
    "bpid",
    "bpn"
  ]
}
```

- Beispiel-Response

```
{
  "endpoints": [
    {
      "type": "bpn",
      "description": "Service to discover connector endpoints based on bpns",
      "endpointAddress": "https://portal-backend.tractus-x-06.arena2036.de/api/administration/Connectors/discovery",
      "documentation": "https://portal-backend.tractus-x-06.arena2036.de/api/administration/swagger/index.html",
      "resourceId": "a0b77e6d-1365-48cd-8f37-fed2e2394489"
    }
  ]
}
```

9.3.5.9.2.3 EDC-Discovery (Portal-Backend)

- Authentifizierung über Token-Generierung für Client-ID **sa-c11-reg-2**
- **POST** auf <https://portal-backend.tractus-x-06.arena2036.de/api/administration/Connectors/discovery>
 - Beispiel-Payload

```
[
  "BPNL00000003AYRE",
  "BPNL00000003B20M"
]
```

- Beispiel-Response

```
[
  {
    "bpn": "BPNL00000003AYRE",
    "connectorEndpoint": [
      "oem-controlplane.tractus-x-06.arena2036.de"
    ]
  },
  {
    "bpn": "BPNL00000003B20M",
    "connectorEndpoint": [
      "tiertwo-1-controlplane.tractus-x-06.arena2036.de"
    ]
  }
]
```

9.3.5.9.2.4 Aufrufen des entsprechenden EDC-Endpoints

Für das Aufrufen des entsprechenden Contracts, wird ebenfalls die Management API des EDC-Controlplanes genutzt. Konkret wird der Endpoint [/catalog/request](#) genutzt. Der Catalog kann folgendermaßen angefragt und gefiltert werden. Der Filter ist optional. Zunächst können wir nach einer registrierten Digital Twin Registry suchen:

- **GET** <https://oem-controlplane.tractus-x-06.arena2036.de/management/v2/catalog/request>

- Beispiel-Payload

```
{
  "@context": {
    "edc": "https://w3id.org/edc/v0.0.1/ns/"
  },
  "@type": "CatalogRequest",
  "counterPartyAddress": "{{provider-dsp-endpoint}}",
```

```

"protocol": "dataspace-protocol-http",
"querySpec": {
    "offset": 0,
    "limit": 50,
    "filterExpression": [
        {
            "@context": {
                "edc": "https://w3id.org/edc/v0.0.1/ns/"
            },
            "@type": "edc:Criterion",
            "operandLeft": "https://w3id.org/edc/v0.0.1/ns/id",
            "operator": "=",
            "operandRight": "{{assetId}}"
        }
    ]
}
}

```

Darauf folgend kann man auch einen Shell-Lookup auf die ausgehandelte Digital Twin Registry durchführen:

- **GET** <https://oem-dtr.tractus-x-06.arena2036.de/api/v3.0/lookup/shells/dXJuOnV1aWQ6MDM5ZDYwNWQtYjZiOC00M2I2LThmMTMtYzUwZDE1YTEzMGY5>

- Beispiel-Response

```
[
{
    "supplementalSemanticIds": [],
    "name": "manufacturerPartId",
    "value": "JJ-55",
    "externalSubjectId": {
        "type": "ExternalReference",
        "keys": [
            {
                "type": "GlobalReference",
                "value": "PUBLIC_READABLE"
            }
        ]
    }
}
]
```

Damit kann man ebenfalls einen spezifischen Shell-Descriptor abfragen. Siehe dafür Kapitel „9.3.5.9.1.2 Registrierung der AAS in DTR“.

9.3.5.9.3 Testdaten

Die in den folgenden Abschnitten beschriebenen Testdaten werden für den Use Case verwendet. Folgende Submodels mussten eigens für das Projekt erstellt werden, da sie vorher noch nicht existierten:

- PartTypeInformation
- SingleLevelBomAsPlanned
- OEM_SupplierRequirements

9.3.5.9.3.1 Testdaten von Kostal (Terminals)

| AAS | Submodels | Benötigte Informationen |
|---|----------------------------|---|
| Kostal_Terminal_MLK_1-2_0,5mm_2024_01_25 | PartTypeInformation | <pre> "catenaXId": "91d54b9c-d950-4309-85e0-2de72d4797eb" "manufacturerPartId": "32140734113" "nameAtManufacturer": "Kostal_MLK_1,2_32140734113" </pre> |
| | SingleLevelBomAsPlanned | Keine ChildItems, da unbekannt |
| | Nameplate [28] | Vom Projekt vorgegeben |
| | ContactInformation [67] | Vom Projekt vorgegeben |
| | HandoverDocumentation [68] | Vom Projekt vorgegeben |
| | TechnicalData [69] | Vom Projekt vorgegeben |
| Kostal_Terminal_PLK_35mm²_HMI_2024_01_25 | PartTypeInformation | <pre> "catenaXId": "248d03b5-9ce1-489d-9c5d-3d158022e3a5" "manufacturerPartId": "23124690760" "nameAtManufacturer": "AAS_MLK_14,5_35" </pre> |
| | SingleLevelBomAsPlanned | Keine ChildItems, da unbekannt |
| | Nameplate | Vom Projekt vorgegeben |
| | ContactInformation | Vom Projekt vorgegeben |
| | HandoverDocumentation | Vom Projekt vorgegeben |
| | TechnicalData | Vom Projekt vorgegeben |
| Kostal_Terminal_PLK_50mm²_HMI_2024_01_25 | PartTypeInformation | <pre> "catenaXId": "4bebcdf4-11c2-4ab9-8092-4b6b6761891a" "manufacturerPartId": "23124690760" "nameAtManufacturer": "AAS_PLK_14,5_50" </pre> |
| | SingleLevelBomAsPlanned | Keine ChildItems, da unbekannt |
| | Nameplate | Vom Projekt vorgegeben |
| | ContactInformation | Vom Projekt vorgegeben |
| | HandoverDocumentation | Vom Projekt vorgegeben |
| | TechnicalData | Vom Projekt vorgegeben |
| | MCAD | Vom Projekt vorgegeben |

9.3.5.9.3.2 Testdaten von Coroplast (Leitungen)

| AAS | Submodels [JSON] | Inhalt |
|-----|------------------|--------|
|-----|------------------|--------|

| | | |
|---|-------------------------|--|
| Coroflex_Leitung_0,5mm²_HMI_2024_02_12 | PartTypeInformation | "catenaXId": "fa120983-4d82-4f94-a06a-c6d648f5bfd9" "manufacturerPartId": "9-2611" "nameAtManufacturer": "COROFLEX_FLR31_0_5" |
| | SingleLevelBomAsPlanned | Keine ChildItems, da unbekannt |
| | Nameplate | Vom Projekt vorgegeben |
| | TechnicalData | Vom Projekt vorgegeben |
| | ProductDesign | Vom Projekt vorgegeben |
| | Documentation | Vom Projekt vorgegeben |
| Coroflex_Leitung_35mm²_HMI_2024_01_25 | PartTypeInformation | "catenaXId": "96be3640-cae3-495d-bd20-0c700fae6ef6" "manufacturerPartId": "9-2611" "nameAtManufacturer": "COROFLEX_180HV_35" |
| | SingleLevelBomAsPlanned | Keine ChildItems, da unbekannt |
| | Nameplate | Vom Projekt vorgegeben |
| | TechnicalData | Vom Projekt vorgegeben |
| | ProductDesign | Vom Projekt vorgegeben |
| | Documentation | Vom Projekt vorgegeben |
| Coroflex_Leitung_50mm²_HMI_2024_01_25 | PartTypeInformation | "catenaXId": "e907e75a-1b8e-4bef-bc6a-0826ac0f273b" "manufacturerPartId": "9-2611 / 50" "nameAtManufacturer": "COROFLEX_9_2611_50" |
| | SingleLevelBomAsPlanned | Keine ChildItems, da unbekannt |
| | Nameplate | Vom Projekt vorgegeben |
| | TechnicalData | Vom Projekt vorgegeben |
| | ProductDesign | Vom Projekt vorgegeben |
| | Documentation | Vom Projekt vorgegeben |

9.3.5.9.3.3 Testdaten von ARENA (Gehäuse)

| AAS | Submodels | Inhalt |
|--|-------------------------|--|
| <u>Kostal GEHAEUSE MLK 1-2 2024 01</u> | PartTypeInformation | "catenaXId": "18a3e5f8-cf00-491b-b690-d176cb9a5848" "manufacturerPartId": "987-654-381" "nameAtManufacturer": "AAS_GEHAEUSE" |
| | SingleLevelBomAsPlanned | Keine ChildItems, da unbekannt |
| | Nameplate | Vom Projekt vorgegeben |
| | HandoverDocumentation | Vom Projekt vorgegeben |
| | TechnicalData | Vom Projekt vorgegeben |
| | MCAD | Vom Projekt vorgegeben |

9.3.5.9.3.4 Testdaten von OEM

| AAS | Submodels | Inhalt |
|--|---------------------|---|
| <u>OEM_SupplierRequirements_2024_08_01</u> | AuthorizedSuppliers | globalAssetId: f446097f-fa75-435b-a88d-a7d80240df0e |

| | | |
|--|--|--|
| | | AuthorizedSuppliersList: BPNL00000003B20M BPNL00000003CSGV BPNL00000003B5MJ |
|--|--|--|

9.3.5.9.3.5 Aggregierte Daten von Tier 1

| AAS | Submodels | | | Inhalt |
|--------------------------------|--------------------------|--------------------------------------|------------------|--|
| <u>ZusammenbauLeitungssatz</u> | PartTypeInformation | | | CatenaXId: d39a99cd-e6b3-4ad1-8890-2f89b7046aa4 |
| | SingleLevel-BomAsPlanned | CatenaXId (Kind-Teil) | BPNL | Anzahl |
| | | fa120983-4d82-4f94-a06a-c6d648f5bfd9 | BPNL00000003B20M | 6 |
| | | 96be3640-cae3-495d-bd20-0c700fae6ef6 | BPNL00000003B20M | 6 |
| | | e907e75a-1b8e-4bef-bc6a-0826ac0f273b | BPNL00000003B20M | 6 |
| | | 91d54b9c-d950-4309-85e0-2de72d4797eb | BPNL00000003CSGV | 3 |
| | | 248d03b5-9ce1-489d-9c5d-3d158022e3a5 | BPNL00000003CSGV | 3 |
| | | 4bebcd4-11c2-4ab9-8092-4b6b6761891a | BPNL00000003CSGV | 3 |
| | | 18a3e5f8-cf00-491b-b690-d176cb9a5848 | BPNL00000003B5MJ | 2 |

9.3.5.9.4 Sequenzieller Ablauf des Use Cases

Im Abschnitt 9.3.3 wurde der generelle sequenzielle Ablauf der Use Cases beschrieben. In diesem Abschnitt wird detailliert auf die einzelnen notwendigen Schritte eingegangen. Die technische Umsetzung wird in Abschnitt 9.3.5.8.4 erklärt.

9.3.5.9.4.1 Bereitstellung der Digitalen Zwillinge der Tier 2 und des OEM

Jeder Komponentenhersteller legt automatisch Digitale Zwillinge zu seinen Produkten an. Der OEM legt seine Anforderungen an den Leitungssatz als Digitalen Zwilling an.

9.3.5.9.4.1.1 Tier 2.1 (Kostal): Data Provisioning

- ContractDefinition (mit Asset & Policies) für DTR im EDC von „Tier 2.1 Kostal“ anlegen
- Submodels aus erhobenen Daten erstellen (Auflistung Testdaten in Abschnitt 0)
- Submodels in das BaSyx Submodel-Repository von „Tier 2.1 Kostal“ hochladen
- Registrieren des Digitalen Zwillinges in der DTR von „Tier 2.1 Kostal“
- ContractDefinition (mit Asset & Policies) für jedes Submodel im EDC von „Tier 2.1 Kostal“ anlegen

9.3.5.9.4.1.2 Tier 2.2 (Coroplast): Data Provisioning

- ContractDefinition (mit Asset & Policies) für DTR im EDC von „Tier 2.2 Coroplast“ anlegen
- Submodels aus erhobenen Daten erstellen (Auflistung Testdaten in Abschnitt 0)
- Submodels in das BaSyx Submodel-Repository von „Tier 2.2 Coroplast“ hochladen
- Registrieren des Digitalen Zwillinges in der DTR von „Tier 2.2 Coroplast“
- ContractDefinition (mit Asset & Policies) für jedes Submodel im EDC von „Tier 2.2 Coroplast“ anlegen

9.3.5.9.4.1.3 Tier 2.3 (ARENA): Data Provisioning

- ContractDefinition (mit Asset & Policies) für DTR im EDC von „Tier 2.3 ARENA“ anlegen
- Submodels aus erhobenen Daten erstellen (Auflistung Testdaten in Abschnitt 0)
- Submodels in das BaSyx Submodel-Repository von „Tier 2.3 ARENA“ hochladen
- Registrieren des Digitalen Zwillinges in der DTR von „Tier 2.3 ARENA“
- ContractDefinition (mit Asset & Policies) für jedes Submodel im EDC von „Tier 2.3 ARENA“ anlegen

9.3.5.9.4.1.4 OEM: Data Provisioning

- ContractDefinition (mit Asset & Policies) für DTR im EDC von „OEM“ anlegen
- Submodels aus erhobenen Daten erstellen (Auflistung Testdaten in Abschnitt 9.3.5.9.3.4)
- Submodels in das BaSyx Submodel-Repository von „OEM“ hochladen
- Registrieren des Digitalen Zwillinges in der DTR von „OEM“
- ContractDefinition (mit Asset & Policies) für jedes Submodel im EDC von „OEM“ anlegen

9.3.5.9.4.2 Zugriff des Tier 1 auf die Daten der Tier 2 & des OEM

Der Tier 1 zieht sich die Informationen zu den autorisierten Suppliern vom OEM. Daraufhin greift er auf die Daten der Tier 2 zu. Die technische Umsetzung wird in Abschnitt 9.3.5.9.1 erklärt.

9.3.5.9.4.2.1 ZuTier 1: Data Consumption von OEM

- Anfrage & Zugriff bei erfolgreicher Contract-Negotiation auf die DTR des OEM
- Aussuchen der entsprechenden Data-Assets des OEM (in diesem Fall gibt es nur das Submodel „AuthorizedSuppliers“)
- Anfrage & Zugriff bei erfolgreicher Contract-Negotiation auf das Submodel „AuthorizedSuppliers“ des OEM

9.3.5.9.4.2.2 *Tier 1: Data Consumption von Tier 2.1 Kostal*

- Herausfinden des Endpoints von Tier 2.1 Kostal mittels EDC Discovery
- Anfrage & Zugriff bei erfolgreicher Contract-Negotiation auf die DTR von Tier 2.1 Kostal
- Aussuchen der entsprechenden Data-Assets von Tier 2.1 Kostal
- Anfrage & Zugriff bei erfolgreicher Contract-Negotiation auf die Submodels von Tier 2.1 Kostal

9.3.5.9.4.2.3 *Tier 1: Data Consumption von Tier 2.2 Coroplast*

- Herausfinden des Endpoints von Tier 2.2 Coroplast mittels EDC Discovery
- Anfrage & Zugriff bei erfolgreicher Contract-Negotiation auf die DTR von Tier 2.2 Coroplast
- Aussuchen der entsprechenden Data-Assets von 2.2 Coroplast
- Anfrage & Zugriff bei erfolgreicher Contract-Negotiation auf die Submodels von 2.2 Coroplast

9.3.5.9.4.2.4 *Tier 1: Data Consumption von Tier 2.3 ARENA*

- Herausfinden des Endpoints von Tier 2.3 ARENA mittels EDC Discovery
- Anfrage & Zugriff bei erfolgreicher Contract-Negotiation auf die DTR von Tier 2.3 ARENA
- Aussuchen der entsprechenden Data-Assets von Tier 2.3 ARENA
- Anfrage & Zugriff bei erfolgreicher Contract-Negotiation auf die Submodels von Tier 2.3 ARENA

9.3.5.9.4.3 Aggregation und Datenbereitstellung von Tier 1

- ContractDefinition (mit Asset & Policies) für DTR im EDC von „Tier 1“ anlegen
- Submodels aus konsumierten Daten erstellen (Auflistung Testdaten in Abschnitt 0)
- Submodels in das BaSyx Submodel-Repository von „Tier 1“ hochladen
- Registrieren des aggregierten Digitalen Zwillings in der DTR von „Tier 1“
- ContractDefinition (mit Asset & Policies) für jedes Submodel im EDC von „Tier 1“ anlegen

9.3.6 Whitespots

Während der Bearbeitung des Projekts konnten Whitespots identifiziert werden, auf die im Folgenden eingegangen wird.

Für den Kontext der identifizierten Whitespots, wie z.B. genutzte Versionen und Annahmen, siehe auch „9.3.4. Aufbau der technischen Infrastruktur“.

9.3.6.1 Funktionalität

Tractus-X

Aktuell gibt es keine Software, die den Austausch digitaler Zwillinge ermöglicht oder Einblick in die aktuellen Angebote von Cofinity-X gewährt.

- ❖ Trace-X⁹³ bietet zwar Funktionen für Data-Provisioning und -Consumption, jedoch ausschließlich mit Tracability-spezifischen Submodellen.
- ❖ Der Simple Data Exchanger⁹⁴ wird zurückgehalten und kann nur begrenzt in Deployments integriert werden.

Catena-X e.V. / Cofinity-X GmbH (als kommerzieller Anbieter)

- ❖ Catena-X ist stark auf Use-Cases ausgerichtet. Das bedeutet, dass Support und Software fehlen, wenn man außerhalb eines bestehenden, nicht in Catena-X enthaltenen Use-Cases agiert.
- ❖ Zudem gibt es mehrere parallele Entwicklungen für Transfer-Konzepte im EDC, wie EDRs, Public HTTP und Cloud, die sich in unterschiedlichen Entwicklungsstadien und mit unterschiedlichem Support befinden.
- ❖ Da nicht alle Komponenten alle Transfer-Konzepte unterstützen, muss der kleinste gemeinsame Nenner gefunden werden.

Catena-X e.V. / IDTA e.V.

In Catena-X gibt es derzeit keine Konzepte für das Asset-Bundling:

- ❖ Der Prozess zur kohärenten Registrierung mehrerer Submodelle befindet sich noch in der Entwicklung ("The process for registering multiple submodels in a cohesive manner is still under development.").
- ❖ Zudem existieren keine EDC-Policies auf der Ebene von Submodel-Properties; aktuell sind diese nur auf der Submodel- oder Asset-Ebene verfügbar.

9.3.6.2 Submodels & Semantic Models

Catena-X e.V. / IDTA

Die Submodel Templates der IDTA sind nicht standardisiert für die Catena-X Use-Cases, lediglich IDTA-02006 (Digital Nameplate) [28] findet sich konzeptionell in [Part Type Information⁹⁵](#) wieder. Catena-X-Technologien nutzen keine bestehenden AASen, sondern es werden „eigene“ Digitale Zwillinge in Form von AAS-Deskriptoren in der DTR registriert. In diesen Deskriptoren wird auf bestehende Submodels verwiesen.

Beispiele für Semantic Models finden sich unter [eclipse-tractusx/sldt-semantic-models⁹⁶](#). Ein Problem dabei ist, dass die JSON-Dateien keine validen IDTA-Submodelle sind. Zudem funktioniert die Import-Funktion des AASX Package Explorers weder für exportierte Submodelle noch für Catena-X Submodelle.

Ein Workaround für Catena-X besteht daher aus mehreren Schritten: Zunächst wird die AASX-Datei heruntergeladen und als XML exportiert. Diese XML-Datei wird dann im AASX Package Explorer geöffnet und das Submodel als JSON exportiert. Anschließend wird die Ziel-AASX geöffnet und als AASX

⁹³ <https://github.com/eclipse-tractusx/traceability-foss>

⁹⁴ <https://github.com/eclipse-tractusx/managed-simple-data-exchanger>

/ <https://github.com/Cofinity-X/cofinity-x-ba-managed-simple-data-exchanger-frontend>

⁹⁵ https://github.com/eclipse-tractusx/sldt-semantic-models/tree/main/io.catenax.part_type_information

⁹⁶ <https://github.com/eclipse-tractusx/sldt-semantic-models>

JSON gespeichert. Im letzten Schritt wird das JSON manuell eingebunden und eine Referenz angelegt, bevor die Datei erneut im AASX Package Explorer geöffnet wird.

9.3.6.3 Security

Catena-X e.V / IDTA / Gaia-X / Eclipse Data Space Group

- ❖ Derzeit existiert kein umfassendes Sicherheitskonzept für die Kommunikation zwischen EDC (Eclipse Data Connector) und AAS. Es mangelt an klaren Konzepten und Richtlinien, die festlegen, wie nach dem EDC-Transfer mit Sicherheits- und Datenschutzaspekten umgegangen werden soll. Weitere Informationen zu den Interaktionsmustern finden sich in der Dokumentation des [Digital Twin Kits](#)⁹⁷.
- ❖ Seitens der IDTA befindet sich der attributbasierte Zugriffskontrollstandard (ABAC) in der Entwicklung und ist noch nicht verabschiedet. Ein Problem ist, dass Dokumente nicht frühzeitig veröffentlicht werden, was die Implementierung erschwert.
- ❖ In BaSyx gibt es derzeit ein rudimentäres RBAC-Konzept und es wird bzgl. Weiterentwicklung auf die Spezifikation der IDTA gewartet. Daher wird die attributbasierte Zugriffskontrolle voraussichtlich (ABAC) erst im neuen BaSyx Release V3 erwartet.
- ❖ Eine mögliche Lösung für die Zertifikatsverwaltung ist die Nutzung von Keycloak, es gibt dafür ein SDK-Anwendungsbeispiel „[BaSyxSecured](#)“⁹⁸.

9.3.6.4 UI

BaSyx

Das [BaSyx-UI](#)⁹⁹ und Tractus-X-Komponenten sind derzeit nur eingeschränkt interaktionsfähig. Zwar existiert ein [Bill of Material Submodel-Plugin](#)¹⁰⁰, jedoch müsste eine Erweiterung implementiert werden, um Catena-X-konforme BOMs (Aspect Model „SingleLevelBomAsPlanned/Built“) zu unterstützen.

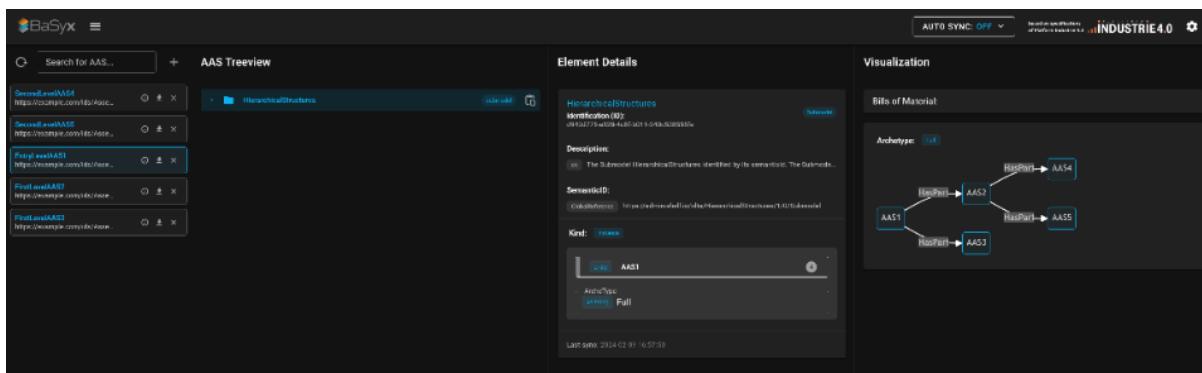


Abbildung 9-25: Darstellung „Bills of Material“ in BaSyx

Tractus-X

Der Simple Data Exchanger¹⁰¹ von Tractus-X wird derzeit noch zurückgehalten und ist nur begrenzt in Deployments integrierbar. Eine Alternative dazu existiert zum aktuellen Zeitpunkt nicht, da das EDC DataDashboard¹⁰² stark veraltet ist und nicht mit Catena-X kompatibel ist.

Trace-X bietet zwar Möglichkeiten für Data-Provisioning und -Consumption, jedoch nur mit Traceability-spezifischen Submodellen.

⁹⁷ <https://eclipse-tractusx.github.io/docs-kits/kits/Digital%20Twin%20Kit/Software%20Development%20View/dt-kit-interaction-patterns/>

⁹⁸ <https://github.com/eclipse-basyx/basyx-java-server-sdk/tree/main/examples/BaSyxSecured>

⁹⁹ <https://github.com/eclipse-basyx/basyx-aas-web-ui>

¹⁰⁰ <https://github.com/eclipse-basyx/basyx-aas-web-ui/blob/main/aas-web-ui/src/components/SubmodelPlugins/BillsOfMaterial.vue>

¹⁰¹ <https://github.com/eclipse-tractusx/managed-simple-data-exchanger-frontend>
<https://github.com/Cofinity-X/cofinity-x-ba-managed-simple-data-exchanger-frontend>

¹⁰² <https://github.com/eclipse-edc/DataDashboard>

9.3.6.5 Helm-Charts

Tractus-X

Der aktuelle Stand der Helm-Charts für Tractus-X zeigt, dass derzeit nur ein Transfer zwischen zwei EDCs unterstützt wird, wie auch auf den Tractus-X Community Days erläutert wurde. Ein weiteres Problem ist, dass Ingresses teilweise fehlen oder nicht getestet sind.

Die Seeding-Daten sind oft hartcodiert und schwer konfigurierbar. Das Erstellen von Docker-Containern, deren Upload in eine Docker-Registry und die Einbindung in Helm-Charts stellen zusätzliche Herausforderungen dar, insbesondere bei der Anpassung von Keycloak.

Ein ordentliches Zertifikatsmanagement ist nicht Teil von Catena-X, und die Helm-Charts bieten nur eine provisorische Lösung für lokale Tests. Die Sandbox-Umgebung ist entweder vollständig oder gar nicht nutzbar. Im letzteren Fall bedeutet dies, dass im Grunde nur zwei EDCs mit einfachem Transfer unterstützt werden.

Die initiale Einstiegshürde ist hoch, und gleichzeitig ist die vollständige Lösung noch nicht einsatzbereit. Diese Punkte verdeutlichen die aktuellen Herausforderungen und den Bedarf an Weiterentwicklungen, um eine nahtlose Integration und Nutzung der verschiedenen Komponenten innerhalb von Catena-X zu gewährleisten.

BaSyx

Hinsichtlich BaSyx gibt es zum aktuellen Zeitpunkt keine öffentlichen Helm-Charts bzw. werden diese derzeit noch unter Verschluss gehalten.

Die IDTA-Spezifikationen werden teilweise von den Tools unterschiedlich implementiert, was sich bspw. in der Namensgebung und der Gruppierung der Services zeigt.

9.3.6.6 Softwareintegration

Tractus-X

Bei Tractus-X ist die Implementierung der IDTA-Spezifikationen nur sehr begrenzt. Die AAS-Registry API (/shell-descriptors), die Discovery API (/lookup/shells) und die Description API sind Beispiele für die begrenzte Umsetzung.

Weitere Informationen dazu finden sich in der Dokumentation zu Digital Twins in Catena-X: <https://catenax-ev.github.io/docs/next/standards/CX-0002-DigitalTwinsInCatenaX>

BaSyx / IDTA

Für BaSyx und IDTA gibt es derzeit keine öffentlichen Konzepte oder Unterstützung für EDCs. Obwohl der Wunsch danach in mehreren Projekten vorhanden ist, sind einzelne Demos und Komponenten aufgrund mangelnder Dokumentation und komplexer Konfiguration schwer zu integrieren. Viele Services funktionieren zwar isoliert, haben jedoch Schwierigkeiten bei der Integration in ein umfassendes System.

9.3.6.7 Dokumentation

Tractus-X / Open-Source Community

Die Dokumentation im Bereich Tractus-X und der Open Source Community weist einige Lücken auf. Die End-to-End-Tutorials sind begrenzt und decken nicht alle notwendigen Aspekte ab. Darüber hinaus fehlt die Dokumentation für viele der oben genannten Punkte vollständig.

BaSyx

Bei BaSyx gibt es ebenfalls erhebliche Dokumentationslücken. Die Dokumentation für das [BaSyx-SDK v2](#) fehlt, während die Version 1 nicht mehr unterstützt wird. Es mangelt an Tutorials sowie an Übersichten und Diagrammen, die für eine umfassende Nutzung und Integration der BaSyx-Komponenten notwendig wären.

Diese Defizite in der Dokumentation erschweren die Implementierung und Nutzung der entsprechenden Technologien erheblich.

9.4 User Interfaces

Dieses Kapitel dient zur Anforderungsaufnahme an vorhandene und zukünftig wünschenswerte Administrationsbezogene und für Anwendungsfallbezogene User Interfaces in der besprochenen digitalen Wertschöpfungskette, als rudimentäre Ideensammlung für zukünftige Umsetzungsprojekte.

Anzeige der AAS und Catena-X-spezifische Submodels:

Aktuell verwendet das Projekt VWS4LS zwei Programme, um Digitale Zwillinge in Form der AAS und die damit verbundenen Submodels anzusehen:

1. [AASX Package Explorer](#)¹⁰³

2. [Eclipse BaSyx](#)¹⁰⁴

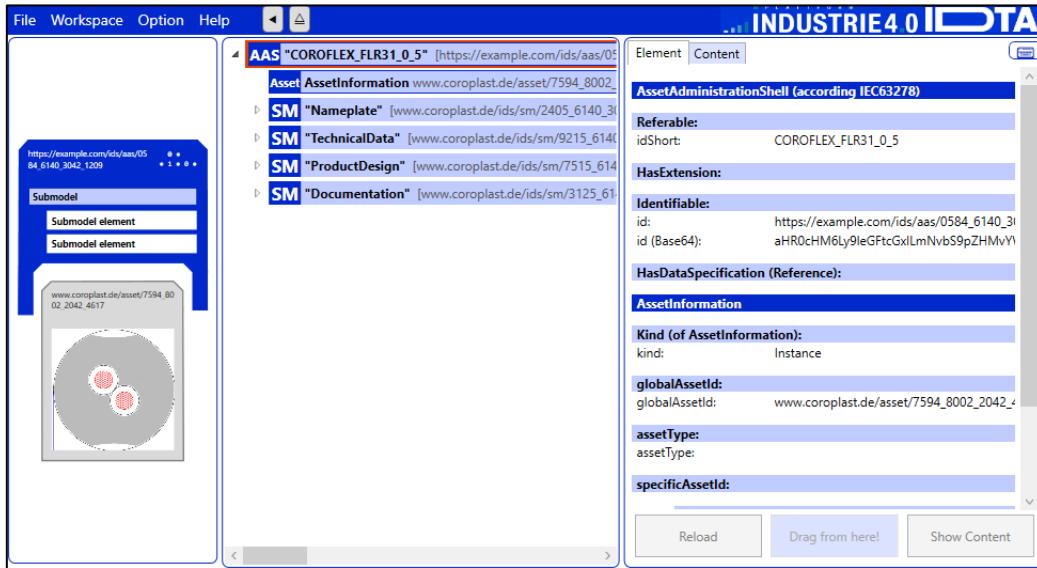


Abbildung 9-26: AASX Package Explorer

Im AASX Package Explorer können auch Catena-X spezifische Submodels („PartTypeInformation“ & „SingleLevelBomAsPlanned“) angezeigt werden (Abbildung 9-27, „[Tier_1_Zusammenbau_Leitungssatz.aaxs](#)“¹⁰⁵):



Abbildung 9-27: Integration des Catena-X Submodels "PartTypeInformation"

¹⁰³ <https://github.com/admin-shell-io/aasx-package-explorer/releases>

¹⁰⁴ <https://eclipse.dev/basyx/>

¹⁰⁵ [https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP08/Testdaten/Tier1\(Aggregation\)/Tier_1_Zusammenbau-Leitungssatz.aaxs](https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/TP08/Testdaten/Tier1(Aggregation)/Tier_1_Zusammenbau-Leitungssatz.aaxs)

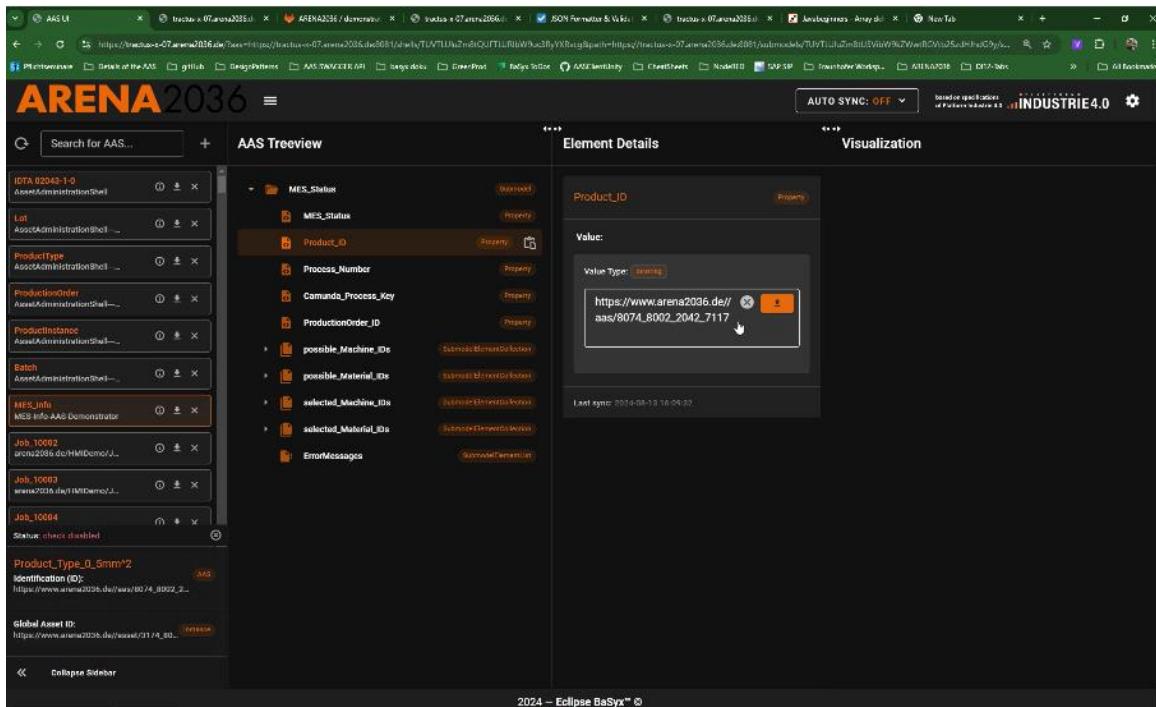


Abbildung 9-28: BaSyx-Frontend zeigt ProductID in AAS MES_Info (Screenshot)

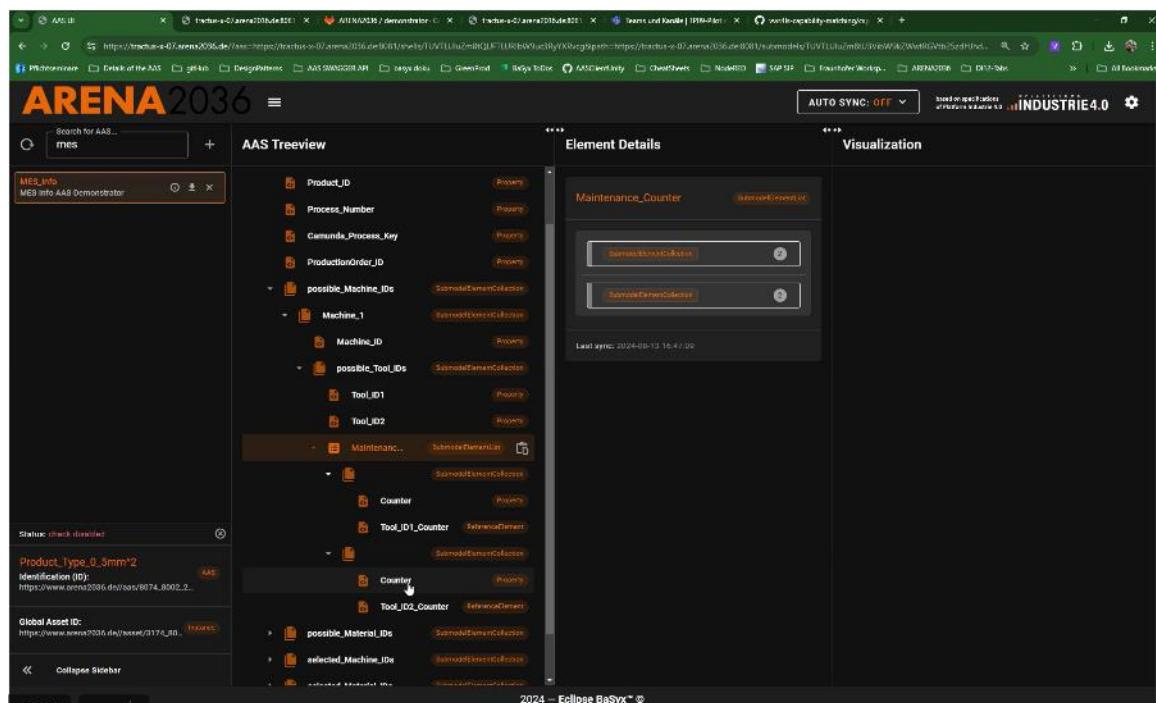


Abbildung 9-29: BaSyx-Frontend zeigt Maintenance_Counter von Machine_1 in AAS MES_Info

Was zum jetzigen Zeitpunkt noch nicht möglich ist, ist dass eines der beiden Programme das Submodel „SingleLevelBomAsPlanned“ interpretiert. In diesem Submodel stehen alle externen Komponenten in Form von Catena-X ID's, die in dem Teil verbaut sind. Das AAS-Programm müsste sich automatisch Zugriff über den EDC auf die enthaltenen Komponenten verschaffen (d.h. Zugriff auf die Externen DTR's und Submodel Server) und anzeigen.

Die Open Source Software „[Trace-X](#)“ kann das, aktuell aber nur mit Catena-X standardisierten Submodels.

9.5 Ausblick

Im Teilprojekt 8 „**Data Storage Policy, Sicherheit, Anbindung an Catena-X**“ konnte eine Aggregation von Informationen für das Produkt Leitungssatz mittels AAS und den Catena-X Komponenten zwischen den Wertschöpfungspartnern in der Leitungssatzwertschöpfung prototypisch durchgeführt werden.

Es mussten bestehende Tractus-X Umbrella Helm Charts umfänglich konfiguriert werden, um eine Testumgebung mit benötigten Catena-X konformen (simulierten) Core Services bereitzustellen. Eine Zusammenarbeit mit der Cofinity-X GmbH als Anbieter eines kommerziellen Systems war zum Zeitpunkt des Projekts noch nicht realisierbar. Zukünftig sollten aber für eine realistische Testumgebung die realen Core Services von Cofinity-X verwendet werden.

Die dezentralen Tractus-X Komponenten für die Enablement Services (EDC & DTR) mussten ebenfalls umfänglich für den Use Case konfiguriert werden. Die Integration eines „Submodel Servers“ unter Verwendung von BaSyx war mit erheblichem Entwicklungsbedarf verbunden, da hier noch keine öffentlichen Helm-Charts vorhanden waren.

Ein Show-Case mit intuitiven User Interfaces, die sowohl Submodels nach Catena-X Standard als auch Submodels der IDTA e.V. gemeinsam verwalten können, war nicht möglich, da diese User Interfaces noch nicht vorhanden sind. Daher erfolgte die Umsetzung des Datenaustausch rein manuell mit API-Befehlen, was eine Demonstration wenig anschaulich macht.

Ein übergreifendes Security-Konzept bestand zum Zeitpunkt des Projekts noch nicht. Die Security-Konzepte des EDC und die des BaSyx-basierten „Submodel-Servers“ sind noch vollständig entkoppelt. Zwar konnten Policies für den EDC verwendet werden, allerdings war seitens BaSyx das RBAC & ABAC noch nicht vollständig implementiert. Ein Overall-Security-Konzept mit EDC und AAS wird benötigt.

Generell erscheint Catena-X als „eigene Welt“ mit eigenen Architekturprinzipien (zumindest was das Zusammenspiel zwischen DTR & AAS angeht). Zudem besteht eine starke Fokussierung auf die bestehenden Catena-X Use Cases.

Aus technischer Sicht gibt es Herausforderungen, die in den Bereichen der Softwareintegration, der Sicherheitskonzepte und der Datenverwaltung liegen. Die unterschiedlichen Implementierungen der IDTA-Spezifikationen und die fehlende Catena-X-Standardisierung der Submodel-Templates, sowie inkonsistenter Entwicklungsstand der Catena-X Use-Cases erschweren die Integration der verschiedenen Komponenten. Die begrenzte Dokumentation und das Entwicklungsstadium der Helm-Charts sowie die limitierten EDC-Konzepte erhöhen die Komplexität der technischen Implementierung. Zudem gibt es bislang kein ausgereiftes Sicherheitskonzept, um eine sichere und robuste Datenübertragung und -verwaltung zwischen den verschiedenen Systemen zu gewährleisten.

Konzeptionell wird es darum gehen, den in einem Teilmodell „Sicherheit“ konfigurierten Rollen und Zugriffsrechte über einen geeigneten Ansatz in dynamisch generierte Policies im Control-Plane des EDC zu generieren. Dies wird eine Überarbeitung der Architektur erfordern.

9.5.1 Bewertung Status Quo

Die konzeptionellen Unterschiede zwischen den beiden Ökosystemen („AAS mit Submodellen“ und „CX-Aspektmodelle“) erfordern geeignete Definitionen, um eine nahtlose Integration zu ermöglichen. Ein wesentlicher Punkt dabei ist der Unterschied in der Handhabung der DTR (*Decentralized Trusted Registry*). Während beide Ökosysteme mit Submodellen und AAS (Asset Administration Shells) arbeiten, liegt der Unterschied darin, dass - anstatt auf bestehende AASen zu verweisen - in der DTR eine neue AAS (Deskriptor) angelegt wird und direkt auf bestehende Submodels verweist.

Dies bedeutet, dass selbst bei bereits vorhandenen AASen diese noch in der DTR angelegt werden müssen, um direkt auf Submodels verweisen zu können. Daher besteht Bedarf an einem Tool, das bestehende AASen automatisch in einer Catena-X-kompatiblen DTR anlegt und die notwendigen Verweise auf die Submodels setzt. Derzeit erfolgt dieser Prozess noch manuell.

Darüber hinaus wurde in beiden Ökosystemen ein Mangel an geeigneten Benutzeroberflächen festgestellt, die sowohl administrative als auch anwendungsbezogene Interaktionen der Anwender mit dem Gesamtsystem unterstützen. Es fehlt insbesondere an UIs, die den Austausch von AASen ermöglichen. Es gibt derzeit keine Benutzeroberfläche, in der digitale Zwillinge nach Catena-X Standards erstellt,

bereitgestellt und konsumiert werden können. Die Entwicklung solcher UI-Lösungen ist entscheidend, um die Anwendungskette abzudecken und somit die Akzeptanz dieser Systeme erheblich zu steigern.

Ein ökosystemübergreifendes Konzept für Identity & Access Management (IAM) ist noch nicht ausdefiniert.

9.5.2 Blick in die Zukunft

Die zukünftige Entwicklung für die Weiterentwicklung der bisherigen Arbeit hängen von einigen grundsätzlichen Entscheidungen ab:

- 1.) Auswahl des zukünftigen Identitätsmanagements. Soll in Zukunft auf eine Alternative zu den in Catena-X verwendeten Business Partner Numbers (BPNs) gesetzt werden, z.B. insbesondere die elektronischen Identitäten der EU (eIDAS2.0) für Personen, Unternehmen und Produkte?
- 2.) Wie wird der Datenzugriff und die Übertragung auf die Verwaltungsschale durchgeführt?
- 3.) An welchen Ort sollen die Souveränität und damit die Zugriffskontrolle erfolgen?
- 4.) Und welche Methoden sollen dabei genutzt werden?

Im Folgenden werden einige grundsätzliche Überlegungen dazu erörtert. Die Entscheidungen erfordern eine weitere Evaluierung auch in Abstimmung mit der Weiterentwicklung mit weiteren Projekten im Rahmen von Manufacturing-X, insbesondere [Catena-X NEXT](#).

Bei der 1. Frage zu den Identitäten ist mit der Einführung von SSI grundsätzlich die Nutzung verschiedener Identitäten und deren Providern möglich. Die entsprechende Weiterentwicklung des Wallets innerhalb von Tractus-X wird sehr wahrscheinlich eine Nutzung von anderen Identitäten ermöglichen. Da anderseits gerade die Leitungssatzbranche per se direkt Bestandteil der Lieferkette der Automobilindustrie ist, ist die Nutzung der von dieser bevorzugten BPNs naheliegend, solange nicht andere Identitäten z.B. durch Produktpässe über die Verwaltungsschale eingeführt werden.

Bei den Fragen zu 2., 3. und 4. geht es darum, wo die Souveränität durch den Datenanbieter (Provider) konfiguriert wird. Folgende mögliche konzeptionelle Lösungsansätze für ein CX-Produktivsystem mit Anbindung an AAS-Infrastrukturen (Abbildung 9-30) kommen dabei in Frage:

- 1) CX-Anwendung greift direkt auf AAS zu, die Zugriffskontrolle erfolgt über die Verwaltungsschale:
Die Catena-X-Infrastruktur wird ausschließlich für die Verwaltung des B2B-Workflows (Kette) über „*non-operational data*“ verwendet. Die eigentlichen Dateninhalte („*operational data*“) des/der Assets im Workflow werden in verteilten AAS-Repositories verwaltet. D.h. aus dem Catena-X-Workflow heraus wird nur eine Verlinkung auf die jeweils relevante AAS mit den Dateninhalten benötigt und bei Bedarf (z.B. Datenmanipulation) aus der CX-Anwendung auf die entsprechende AAS-Anwendungstoolkette „abgesprungen“. Die Rechteverwaltung für Submodelle und einzelne Datenpunkte findet dabei ausschließlich im AAS-Repository statt. So können die beiden Welten parallel miteinander koexistieren und interagieren, sind aber technisch stark entkoppelt.
- 2) CX-Anwendung greift über ein Aspekt-Modell des EDC auf Teilmodelle der AAS zu, die Zugriffskontrolle erfolgt nur über angepasste Policies im EDC: Grundsätzlich getrennte Datenhaltung wie oben beschrieben, aber mit Anpassung des EDC, der dann als Gateway für den Zugriff auf die AAS-Infrastruktur dient. Dies würde einen tieferen Eingriff in dem EDC-System bedeuten, könnte aber die Komplexität auf Seiten der CX-Anwendungen, die mit einer VWS-Infrastruktur interagieren möchten, deutlich verringern.

- 3) CX-Anwendung greift über ein angepasstes Asset des EDC auf die komplette AAS zu, die Zugriffs-kontrolle erfolgt über die AAS: Es ist eine Architekturanpassung im Control-Plane des EDC vorzunehmen, die erlaubt, dass die in einem Teilmodell "Sicherheit" konfigurierten Rollen und Zugriffsrechte über einen geeigneten Ansatz in dynamisch generierte Policies in den Control-Plane des EDC übertragen werden. Dafür wird man neben dem EDC auch die entsprechende dDTR konzeptionell erweitern müssen. Die Änderungen würde auch das Protokoll der "Vertragsaushandlung" zwischen zwei EDC betreffen das Protokoll würde die dDTR einbeziehen müssen (Abbildung 9-30).

Operational View

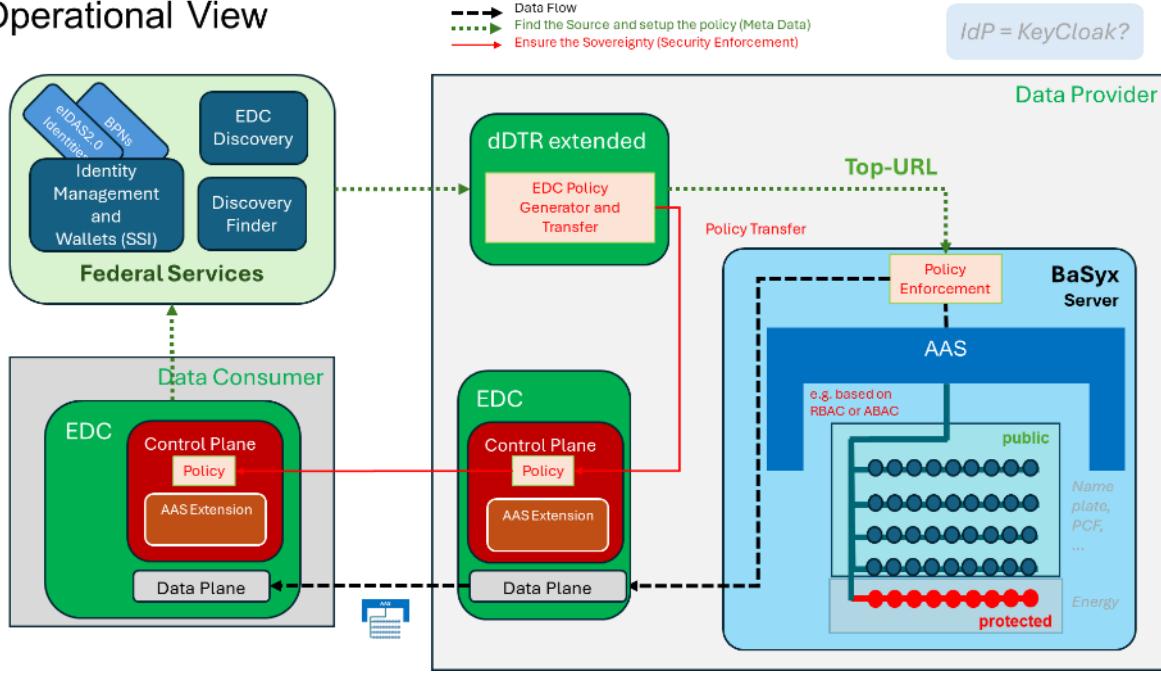


Abbildung 9-30: Interaktion Tractus-X und AAS via EDC

Für die Anwendung in der Leitungssatz-Branche müsste eine der oben aufgezählten Optionen mit den anderen Manufacturing-X-Projekten abgestimmt, weiter detailliert und dann zukünftig umgesetzt werden.

10 TP9 - Pilotierung, Erprobung und Demonstrator

Im TP9 „**Pilotierung, Erprobung, Demonstrator**“ fand die partnerübergreifende Pilotierung und Erprobung für die Lösungskonzepte zur Umsetzung der Verwaltungsschale statt. Dazu wurden über die Projektlaufzeit mehrere Demonstratoren aufgebaut, welche die erarbeiteten Ergebnisse und Konzepte aus den verschiedenen TPs integrieren und vor allem die Mehrwerte der VWS darstellen sollen.

Das TP9 war in folgende Arbeitspakete gegliedert:

10.1 AP 9.1: Konzeption und Bauplan des Demonstrators

Im AP 9.1 „**Konzeption und Bauplan des Demonstrators**“ wurde die Ausgestaltung des Demonstrators erarbeitet. Insbesondere wurden hier die zeitlichen Ausbaustufen geplant (Abbildung 10-1). Wie in der Abbildung dargestellt, wurden an vielen Ecken und Enden die bisherige Planung an das Projektgeschehen angepasst. Die verschiedenen Demonstrator-Stufen werden in den beiden nachfolgenden Kapiteln dargestellt.

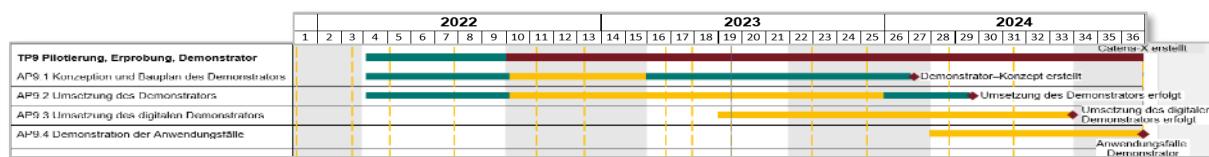


Abbildung 10-1: Zeitplan TP9

Die Konzeption der Demonstratoren wurde bereits kurz nach Projektbeginn im Dezember 2021 gestartet und über die gesamte Projektlaufzeit fortgesetzt. Insgesamt wurden 4 Demonstratoren entwickelt und aufgebaut, welche im Folgenden beschrieben werden. Der physische Aufbau des Demonstrators hat sich nach der Fertigstellung von Stufe 2 kaum mehr verändert. Der Hauptteil der Veränderung erfolgte anschließend vor allem softwareseitig. Daher werden die Stufen 1 und 2 des Demonstrators in dem Kapitel 10.2 dargestellt wo es um die physische Demonstration geht und die Demonstrator-Stufen 3 und 4 in dem Kapitel 10.3 dargestellt.

10.2 AP 9.2 - Umsetzung des physischen Demonstrators

Im AP 9.2 „**Umsetzung des physischen Demonstrators**“ wurde die Umsetzung der physischen Elemente (Sachmittel) des Demonstrators definiert. Dies umfasste geeignete Beispiele für Komponenten des Leitungssatzes, ganze Teilleitungssätze und Produktionsmittel zur Bearbeitung und Montage des Leitungssatzes. Für alle diese Komponenten haben die Partner des Konsortiums Beiträge geleistet.

Die Konzeption der Demonstratoren, fing bereits kurz nach dem Start des Projektes im Dezember 2021 an und wurde über die gesamte Projektlaufzeit fortgesetzt. Insgesamt wurden 4 verschiedenen Demonstratoren entwickelt und aufgebaut, welche im Folgenden kurz beschrieben und im weiteren Verlauf des Kapitels öfter referenziert werden.

10.2.1 Demonstrator Ausbaustufe 1 (2022) – HMI 2022

Ziel des ersten Demonstrators war es, auf die Komplexität des Leitungssatzes aufmerksam zu machen, und erste statische Produktverwaltungsschalen selbst mittels des AASX-Package-Explorers aufzubauen.

Hierzu wurde ein Leitungssatz einer bereits auf dem Markt befindlichen Mercedes C-Klasse (2016) von dem Projektpartner Dräxlmaier zur Verfügung gestellt. Eine Konstruktion aus Monopanplatten, Klavierschraubern und Abschlussleisten aus Edelstahl wurde zurechtgeschnitten und montiert, damit der Leitungssatz darauf befestigt und auf Messen präsentiert werden kann (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.**).



Abbildung 10-2 Physische Leitungssatzpräsentation inkl. Produktverwaltungsschalen (Bildschirme)

Zur Präsentation erster Produktverwaltungsschalen wurden 4 ausgewählte Leitungssatzkomponenten (Clip, Terminal, Tape und Leitung) als VWS mit dem AASX-Package-Explorer erstellt. Hierfür wurden die bereits vorhandenen Teilmodelle der IDTA verwendet (d.h. *DigitalNameplate*, *ContactInformation*, *TechnicaData* und *HandoverDocumentation*).

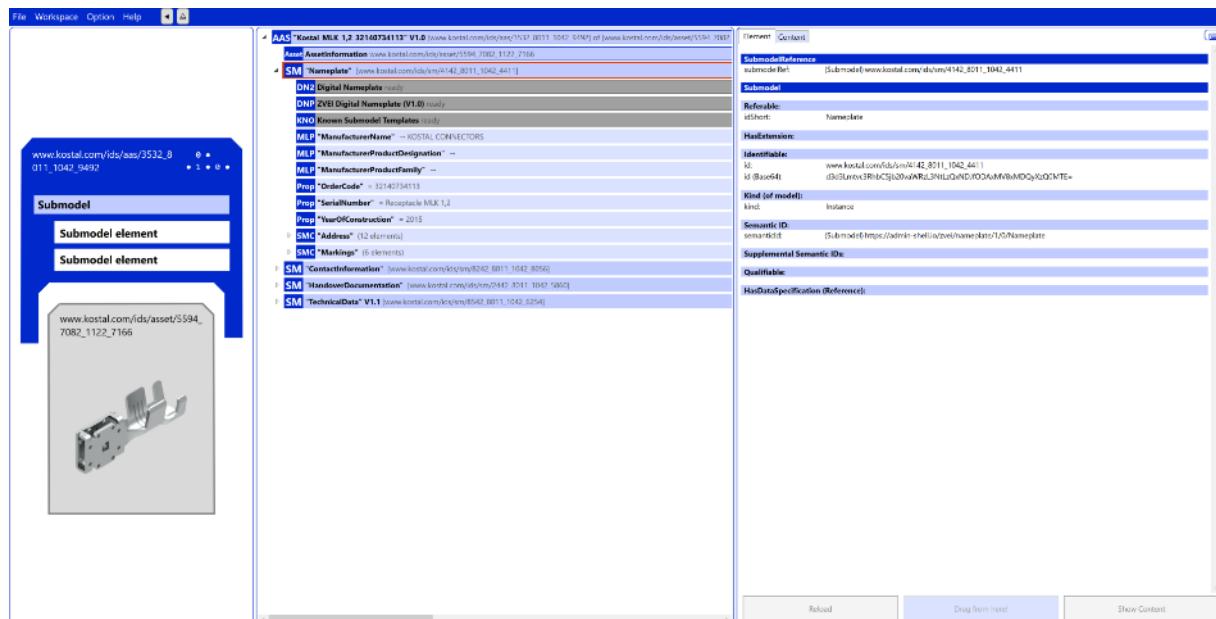


Abbildung 10-3 Beispiel VWS-Terminal

10.2.2 Demonstrator Ausbaustufe 2 (2023) - HMI 2023



Abbildung 10-4 Projektdemonstrator mit dem Fokus auf der Implementierung der VWS in ein Produktionsprozess

Ziel des zweiten Demonstrators war es, die dynamische Einbindung der VWS in einen Leitungssatzproduktionsprozess umzusetzen und vor allem die Mehrwerte der VWS und deren Möglichkeiten in Bezug zur durchgängigen Digitalisierung durch einen Standard in den Vordergrund stellen. Der grundsätzliche Aufbau des Demonstrators besteht aus sechs Hauptkomponenten, die im Folgenden kurz erläutert werden:

- **MES-Applikation**
Die Manufacturing Execution System (MES)-Applikation dient dazu, den Produktionsauftrag zu initiieren und die verschiedenen Informationsquellen sowie -ströme zu koordinieren.
- **Einzelkomponente Leitung**
Die Leitung besitzt eine eigene Verwaltungsschale, in der wesentliche Informationen wie *WireNumber* und *WireBatch* hinterlegt sind. Diese Daten werden durch das Scannen eines QR-Codes in den Produktionsprozess integriert.
- **Einzelkomponente Terminal**
Auch das Terminal verfügt über eine eigene Verwaltungsschale. Deren Informationen, wie *TerminalNumber* und *TerminalBatch*, werden ebenfalls über das Scannen eines QR-Codes in den Prozess eingebracht.
- **Crimp-Maschine**
Die Aufgabe der Crimp-Maschine ([Wezag_UP150](#)) ist die Durchführung des Crimp-Prozesses. Die im Auftrag enthaltenen Informationen werden in den Produktionsprozess der Maschine eingespeist, während gleichzeitig Produktionsdaten an die MES-Applikation zurückübermittelt werden.
- **Shopfloor-Monitoring**
Das Monitoring-System erfasst verschiedene Informationen von der Maschine sowie den Typ- und Instanz-Verwaltungsschalen des jeweiligen Auftrags, wie etwa den Verlauf der Crimpkraft.
- **BaSyx-Middleware**
In der BaSyx-Middleware sind die Verwaltungsschalen der Maschine und der jeweiligen Jobs hinterlegt, auf die während des Produktionsprozesses zugegriffen wird.

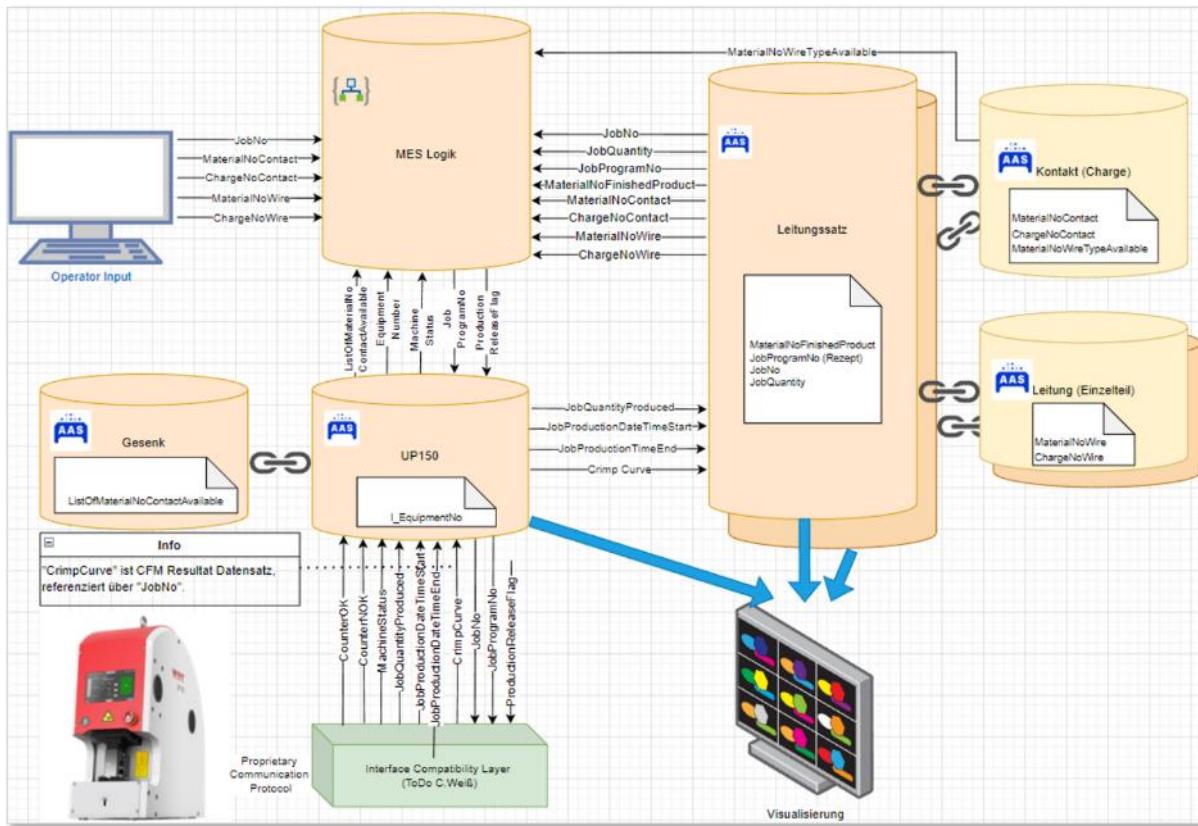


Abbildung 10-5 Bestandteile Demonstrator und Informationsfluss

Die Initialisierung eines Produktionsauftrags erfolgt durch das Scannen eines QR-Codes. Dieser enthält grundlegende Informationen zum Auftrag, wie die Job- und Produktnummer sowie eine ID für die zu verwendende Maschine. Im nächsten Schritt wird der Auftrag um weitere Informationen ergänzt, indem QR-Codes der Einzelkomponenten (Verwaltungsschalen von Leitung und Terminal) gescannt werden. Diese Informationen werden im Auftrag hinterlegt und vervollständigen die Auftragsbeschreibung.

Die MES-Applikation übermittelt anschließend den Auftrag mit der entsprechenden Produktionsfreigabe an die Maschine, woraufhin der Crimp-Prozess gestartet wird. Während der Produktion erzeugt die Maschine verschiedene Daten, wie den Verlauf der Crimpkraft oder die Produktionsendzeit. Diese Informationen werden nach Abschluss des Prozesses an die MES-Applikation zurückgesendet und in der entsprechenden Job-Verwaltungsschale gespeichert. Dieser Schritt veranschaulicht sowohl das Push- als auch das Pull-Prinzip, indem die Verwaltungsschale in der Lage ist, Informationen zu liefern und diese an vorgesehenen Speicherorten abzulegen.

10.3 AP 9.3 - Umsetzung des digitalen Demonstrators

Im Arbeitspaket 9.3 „Umsetzung des digitalen Demonstrators“ wurde eine mehrstufige Demonstration der Entstehung und der Anreicherung der Verwaltungsschale entlang der Leitungssatz-Wertschöpfungskette erarbeitet. In der ersten Stufe wurde nach jedem Prozessschritt entlang des gesamten Entstehungsprozesses das Entwicklungsstadium des Assets „Leitungssatz“ (oder der Sub-Komponenten) und die dazugehörige Verwaltungsschale dargestellt

10.3.1 Demonstrator Ausbaustufe 3 (2024) – HMI 2024

Ziel der Ausbaustufe 3 war die dynamische Einbindung der Verwaltungsschalen (VWS) in einen Leitungssatzproduktionsprozess mit mehreren Produktionsressourcen. Der physische Aufbau des Demonstrators hatte sich im Vergleich zu Kapitel 10.2 lediglich dahingehend lediglich dahingehend verändert, dass eine weitere Produktionsressource nahezu nicht verändert, dass eine weitere Produktionsressource ([Komax Alpha](#)) mit eingebunden wurde (siehe Abbildung 10-6). Der Produktionsressource ist aufgrund der Größe und Transportabilität als Simulation verfügbar und steht physisch nicht zur Verfü-

gung. Darüber hinaus, besteht der Demonstrator weiterhin aus den Grundkomponenten – MES-Applikation, Produktionsmaschine, BaSyx und den Verwaltungsschalen der Einzelkomponenten (Leitungen und Terminals).

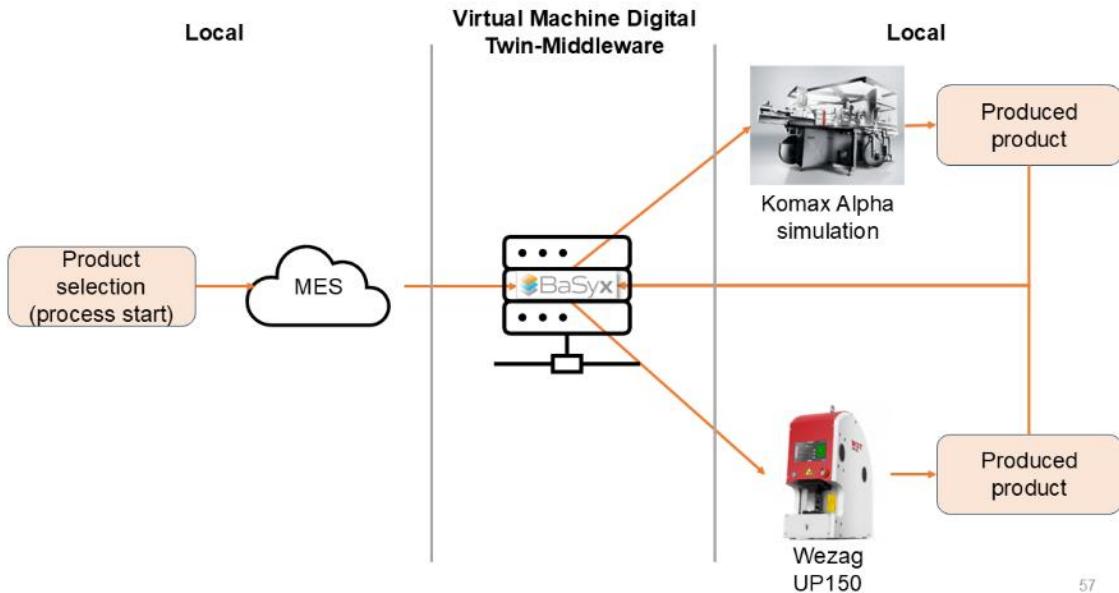


Abbildung 10-6 Aufbau des Demonstrators

Der Fokus der Veränderung lag in der Nutzung von lokalen Rechnern und dezentralen Servern in der ARENA2036¹⁰⁶. Lokal laufen die MES-Applikation und die gesamte Shopfloor-Anbindung, während die Daten dezentral auf den Servern der ARENA2036 gesammelt werden. Darüber hinaus wurde die Anzahl der Produktionsressourcen erweitert, indem eine physische Maschine ([Wezag UP150](#)) und eine simulierte Maschine ([Komax Alpha](#)) eingebunden wurden. Zusätzlich werden verschiedene Anbindungsprotokolle (OPC-UA Companion Specification und eine proprietäre Schnittstelle) unterstützt. Im Folgenden wird der Ablauf des Demonstrators beschrieben, bei dem aus verschiedenen Produktvarianten eine Auswahl getroffen und der Crimp-Prozess zur Herstellung einer gecrimpten Leitung durchgeführt wird. Die einzelnen Prozessschritte finden sowohl vor Ort auf dem Shopfloor als auch auf dezentralen Servern statt.

Abschnitt 1: Auf dem Shopfloor – Produktauswahl und Auftragserstellung

Der Prozess startet mit der Produktauswahl. Es stehen drei verschiedene Leitungsvarianten zur Verfügung, die sich jeweils durch unterschiedliche Querschnitte unterscheiden. Zu jeder Leitung muss das passende Terminal ausgewählt werden. Das Ziel ist es, eine gecrimpte Leitung auf einer geeigneten Maschine herzustellen.

Sobald die Produktauswahl getroffen wurde, erstellt die MES-Applikation automatisch den Produktionsauftrag. Dazu fordert das MES zunächst die erforderlichen Produkt- und Prozessdaten an und legt fest, welche Ressource – also welche Maschine – für die Durchführung des Crimp-Prozesses genutzt wird. Sobald alle Informationen erfasst sind, ist der Auftrag vollständig beschrieben und wird zur weiteren Bearbeitung an die BaSyx-Middleware übertragen.

Abschnitt 2: Zentral auf einem Server in der ARENA2036 – Erstellung der Verbundkomponente Verwaltungsschale

Im nächsten Schritt wird auf den Servern die Job-Verwaltungsschale des ausgewählten Produkts erstellt. Diese Verwaltungsschale enthält alle für die Produktion relevanten Informationen, einschließlich der spezifischen Daten der gewählten Leitung und des Terminals. Nachdem der Auftrag und alle damit

¹⁰⁶ <https://tractus-x-07.arena2036.de/>

verbundenen Informationen digital erfasst sind, werden die Auftragsdetails an die ausgewählte Maschine übermittelt, die den Crimp-Prozess durchführen soll.

Abschnitt 3: Zurück auf dem Shopfloor – Prozessausführung

Die ausgewählte Maschine auf dem Shopfloor übernimmt die Ausführung des Crimp-Prozesses. Hierbei kommen verschiedene Kommunikationsprotokolle wie die OPC-UA Companion Specification zum Einsatz, um eine reibungslose Steuerung und Datenübertragung sicherzustellen. Während des Produktionsprozesses werden sämtliche Prozessdaten – wie der Verlauf der Crimpkraft und die Endzeiten der Produktionsschritte – in die zuvor erstellte Verwaltungsschale zurückgespielt. Dieser Vorgang ermöglicht eine vollständige digitale Abbildung und Rückverfolgbarkeit des Produktionsprozesses.

Am Ende des Prozesses entsteht eine gecrimpte Leitung, die das Zielprodukt des Demonstrators darstellt. Dieser Ablauf verdeutlicht die enge Verzahnung von lokaler Produktion und digitaler Beschreibung sowie das Zusammenspiel zwischen physischen Produkten und ihren digitalen Abbildern.

10.3.2 Demonstrator Ausbaustufe 4 (2024) - Ergebnistagung

Ziel dieser Ausbaustufe war, den statischen Teil des Demonstrators weiter zu reduzieren und somit ein weiterer Schritt in Richtung dynamischer VWS-Nutzung einer realistischen industriellen Anwendung zu tätigen. Hierfür wurden die im Projekt entwickelten Funktionen des *Capabilitychecks* (TP5), dem *Automatisiertenverhandlungsprozess* (TP6), *Tool-Wear Funktion* (Ansatz zur Predictiv Maintanance → hochzählen der Produktionszyklen) und die Submodellstruktur von *ProductionOrder/Batch/Lot* (TP3) implementiert. Somit hat sich der physische Aufbau des Demonstrators, im Vergleich zum Kapitel 10.3.1 nicht verändert, sondern die Veränderungen finden softwareseitig statt.

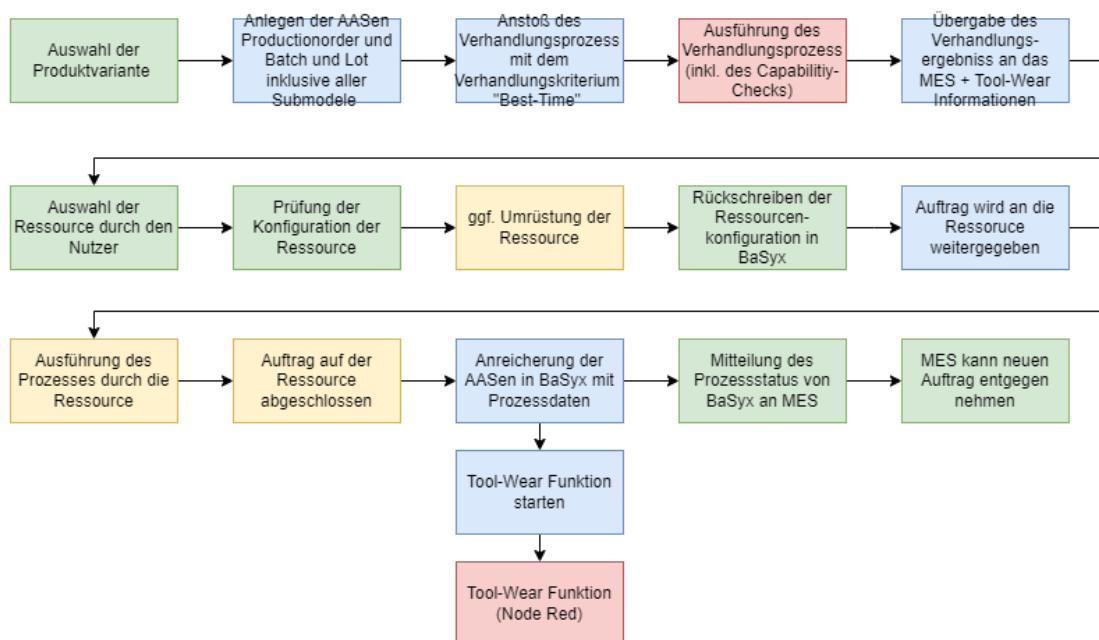


Abbildung 10-7 Schematischer Prozessablauf

In Abbildung 10-7 ist der Prozessablauf des finalen Demonstrators zusehen, auf den im weiteren Verlauf des Kapitels eingegangen wird. Die farblichen Markierungen der Aktivitäten, bedeutet eine Unterscheidung der Hauptkomponenten des Demonstrators.

- Blau → Eine Aufgabe, die durch die BaSyx-Middleware umgesetzt werden muss
- Grün → Eine Aufgabe, die durch die MES-Applikation umgesetzt werden muss
- Rot → Eine Aufgabe die durch den Verhandlungsprozess umgesetzt werden muss
- Gelb → Eine Aufgabe, die durch den Werker umgesetzt werden muss

Zu Beginn wird eine Produktauswahl durch den Werker in der MES-Applikation getroffen. Hierfür stehen wieder die verschiedenen Kombinationen aus Leitung und Terminal in unterschiedlichen Querschnitten zur Auswahl, die für die Produktion unterschiedliche Capabilities benötigen. Nach der Auswahl des Produktes, wird automatisiert in der *BaSyx-Middleware* eine VWS-Kombination für die Produktinstanz aus den *ProductionOrder*, *Batch* und *Lot* erzeugt, welche im folgenden Prozessablauf mit weiteren Prozessinformationen gefüllt werden. Anschließend wird der Werker durch das MES aufgefordert, die entsprechenden Produktinformationen, die für die spätere Ausführung des Prozesses notwendige sind, hinzuzufügen – dieser Schritt wird durch das Scannen der QR-Codes stellvertretend für den Verweis auf Einzelkomponenten umgesetzt. Da jedes Produkt unterschiedliche Capabilities benötigt, müssen nun im nächsten Schritt die potenzielle möglichen Produktionsressourcen ausfindig gemacht werden. Hierfür wurde der in Kapitel 7.2 bereits beschriebene Verhandlungsprozess implementiert, der genau Ablauf ist in Abbildung 10-8 nochmal dargestellt. Ziel hierbei ist es, eine Produktionsstrategie zu definieren, in Abhängigkeit der notwendigen Required Capabilities auf Produktseite, mit dem Ziel „Best-Time“ (schnellstmögliche Produktion).

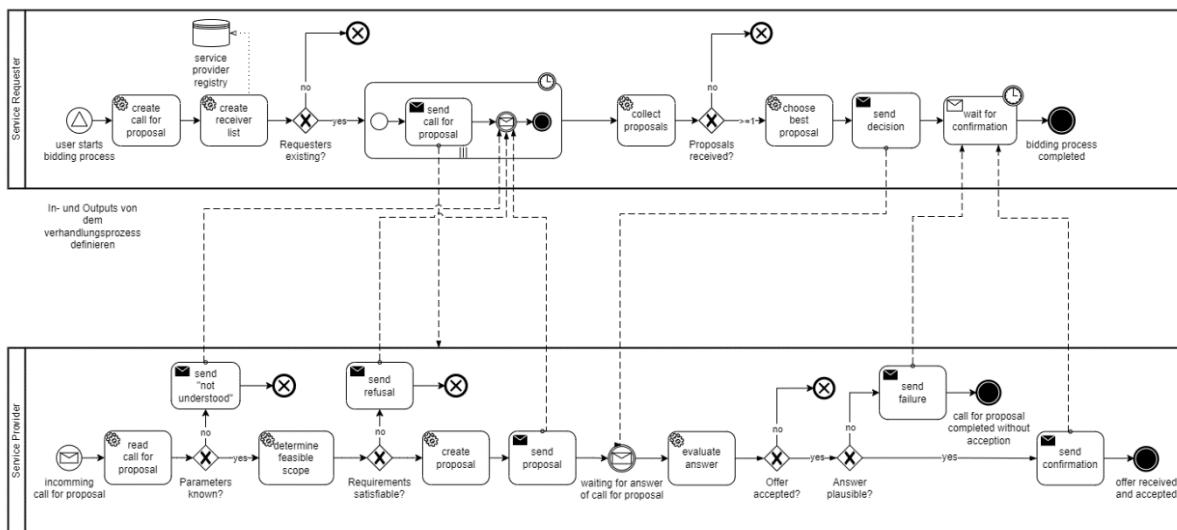


Abbildung 10-8 Verhandlungsprozess zwischen Service Requester und Provider

Als Ergebnis des Verhandlungsprozesses werden alle potenziellen Produktionsressourcen ausgegeben, die den Capabilities entsprechen würden. Die Informationen werden anschließend an das MES weitergegeben und für den Werker in der MES-Applikation visuell zur Verfügung gestellt. Der Werker wählt nun eine der möglichen Ressourcen und muss der MES-Applikation die aktuelle Ressourcen Konfiguration (Werkzeugkonfiguration) mitteilen. Dieser Vorgang wird durch Abscannen der entsprechenden QR-Codes durchgeführt, die repräsentativ für die einzelnen Tools stehen. Sollte nun eine nicht passende Ressourcenkonfiguration aktuell auf der Ressource sein, wird der Werker aufgefordert die passenden Topols zum einen zu scannen und anschließend die Ressource neu zu Rüsten. Die neue Ressourcenkonfiguration wird anschließend in die Instanz VWS der *ProductionOrder* eingefügt und der Auftrag wird nun an die ausgewählte Ressource weitergeleitet. Die Ressource wird nun vom System freigeschaltet und kann den vorgesehenen Prozess durchlaufen. Nach Abschluss des Produktionsprozesses werden die während des Prozesses erzeugten Prozessdaten direkt in das Entsprechende *Batch* bzw. *Lot* eingefügt. Weiter wird die Information des „Prozessabschlusses“ an das MES weitergegeben und der Count für die vom Werkzeug durchlaufenen Produktionszyklen (Ansatz *Predictiv Maintenance*) um 1 erhöht. Das MES setzt in diesem Fall den Status auf IDLE und ist nun bereit für einen neuen Auftrag.

10.4 AP 9.4 - Demonstration der Anwendungsfälle

Im AP 9.4 „**Demonstration der Anwendungsfälle**“ wurden dedizierte Szenarien für die jeweiligen Zusammenhänge der fünf Hauptanwendungsfälle gezeigt:

1. Kollaborative Entwicklung (im Engineering entstehen Typ-Daten, diese werden beim Tier 1 und 2 ergänzt),

- Durch die Umsetzung bzw. die Anbindung der VWS an das Catena-X Ökosystem unter Verwendung der Tractus-X Komponenten, wurde ein solches Szenario in Kapitel 9.3 durchgespielt und detailliert beschrieben. Aufgrund technischer Differenzen (siehe Kapitel 9.3.6) und einiger fundamentaler Fragestellungen konnte eine Kombination aus den Demonstratoren aus Kapitel 9.3 und 10.3.2 nicht umgesetzt werden.
2. Fähigkeiten und Randbedingungen der Produktion im Engineering berücksichtigen (Varianten-Konfiguration)
-
3. Daten- und Informationsbereitstellung für die Rüstung und Parametrisierung der Produktionsmittel je Instanz (Anpassung der Komponenten, Werkzeuge, Bahnen etc.),
- Durch die im TP3 beschriebene Prozessliste (Kapitel 4.3.2) wurden alle notwendigen Parameter, die für die Prozessausführung erforderlich sind, im Detail beschrieben. Für den Crimp-Prozess wurde der dort erläuterte Zusammenhang zwischen **ProductionOrder**, Batch und Lot genutzt, um eine exemplarische Umsetzung innerhalb des Demonstrators zu ermöglichen. Die Prozessliste diente dazu, die erforderlichen Prozessparameter sowie die notwendigen Informationen für die Ressourcen bereitzustellen und zu aggregieren.
4. Automatisiertes Änderungsmanagement entlang der Wertkette (Änderungen im Engineering führen zu automatisierten Anpassungen der Komponenten und der Prozesse),
- Dieser Anwendungsfall wird sowohl konzeptionell als auch demonstrativ in den Demonstratoren präsentiert. Die Untergruppe „Änderungsmanagement“ aus dem Architekturteam (Kapitel 12.8) beschreibt einen konzeptionellen Prozess, wie eine Änderung beschrieben sein muss, und welche Stakeholder in dem Prozess involviert sind. Hingegen die Untergruppe „Synchronisation“ ebenfalls aus dem Architekturteam (Kapitel 12.6) eine technische Änderung zweier VWS beschreibt und wie diese umgesetzt werden können. Teile aus den Ergebnissen der Synchronisationsgruppe sind bereits in dem Demonstrator im Kapitel 10.3.2 verwertet worden, da der Mechanismus, für eine Verhandlung nahezu identisch mit der Synchronisation ist.
5. Prozess- und Qualitätsdaten erfassen, dokumentieren und rückverfolgbar machen.
- Dieser Anwendungsfall ist sowohl konzeptionell als auch demonstrativ im Demonstrator, welcher im Kapitel 10.3.2 beschrieben wurde, abgedeckt. In der VWS, werden nach der Prozessausführung alle Qualitätsdaten, wie bspw. die Crimpkraftkurve und weitere Prozessdaten aggregiert.

Für die Anwendungsfälle wurden typische Bearbeitungsvorgänge dargestellt, bspw. eine Platzierung von Steckern auf dem Kabelformbrett mittels Roboter, das Crimpen eines Leiters, das Bestücken des Steckers sowie das Zusammenstecken eines Steckers mit einem anderen oder einem Steuergerät.

11 TP10 – Transfer und Koordination

Das TP 10 umfasste alle Aktivitäten zur Sicherstellung des Transfers und der Koordination der Projekt ergebnisse. Im Fokus standen der kontinuierliche Basistransfer, die Entwicklung und Bereitstellung von allgemeinen und leitungssatzspezifischen Implementierungspaketen sowie die Ergebnisintegration und Koordination. Damit konnten die Ergebnisse des Projekts effektiv verbreitet und nachhaltig in den industriellen und öffentlichen Raum überführt werden. Für VWS4LS gibt Abbildung 11-1 hierzu einen kontextuellen Überblick.

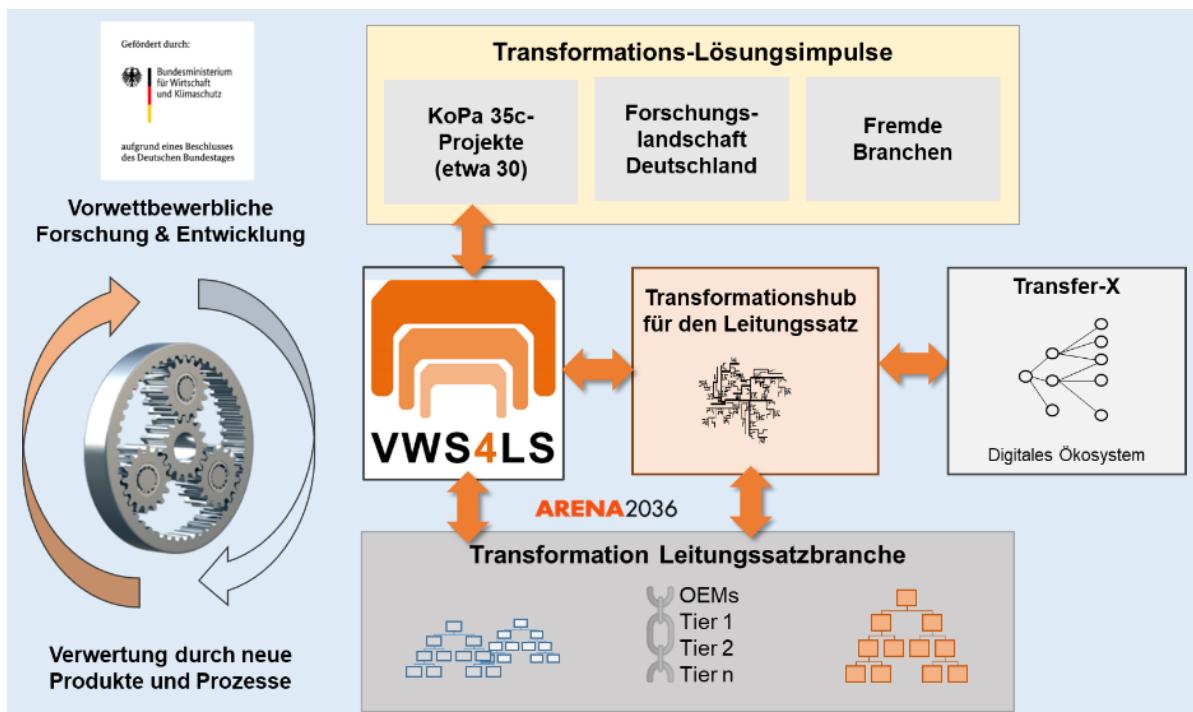


Abbildung 11-1: [Transferkontext](#)

11.1 AP 10.1 - Kontinuierlicher Basistransfer

Der kontinuierliche Basistransfer hat sichergestellt, dass alle relevanten Projektergebnisse auf geeignete Weise verbreitet und öffentlich zugänglich gemacht wurden. Ein zentraler Bestandteil war der Aufbau einer [VWS4LS-Webseite](#)¹⁰⁷, die als öffentlicher Einstieg in die VWS4LS-Wissensbasis dient. Die Webseite wurde kontinuierlich mit neuen Ergebnissen und Fortschritten ergänzt und diente als Plattform, um alle relevanten Einzelergebnisse des Projekts anschaulich darzustellen und auf die Detaildokumentationen in den [VWS4LS-Github-Repositories](#)¹⁰⁸ zu verlinken.

Zur Grundlagenvermittlung wurden 2 von 16 Lernmodule erstellt und über [Transfer-X](#)¹⁰⁹ veröffentlicht:

[ILIAS: Verwaltungsschale Beispielanwendung – VWS4LS \(Teil I\)](#)

[ILIAS: Verwaltungsschale Beispielanwendung – VWS4LS \(Teil II\)](#)

Zu weiteren relevanten Begleithemen (z.B. Catena-X) finden sich hier noch weitere hochwertige Lernmodule:

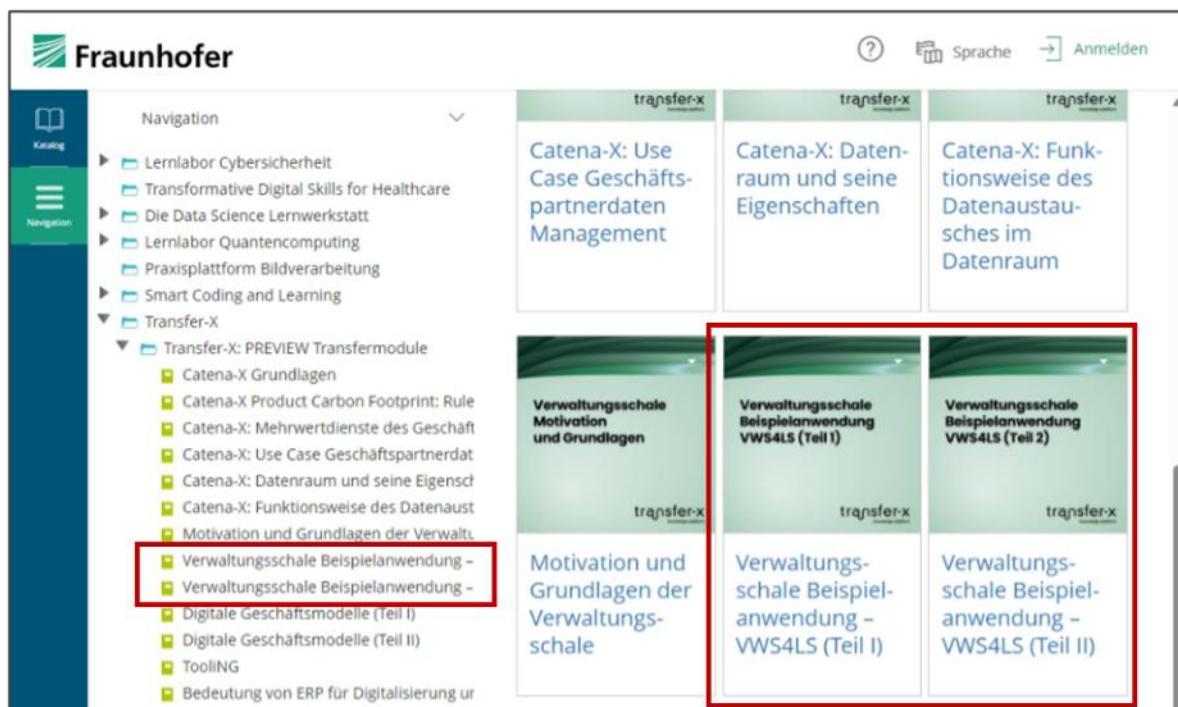
[Inhalt: Transfer-X: PREVIEW Transfermodule: Fraunhofer ILIAS](#)¹¹⁰

¹⁰⁷ <https://arena2036.de/de/vws4ls>

¹⁰⁸ <https://github.com/orgs/VWS4LS/repositories>

¹⁰⁹ <https://transfer-x.de/>

¹¹⁰ https://ilias.fraunhofer.de/ilias.php?baseClass=ilrepositoryqui&ref_id=29387



Fraunhofer

Navigation

- ▶ Lernlabor Cybersicherheit
- ▶ Transformative Digital Skills for Healthcare
- ▶ Die Data Science Lernwerkstatt
- ▶ Lernlabor Quantencomputing
- ▶ Praxisplattform Bildverarbeitung
- ▶ Smart Coding and Learning
- ▼ Transfer-X
 - ▶ Transfer-X: PREVIEW Transfermodule
 - Catena-X Grundlagen
 - Catena-X Product Carbon Footprint: Rule
 - Catena-X: Mehrwertdienste des Geschäfts
 - Catena-X: Use Case Geschäftspartnerdaten
 - Catena-X: Datenraum und seine Eigenschaften
 - Catena-X: Funktionsweise des Datenaustausches im Datenraum
 - Motivation und Grundlagen der Verwaltungsschale
 - Verwaltungsschale Beispielanwendung – VWS4LS (Teil I)
 - Verwaltungsschale Beispielanwendung – VWS4LS (Teil II)
 - Digitale Geschäftsmodelle (Teil I)
 - Digitale Geschäftsmodelle (Teil II)
 - ToolING
 - Bedeutung von ERP für Digitalisierung und Prozessmanagement

transfer-x

Catena-X: Use Case Geschäftspartnerdaten Management

transfer-x

Catena-X: Datenraum und seine Eigenschaften

transfer-x

Catena-X: Funktionsweise des Datenaustausches im Datenraum

Verwaltungsschale Motivation und Grundlagen

transfer-x

Motivation und Grundlagen der Verwaltungsschale

Verwaltungsschale Beispielanwendung VWS4LS (Teil 1)

transfer-x

Verwaltungsschale Beispielanwendung VWS4LS (Teil 2)

transfer-x

Verwaltungsschale Beispielanwendung – VWS4LS (Teil I)

Verwaltungsschale Beispielanwendung – VWS4LS (Teil II)

Abbildung 11-2: [Transfer-X Lernmodule: Beiträge von VWS4LS](#)

Neben der digitalen Wissensbasis war die Teilnahme an Veranstaltungen von zentraler Bedeutung. Alle relevanten Veranstaltungen, wie die Bordnetzkongresse in Landshut und Ludwigsburg, wurden umfassend dokumentiert und auf der VWS4LS-Webseite vorgestellt.

Verwaltungsschale für den Leitungssatz

- VWS4LS Überblick
 - VWS4LS Ergebnisse
 - VWS4LS Ergebnistagung
 - Konzept, Informationsmodelle und Produktbeschreibung
 - Entwicklungsprozesse des Leitungssatzes
 - Produktionsprozesse des Leitungssatzes
 - Montageprozesse des Leitungssatzes in der Karosserie
 - Integration von Verwaltungsschalen (Verbundkomponente)
 - Automatisierte Verhandlungsprozesse
 - Data Business Policy, Data Governance und Monetarisierung
 - Data Storage Policy, Sicherheit, Anbindung an Catena-X
 - Pilotierung, Erprobung, Demonstrator
 - **Veranstaltungen von VWS4LS**
- Alle Projekte
► Abgeschlossene Projekte

ANSPRECHPARTNER



Georg Schnauffer
ARENA2036

23. Mai 2024 – VWS4LS-Konferenzveranstaltung

VWS4LS-Konferenzveranstaltung am 23.05.2024

Am 23.05.2024 fand von 9:00 Uhr bis 12:00 Uhr die vierte öffentliche Konferenzveranstaltung von Verwaltungsschale für den Leitungssatz (VWS4LS) statt. Die Veranstaltung bot eine Plattform für den Austausch von Informationen und die Vorstellung der Fortschritte in den verschiedenen Teilprojekten zu den Hauptthemen "Leitungssatzwertschöpfung" und "Verwaltungsschale". Die Teilnehmer:innen hatten die Möglichkeit, sich über die Fortschritte in den Teilprojekten zu informieren und wertvolle Einblicke in die digitale Transformation der Leitungssatz-Branche zu gewinnen. Die Präsentationen der verschiedenen Teilprojekte zeigten das Engagement und die Expertise der beteiligten Unternehmen und boten einen Ausblick, wie die Wertschöpfung von morgen in der Leitungssatzindustrie aussehen kann.

Für diejenigen, die nicht live dabei sein konnten, bieten wir die Möglichkeit, die präsentierten Folien herunterzuladen und die Vortragsaufzeichnung anzusehen.

[Zum Foliensatz \(PDF\)](#)

Veranstaltungsaufzeichnung:



Im Folgenden können Sie die detaillierte Agenda für die Veranstaltung einsehen:

[Agenda VWS4LS- Konferenzveranstaltung am 23.05.2024 von 09:00 Uhr bis 12:00 Uhr](#)

Abbildung 11-3 Darstellung der Konferenzveranstaltung vom 23.05.2024 mit Folien und Aufnahme

Ebenso wurden die Messeauftritte des Projekts, insbesondere auf der [Hannover Messe \(HMI\)](#) und der [Productronica](#), als wichtige Kommunikationsplattformen genutzt, um die Ergebnisse einem breiten Publikum zu präsentieren.

Ein Eckpfeiler des Transfers war die Durchführung von dreimal jährlich stattfindenden öffentlichen Schaufensterveranstaltungen, in denen das Projekt der breiten Öffentlichkeit präsentiert wurde. Diese virtuellen Veranstaltungen, die im Frühjahr und Herbst durchgeführt wurden, boten einen praxisnahen Einblick in die Anwendungen der Verwaltungsschale sowie des Leitungssatzes und trugen dazu bei, das Konzept in der Industrie zu verankern. Die Präsentationen wurden aufgezeichnet und ebenfalls öffentlich zugänglich gemacht.

Darüber hinaus, wurde ein AAS-Expertenworkshop mit der Smartfactory Kaiserlsuatern ins Leben gerufen, um sich rund um das Thema VWS austauschen zu können. Dieser Workshop hat mittlerweile zum 3. Mal stattgefunden und soll auch weiterer Bestand des Transfers über das Projekt VWS4LS hinaus sein. Fokus der regelmäßigen treffen ist vor allem die Anwendung der VWS und neue Use Cases kennen zu lernen. Aus dem Projekt VWS4LS heraus wurden in allen Veranstaltungen bisher die Ergebnisse präsentiert und diskutiert.

11.1.1 Veranstaltungen

Folgende Veranstaltungen wurden im Rahmen des Projekts VWS4LS durchgeführt bzw. mit Beiträgen unterstützt:

| Datum | Veranstaltung | Kurzbeschreibung/Tätigkeiten |
|-------------------|---|--|
| 07.03.2022 | Schulungsveranstaltung | Virtuelle Verwaltungsschalen-Schulung durch University4Industry im Verbund mit dem Projekt „SDM4FZI“ |
| 26.- 27.04.2022 | D3con Hamburg | Kooperation mit dem Veranstalter zur Sondierung der Potenziale von Programmatic Advertising-Ansätzen für automatisierte Verhandlungsprozesse auf einer der weltweit führenden Online Advertising Konferenzen |
| 30.5.- 02.06.2022 | Hannover Messe 2022 | Projektvorstellung auf dem Stand der Plattform Industrie 4.0 mit VWS4LS-Demonstrator. Zusätzliche Vorträge: <ul style="list-style-type: none"> • Projektvorstellung auf der Konferenzstage der Plattform Industrie 4.0 • Projektvorstellung auf der Bühne des BMWK |
| 30.5.- 02.06.2022 | Bordnetzkongress Ludwigsburg | Vortrag mit Projektvorstellung auf dem wichtigsten Fachkongress für das Bordnetz in Europa |
| 28.06.2022 | 1. Transferkonferenz | Öffentliche virtuelle Projektvorstellung von VWS4LS, Dauer: 3 Stunden, die Aufzeichnung wurde veröffentlicht. |
| 20.- 21.07.2022 | Digital Product Forum (Mercedes-Benz) | Projektvorstellung mit eigenem Stand und VWS4LS-Demonstrator auf der globalen Zulieferer-Messe der Mercedes-Benz AG auf der Messe Stuttgart |
| 12.09.2022 | Pitchday Datenraum Industrie 4.0 | Vorstellung des Zusammenhangs zwischen Datenräumen und dem Projekt VWS4LS in Form eines neuen Projektansatzes basierend auf der Integration von Datenraum-basierter Kollaboration (Schwerpunkt des TP 8) |
| 20.09.2022 | Bordnetzkongress Landshut | Vortrag auf dem Fachkongress für das Bordnetz an der Hochschule Landshut |
| 25.- 26.10.2022 | Klausur der Plattform Industrie 4.0 | Austausch mit Experten und Vernetzung mit Experten der Plattform Industrie 4.0, insbesondere der AG 1 |
| 16.- 18.11.2022 | Stuttgarter Conference on Automotive Production | Teilnahme und Vorstellung des Projektes auf der wissenschaftlichen Konferenz SCAP in Stuttgart mit Fachpublikation im Tagungsband |
| 22.11.2022 | 2. Transferkonferenz | Öffentliche virtuelle Projektvorstellung von VWS4LS, Dauer: 3 Stunden, die Aufzeichnung wurde veröffentlicht. |
| 08.- 09.12.2022 | AAS-Techdays der IDTA | Austausch mit Experten und Vernetzung zu Herausforderungen zur Verwaltungsschale mit anderen 35c-Vertetern |
| 17.-21.04.2023 | Hannover Messe 2023 | Eigener Stand von VWS4LS in Halle 8, Vorstellung des Projektes mit erweiterten VWS4LS-Demonstratoren zusammen mit Vertretern der Partner als Standpersonal |
| 09.- 10.5.2023 | Bordnetzkongress Ludwigsburg | Vortrag mit Projektvorstellung sowie eigener Ausstellungsstand mit VWS4LS Demonstrator auf dem wichtigsten Fachkongress für das Bordnetz in Europa |

| Datum | Veranstaltung | Kurzbeschreibung/Tätigkeiten |
|-----------------|--|--|
| 25.05.2023 | 3. Transferkonferenz | Öffentliche virtuelle Projektvorstellung von VWS4LS, Dauer: 3 Stunden, die Aufzeichnung wurde veröffentlicht. |
| 13.06.2023 | Halbzeit Meilenstein-treffen | Aufbereitung und Vorstellung bisheriger Projektergebnisse zur Fortsetzung der 2. Projekthälfte |
| 21.09.2023 | Bordnetzkongress Landshut | Teilnahme auf dem Fachkongress für das Bordnetz an der Hochschule Landshut |
| 09.11.2023 | AAS-Experten-workshop | Ausrichtung des Netzwerktreffens mit der SmartFactory Kaiserslautern und Vorstellung einiger Projektergebnisse |
| 14. -17.11.2023 | VWS4LS auf der productronica | Eigener Stand von VWS4LS in Halle B4, Vorstellung des Projektes mit erweitertem VWS4LS-Demonstrator |
| 21.03.2024 | AAS-Experten-workshop | Ausrichtung des Netzwerktreffens mit der SmartFactory Kaiserslautern und Vorstellung einiger Projektergebnisse |
| 11.04.2024 | Innovationsforum Leitungssatz 2024 | Teilnahme auf dem Fachkongress für das Bordnetz in der ARENA2036 |
| 22.-26.04.2024 | VWS4LS auf der Hannover Messe 2024 | Eigene Standfläche auf dem Gemeinschaftstand der ARENA2036. Darstellung neuer Projektergebnisse und Vorstellung eines Demonstrators, sowie der Projektvorstellung auf der Conference-Stage Industry 4.0 zusammen mit Vertretern der Partner als Standpersonal. |
| 23.05.2024 | 4. Transferkonferenz | Öffentliche virtuelle Projektvorstellung von VWS4LS, Dauer: 3 Stunden, die Aufzeichnung wurde veröffentlicht |
| 25.09.2024 | AAS-Experten-workshop | Ausrichtung des Netzwerktreffens mit der SmartFactory Kaiserslautern und Vorstellung einiger Projektergebnisse |
| 26.09.2024 | Bordnetze Digital 2024 | Teilnahme und Projektvortrag an dem Fachkongress zum Thema Anbindung an Catena-X |
| 15.11.2024 | VWS4LS Ergebnista-gung | Vorstellung der Projekthighlights der Öffentlichkeit |

11.1.2 Publikationen

Folgende Publikationen erfolgten im Rahmen des Projekts VWS4LS:

- Salinas Segura, A., Angos Mediavilla, M., Braun, L., Freund, M., Kosel, C., Rodriguez, M. (2024). **"A Process Model for Deriving Asset Administration Shells for Inter-company Collaboration – A Practical Approach"**; In: Yilmaz, M., Clarke, P., Riel, A., Messnarz, R., Greiner, C., Peisl, T. (eds) Systems, Software and Services Process Improvement. EuroSPI 2024. Communications in Computer and Information Science, vol 2180. Springer, Cham. https://doi.org/10.1007/978-3-031-71142-8_16
- Melanie Stolze, Alexander Belyaev, Christian Kosel, Christian Diedrich, Alfred Barnard, "Realizing automated production planning via proactive AAS and Business Process Models", 2024 29th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Padua, Italy, 2024. [71]

- Melanie Stolze, Gustavo Cainelli, Christian Kosel, Alexander Belyaev, Christian Diedrich (2024); **Entwurf komplexer Automatisierungssysteme: Konzepte zur Realisierung proaktiver Verwaltungsschalen und deren Kommunikation**; In *EKA 2024 - Entwurf komplexer Automatisierungssysteme, 18. Fachtagung (Jumar, Ulrich et al.)*. [72] [Link](#)
- Angos-Mediavilla, M.; Gorenzweig, M.; Pahnke, G.; Pomp, A.; Freund, M. and Meisen, T. (2024). **Advancing Industry 4.0: Integrating Data Governance into Asset Administration Shell for Enhanced Interoperability**. In *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS*; ISBN 978-989-758-692-7; ISSN 2184-4992, SciTePress, pages 128-140. DOI: 10.5220/0012632900003690 [Link](#) [PDF](#)
- Bernd Kärcher (2024), **VWS4LS: Datengetriebene Leitungssatzproduktion mit OPC UA**; Fachartikel VDMA. [Link](#)
- Mario Angos Mediavilla, Michele Lagnese, André Pomp, Tobias Meisen (2023); **Asset administration shell-based engineering change management process: Challenges and ways forward**; Procedia CIRP, Volume 120, 2023, Pages 1010-1015; ISSN 2212-8271; <https://doi.org/10.1016/j.procir.2023.09.116>. [Link](#)
- Schnauffer, G., Görzig, D., Kosel, C., Diemer, J. (2022). **Asset Administration Shell for the Wiring Harness System**. In: Kiefl, N., Wulle, F., Ackermann, C., Holder, D. (eds) In *Advances in Automotive Production Technology – Towards Software-Defined Manufacturing and Resilient Supply Chains*. SCAP 2022. ARENA2036. Springer, Cham. https://doi.org/10.1007/978-3-031-27933-1_30 [Link](#)

11.2 AP 10.2 - Allgemeines Implementierungspaket

Möglichst viele Vorgehensweisen im Projekt wurden von vornherein auf den Transfer ausgerichtet, sowohl für den internen Transfer (Implementierungsvorbereitung bei den Partnern), als auch für den Transfer in den ungeförderten Raum (Breitentransfer).

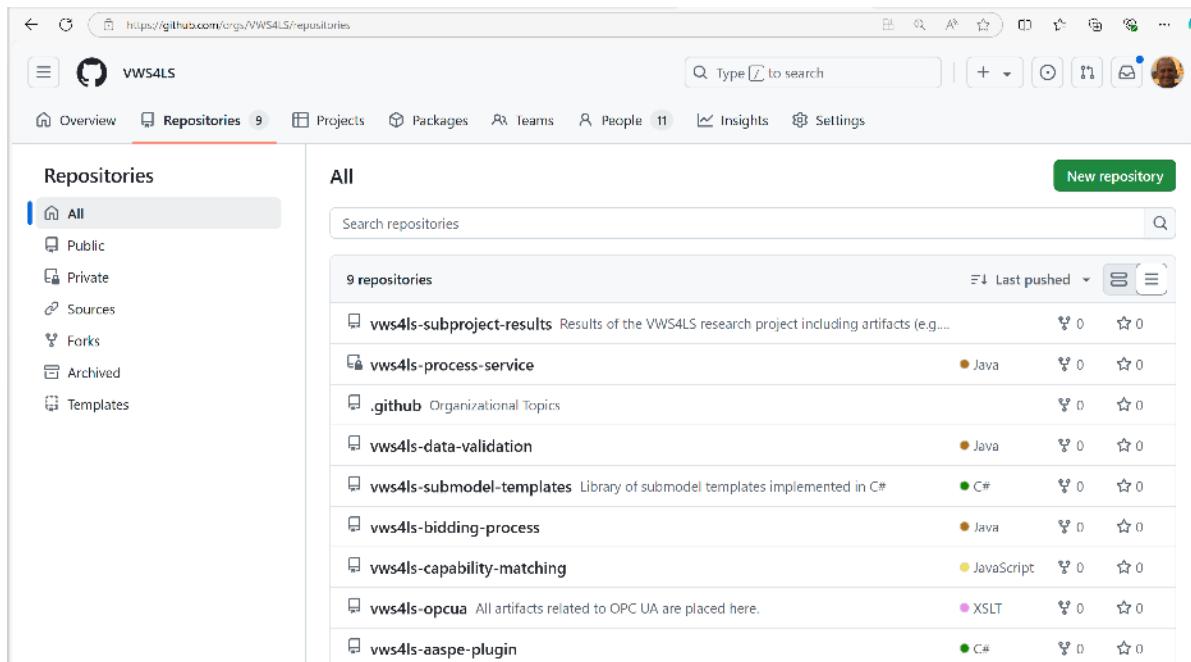


Abbildung 11-4: [VWS4LS auf Github](#)

Insbesondere wurden die Ergebnisse des Projekts systematisch auf einer eigenen [VWS4LS-GitHub-Präsenz](#)¹¹¹ [24] veröffentlicht, um die technischen Ergebnisse für Entwickler, Unternehmen, Forschungseinrichtungen und die allgemeine Öffentlichkeit frei zugänglich zu machen, einschließlich Quellcodes, Informationsmodellen und Implementierungsanleitungen. Diese offene Plattform förderte die Nachnutzung der erarbeiteten Lösungen und erleichterte den Wissenstransfer in die Praxis.

¹¹¹ <https://github.com/VWS4LS>

11.3 AP 10.3 - Implementierungspaket Leitungssatz

Aufbauend auf dem allgemeinen Implementierungspaket wurden die Spezifika der Umsetzung in der Domäne des Leitungssatzes aufbereitet, bspw. die Informationsmodellierung und Anwendung der spezifischen Submodels oder die Integration domainspezifischer Datenformate, wie z.B. KBL und VEC.

Fachlich-inhaltliche Quellen, insbesondere der Plattform Industrie 4.0, wurden referenziert und in einen Implementierungskontext gesetzt.

Ein wesentlicher Bestandteil der Koordination war auch die Vernetzung mit Multiplikator-Organisationen, darunter die [Plattform Industrie 4.0](#), die [Industrial Digital Twin Association \(IDTA\)](#) und dem [Transformationshub Leitungssatz](#)¹¹², um eine breite Anwendung der Verwaltungsschale in der Industrie zu fördern (siehe Abbildung 11-1: Transferkontext).

11.4 AP 10.4 - Ergebnisintegration und Koordination

Die Ergebnisintegration und Koordination des Projekts wurde maßgeblich durch die ARENA2036 gesteuert. Das Projektmanagement wurde von der ARENA2036 übernommen, die auch die Koordination der Projektpartner und deren Ergebnisse verantwortete. Durch die enge Vernetzung und Steuerung wurde sichergestellt, dass alle Teilprojekte auf die übergeordnete Zielstellung ausgerichtet und die verschiedenen Arbeitspakete erfolgreich integriert wurden.

Durch die enge Zusammenarbeit zwischen Projektpartnern, Multiplikatoren und externen Stakeholdern wurde die Integration der Ergebnisse entlang der übergeordneten Zielstellungen des Projekts effizient gesteuert und koordiniert. Diese Maßnahmen trugen maßgeblich dazu bei, dass die Projektergebnisse auf breiter Ebene verwertet und nachhaltig in die Industrie transferiert werden konnten.

Es fehlt noch ein expliziter Abgleich mit den Versprechungen in der GVB.

¹¹² <https://www.leitungssatz-hub.de/>

12 Architektur AAS

In diesem Kapitel wurden TP-übergreifende Architekturthemen dokumentiert.

12.1 Einführung

Im Rahmen eines projektinternen Workshops am 16.02.2023 wurde ein Big-Picture zur Orientierung für das Gesamtprojekt erstellt. Als Orientierungsrahmen diente dabei das Themenpuzzle in Abbildung 12-1. Die Teilnehmer, sowohl Experten aus dem Bereich der Leitungssatzbranche als auch aus dem Bereich der Verwaltungsschale, sollten klären, welche Lücken für die Anwendbarkeit der Verwaltungsschale in einer realen Wertkette geschlossen werden müssen und was aus Sicht der führenden Institute aus dem Bereich der Verwaltungsschalenanwendung schon möglich ist, bzw. in naher Zukunft umgesetzt werden kann.

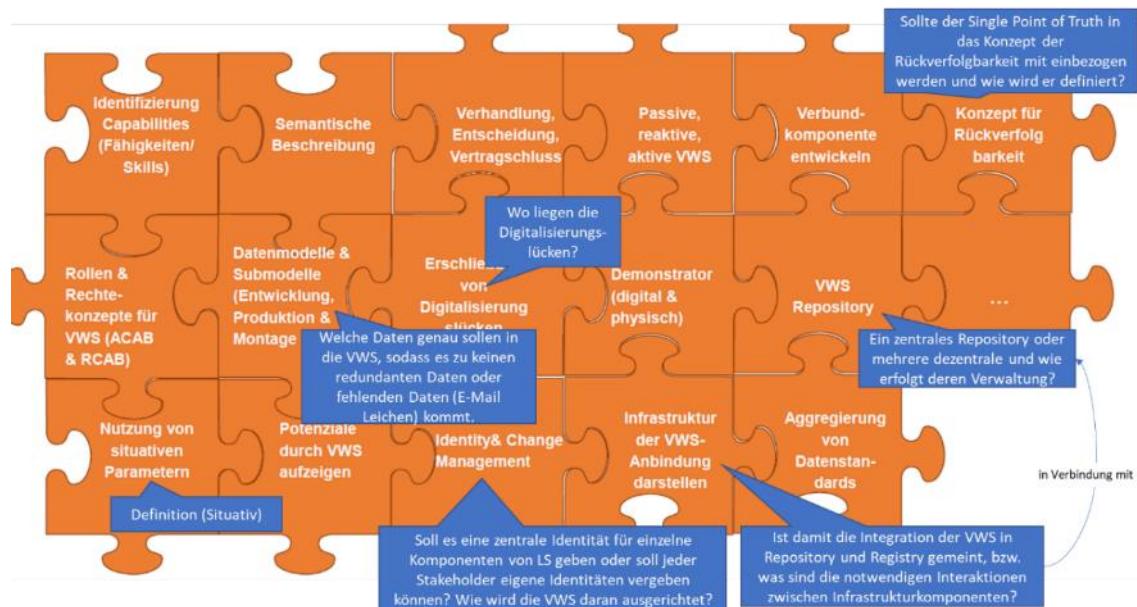


Abbildung 12-1: Big Picture Workshop - Fragen

Um die Ausarbeitung voranzutreiben, wurde zu Beginn als gemeinsamer Startpunkt ein konkreter Use Case festgelegt, hier der Crimp-Prozess, anhand dessen die relevanten umsetzungstechnischen Fragen an die VWS formuliert wurden.

Wesentliche identifizierte Kernpunkte für die architektonischen Gestaltungsfelder waren zum einen die IT-Architektur bzgl. der VWS-Verwaltung, sowie der sog. „Single Point of Truth“ [73], auf welche in den Folgekapiteln näher eingegangen wird.

12.1.1 Zentrale vs. dezentrale Architektur

Ein typischer **zentral orientierter Architekturansatz** ist in Abbildung 12-2 dargestellt. Bei der Diskussion dieses Ansatzes wurde jedoch deutlich, dass er nicht in jeder Situation der Leitungssatzentwicklung und -produktion genutzt würde und sich außerdem die Frage stellt, wer für die Verwaltung des zentralen VWS-Repositories zuständig wäre.

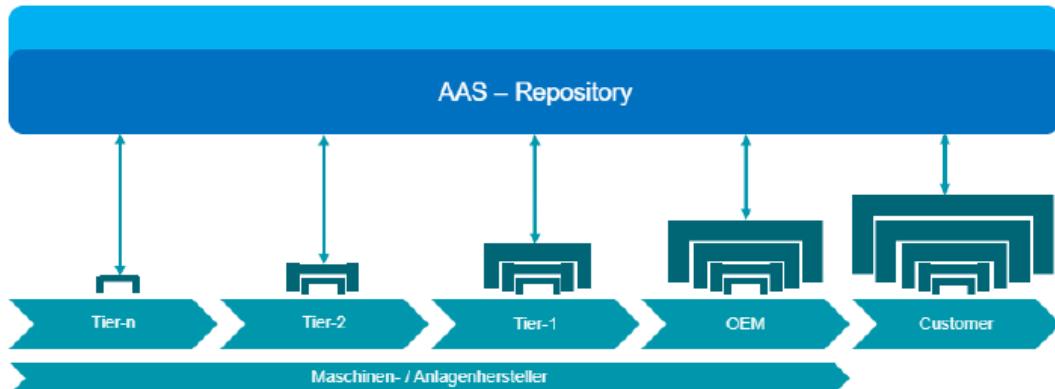


Abbildung 12-2: Zentrale VWS-Architektur

Bei einer **dezentralen Architektur** wird davon ausgegangen, dass der OEM und jeder Zulieferer ihr eigenes Repository hosten, d.h. es im System mehr als eine VWS-Registry und ein VWS-Repository gibt [74]. Jeder Auftraggeber in der Wertkette betreibt eine Registry, um es den Auftragnehmern zu ermöglichen, sich an der jeweiligen VWS des Auftraggebers anzumelden. Die resultierende IT-Architektur könnte sich wie in Abbildung 12-3 dargestellt an der entwickelten VWS-Struktur des Bordnetzes orientieren und eine entsprechende Vernetzung von VWS-Repositorien und -Registries in einem vermaschten Netzwerk ergeben, in dem jedes Unternehmen seine VWS anderen Unternehmen zur Verfügung stellen kann. Das geschieht über die Bereitstellung von VWS-Registries auf der Seite der Unternehmen, die Daten nach außen geben wollen. Durch entsprechende Zugriffsbeschränkungen können dann externe Unternehmen auf die für sie freigegebenen VWS zugreifen. Diese Informationen aus der Referenzierung können später für die Synchronisation genutzt werden.

Anmerkung: Abbildung 12-3 zeigt die erweiterte dezentrale Architektur beispielhaft mit nur einem OEM

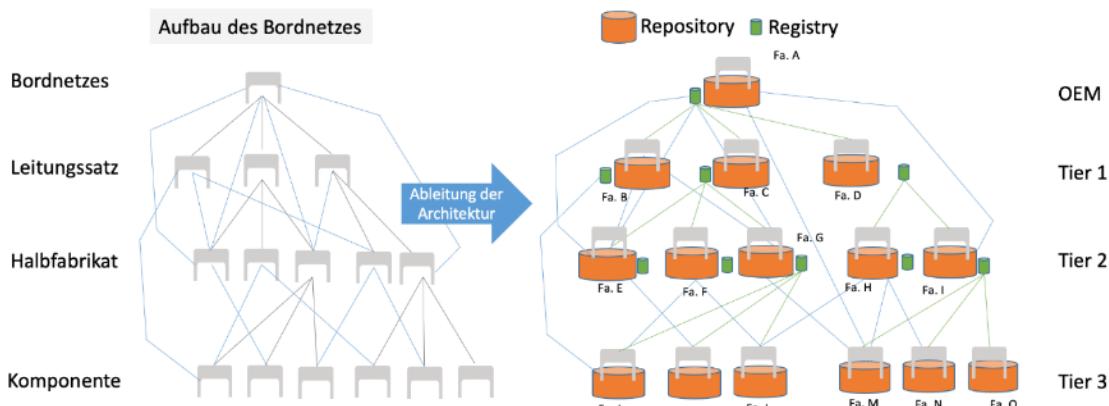


Abbildung 12-3: Dezentrale VWS-Architektur

Bei diesem dezentralen Ansatz setzen sich die Informationen der VWS einer Leitungssatz-Komponente (LS-Komponente) also über die Hierarchiestufen mehrerer Unternehmen zusammen, deren VWS aufeinander verweisen. Beispielsweise werden die Daten für LS-Komponenten von den Komponentenherstellern initial erzeugt und später von dem Konfektionär oder OEM genutzt, jedoch mit einer reduzierten Sicht auf die Inhalte (bspw. werden lediglich Informationen wie bzw. Digital Nameplate [28] und Produktdaten [69] weitergegeben).

12.1.2 Single Point of Truth

Ein wichtiger Punkt im Kontext der Architektur ist die Definition des sog. „Single Point of Truth“ (SPoT) [73], also wo die aktuell gültige Version der VWS abgelegt ist, bzw. bezogen werden kann.

Aus den bisherigen Ergebnissen des VWS4LS-TP2 „Entwicklung des Leitungssatzes“, geht hervor, dass der SPoT in den verschiedenen Lebenszyklen des Leitungssatzes immer bei einem anderen Teilnehmer liegt und deshalb eine entsprechende Rückverfolgbarkeit durch die Schichten gewährleistet sein muss. In der Entwicklungsphase bspw. besitzt der OEM den SPoT, während in der Produktion der Konfektionär den SPoT besitzt (Abbildung 12-4).



Abbildung 12-4: Wechselnder SPoT im Lebenszyklus des Leitungssatzes

Wobei sich die Frage stellt, ob diese Festlegung über alle Schichten (Tier-1 bis Tier-n) hinweg gültig ist, oder es in jeder Schicht SPoT's geben kann, die nur an die dortige LS-Komponente gebunden sind.

Der SPoT wurde anhand des folgenden konkreten Beispiels näher betrachtet, der Reaktionskette in einem Schadensfall an einem Auto:

Ist ein Schaden in der Elektronik an einem Auto aufgetreten, wendet sich der Kunde zuerst mit der Seriennummer des Autos an den OEM. Der OEM ist für den Kunden die erste Anlaufstelle, da er alle Herstellungsdaten zusammenführt und somit sein SPoT ist. Der OEM wiederum betrachtet den Schadensfall und kann den betroffenen Leitungssatz einem Konfektionär zuordnen. Hier ist der Konfektionär der SPoT für den OEM. Dieses Verfahren geht bis in die unterste Ebene.

Das Beispiel in Abbildung 12-5 verdeutlicht, wie die VWS-Struktur den Aufbau des Leitungssatzes abbildet und wo die kontextabhängigen SPoTs liegen, die zusammen ein sog. „Network of Truth“ bilden.

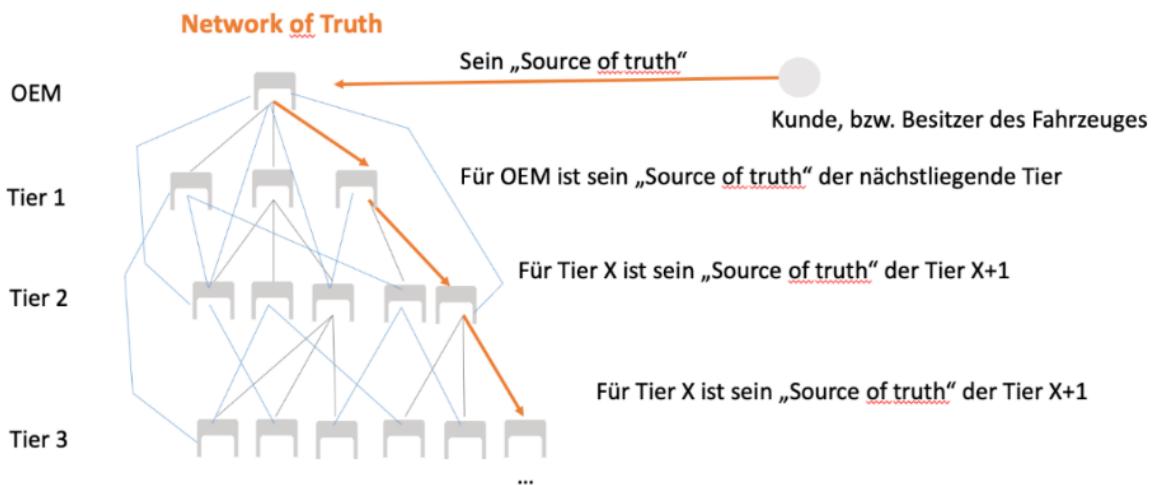


Abbildung 12-5: Network of Truth

12.1.3 Gestaltungsfelder

Es wurden sechs grundsätzliche architektonische Gestaltungsfelder in Verbindung mit der VWS identifiziert, diese sind in Tabelle 12-1 aufgelistet.

Tabelle 12-1 Architektonische Grundfragestellungen

| Gestaltungsfelder für Architektur der VWS | Kurzerklärung |
|---|---|
| Modularisierung | Strukturierung von VWS und Teilmodellen, d.h. die Partitionierung und Vernetzung von VWS untereinander. |
| Verlinkung | Verlinkung als technische Lösung zur Vernetzung von Teilmodellen, wie zum Beispiel dem Teilmodell „Leitungssatzspezifikation“ und dem Teilmodell „Crimpprozess“, sowie Teilmodell-Versionen und VWS. Verlinkungen können sowohl innerhalb einer VWS als auch zwischen mehreren Verwaltungsschalen entstehen. |
| Versionierung | Versionierung in der VWS dient zur Verwaltung und Nachverfolgung von Änderungen an Daten und Teilmodellen. Damit können unterschiedliche Versionen erstellt und verwaltet werden, denn es wird sichergestellt, dass alle Änderungen nachvollziehbar und abgrenzbar sind. Dies ist besonders wichtig, wenn mehrere Akteure beteiligt sind, um Klarheit darüber schaffen zu können, welche Variante gültig ist und welche Version zu einem bestimmten Zeitpunkt für bestimmte Anwendungen oder Benutzer relevant ist. |
| Synchronisation | Synchronisation im Bereich der VWS bezieht sich auf die technische Lösung zur Aktualisierung abhängiger Daten, um sicherzustellen, dass alle Informationen konsistent und aktuell sind. Dabei ergeben sich einige wichtige Fragestellungen: Wie identifiziert man die Verlinkungen oder referenzierten Daten? Wie wird die Übereinstimmung zwischen dem physischen Asset und den Daten in den VWS gewährleistet? Diese Aspekte sind entscheidend, um eine verlässliche Synchronisation sicherzustellen. |
| Rückverfolgbarkeit | Wie kann die durchgängige Rückverfolgbarkeit von Daten in der Wertschöpfungskette auch über Unternehmensgrenzen hinweg sichergestellt werden? |
| Änderungsmanagement | Wie können in Bezug auf die Engineering-Prozesse in der Leitungssatzentwicklung Änderungen an VWS teilautomatisiert umgesetzt werden? Wie werden Änderungen einer übergeordneten Instanz lokalisiert und dokumentiert? |

Diese sechs Grundbausteine werden in den Folgekapiteln besprochen und sollen anschließend den führenden VWS-Initiativen zur Verfügung gestellt und von diesen bewertet werden.

12.2 Modularisierung

Da jede VWS über spezifische Teilmodelle verfügt, befasst sich die Modularisierung mit der inhaltlichen Struktur der definierten VWS.

Der Leitungssatz wird dabei sinnvoll in mehrere, eigenständige Module aufgeteilt, die ihre eigenen VWS und Teilmodelle besitzen können. Jede im Leitungssatz verbaute Komponente oder Halbfabrikat kann als Modul definiert werden.

- Ein **Halbfabrikat im Bordnetz** ist ein vorgefertigtes, teilweise bearbeitetes Bauteil oder eine Baugruppe, die im Herstellungsprozess eines Fahrzeugkabelbaums verwendet wird.
- Ein **Funktionsmodul** ist eine vorkonfigurierte, eigenständige Komponente oder Einheit innerhalb eines Kabelbaums, die spezifische elektrische oder elektronische Funktionen erfüllt. Diese Module sind so gestaltet, dass sie sich nahtlos in den Gesamtleitungssatz integrieren lassen und bestimmte Aufgaben wie die Verbindung von Sensoren, Aktoren, Steuergeräten und anderen elektrischen Komponenten übernehmen.

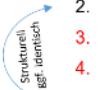
In der Analyse der erforderlichen VWS und Teilmodelle für die Abbildung des Leitungssatzes ist es von entscheidender Bedeutung, die verschiedenen Stakeholder im Entwicklungs- und Produktionsprozess zu berücksichtigen für eine detaillierte Betrachtung darüber,

- welche Informationen über den Leitungssatz benötigt werden,
- welche Stakeholder VWS mit diesen Informationen besitzen sollen, und
- welche spezifischen Inhalte diese VWS für die Stakeholder haben sollen.

Ebenso wichtig ist die Klärung, mit wem diese Inhalte geteilt werden müssen und in welchem Umfang dies geschehen muss.

Als Orientierungshilfe diente auch das PPR-Modell) [4], um festzulegen, welches Asset eine VWS erhält und wer im Lebenszyklus des Leitungssatzes Informationen dieses Assets benötigt bzw. beiträgt (Abbildung 12-6).

Legende:
Asset / Nicht-Asset



1. Komponenten (Typ) (Gehäuse, Kontakte, ...)
2. Leitungssatz-Def (Typ) (150% KBL/VEC)
3. Bill of Process (ergänzend zu 2.)
4. Manufacturing BOM (ergänzend zu 2.)
5. Halbfabrikat (Typ) (z.B. geschnitten + crimped Leitung)
6. Prozess-Schritt (ergänzend zu 5.)
7. Ressourcentyp
8. Fähigkeitsdefinition (ergänzend zu 7.)
9. Ressourcen-Instanz
10. Werk / Maschinenpark (Instanz)
11. Komp. Instanz (Charge od. Einzelteil)
12. Halbfabrikat (Instanz) (Charge od. Einzelteil)
13. Job / Prozessschritt Instanz (ergänzend zu 12, 14)
14. Leitungssatz Instanz
15. Order (ergänzend zu 15)
16. Ergebnisdaten (ergänzend zu 14)

| OEM | Konf. | Komp. Herst. | M/C |
|-----|-------|--------------|-----|
| o | o | o | o |
| o | o | | |
| | | | |
| | | o | |
| | | | |
| | | o | o |
| | | | |
| | o | | o |
| | | o | o |
| | | o | |
| o | o | | |
| | | | |
| | | | |

Abbildung 12-6: Ermittlung der Assets und Nicht-Assets mit Stakeholderzuweisung

Es hat sich dabei als zielführend herausgestellt, zwischen den beiden im Projekt überwiegend betrachteten Lebensphasen des Leitungssatzes zu unterscheiden: Im **Entwicklungsprozess** stehen die Typ-Beschreibungen von Modulen des Leitungssatzes im Mittelpunkt und dementsprechend deren VWS. Im **Produktionsprozess** geht es hingegen um die Instanzen der verbauten Module, also die konkreten Umsetzungen der im Entwicklungsprozess definierten Assets. Dementsprechend werden für den Produktionsprozess Verwaltungsschalen für die Instanzen von Modulen benötigt.

12.2.1 Aufteilung des Leitungssatzes in Module

Die Analyse beginnt mit der Identifizierung der verschiedenen Funktionsmodule und ihrer Beziehungen zueinander, aus denen der Leitungssatz besteht. Ein **Funktionsmodul** ist eine vorkonfigurierte, eigenständige Komponente oder Einheit eines Bordnetzes, die spezifische elektrische oder elektronische Funktionen erfüllt. Diese Module sind so gestaltet, dass sie sich nahtlos in den Gesamtleitungssatz integrieren lassen und bestimmte Aufgaben wie die Verbindung von Sensoren, Aktoren, Steuergeräten und anderen elektrischen Komponenten übernehmen.

Ein Funktionsmodul lässt sich wiederum in mehrere Halbfabrikate unterteilen. Ein **Halbfabrikat** im Bordnetz ist ein vorgefertigtes, teilweise bearbeitetes Bauteil oder eine Baugruppe, die im Herstellungsprozess eines Fahrzeugkabelbaums verwendet wird. Diese Halbfabrikate sind im Gegensatz zu Funktionsmodulen keine Endprodukte, sondern Zwischenstufen, die noch weiterverarbeitet werden müssen, um in das Gesamtsystem des Fahrzeubordnetzes eingebaut zu werden.

Zudem werden die **Ressourcen** mit ihren Informationen betrachtet, die im Produktionsprozess zur Herstellung des Leitungssatzes genutzt werden (siehe Abbildung 12-7).

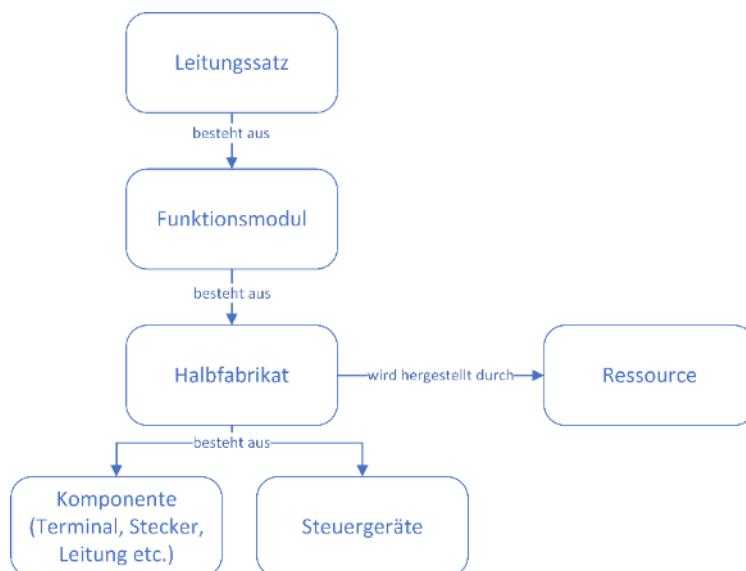


Abbildung 12-7: Modularisierung des Leitungssatzes

12.2.2 Verwaltungsschalen im Entwicklungsprozess

Im Entwicklungsprozess des Leitungssatzes entstehen bei den verschiedenen Stakeholdern eigene VWS für ein und dasselbe Asset, was in Abbildung 12-8 schematisch dargestellt ist.

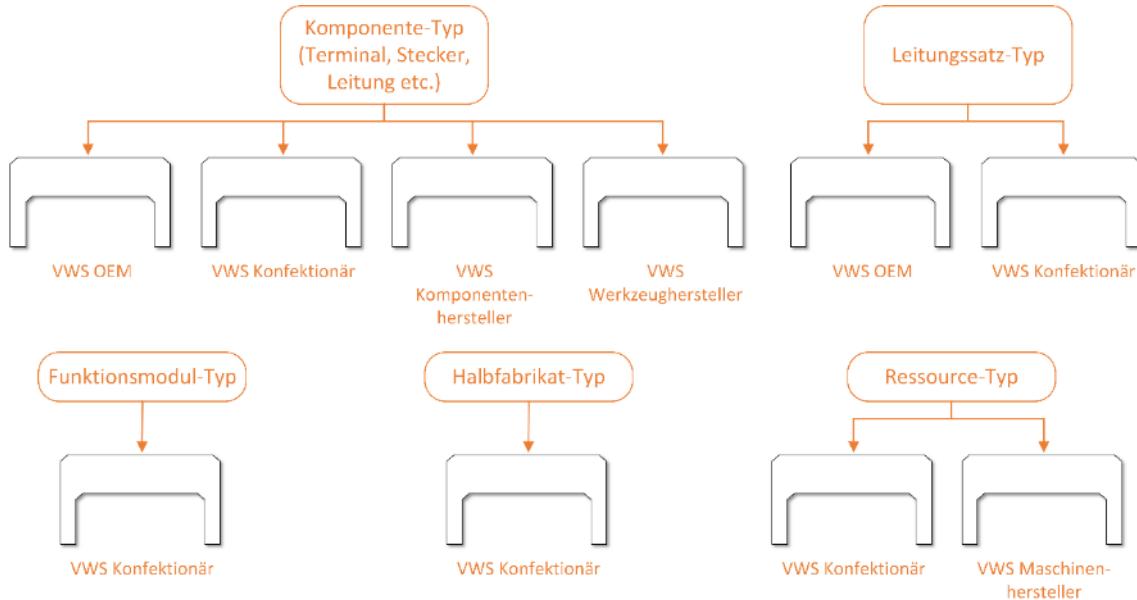


Abbildung 12-8: Übersicht über VWS der jeweiligen Stakeholder im Entwicklungsprozess

Hierbei kann davon ausgegangen werden, dass die Stakeholder sowohl Asset-Informationen aus den VWS anderer Stakeholder beziehen als auch selbst Informationen zum Asset hinzufügen wollen. Deshalb wurde entschieden, die Möglichkeit der Existenz mehrerer VWS zu einem Asset zuzulassen und die Zuordnung der VWS untereinander über Versionierung und Verlinkung zu lösen.

12.2.3 Verwaltungsschalen im Produktionsprozess

Ähnlich wie im Entwicklungsprozess, gibt es auch für den Produktionsprozess des Leitungssatzes mehrere Stakeholder-spezifische VWS für ein und dasselbe Asset. Im Produktionsprozess entstehen somit mehrere VWS zu einem Asset, was in Abbildung 12-9 dargestellt ist.

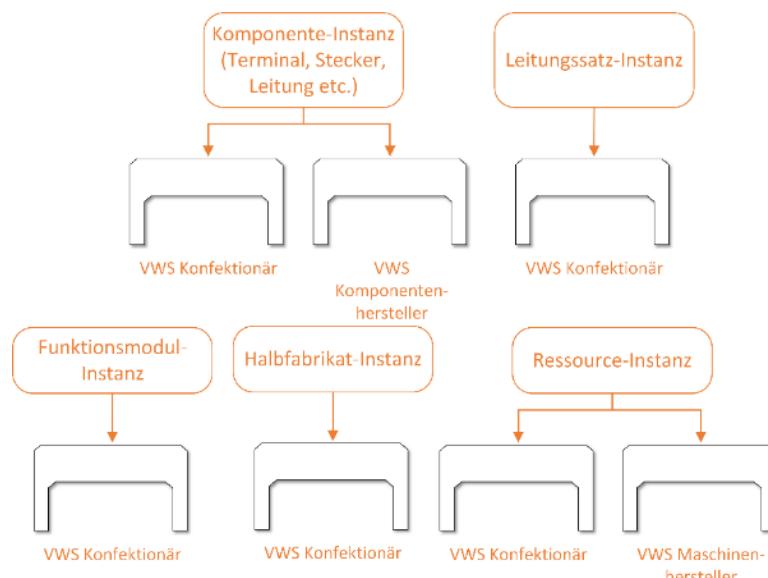


Abbildung 12-9: Übersicht über VWS der jeweiligen Stakeholder im Produktionsprozess

12.2.4 Inhalte und Verknüpfung der Stakeholder-Verwaltungsschalen

Die Stakeholder-spezifischen VWS enthalten Informationen, die den Anforderungen und Erwartungen des jeweiligen Stakeholders gerecht werden. Dies können technische Spezifikationen, Sicherheitsrichtlinien, Qualitätsstandards, Produktionsinformationen, Dokumentationen und andere relevante Informationen sein, die in Form von Teilmodellen oder verlinkten proprietären Dateiformaten vorliegen. Ein Abgleich der abzubildenden Informationen mit den bereits bei der IDTA vorhandenen Teilmodellen [27], zeigt die Lücken für weiteren Entwicklungsbedarf seitens des Projektes auf. Neu zu entwickelnde Teilmodelle werden in den folgenden Abschnitten explizit genannt.

Durch die kollaborative Zusammenarbeit der Stakeholder war zu klären, welche VWS-Inhalte untereinander geteilt werden. Dazu wurde eine tabellarische Anforderungsanalyse durchgeführt, die bis zur Ebene des Inhalts eines Teilmodells die Zugriffsrechte definiert (siehe Datei „[Anforderungsanalyse Architektur 20230713.xlsx](#)“¹¹³.

In den folgenden Abbildungen werden die VWS aus Abbildung 12-8 und Abbildung 12-9 mit Teilmodellen angereichert. Grüne und rote Markierungen vor den Teilmodellen kennzeichnen die Zugriffsrechte. Grüne Teilmodelle werden extern geteilt, die Roten dürfen nur durch den Stakeholder gelesen und bearbeitet werden. Zu einigen Teilmodellen, wie bspw. dem Teilmodell „IDTA 02204: Handover Documentation“ [68] [68] ist zusätzlich beschrieben, welche Dokumente dort abgelegt werden. Weiterhin zeigen die Abbildungen die Verknüpfung der unterschiedlichen Stakeholder-VWS in dem Entwicklungs- und Produktionsprozess des Leitungssatzes. Ein Hosting der VWS auf verschiedenen Servern wird ebenfalls angedeutet.

12.2.4.1 VWS für die Ressource

Sowohl die VWS zu den Ressourcen-Typen, aus denen sich der Konfektionär eine Ressource konfigurieren kann, als auch die VWS zu den Ressourcen-Instanzen, die später beim Konfektionär betrieben werden, erhalten ein Repository (VWS Repo-Typen und VWS Repo-Instanzen) bei dem Maschinenhersteller (Abbildung 12-10).

Die Abbildung des modularen Aufbaus der Ressourcen wird in der Ressourcen-VWS mit Hilfe des Teilmodells „IDTA 02011: Hierarchical Structures enabling Bills of Material“ [30] [30] vorgenommen, dass Referenzen auf die einzelnen VWS der Ressourcenmodule beinhaltet. In den Modul-VWS ist lediglich das digitale Typenschild [28] zur Identifizierung des Moduls vorhanden.

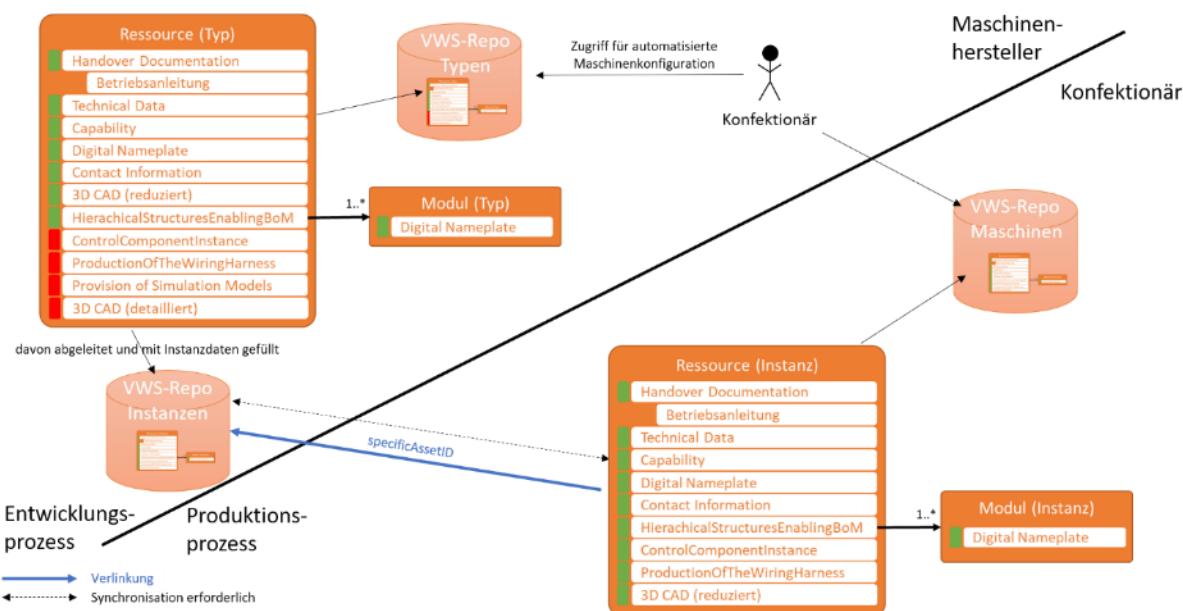


Abbildung 12-10: VWS und deren Verknüpfung für die Ressource

¹¹³ https://github.com/VWS4LS/vws4ls-subproject-results/blob/main/General/Anforderungsanalyse_Architektur_20230713.xlsx

Hat der Konfektionär mit Hilfe der VWS der Ressourcen-Typen seine Maschine im Entwicklungsprozess des Leitungssatzes konfiguriert und wurde diese errichtet, erhält der Konfektionär mit Anlieferung der Maschine auch die VWS der Ressourcen-Instanz und kann diese in seinem eigenen Repository abspeichern. Hierbei erhält der Konfektionär nur die Teilmodelle, die für ihn vom Maschinenhersteller freigegeben sind. Damit bei Änderungen durch den Maschinenhersteller auch die Informationen bei dem Konfektionär aktuell gehalten werden können, wird mit Hilfe der Verlinkung zwischen den beiden VWS der Ressourcen-Instanz eine Beziehung gelegt.

Möchte der Konfektionär eigene Informationen zu den Maschinen hinzufügen ist eine Möglichkeit, eine weitere VWS für die Ressource im eigenen System anzulegen und diese über die *specificAssetID* des VWS-Metamodells Version 3 zu verknüpfen. Zu sehen ist der beschriebene Aufbau in Abbildung 12-10.

12.2.4.2 VWS für die Komponenten

Die Informationen sowohl zu den Komponenten-Typen als auch -Instanzen beinhalten das *Produktmodell*, bei dem die Branchenstandards KBL [5] bzw. VEC [6] in Dateiform als *Product Specification* an die Komponente angehangen wird. Das bei der IDTA in Entwicklung befindliche Teilmodell „*BillOfProcess*“ [26] wird für die *Produktionsdaten* verwendet.

Im **Entwicklungsprozess** des Leitungssatzes stellt der Komponentenhersteller seine VWS zum Komponenten-Typen dem OEM, den Konfektionär und den Werkzeugherstellern zur Verfügung. Die betroffenen Stakeholder legen sich diese VWS typischerweise in einem internen Repository ab („VWS-Repository fremde Komponenten-Typen“). Die Verbindung mit der originalen VWS erfolgt mit Hilfe der Verlinkung.

Der OEM möchte im Entwicklungsprozess Ergänzungen (z.B. Freigabe von Komponenten für eine spezielle Baureihe) oder Einschränkungen (z.B. ein Kontaktteil wurde für einen bestimmten Querschnittsbereich ausgelegt, darf bei einem OEM jedoch nur für einen geringeren Querschnittsbereich eingesetzt werden) zu der Komponente vornehmen. Dafür legt er eine eigene VWS an und speichert diese in einem Repository mit den eigenen Komponenten-VWS. Durch die *specificAssetID* kann er die VWS des Komponentenherstellers, die bereits in seinem System bekannt ist, mit seiner eigenen VWS verknüpfen und somit die Informationen erweitern.

Im Austausch mit dem Konfektionär gibt der OEM seine VWS *OEM-Komponente* an den Konfektionär weiter. Diese VWS kann der Konfektionär sich in seinem System speichern. Durch die vom OEM bereitgestellte *specificAssetID* kann der Konfektionär die VWS-Komponente vom Komponentenhersteller direkt anfragen. Ähnlich dem OEM kann dann auch der Konfektionär seine VWS für die Komponenten hinzufügen.

Der Werkzeughersteller, der nur an den Komponenteninformationen interessiert ist, ohne diese verändern zu wollen, kann die Informationen immer direkt vom Server des Herstellers abgreifen oder sie sich in seinem eigenen System ablegen. Der beschriebene Aufbau für den Entwicklungsprozess ist in Abbildung 12-11 zu sehen.

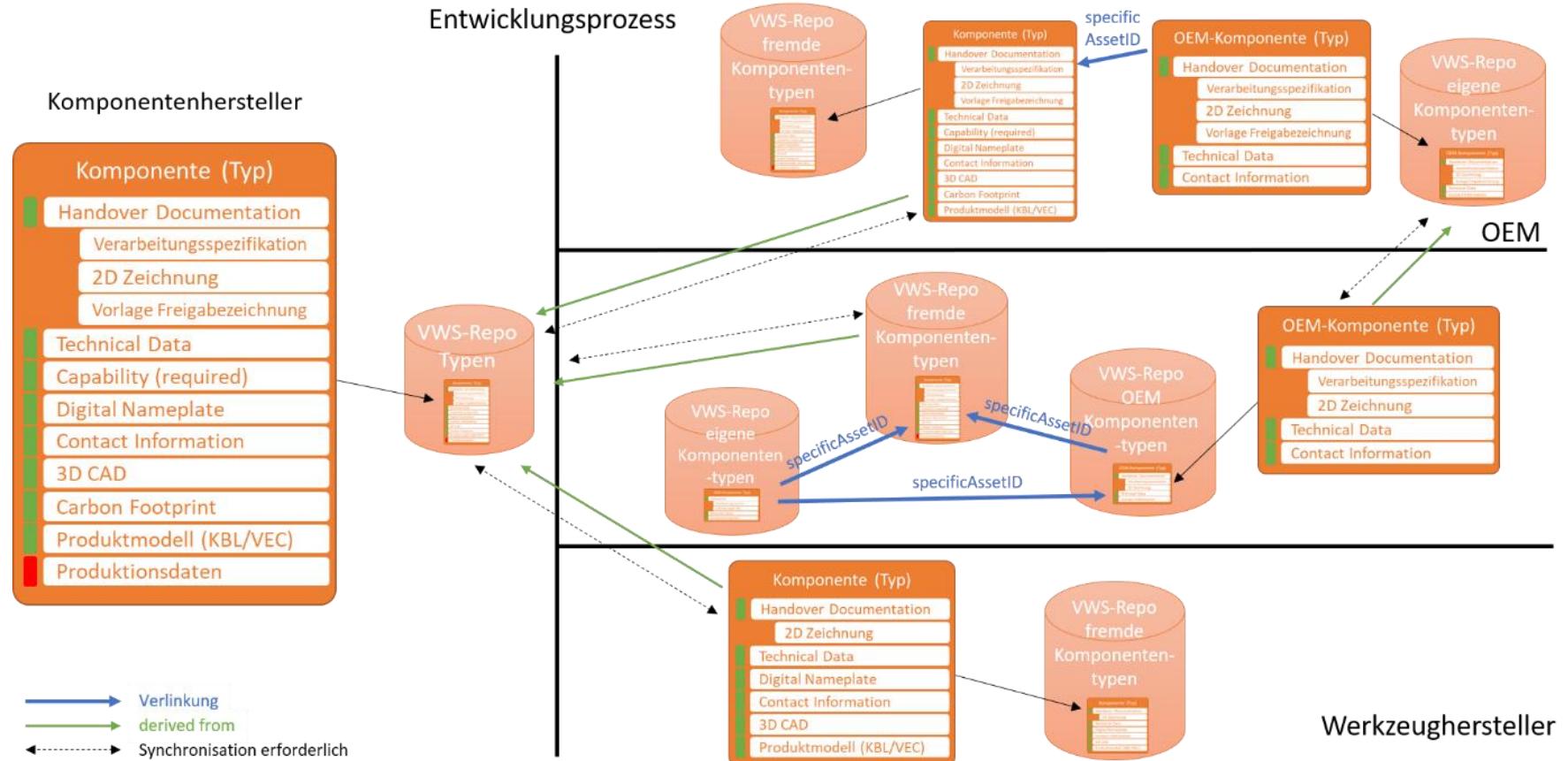


Abbildung 12-11: VWS und deren Verknüpfung für die Komponente im Entwicklungsprozess

Im **Produktionsprozess** des Leitungssatzes sind nur noch der Komponentenhersteller und Konfektionär beteiligt, weshalb sich die Systemarchitektur weniger komplex als im Entwicklungsprozess gestaltet. Jede VWS der Komponenten-Instanzen werden beim Komponentenhersteller in einem dafür vorgesehenen Repository abgelegt. Bei Auslieferung der Komponente, wird eine Kopie dieser VWS an den Konfektionär übergeben und zusätzlich eine Verlinkung zwischen dem Original und der Kopie eingefügt. Eine Synchronisation der kopierten VWS ist hierbei nicht vorgesehen, da Änderungen an den Informationen zu einer bereits produzierten Komponente unwahrscheinlich sind. Prinzipiell ist jedoch eine manuelle Synchronisation möglich. Zu sehen ist der Aufbau in Abbildung 12-12.

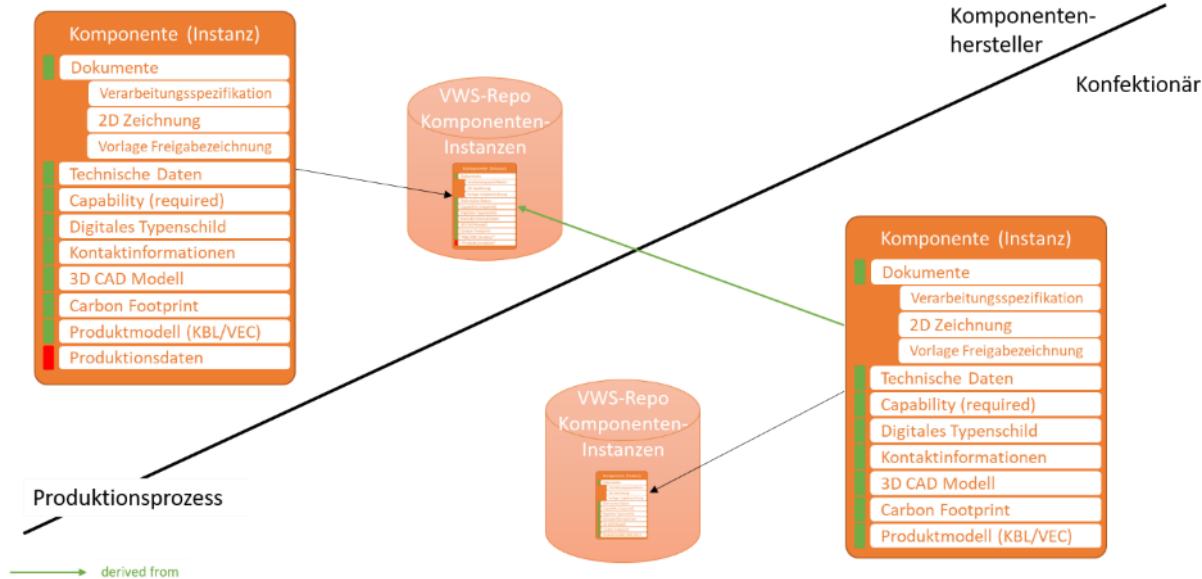


Abbildung 12-12: VWS und deren Verknüpfung für die Komponente im Produktionsprozess

12.2.4.3 VWS für den Leitungssatz

Im Entwicklungsprozess übergibt der OEM dem Konfektionär einen vordefinierten Leitungssatz-Typ, den der Konfektionär zum Produkt entwickeln soll. Die Informationen zu dem Leitungssatz-Typ beinhalten neben den Standardinformationen wie den Dokumentationen und Kontaktdaten auch das selbst entwickelte *Produktmodell* mit der VEC- bzw. KBL-Datei und verschiedene Stücklisten (*Configuration BoM, Product BoM*) zu der Zusammensetzung des Leitungssatzes. Die Stücklisten wurde dabei von dem *Hierarchical Structures Enabling Bill of Material* Teilmodell abgeleitet.

Der OEM stellt über sein eigenes Repository die VWS zu dem Leitungssatz mit verringertem Informationsgehalt dem Konfektionär bereit. Dieser wiederum speichert diese VWS *OEM-Leitungssatz* bei sich im System ab. Danach kann der Konfektionär die Entwicklung des Leitungssatzes, basierend auf der Vorgabe des OEM, beginnen. Dabei entstehen neben der Leitungssatz-VWS auch VWS zu den einzelnen Funktionsmodulen und Halbfabrikaten. Die Verlinkung unterhalb dieser VWS ist in dem Kapitel 12.3 näher beschrieben.

In den VWS, die der Konfektionär erstellt, kommt als Teilmодell die Manufacturing BoM hinzu. Der Zweck dieses Teilmодells ist ebenso in Kapitel 12.3 beschrieben. Zudem wird das BoM-Teilmодell (150%) eingeführt, dass eine Modellierung der notwendigen Prozessschritte zur Herstellung der Leitungssatzinstanzen erlaubt. Das Teilmодell steht eng mit dem Teilmодell *Bill of Process* (ID: 02031-1) in Verbindung. Der beschriebene Aufbau ist in Abbildung 12-13

Abbildung 12-14: VWS und deren Verknüpfung für den Leitungssatz im Produktionsprozess

zu sehen.

Während des Produktionsprozesses, werden die VWS für die Instanzen aus den VWS des Leitungssatzes und seiner Funktionsmodule und Halbfabrikate aus dem Entwicklungsprozess abgeleitet und mit Instanz-Informationen angereichert. Die VWS zu den Instanzen werden in einem separaten Server ge-

speichert, auf den der OEM später Zugriff hat. Außer dem Produktmodell und der dazugehörigen Dokumentation, verbleiben alle weiteren Informationen bei dem Konfektionär allein. Zu sehen ist der Aufbau in Abbildung 12-14.

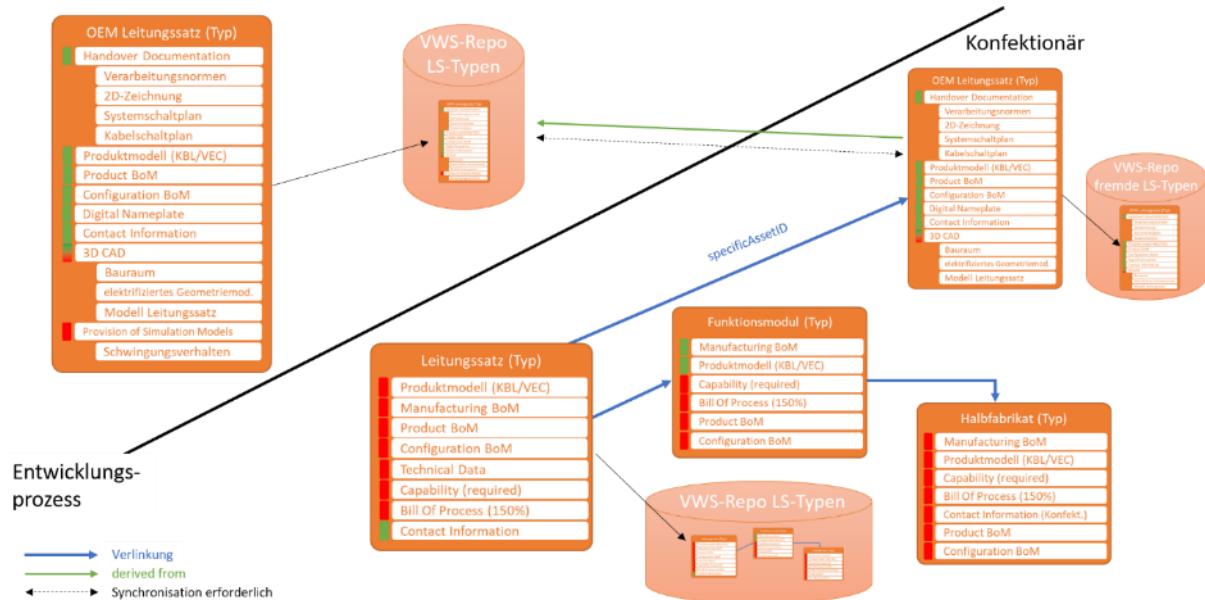


Abbildung 12-13: VWS und deren Verknüpfung für den Leitungssatz im Entwicklungsprozess

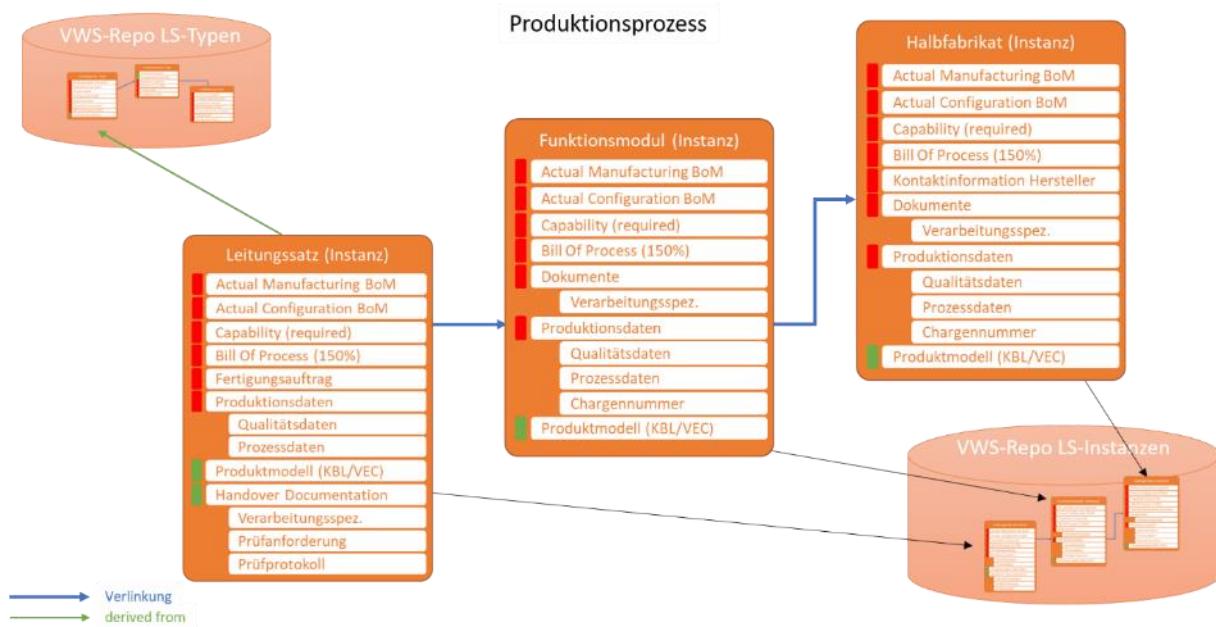


Abbildung 12-14: VWS und deren Verknüpfung für den Leitungssatz im Produktionsprozess

Wird mit Auslieferung des Leitungssatzes der Montageprozess für diesen angestoßen, erhält der OEM eine VWS mit lediglich dem Produktmodell und der Übergabedokumentation (siehe Abbildung 12-15). Diese kann der OEM in seinem eigenen System ablegen. Alle anderen Informationen stehen dem OEM nicht direkt zur Verfügung. Jedoch kann der OEM in Spezialfällen einzelne dieser Informationen beim Konfektionär anfordern.

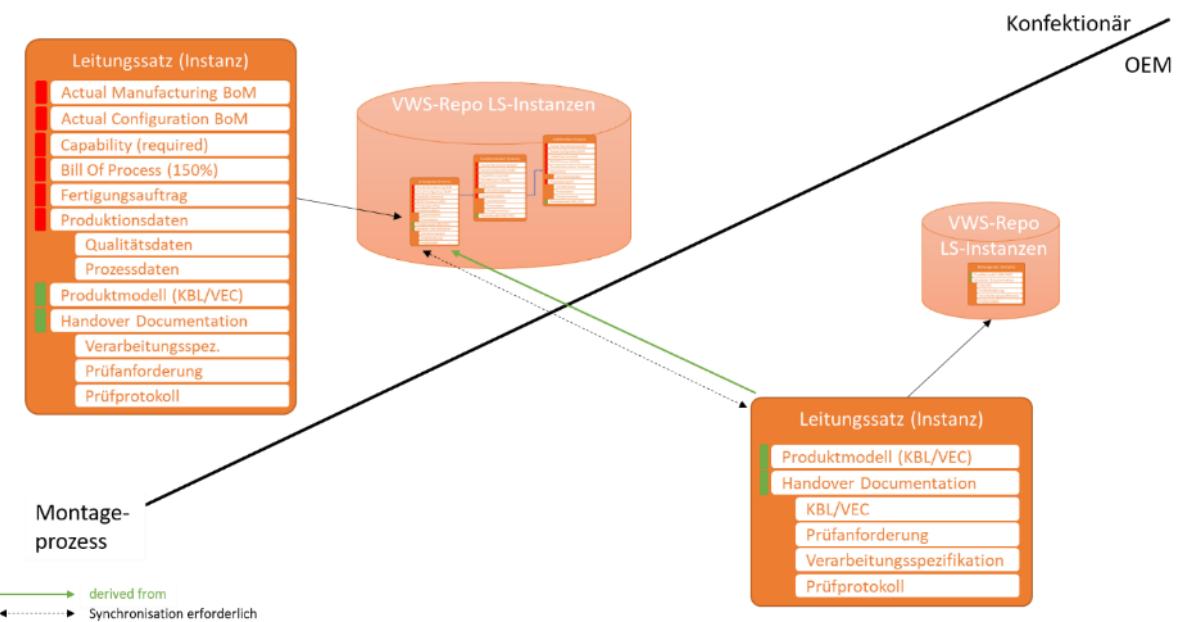


Abbildung 12-15: VWS und deren Verknüpfung für den Leitungssatz im Montageprozess

12.3 Verlinkung

Das Produktmodell beschreibt den Aufbau des Leitungssatzes. Ausgehend von der Produktspezifikation erzeugt der OEM eine Verwaltungsschale für den sog. „150% Leitungssatz“ und verlinkt diesen mit seinen Komponenten (vgl. Kapitel 6).

Der Konfektionär erzeugt eine Kopie dieser Verwaltungsschale und verlinkt diese mit der Verwaltungsschale des OEM sowie den eigenen Verwaltungsschalen der Komponenten („Umschlüsselung“). Auf dieser Grundlage leitet der Konfektionär eine mehrstufige Zerlegung in weitere Halbfabrikate nach Bedarf ab. Die Definition der Halbfabrikate kann ebenfalls varianzbehaftet sein (150% Halbfabrikate).

Im Falle einer Bestellung erhält der Konfektionär vom OEM eine Beauftragung mit genauen Informationen über die gewünschte Ausstattung. Der Konfektionär erzeugt eine Verwaltungsschale für jede Leitungssatzinstanz und verknüpft diese mit der Verwaltungsschale des OEM, um eine durchgehende Rückverfolgbarkeit zu gewährleisten.

Alle Verlinkungen, die hierbei zwischen Konfektionär und OEM entstehen, zeigen vom Konfektionär zum OEM. Damit wird sichergestellt, dass der Konfektionär unabhängig vom OEM auf Informationselemente verlinken kann und somit keinen Schreibzugriff auf die Verwaltungsschale des OEM benötigt.

Das Produktmodell besteht aus den folgenden Teilmustern:

- *ProductSpecification*
- *ProductBoM*
- *ProductConfigurationBoM*
- *ManufacturingBoM*
- *ManufacturingConfigurationBoM*

12.3.1 Typ-Verwaltungsschale des OEM

Die *ProductSpecification*, *ProductConfigurationBoM* und *ProductBoM* werden in der Typ-Verwaltungsschale des Leitungssatzes zusammengefasst (Abbildung 12-16).

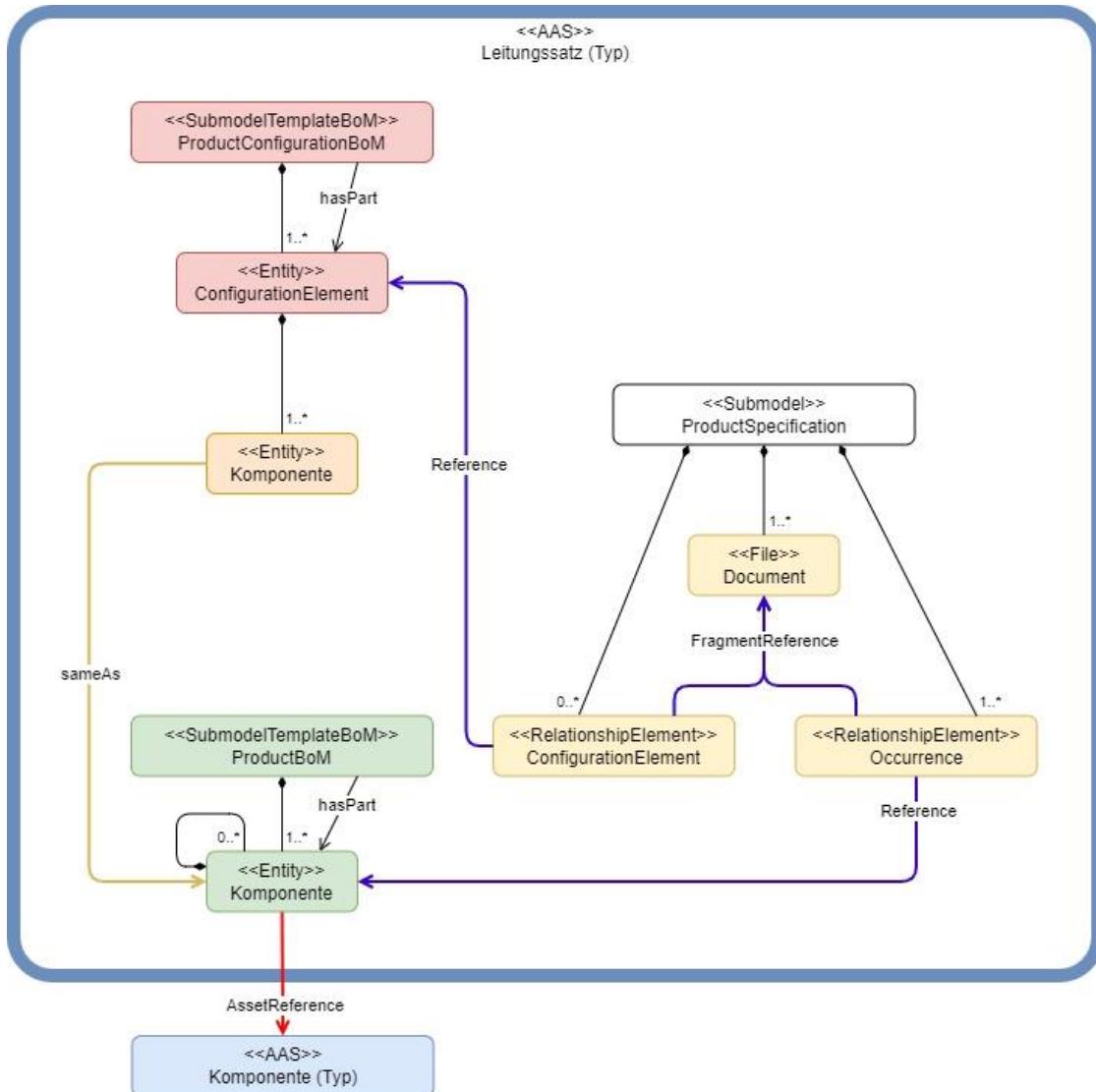


Abbildung 12-16: Typ-VWS des Leitungssatzes aus Sicht des OEM

Diese VWS beschreibt den kompletten Aufbau des 150% Leitungssatzes aus Sicht des OEM. Der Konfektionär erstellt von dieser VWS eine Kopie und reichert diese mit seiner *ManufacturingBoM* an.

12.3.1.1 Teilmodell ProductSpecification

Die *ProductSpecification* beinhaltet eine vollständige Spezifikation des Assets als KBL-, VEC- oder vergleichbares Informationsmodell mit *Relationship-Elementen*, um Beziehungen zwischen dem Informationsmodell und anderen Teilmodellen herzustellen (Abbildung 12-17). Es dient in erster Linie als Bindeglied zwischen Informationsfragmenten des serialisierten Informationsmodells (z.B. XML) und anderen Teilmodellen der Verwaltungsschale. Da für die *ProductSpecification* zum aktuellen Zeitpunkt keine geeignete Teilmodellvorlage der IDTA existiert, wird hilfsweise direkt die KBL- oder VEC-Datei als solche in der VWS mitgeführt. Das könnte zukünftig als Best-Practice-Ansatz in eine IDTA-Submodell-Definition übernommen.

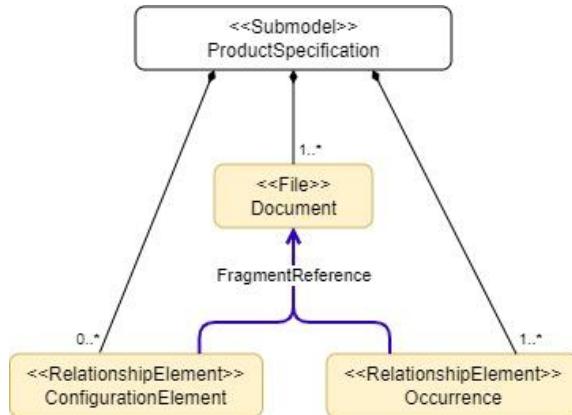


Abbildung 12-17: Teilmodell "ProductSpecification"

12.3.1.2 Teilmodell ProductBoM

Ausgehend von der Produktspezifikation werden die *ProductBoM* und die *ProductConfigurationBoM* abgeleitet.

Die *ProductBoM* beinhaltet alle Komponenten, die in einem Asset verwendet werden. Jedes Vorkommen einer Komponente wird in der *ProductBoM* einzeln aufgeführt, um eine eindeutige Identifikation für jede Komponentenverwendung zu ermöglichen (Abbildung 12-18). Für dieses Teilmodell wird die Teilmodellvorlage „IDTA 02011: Hierarchical Structures enabling Bills of Material“ verwendet. Gemäß dieser Struktur wird für jede Komponentenverwendung eine *Self-Managed Entity* erzeugt und im Teilmodell hinterlegt. Diese verweisen auf die jeweiligen Verwaltungsschalen des Komponententyps.

Die *ProductBoM* kann auch zusammengesetzte Komponenten enthalten. In diesem Fall besteht eine zusammengesetzte Komponente aus einer oder mehreren Teilkomponenten, die jeweils auf die Verwaltungsschalen des Komponententyps verweisen.

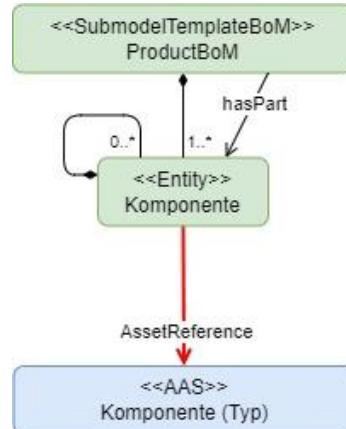


Abbildung 12-18: Teilmodell "ProductBoM"

12.3.1.3 Teilmodell ProductConfigurationBoM

Die *ProductConfigurationBoM* beinhaltet alle Optionen bzw. Varianten, aus denen spätere Asset-Instanzen zusammengesetzt werden können. Hierfür wird die Teilmodellvorlage „IDTA 02011: Hierarchical Structures enabling Bills of Material“ verwendet und für jede Option bzw. Variante eine *Co-Managed Entity* erzeugt, die wiederum aus einer oder mehreren Halbfabrikaten oder Komponenten besteht (Abbildung 12-19).

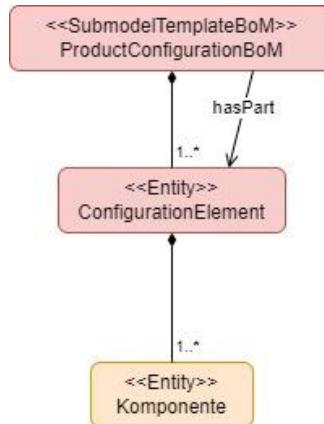


Abbildung 12-19: Teilmodell "ProductConfigurationBoM"

Sowohl die *ProductBoM* als auch die *ProductConfigurationBoM* werden durch die *ProductSpecification* referenziert. Des Weiteren bestehen zwischen den Entitäten der *ProductConfigurationBoM* und der *ProductBoM* *sameAs*-Beziehungen, um die semantische Äquivalenz der Komponenten eindeutig zu definieren.

12.3.2 Typ-Verwaltungsschale (Konfektionär)

Aus den Teilmustern *ProductSpecification*, *ProductBoM*, *ManufacturingBoM* und *ManufacturingConfigurationBoM* ergibt sich ein Gesamtmodell der Typ-Verwaltungsschale des 150% Leitungssatzes aus Sicht des Konfektionärs (Abbildung 12-20).

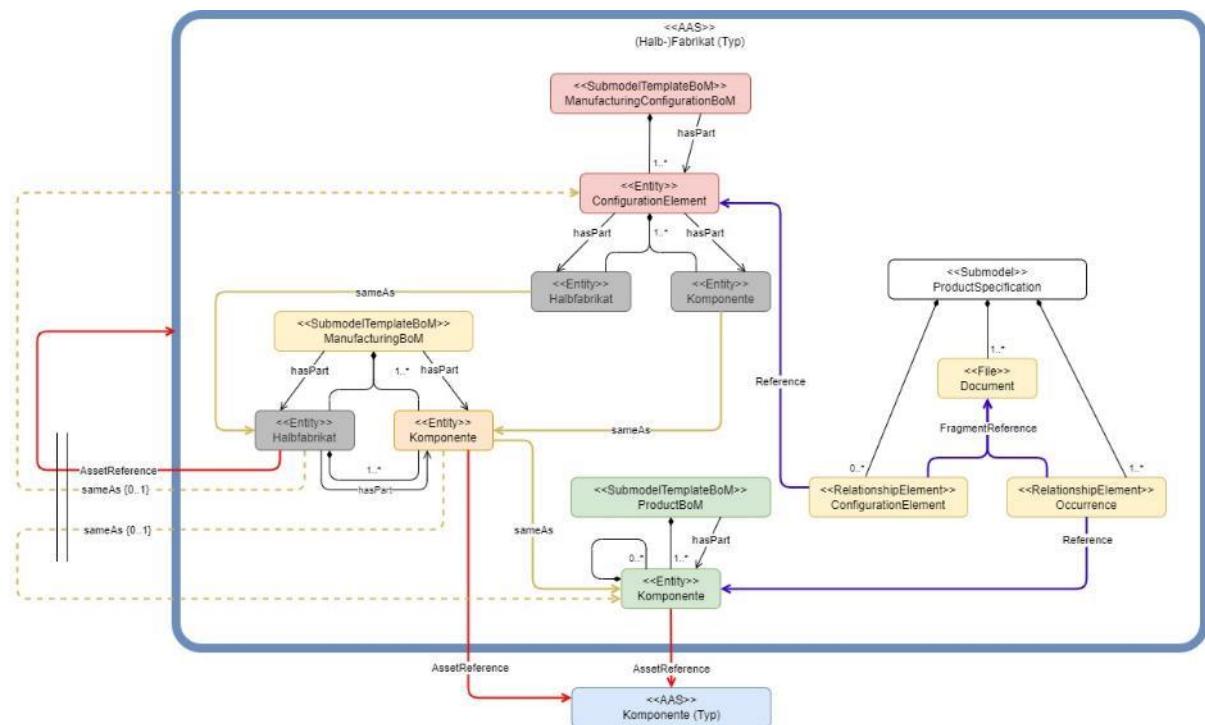


Abbildung 12-20: Typ-VWS des Leitungssatzes aus Sicht des Konfektionärs

Die hier dargestellte Struktur wird rekursiv für alle Halbfabrikate des Konfektionärs angewendet. Bei Halbfabrikaten ohne Varianz kann die *ManufacturingConfigurationBoM* entfallen (die *sameAs*-Beziehung zwischen Halbfabrikat der *ManufacturingBoM* und *ConfigurationElement* der *ManufacturingConfigurationBoM* des nächsten Halbfabrikats ist optional).

Aus Gründen der Wiederverwendbarkeit von Typ-Verwaltungsschalen (beispielsweise eines Halbfabrikats, welches bereits für einen anderen Leitungssatz beschrieben wurde), müssen zwischen den Komponenten der Verwaltungsschalen entsprechende *sameAs*-Beziehungen hergestellt werden (*sameAs*-

Beziehung zwischen Komponente der *ManufacturingBoM* und Komponente der *ProductBoM* des nächsten Halbfabrikats). Für Halbfabrikate der untersten Ebene (nicht weiter aufgeteilt in weitere Halbfabrikate) gilt diese Regel nicht. Deswegen ist die *sameAs*-Beziehung zwischen den Komponenten der *ManufacturingBoM* und *ProductBoM* des nächsten Halbfabrikats optional.

Aufgrund der *sameAs*-Beziehungen kann bei wiederverwendeten Halbfabrikaten von einer Äquivalenz der Produktspezifikation ausgegangen werden, auch wenn sich die konkreten Bezeichnungen (z.B. Komponentenbezeichnungen) unterscheiden.

Um den Zusammenhang mit den Komponenten und Entitäten des OEM darzustellen, hinterlegt der Konfektionär bei seinen Verwaltungsschalen die *specificAssetId* des OEM und verlinkt die Komponenten der *ProductBoM* und die Module der *ConfigurationBoM* mittels *sameAs*-Beziehungen (Abbildung 12-21). Da der Konfektionär zusätzliche Komponenten zur *ProductBoM* hinzufügen kann, ist die *sameAs*-Beziehung zwischen der *ProductBoM* des Konfektionärs und OEM optional.

Sollte vom OEM keine Verwaltungsschale zur Verfügung stehen, hat der Konfektionär grundsätzlich die Möglichkeit auf diese Verknüpfungen zu verzichten und unabhängig zu agieren.

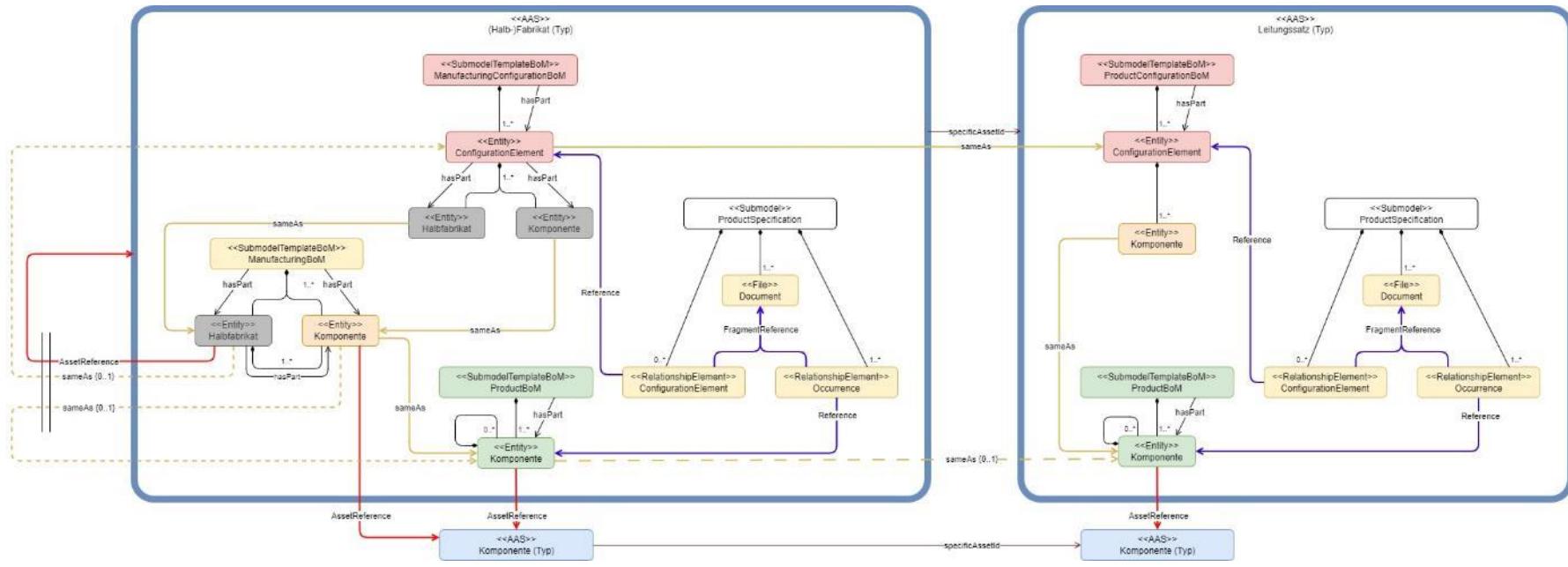


Abbildung 12-21: Typ-VWS des Leitungssatzes aus Sicht des Konfektionärs

12.3.2.1 Teilmodell ManufacturingBoM (Konfektionär)

Die *ManufacturingBoM* beschreibt den Zusammenbau des Leitungssatzes bzw. eines Halbfabrikats aus Komponenten und Halbfabrikaten (Abbildung 12-22). Hierfür wird die Teilmodellvorlage „IDTA 02011: Hierarchical Structures enabling Bills of Material“ [30] verwendet. Für jedes Halbfabrikat und für jede Komponente werden entsprechende *Self-Managed Entities* erzeugt. Diese verweisen auf die VWS der Halbfabrikate bzw. Komponenten.

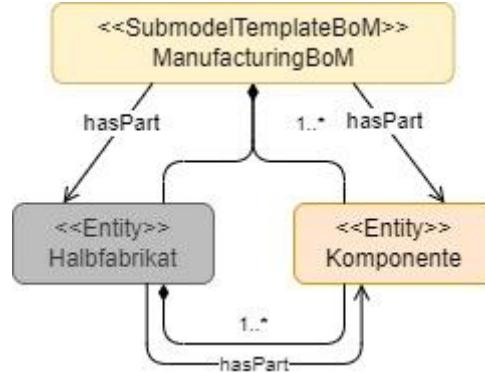


Abbildung 12-22: Teilmodell "ManufacturingBoM"

12.3.2.2 Teilmodell ManufacturingConfigurationBoM (Konfektionär)

Die *ManufacturingBoM* wird mit der *ManufacturingConfigurationBoM* verlinkt, die an Stelle der *Product-ConfigurationBoM* des OEM tritt (Abbildung 12-23). Hierdurch kann von einer konkreten Leitungssatz-Konfiguration auf die notwendigen Halbfabrikate und Komponenten der *ManufacturingBoM* geschlossen werden.

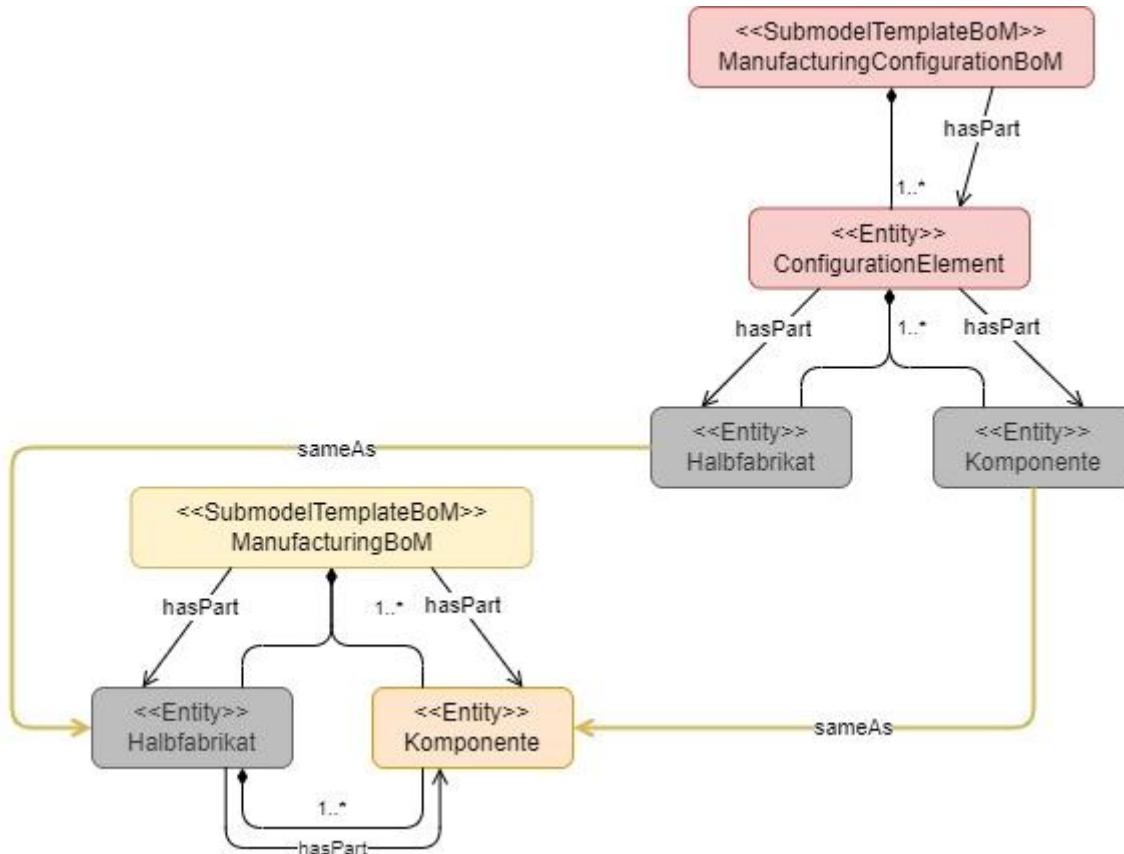


Abbildung 12-23: Beziehungen "ManufacturingConfigurationBoM" und "ManufacturingBoM"

12.3.2.3 Teilmodell ProductBoM (Konfektionär)

Die *ProductBoM* des OEM bleibt weitestgehend unverändert bestehen. Der Konfektionär fügt bei Bedarf zusätzliche Komponenten hinzu und ändert die *AssetReferenzen* auf entsprechende Typ-Verwaltungsschalen seiner eigenen Komponenten, die wiederum mittels *specificAssetId* auf die Typ-Verwaltungsschalen des OEM verweisen. Des Weiteren werden *sameAs*-Beziehungen zwischen den Komponenten der *ManufacturingBoM* und der *ProductBoM* hergestellt, um semantisch klarzustellen, dass es sich hierbei um dieselbe Komponente handelt (Abbildung 12-24).

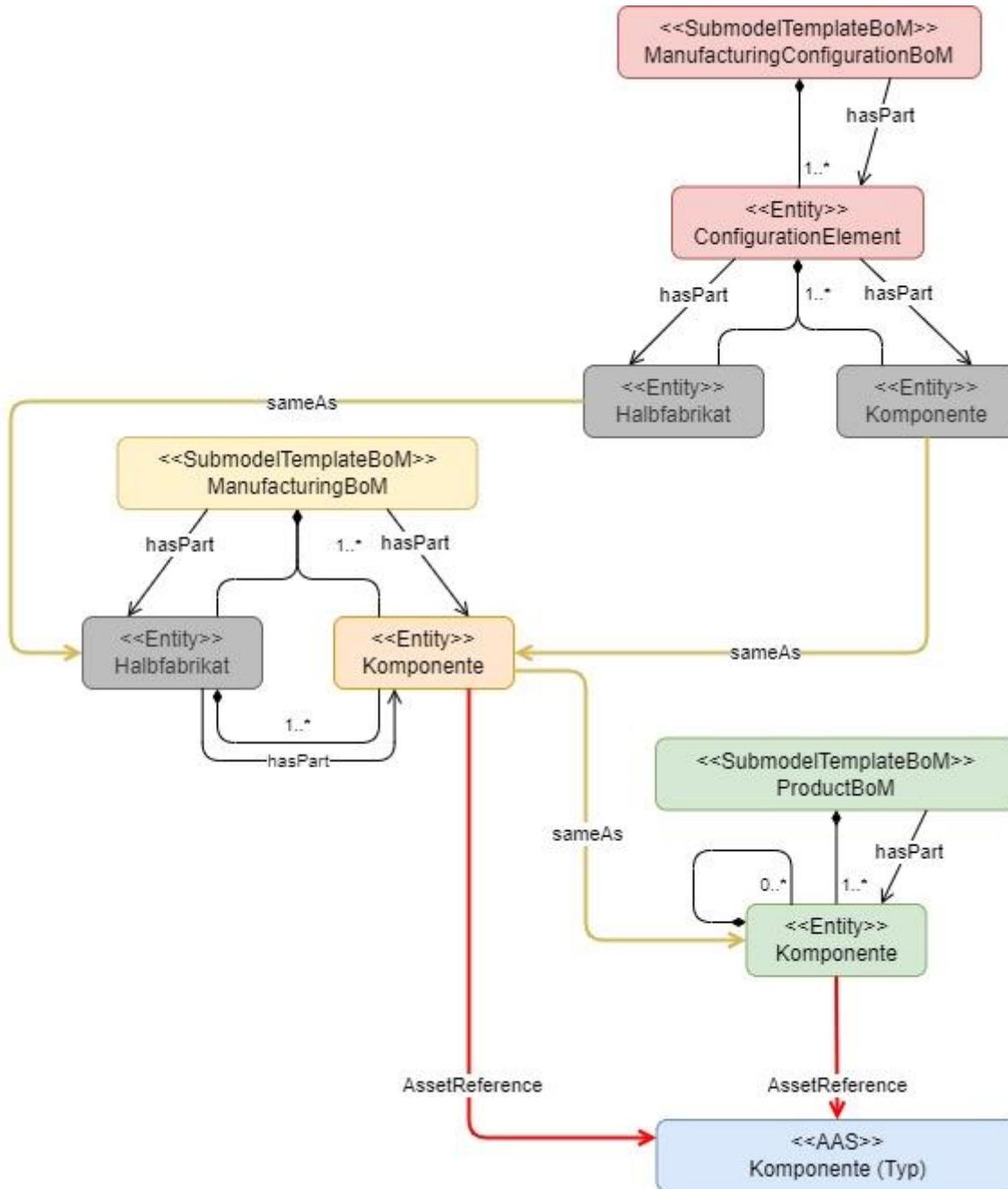


Abbildung 12-24: Verlinkung mit dem Teilmodell "ProductBoM"

12.3.3 Instanz-Verwaltungsschale des OEM

Im Auftragsfall erzeugt der OEM eine Instanz-Verwaltungsschale des Leitungssatzes und hinterlegt eine *ConfigurationBoM* mit der gewünschten Ausstattung. Mittels *sameAs*-Beziehung werden die *Configuration*-Elemente mit den entsprechenden *Configuration*-Elementen der Typ-Verwaltungsschale verlinkt (Abbildung 12-25).

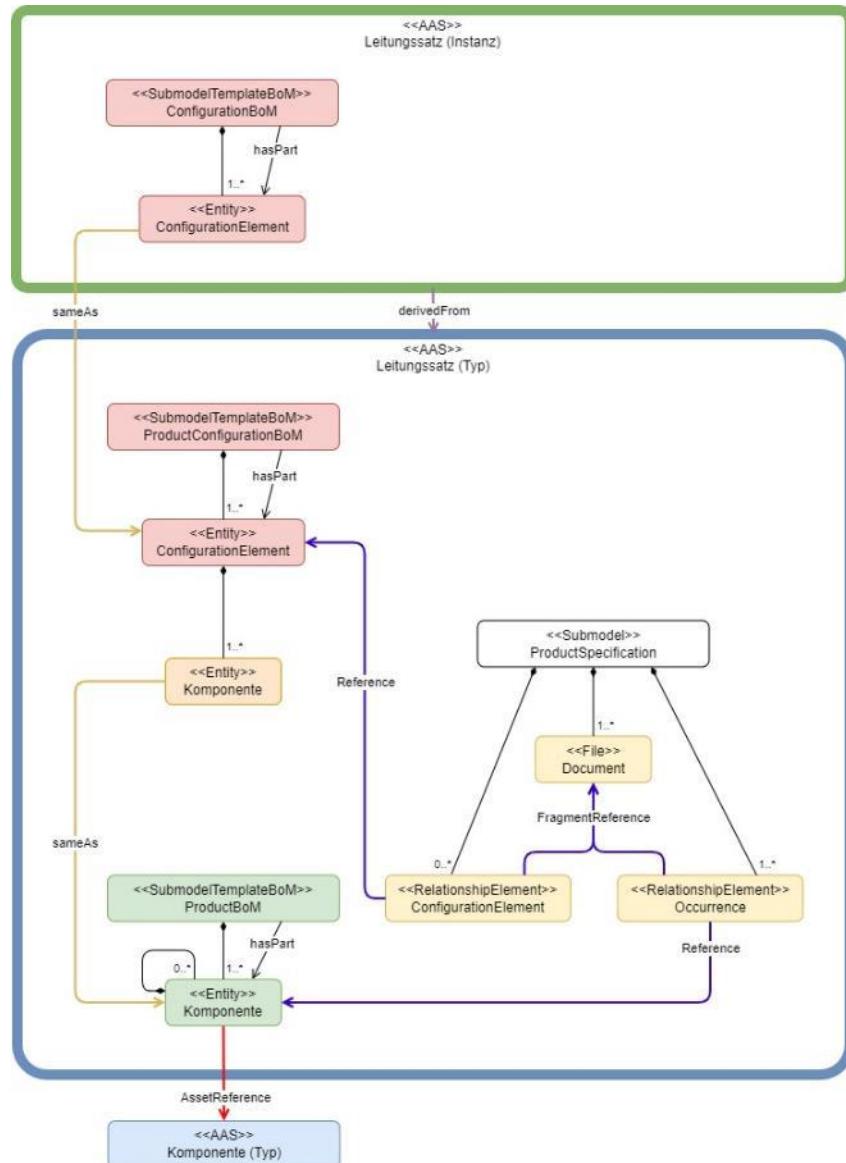


Abbildung 12-25: Verlinkung Typ/Instanz-VWS beim OEM

12.3.4 Instanz-Verwaltungsschale des Konfektionärs

Der Konfektionär erstellt eine Kopie der Instanz-Verwaltungsschale mit Beziehung auf seine eigene Typ-Verwaltungsschale. Zudem wird mittels *specificAssetId* auf die Verwaltungsschale des OEM verwiesen. Die *ConfigurationBoM* der Instanz-Verwaltungsschale wird sowohl mit der *ConfigurationBoM* der Typ-Verwaltungsschale als auch mit der Instanz-Verwaltungsschale des OEM verknüpft. Damit wird sichergestellt, dass der bestellte Umfang vollständig abgebildet ist.

Die Instanz-Verwaltungsschale des Konfektionärs beinhaltet zudem eine *ManufacturingBoM*, welche auf die entsprechenden Instanz-Verwaltungsschalen der Halbfabrikate und Komponenten verweist. Die *ManufacturingBoM* der Instanz-Verwaltungsschale ist wiederum mittels *sameAs*-Beziehungen mit der *ManufacturingBoM* der Typ-Verwaltungsschale verknüpft.

Sollte vom OEM keine Verwaltungsschale zur Verfügung stehen, hat der Konfektionär grundsätzlich die Möglichkeit auf diese Verknüpfungen zu verzichten.

Die Zusammenhänge der verschiedenen Typ- und Instanz-Verwaltungsschalen sind in Abbildung 12-26 dargestellt.

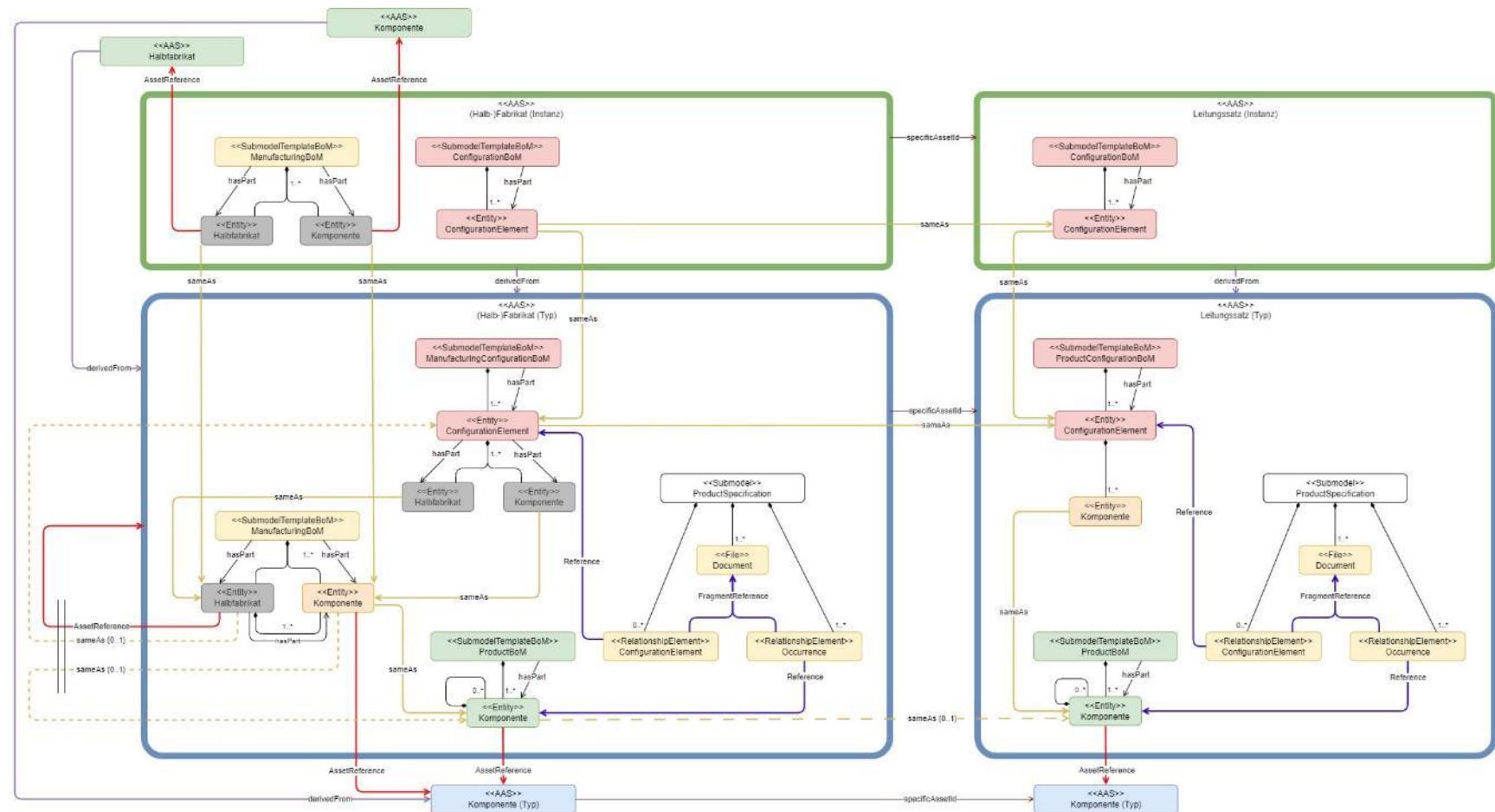


Abbildung 12-26: Typ- und Instanz-VWS für den Leitungssatz

12.3.5 Verlinkung des VEC

Ziel der Verlinkung des VEC ist es, die Typ- und Instanzinformationen mit entsprechenden Assets bzw. Verwaltungsschalen zu verknüpfen, sodass der VEC in einer VWS integriert und die Notwendigkeit zur Interpretation des VEC reduziert werden kann.

Im VEC sind identifizierbare Elemente mandatorisch und im jeweiligen Namensraum einzigartig und persistent. Das bedeutet, dass diese Elemente für eine eindeutige Zuordnung genutzt und auch bei Änderungen beibehalten werden können.

Unterschieden werden muss hierbei zwischen Typ- und Instanzinformationen, nachfolgend am Beispiel von Komponenten (engl. Parts) erläutert:

Bauteile werden im VEC durch Elemente vom Typ *PartVersion*¹¹⁴ identifiziert. Dazu gehört potenziell alles, was in klassischen PDM-Prozessen durch eine Sachnummer identifizierbar ist. Dies können zum Beispiel Komponenten und Leitungssatzmodule sein, aber auch andere Zusammenbauten bzw. Stücklistenteile. Alle Beschreibungen eines Teils (z.B. Stückliste und technische Daten) beziehen sich auf eine *PartVersion*.

Komponentenvorkommen werden im VEC durch Elemente vom Typ *PartOccurrence*¹¹⁵ dargestellt. Hierbei handelt es sich um jeweils ein Vorkommen einer Komponente eines spezifischen Typs (*PartVersion*). Alle Komponenten eines (150%-)Leitungssatzes sind in einer *CompositionSpecification*¹¹⁶ enthalten. Die *PartOccurrences*, die einem Modul enthalten sind, werden durch eine *PartStructureSpecification* definiert. Diese referenziert die *PartVersion* des Moduls als beschriebenes Teil.

Um die Typ- bzw. Instanzinformationen der VEC nutzen zu können, müssen einige Fragestellungen beantwortet werden:

1. Wie können spezifische Informationsfragmente des VEC referenziert werden?
2. Wie können Typeninformationen des VEC mit Assets verknüpft werden?

12.3.5.1 Referenzieren von Informationsfragmenten des VEC

Grundlage für die Verlinkung des VEC ist die Möglichkeit Informationsfragmente zu referenzieren. Da es sich beim VEC um eine XML-Struktur handelt, bietet sich hierfür die Abfragesprache XPath¹¹⁷ an.

Auf Seiten der Verwaltungsschale kann mittels *RelationshipElement* in eine entsprechende Zielfile referenziert werden. Hierfür wird zunächst die VEC als *File*-Element (oder alternativ als *Blob*-Element) in der Verwaltungsschale hinterlegt. Anschließend wird ein *RelationshipElement* verwendet, um als *FragmentReference* mittels XPath in die Datei zu verweisen.

Das *File*- (oder *Blob*-)Element sollte dabei den Medientyp (*mimeType*) *text/xml* besitzen, damit entsprechende Softwarewerkzeuge die Datei korrekt einlesen können.

Anmerkung: In einer der nächsten Versionen des VEC sollen Elemente die Fähigkeiten erhalten global eindeutige IDs zu tragen; voraussichtlich URNs. Ab dann können diese als Referenzierungssanker verwendet werden und die Notwendigkeit zur Bildung hierarchischer XPath Ausdrücke entfällt.

Beispiel

Zunächst muss der XPath auf Grundlage der VEC-Datei abgeleitet werden. Die folgende Struktur soll dabei das exemplarische Vorgehen demonstrieren. In diesem Beispiel könnten die Elemente *DocumentVersion*, *Specification* sowie *Component* prinzipiell mehrmals vorkommen, worauf aus Gründen der Lesbarkeit verzichtet wurde:

¹¹⁴ <https://ecad-wiki.prostep.org/specifications/vec/v202/classes/partversion/>

¹¹⁵ <https://ecad-wiki.prostep.org/specifications/vec/v202/classes/partoccurrence/>

¹¹⁶ <https://ecad-wiki.prostep.org/specifications/vec/v202/classes/compositionspecification/>

¹¹⁷ <https://www.w3.org/TR/xpath-31/>

```

<DocumentVersion>
    <CompanyName>Acme Inc.</CompanyName>
    <DocumentNumber>HARNESS-1</DocumentNumber>
    <DocumentVersion>1</DocumentVersion>
    <Specification>
        <Identification>HARNESS-1</Identification>
        <Component>
            <Identification>A1</Identification>
            <Role>
                <Identification>A1</Identification>
            </Role>
        </Component>
    </Specification>
</DocumentVersion>

```

Um in dieser Struktur auf das *Component*-Element zu verweisen, muss eine eindeutige Identifikation ermöglicht werden. Das Geschieht mittels der jeweiligen *Identification*-Elemente, bzw. der Dokumenteninformationen (*CompanyName*, *DocumentNumber*, *DocumentVersion*). Nachfolgend ist der zugehörige XPath definiert:

```

//DocumentVersion
    [ ./CompanyName='Acme Inc.' ]
    [ ./DocumentNumber='HARNESS-1' ]
    [ ./DocumentVersion='1' ]
/Specification
    [ ./Identification='HARNESS-1' ]
/Component
    [ ./Identification='A1' ]

```

Um in der Verwaltungsschale auf das Element zu verweisen, wird die VEC-Datei zunächst als *File*-Element hinterlegt (Abbildung 12-27). Anschließend wird ein *Entity*-Element erstellt, welches als Ankerpunkt dient und auf ein Asset verweist (Abbildung 12-28). Zuletzt wird mittels eines *Relationship*-Elements eine Verlinkung zwischen dem *Entity*-Element und der VEC-Komponente hergestellt (Abbildung 12-29).

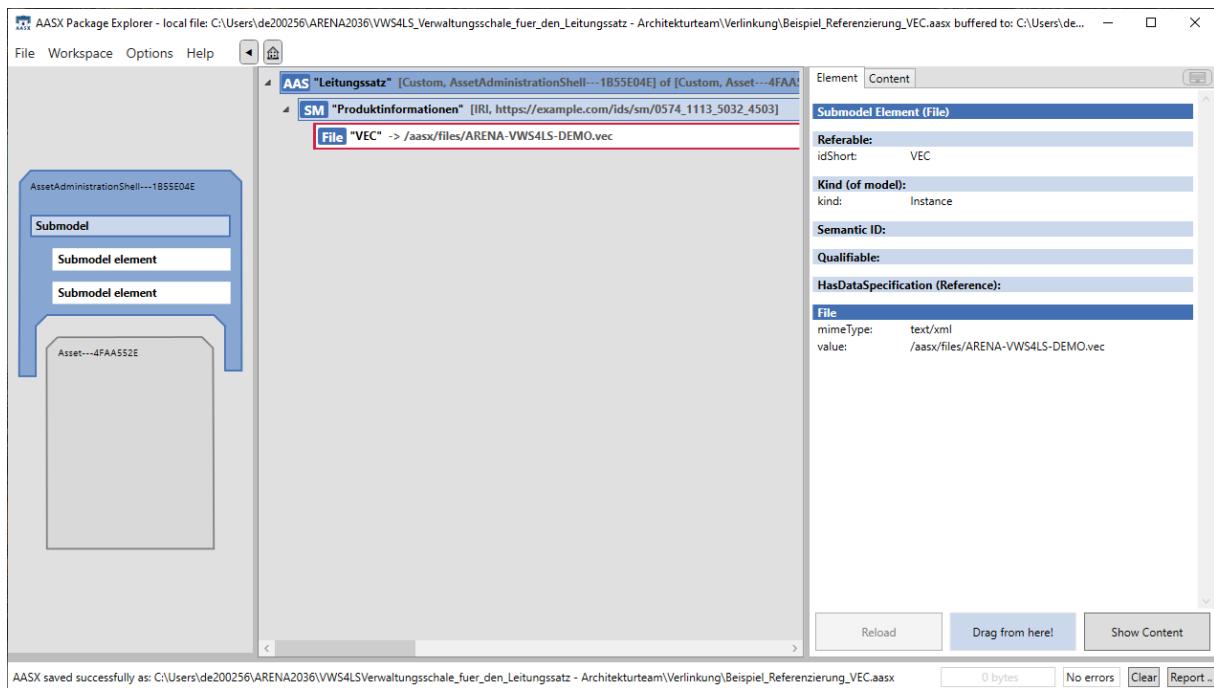


Abbildung 12-27: VEC-Datei als File-Element in der Verwaltungsschale.

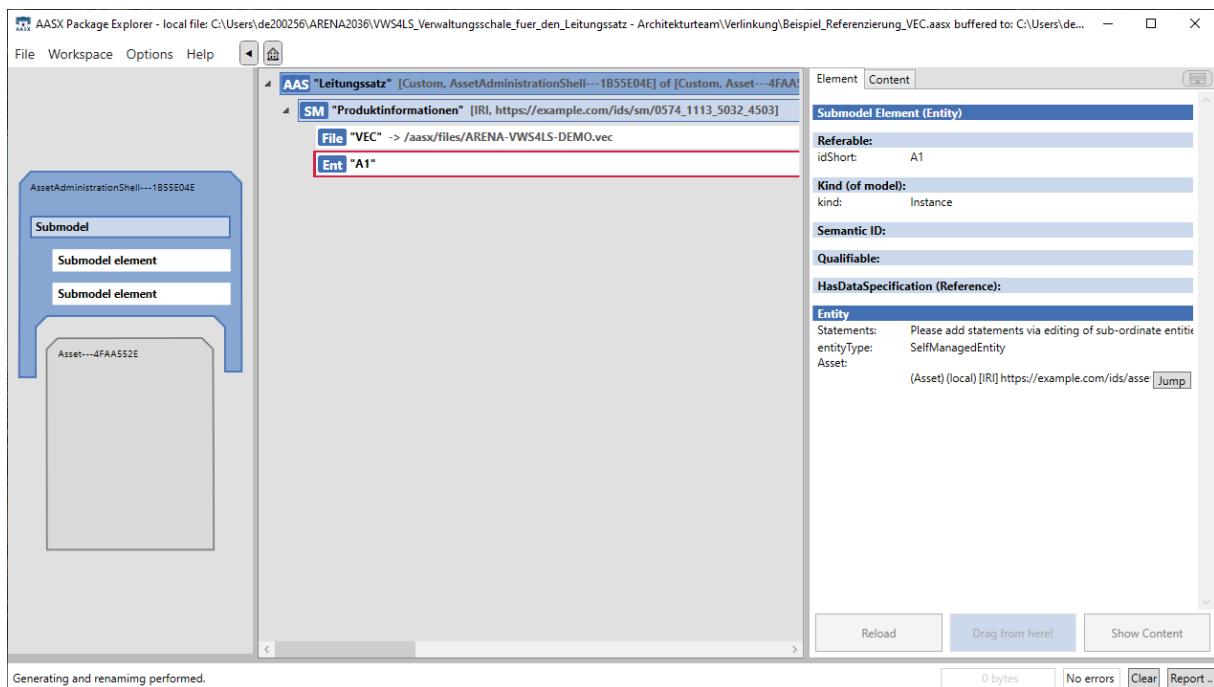


Abbildung 12-28: Entity-Element einer Komponente.

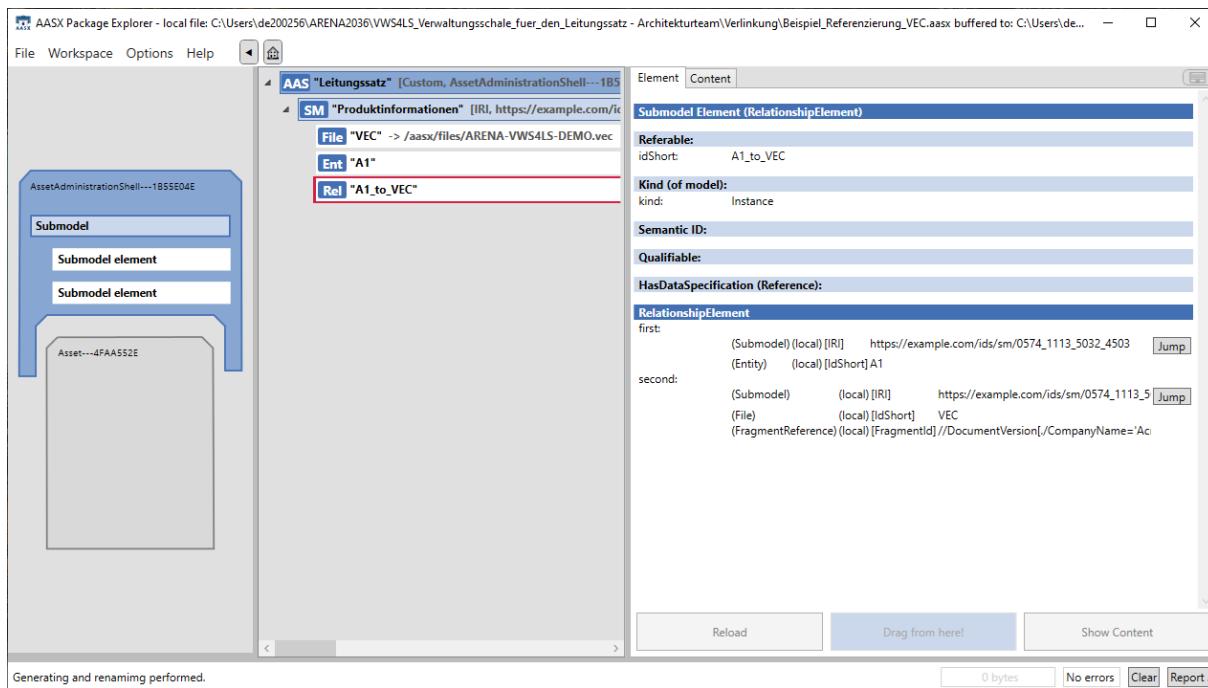


Abbildung 12-29: Relationship-Element mit FragmentReference in VEC-Datei

12.3.5.2 Verlinkung von Typinformationen

Um die Typinformationen des VEC zu verlinken, bestehen grundsätzlich drei Möglichkeiten:

- 1) Ableitung globaler Asset Identifier aus dem VEC

Diese Möglichkeit setzt voraus, dass die globalen Asset IDs mithilfe eines Logikbausteins und definierten Konventionen aus den Komponenteninformationen errechnet werden können. Die hierfür notwendigen Konventionen können dabei, je nach gewünschter Granularität, unternehmens- oder projektspezifisch ausgelegt werden.

Beispiel

`http://<CompanyName>/<PrimaryPartType>/<PartNumber>/<PartVersion>`

- 2) Verwendung des Attributs `aliasId` aus dem VEC

Diese Möglichkeit setzt voraus, dass der Ersteller der VEC für alle Komponenten die zugehörigen Asset Identifier pflegt.

`aliasId` ist ein optionales Attribut der `PartVersion118`-Klasse im VEC . Es dient in erster Linie dazu, einer Komponente zusätzliche Identifizierungsmerkmale mitzugeben.

Für unsere Zwecke kann dieses Attribut verwendet werden, um die global eindeutige Identifikation des Assets zu hinterlegen. Dadurch besteht kein Interpretationsbedarf bei einem Import des VEC in die Verwaltungsschale.

`aliasId` ist vom Typ `AliasIdentification119`, welches mehrere Attribute für die Beschreibung eines Identifikationsmerkmals bietet . Für unsere Zwecke sind zwei Attribute entscheidend:

- `identificationValue`
Dieses Attribut enthält das Identifikationsmerkmal als Zeichenfolge. Für unsere Zwecke muss dieses Attribut die global eindeutige Asset ID beinhalten.
- `type`
Dieses Attribut definiert den Typ des Identifikationsmerkmals. Es handelt sich hierbei um eine Enumeration vom Typ `AliasIdentificationType120` . Da wir grundsätzlich auf die

¹¹⁸ <https://ecad-wiki.prostep.org/specifications/vec/v202/classes/partversion/#Attributes>

¹¹⁹ <https://ecad-wiki.prostep.org/specifications/vec/v202/classes/aliasidentification/>

¹²⁰ <https://ecad-wiki.prostep.org/specifications/vec/v202/classes/aliasidentificationtype/>

global eindeutige Asset ID verweisen wollen, ist dieses Attribute immer mit dem Typ *UUID* (global einzigartiges Identifikationsmerkmal) zu befüllen.

Beispiel

- *identificationValue*
"http://www.example.com/vws4ls/aas/1/0/demo_vec_asset"
- *type*
"UUID"

3) VEC Identifier als spezifische Asset ID in der VWS

Diese Möglichkeit setzt voraus, dass Elemente im VEC eindeutig identifizierbar sind und mindestens das VWS-Metamodel in der Version 3 verwendet wird.

Hierfür wird ein einzigartiges Identifikationsmerkmal sowohl im VEC als auch in der VWS definiert, sodass diese zugeordnet werden können. Dieses Identifikationsmerkmal kann dann auch als Filterkriterium in einer VWS-Registry genutzt werden.

12.5 Versionierung

Ein erster Ansatz zur Versionierung der Verwaltungsschale (VWS) wäre es, eine Versionsnummer zu den jeweiligen Verwaltungsschalen hinzuzufügen (Version/Revision) und immer auf eine spezifische Version zu verweisen. Bei Änderungen müsste dann zunächst ein manueller Eingriff erfolgen, um entlang der Lieferkette die neuen Versionen einzubinden, sofern dies notwendig oder sinnvoll ist. Dieser Ansatz kann durch die Einführung eines standardisierten Versionierungsschemas, wie zum Beispiel der semantischen Versionierung (MAJOR.MINOR.PATCH), weiter verbessert werden. Durch dieses strukturierte Schema lassen sich unterschiedliche Arten von Änderungen klar kennzeichnen: Major-Versionen signalisieren tiefgreifende Änderungen, die die Rückwärtskompatibilität brechen, während Minor-Versionen neue Funktionen hinzufügen, aber weiterhin kompatibel bleiben. Patch-Versionen hingegen beschränken sich auf Fehlerkorrekturen, ohne die Kompatibilität zu beeinflussen. Eine solche Struktur würde nicht nur die Nachvollziehbarkeit von Änderungen erhöhen, sondern auch die Implementierung von Updates in der gesamten Lieferkette erleichtern.

Darüber hinaus könnte dieser Mechanismus automatisiert werden, indem die Versionen anhand der Fähigkeiten (Capabilities/Skills) des Assets abgeglichen werden. Ein solches System könnte automatisch feststellen, ob ein Asset die Anforderungen einer neuen Version erfüllt oder ob ein Update notwendig ist. Dies könnte durch die Integration von Versionierungssystemen in digitale Plattformen unterstützt werden, die die Fähigkeiten der Assets mit den aktuellsten Versionen der Verwaltungsschale synchronisieren. Digitale Zertifikate oder Kompatibilitätsprüfungen könnten sicherstellen, dass nur getestete und kompatible Versionen verwendet werden, wodurch die Zuverlässigkeit und Stabilität des Systems gewährleistet wird.

Zudem könnte bei großen Änderungen ein automatisiertes System eingeführt werden, das Änderungs- und Versionierungsprotokolle führt, um die gesamte Historie einer Verwaltungsschale oder eines Teilmodells zu dokumentieren. Dadurch könnte bei Bedarf auf ältere Versionen zurückgegriffen werden (Version Rollback), falls eine neue Version nicht wie erwartet funktioniert. Ein solches System würde die Flexibilität und Resilienz der Verwaltungsschale deutlich erhöhen und könnte als zusätzlicher Sicherheitsmechanismus im Falle unerwarteter Probleme mit neuen Versionen fungieren.

Idee: Automatische Validierung des Assets anhand gegebener Parameter (Wird u.a. im BaSys4Transfer-Projekt ebenfalls verfolgt.) Siehe auch [75].

12.5.1 VWS-Aufbau für den Leitungssatz

Bei der Umsetzung der Versionierung von Leitungssatz-Spezifikationen ist es wichtig zu berücksichtigen, dass die Entwicklung und die Produktion zeitlich und organisatorisch entkoppelt sind. Änderungen entstehen in der Regel zuerst in der Entwicklung und fließen anschließend in die Produktion. Diese Übermittlung erfolgt üblicherweise in Form von halbjährlichen Releases. Ein zentraler Punkt dabei ist der definierte Freigabezustand zu einem bestimmten Zeitpunkt (z. B. zum Entwicklungsstart). Dieser Fahrzeugfreigabezustand (Fzg-Freigabezustand) bildet die Grundlage für die weitere Entwicklung und wird im Laufe der Zeit durch detaillierte Änderungen weiter verfeinert (siehe Abbildung 12-30).

Anstatt für jede kleine Änderung eine neue Version zu erzeugen, werden die Änderungen über einen definierten Zeitraum gesammelt und dann zu bestimmten Zeitpunkten als Sets freigegeben. Dadurch wird der Aufwand minimiert, indem nicht jede kleine Anpassung sofort zu einer neuen Version führt.

Es ist jedoch auch vorgesehen, dass dringende oder wichtige Änderungen, die nicht dem regulären Release-Zeitplan entsprechen, als Change Sets „out of order“ in die Produktion eingespielt werden können. Diese Änderungen müssen nicht zwingend in der Reihenfolge der Entwicklung umgesetzt werden, sondern können je nach Dringlichkeit priorisiert werden.

Die verschiedenen Änderungen sind dabei potenziell unabhängig voneinander und können sich in ihrer Komplexität stark unterscheiden, was zusätzliche Flexibilität erfordert, um sie effizient in die Produktion zu integrieren. Dieser Ansatz stellt sicher, dass die Produktion jederzeit mit den notwendigen Informationen versorgt ist, während gleichzeitig die Entwicklung weiterhin flexibel und detailliert am Produkt arbeiten kann.

- AEM → Beschreibt eine definierte Änderung zu einem bestimmten Freigabezustand

- Change-Set → Besteht aus mehreren AEMs und Bündelt diese zu einem Set

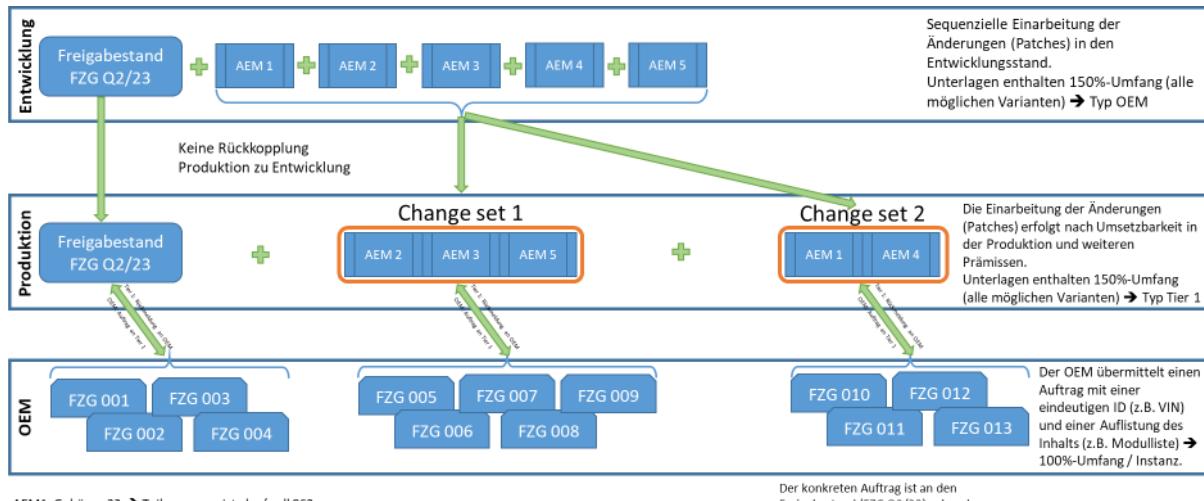


Abbildung 12-30: Entwicklungsversionen und korrespondierende Change-Sets in der Produktion

Da die jeweils eingespielten Change Sets nicht direkt aufeinander aufbauen, ist die gängige semantische Versionierung mit aufsteigender Nummerierung (wie etwa 1.1.4 → 1.2.0 → 1.2.1 → 2.0.0) nicht anwendbar. In einem ersten Konzept sehen wir daher vor die „Versionsnummer“ als Textbaustein basierend auf den Change Sets der Releases zu gestalten, etwa durch Verkettung der Bezeichner/Versionsnummern der einzelnen im Release enthaltenen Änderungen. Diese Versionsnummer ist dann auch Teil der ID der entsprechenden Verwaltungsschale (siehe Abbildung 12-31). Abbildung 12-31 zeigt dabei beispielhaft drei Teilmodelle (CAD-Modell, Fähigkeiten, Prozesse). Vorteil dieses Aufbaus ist die menschenlesbare Dokumentation, die direkt von Experten der Branche verstanden werden kann.

12.5.2 Versionierungskonzept der Verwaltungsschale

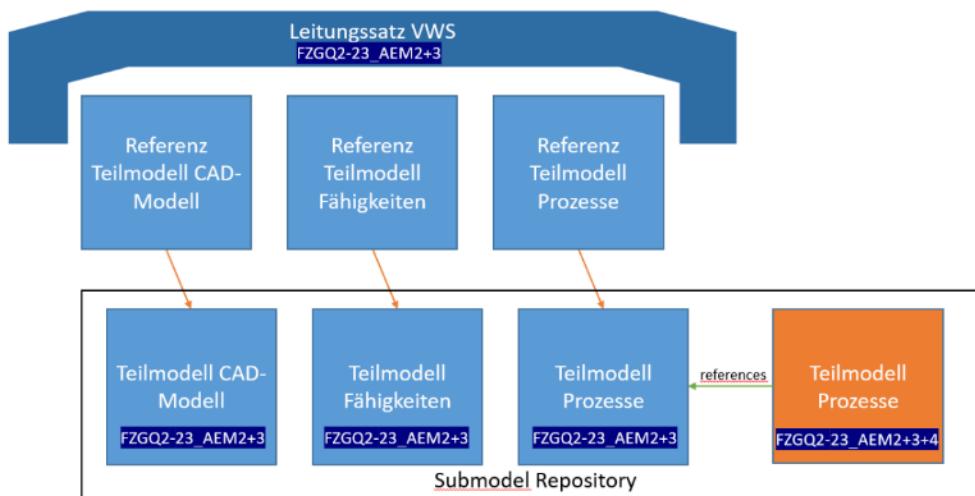


Abbildung 12-31: Vorgeschlagenes Versions-ID-Konzept

Abbildung 12-31 zeigt auch, dass die Teilmodelle der Change Sets nicht direkt der zugehörigen VWS zugeordnet sind, sondern lediglich indirekt über Referenz-Teilmodelle. Dadurch können unveränderte Teilmodelle weiter referenziert werden in neuen Versionen des Teilmodells. Dies passt ebenfalls zur Struktur der Verwaltungsschale in Version 3, in der Teilmodelle als unabhängige Elemente eigenständig, etwa in einem Submodel Repository, existieren können. Somit ist es möglich ein Typ-Submodel, was in mehreren Verwaltungsschalen exakt gleich ist, zu referenzieren und bei Änderungen direkt alle Abhängigkeiten automatisch mitzupflegen.

Das orange markierte Prozesse-Teilmodell in der Abbildung stellt eine neue Version des alten Teilmodells (links) dar, auf das die zusätzliche Änderung 4 angewandt wurde, und dass eine Referenz auf die Vorgängerversion enthält. Auf diese Weise lassen sich alle übernommenen Änderungen bis zum ursprünglichen Referenz-Teilmodell zurückverfolgen.

Die Versionsnummer der Verwaltungsschalen und Teilmodelle sind dabei abhängig von den eingepflegten Änderungspaketen. Im Beispiel in Abbildung 12-31 wurden durch die Änderungspakete 2 und 3 alle Teilmodelle angepasst, weshalb diese als „FZGQ2-23_AEM2+3“ versioniert sind. Durch Änderungspaket 4 wurde jedoch lediglich das Teilmodell Prozesse verändert, weshalb dazu eine neue Version erstellt, und die Versionsnummer verändert wird. Keine weiteren Teilmodelle werden in ihrer Version daher die Referenz zu Änderungspaket 4 erhalten. Änderungen für etwa Fähigkeiten müssten somit im nächsten Paket (Nummer 5) enthalten sein, sodass eine inkonsistente Versionsbezeichnung (etwa durch Doppelnutzung) nicht entstehen kann. Abbildung 12-32 zeigt Beispiele für entsprechend versierte Verwaltungsschalen.

Dateiname: Versionierung_Neu

| Element | Content |
|---|--|
| AssetAdministrationShell (according IEC63278) | |
| Referable: | |
| idShort: | Leitungssatz_OEM |
| HasExtension: | |
| Identifiable: | |
| id: | AssetAdministrationShell--72A03611 |
| id (Base64): | QXNzZXRBZG1pbmldHJhdGlvbNoZWxsLS0nzbMDM2MTE= |
| administration: | |
| version: | FZGQ2-23_AEM1_AEM2_AEM3 |
| revision: | |
| HasDataSpecification (Reference): | |
| Derived From | (AssetAdministrationShell)AssetAdministrationShell--72A03611 |
| AssetInformation | |
| Kind (of AssetInformation): | |
| kind: | Instance |
| globalAssetId: | |
| globalAssetId: | Asset--03336C9E |
| assetType: | |
| specificAssetId: | |
| DefaultThumbnail: | Resource element |
| AAS "Leitungssatz_OEM" VFZGQ2-23. [AssetAdministrationShell---59451790] of [Asset---03336C9E, Instance] | |
| Asset AssetInformation | Asset---03336C9E |
| SM "VEC" VFZGQ2-23. | [https://example.com/ids/sm/7303_7021_7032_7370] |
| AAS "Leitungssatz_OEM" VFZGQ2-23_AEM1. [AssetAdministrationShell---29BE5257] of [Asset---03336C9E, Instance] | |
| Asset AssetInformation | Asset---03336C9E |
| SM "VEC" VFZGQ2-23_AEM1. | [https://example.com/ids/sm/0143_7021_7032_5515] |
| AAS "Leitungssatz_OEM" VFZGQ2-23_AEM1_AEM2. [AssetAdministrationShell---1453EB07] of [Asset---03336C9E, Instance] | |
| Asset AssetInformation | Asset---03336C9E |
| SM "VEC" VFZGQ2-23_AEM1_AEM2. | [https://example.com/ids/sm/6353_7021_7032_8355] |
| AAS "Leitungssatz_OEM" VFZGQ2-23_AEM1_AEM2_AEM3. [AssetAdministrationShell---72A03611] of [Asset---03336C9E, Instance] | |
| Asset AssetInformation | Asset---03336C9E |
| SM "VEC" VFZGQ2-23_AEM1_AEM2_AEM3. | [https://example.com/ids/sm/5363_7021_7032_7097] |
| AAS "Leitungssatz_Tier1" VFZGQ2-23. [AssetAdministrationShell---026F9C85] of [Asset---47F6D15C, Instance] | |
| Asset AssetInformation | Asset---47F6D15C |
| SM "VEC" VFZGQ2-23. | [https://example.com/ids/sm/4244_7021_7032_3296] |
| SM "Capabilities" VFZGQ2-23. | [https://example.com/ids/sm/9044_7021_7032_5045] |
| AAS "Leitungssatz_Tier1" VFZGQ2-23_AEM2_AEM3. [AssetAdministrationShell---17DA687C] of [NotApplicable] | |
| Asset AssetInformation | |
| SM "VEC" VFZGQ2-23_AEM2_AEM3. | [https://example.com/ids/sm/0224_7021_7032_4860] |
| SM "Capabilities" VFZGQ2-23. | [https://example.com/ids/sm/9044_7021_7032_5045] |
| AAS "Leitungssatz_Tier1" VFZGQ2-23_AEM1_AEM2_AEM3. [AssetAdministrationShell---5A44B885] of [NotApplicable] | |
| Asset AssetInformation | |
| SM "VEC" VFZGQ2-23_AEM1_AEM2_AEM3. | [https://example.com/ids/sm/5363_7021_7032_7097] |
| AAS "Leitungssatz_Tier1_Instanz_0815". [AssetAdministrationShell---0451F76C] of [Asset---42EC4085, Instance] | |
| Asset AssetInformation | Asset---42EC4085 |
| AAS "Leitungssatz_Tier1_Instanz_0816". [AssetAdministrationShell---66A19FDC] of [Asset---3D868F57, Instance] | |
| Asset AssetInformation | Asset---3D868F57 |

Abbildung 12-32: Beispiel versionierter Verwaltungsschalen (Teilmodell)

Für den standardisierten Zugriff sehen wir eine Erweiterung des Discovery/Registry Service vor, der als Default immer die aktuelle Version einer gegebenen Verwaltungsschale bereithält. Somit ist kein Bruch zur Konvention der Verwaltungsschalendefinition vorhanden. Dennoch könnte eine Erweiterung der API, um ebenfalls auf die älteren Versionen zugreifen zu können, in Betracht gezogen. Dazu wird im ersten Entwurf ein URL-Parameter erprobt, welcher optional ist und bei nicht-verwendung das standardmäßige Verhalten des entsprechenden Services laut VWS-Definition bereithält.

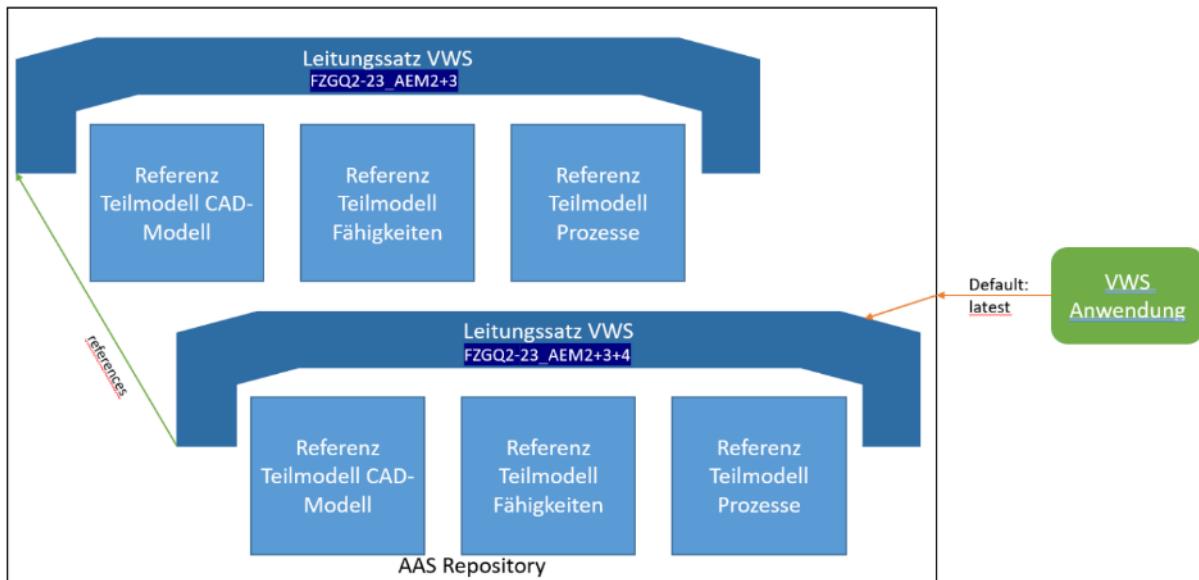


Abbildung 12-33: "Latest" default

Technisch wird Git¹²¹ der auf einem davon abgeleiteten Mechanismus aufbauen, so dass Änderungen letzten Endes wie Patches behandelt werden und ein „Git-Branching“ resultieren.

Bezüglich des konkreten Zugriffs auf eine versionierte VWS eines Leitungssatzes wurden drei mögliche Varianten identifiziert (das Asset ist hierbei jeweils der zu bauende Leitungssatz):

12.5.3 Zugriff mit Registry und Discovery

Hier kennt die Anwendung die Asset-ID und fragt damit beim Discovery Service ab, welche VWS zu der Asset-ID assoziiert sind. Hierzu fragt die Discovery den Versionierungsservice an mittels eines *GET-Request* (Abbildung 12-34), welcher die *VWS-ID* mit Versionsnummer zurückgibt, die dann an die Anwendung weitergeleitet wird. Anschließend fragt die Anwendung das *VWS-Registry* nach dem Endpunkt der VWS mit der entsprechenden *VWS-ID*. Die *VWS-Registry* gibt der Anwendung einen *VWS-Dekskriptor* zurück. Danach holt die Anwendung mit diesem Dekskriptor die VWS aus dem *VWS-Repository*.

Bei dieser Lösung muss ein Discovery- und ein Versionierungsservice entwickelt werden. Ein Discovery Service ist durch Version 3 der VWS allgemein vorgesehen und zuständig für das Auffinden von VWS zu einer Asset-ID bzw. das bidirektionale Mapping zwischen Asset und VWS.

¹²¹ <https://git-scm.com/>

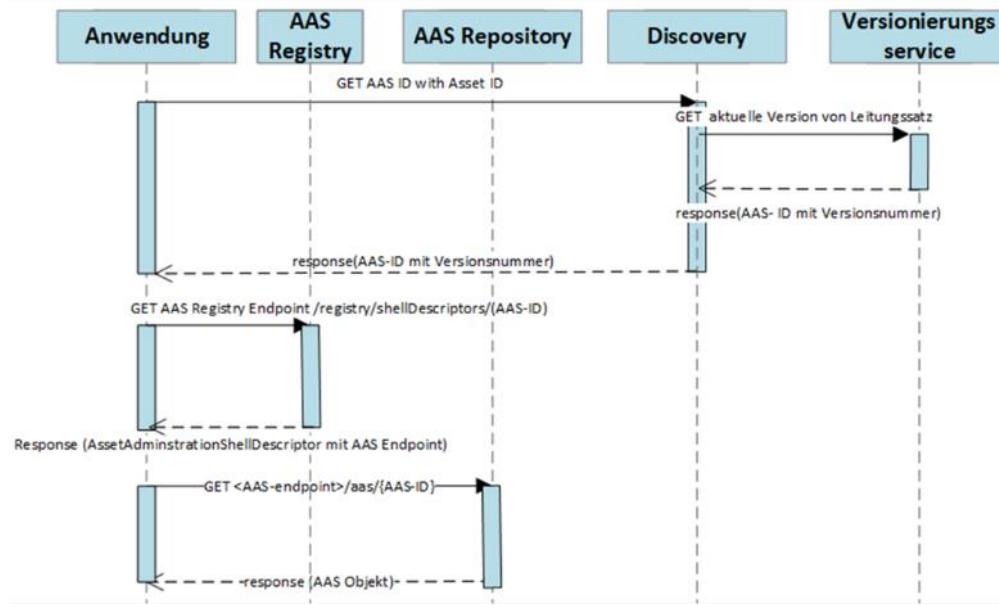


Abbildung 12-34: Zugriff mit Discovery und Registry

12.5.4 Zugriff ohne Discovery

In diesem Fall sendet die Anwendung ein *GET-Request* zu der Registry mit der Asset-ID (Abbildung 12-35). Die Registry leitet die Anfrage weiter an den Versionierungsservice und fragt nach der aktuellen Version des Leitungssatzes. Der Versionierungsservice gibt der Registry die VWS-ID mit Versionsnummer zurück. Die Registry macht einen *GET-Aufruf*, um den Endpunkt der VWS zu ermitteln. Danach gibt die Registry der Anwendung den VWS-Deskriptor zurück, welcher den VWS-Endpoint beinhaltet. Die Anwendung macht anschließend einen *GET-Aufruf* an das VWS-Repository, um das VWS-Objekt zu erhalten.

Bei dieser Lösung übernimmt die Registry die Discovery-Funktion, und hat direkte Interaktion mit dem Versionierungsservice. Für diese Variante muss eine VWS-Registry ebenfalls das Auflösen der Versionsnummer übernehmen und entsprechend erweitert werden.

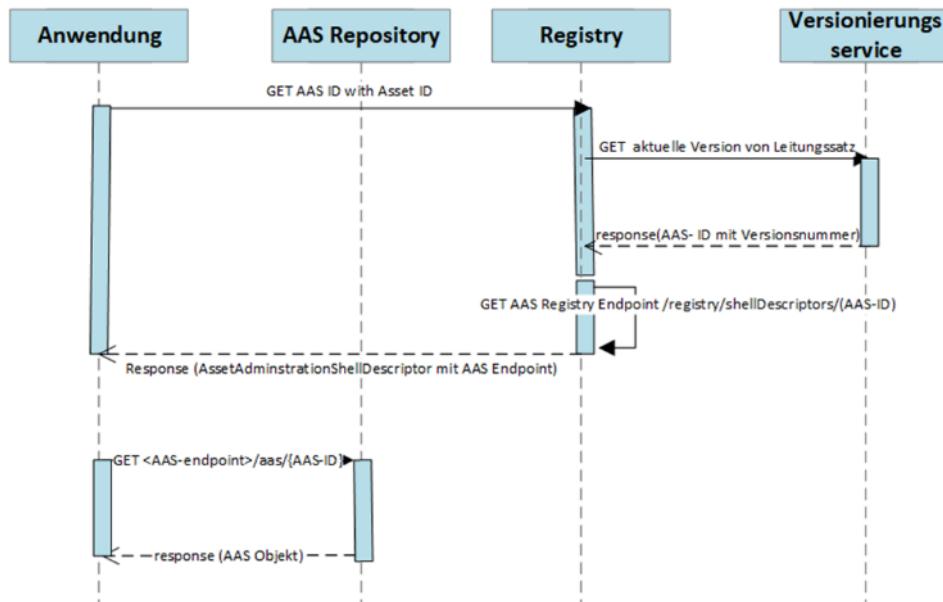


Abbildung 12-35: Zugriff ohne Discovery

12.5.5 Zugriff ohne Registry

In dieser Variante fragt die Anwendung den Discovery Service mit der Asset-ID nach der VWS-ID (Abbildung 12-36). Die Discovery gibt der Anwendung eine Teil-VWS-ID zurück. Mit der Teil-VWS-ID macht

die Anwendung einen *GET-Aufruf* beim Versionierungsservice, um die aktuelle VWS-Version zu bekommen, und baut die vollständige spezifische VWS-ID mit der Versionsnummer zusammen, bevor die Anfrage an das VWS Repository gesendet wird, sodass eine VWS statt aller vernetzter Verwaltungsschalen gefunden werden kann. Das *VWS-Repository* gibt der Anwendung somit das spezifische VWS-Objekt zurück.

Bei dieser Lösung muss die Anwendung die Logik zu Auflösung der vollständigen ID kennen, um die relevante VWS zu finden. Dies hat den Nachteil, dass die Entwickler von Anwendungen Kenntnisse über diesen Mechanismus haben müssen, um die relevante VWS zu finden.

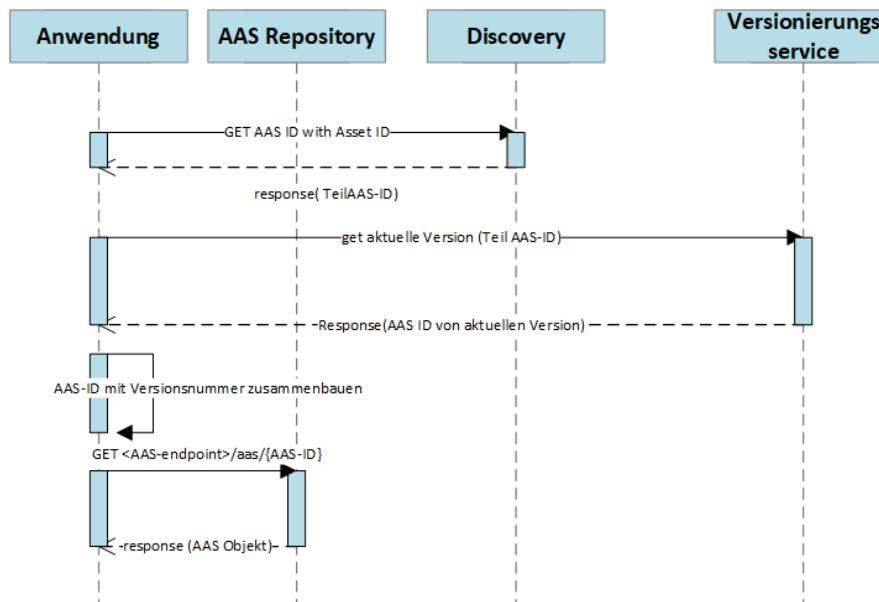


Abbildung 12-36: Zugriff ohne Registry

12.6 Synchronisation

Synchronisation zwischen Asset und seiner Verwaltungsschale

Mit der Definition der Verwaltungsschale als Digitaler Zwilling kommt direkt die Anforderung an die Äquivalenz der Daten, die im bzw. zum Asset vorliegen, mit den Daten, die durch die Verwaltungsschale repräsentiert werden. Dabei ist es unerheblich, ob die Daten z.B. durch einen Sensor erfasst und in einer Speicherprogrammierbare Steuerung (SPS) des Assets vorliegen oder ob die Daten durch eine Person in einer Datenbank außerhalb der Systemgrenzen des Assets erfasst wurden. In beiden Fällen sollen die Daten widerspruchsfrei sein.

Man kann hier zwischen zwei Ansätzen unterscheiden. Zum einen können die Daten zusätzlich in den Teilmustern der Verwaltungsschale abgelegt werden, die in einem Submodel Repository gehostet werden. Das Submodel Repository agiert in diesem Fall als Datenspeicher und bei einer Änderung ist auch das Datum im Repository zu aktualisieren. Man kann in diesem Fall von einem Push-Vorgang in Richtung Teilmuster der Verwaltungsschale sprechen. Im zweiten Fall wird bei dem Zugriff auf die Verwaltungsschale der aktuelle Wert aus der SPS oder dem Drittsystem geladen und weitergereicht. In diesem Fall agiert das Submodel Repository als Proxy und es handelt sich nicht um einen Synchronisationsvorgang im engen Sinn, da das Datum nur einmal gespeichert ist. Man kann von einem Pull-Vorgang durch die Implementierung der Verwaltungsschale sprechen.

Für beide Vorgehensweise gibt es bereits Konzepte und auch praktische Umsetzungen:

Als Beispiel für das Push-Prinzip sei die Data Bridge-Komponente des Eclipse BaSyxTM¹²² Framework genannt, welche die Verwaltungsschalenmiddleware in beide Richtungen mit weiteren Systemen verbinden kann.

Für die Umsetzung eines Pull-Prinzip existiert in Eclipse BaSyxTM das Property Delegation Feature.

Beide genannten Lösungen wurden im Projekt VWS4LS untersucht und werden als valide, funktionierende Lösung eingeschätzt. Die Synchronisation zwischen Asset und Verwaltungsschale wird damit als Stand der Technik betrachtet und der Schwerpunkt wurde im Projekt auf die Synchronisation zwischen Verwaltungsschalen gelegt.

Synchronisation zwischen Verwaltungsschalen

Die Ergebnisse aus TP 2 und der Architekturgruppe „Verlinkung“ zeigen, dass Verwaltungsschalen (im Folgenden als „abhängige VWS“) auf Basis anderer VWS (im Folgenden „übergeordnete VWS“) definiert werden. Dafür gibt es folgende Fallbetrachtungen:

Szenario 1: Unterschiedliche „Wahrheiten“ zum selben Asset

Zum selben Asset-Typ gibt es für die Rollen im Prozess jeweils eigene Typ-Verwaltungsschalen. Diese können einerseits unterschiedliche Detailstufen darstellen (vgl. Typ Leitungssatzkomponente OEM / Tier1), aber auch voneinander abweichende Angaben definieren (vgl. Freigabebereiche für Komponenten).

Szenario 2: Ableitungen

Basierend auf einer VWS werden andere abgeleitete VWS erzeugt, z.B. Verwendung von Komponenten im Leitungssatz oder Ableitung von Halbfabrikaten aus der Leitungssatz-VWS. Bei beiden genannten Ableitungen handelt es sich eigentlich um die gleiche Beziehung in entgegengesetzter Richtung gedacht.

In beiden Fällen ergibt sich ein Synchronisationsbedarf zwischen den abhängigen VWS und der übergeordneten VWS im Falle von Änderungen an der übergeordneten VWS. Im Folgenden wird ein Konzept für die Synchronisation schrittweise beschrieben.

12.6.1 Rahmenbedingungen für den gewählten Lösungsansatz

Bei den Überlegungen zum SPoT (siehe Kapitel 12.1.2) wurde aufgezeigt, dass eine Verteilung der Informationen über mehrere Repositories, jeweils betrieben durch verschiedene Partner der Wertschöp-

¹²² <https://github.com/eclipse-basyx/basyx-databridge>

fungskette, angestrebt wird. Wenn man in der Praxis aber an Bibliotheken für Leitungssatzentwicklungs werkzeuge, mit mehreren tausend Komponenten von mehr als 100 verschiedenen Herstellern denkt, liegt es nahe, lokale Kopien der Verwaltungsschalen anzulegen, selbst wenn diese ein 1:1 Abbild der entfernten Verwaltungsschale sind. Dieses lokale "Caching" verringert die Antwortzeit der Systeme und erhöht die Ausfallsicherheit.

Der Bedarf nach zuverlässigen Synchronisationsmechanismen besteht beim lokalen Caching, um auch bei redundanter physikalischer Datenspeicherung nach wie vor einen logischen SPoT zu realisieren. Der Begriff der führenden Verwaltungsschale kann hier auf die ursprüngliche Verwaltungsschale angewendet werden, die den SPoT darstellt, wobei die abhängige Verwaltungsschale das "gecachte" Objekt ist, also die lokale Kopie.

Aus den Überlegungen zum SPoT geht hervor, dass es, bezogen auf ein bestimmtes Teilmodell, ein konkreter Datenbestand als SPoT definiert werden kann. Es wird angenommen, dass sämtliche Änderungen an diesem führenden Datenbestand vorgenommen werden und dass eine Synchronisation nur in eine Richtung, ausgehend vom führenden Datenbestand erfolgt. Eine Synchronisation in die umgekehrte Richtung kann zu Konflikten führen, wenn z.B. zeitgleich dasselbe Datenelement unterschiedlich geändert wird. Solche Konflikte sind nur schwierig aufzulösen und widersprechen der Idee des SPoT.

Es wird also vorausgesetzt, dass sämtliche Änderung in dem führenden Teilmodell vorgenommen werden und diese Änderungen anhand von Synchronisationsnachrichten an einen oder mehrere Empfänger verteilt wird. Bewusst offen bleibt jedoch, was mit den Synchronisationsnachrichten auf der Empfängerseite passiert. Bleibt es beim Anlegen einer exakten Kopie des ursprünglichen Teilmodells, wird es um zusätzliche Informationen ergänzt oder werden einzelne Werte mit eigenen "Wahrheiten" auf der Empfängerseite überschrieben?

All dies sind mögliche Anwendungsfälle. Es ist aber immer davon auszugehen, dass eine Änderung am führenden Teilmodell für den Empfänger von Interesse ist. Aber selbst wenn das nicht der Fall ist, kann die Synchronisationsnachricht der Ausgangspunkt sein für eine Überprüfung der eigenen Wahrheit auf der Empfängerseite.

Es wird angenommen, dass die Synchronisationsbeziehung im Einverständnis und mit Kenntnis aller beteiligten Parteien eingerichtet wird. Das Konzept geht davon aus, dass die Empfänger aus technischer Sicht und aus Sicht einer etwaigen Berechtigungssteuerung Zugriff haben auf das führende Teilmodell der Senderseite. Dieser Zugriff ist notwendig, damit initial der Empfänger seine lokale Kopie (abhängige VWS) anlegen kann. Diese abhängige VWS wird daraufhin durch die Synchronisationsnachrichten, welche die führende Verwaltungsschale bei Wertänderungen aktiv aussendet, aktuell gehalten. Fehler in der Übertragung können einen kompletten Abgleich des Teilmodells bzw. Eine Neu-Initialisierung erforderlich machen, sodass der direkte Zugriff auf das Teilmodell durch den Empfänger auch zu einem späteren Zeitpunkt noch notwendig und sinnvoll ist.

12.6.2 Konfiguration der Synchronisationsbeziehung

Es wird davon ausgegangen, dass eine Synchronisationsbeziehung immer auf Ebene ganzer Teilmodelle eingerichtet wird. Eine feinere Steuerung erhöht den Konfigurations- und Verwaltungsaufwand. Eine Synchronisation aller Teilmodele einer Verwaltungsschale ohne Möglichkeit zu Einschränkungen scheint aus fachlicher Sicht nicht sinnvoll; es sollen selten alle Informationen zu einem Asset mit allen Partnern der Wertschöpfungskette geteilt werden.

Zum Einrichten der Synchronisationsbeziehung registriert sich die Verwaltungsschale, die Änderungen bezüglich relevanter Teilmodele empfangen möchte, bei der führenden Verwaltungsschale. Die Initiative, dass eine Synchronisationsbeziehung eingerichtet wird, geht also von der Empfängerseite aus. Die Synchronisationsnachrichten werden direkt an den jeweiligen Empfänger gesendet. So behält die sendende Verwaltungsschale die Kontrolle darüber, wer die Nachrichten empfängt und man kann die Mechanismen der Zugriffssteuerung auf Teilmodele auch auf die Synchronisationsnachrichten anwenden. Würde man z.B. einen Message-Broker einbeziehen, der die Zustellung der Nachrichten an die einzelnen Empfänger übernimmt, müsste dieser auch die Zugriffssteuerung auf die Teilmodele der Verwaltungsschale berücksichtigen und anwenden. Wir gehen im Konzept hier jedoch davon aus, dass eine Komponente innerhalb der Systemgrenzen der Verwaltungsschalenimplementierung das zustellen der Nachrichten übernimmt.

Folgende Informationen sind im Rahmen des Verhandlungsprozesses für die Synchronisationsbeziehung festzulegen:

1. URL zur Registry, in der die empfangende Verwaltungsschale registriert ist
2. VWS-Id der empfangenden Verwaltungsschale
3. Liste der relevanten Teilmodelle, deren Änderungen gesendet werden sollen
4. Zeitintervall, das mindestens zwischen zwei Änderungsnachrichten liegen soll
5. Zeitintervall, nachdem mindestens eine Heartbeat-Nachricht gesendet werden soll, falls keine Änderung vorliegt
6. Weitere Parameter für die Steuerung des Fehlerhandlings wie z.B. Mindestzeitintervall für die Pufferung bei fehlerhaften Zustellversuchen

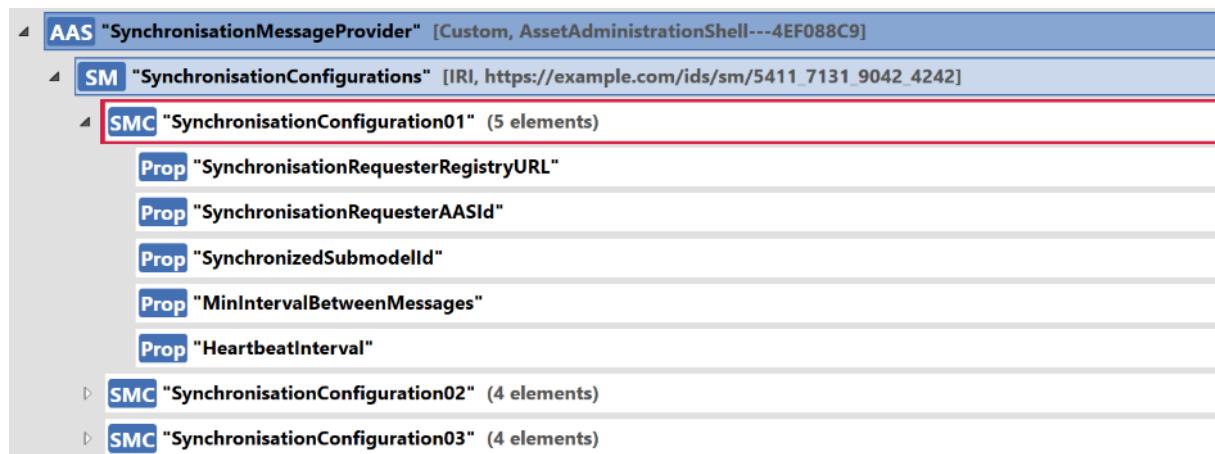


Abbildung : VWS-Synchronisation: Teilmodell zur Konfiguration der Synchronisationsbeziehung

Der 4. Parameter kann bei sich häufig ändernden Werten sinnvoll sein, um das Nachrichtenaufkommen zu regulieren. So kann man die Häufigkeit an die Anforderungen der Empfängerseite und an die gegebenen technischen Rahmenbedingungen der Netzwerkkommunikation anpassen. Dieser Parameter ist optional zu sehen. Erfolgt eine Änderung am Teilmodell innerhalb des Zeitintervalls wird zunächst keine Synchronisationsnachricht gesendet. Erst nachdem das Zeitintervall verstrichen ist, wird wieder eine Synchronisationsnachricht gesendet, die in einer Nachricht alle geänderten Teilmodellelemente mit dem jeweils aktuellen Wert beinhaltet.

Ändert sich z.B. innerhalb des Zeitintervalls ein bestimmtes Teilmodellelement zehn Mal, so wird nur eine Nachricht gesendet, die den jüngsten Wert des Elements beinhaltet.

Ist das Zeitintervall nicht angegeben oder auf 0 gesetzt, wird jede Änderung unmittelbar als Synchronisationsnachricht gesendet.

Es sollen beim Senden der Synchronisationsnachrichten die gleiche Berechtigungssteuerung greifen wie beim direkten Zugriff auf das Teilmodell über die Verwaltungsschalen-API. Um dies realisieren zu können, muss sich der Empfänger beim Einrichten der Synchronisationsbeziehung gegenüber der führenden Verwaltungsschale authentifizieren und für den Teilmodell-Zugriff autorisieren. Die führende Verwaltungsschale sollte innerhalb der Konfiguration zur Synchronisationsbeziehung die Informationen ablegen die notwendig sind, um auch zu einem späteren Zeitpunkt prüfen zu können ob der jeweilige Empfänger immer noch berechtigt ist auf das Teilmodell zuzugreifen. Welche Information hier notwendig ist, hängt von der technischen Implementierung der Zugriffsteuerung ab und deshalb sollen an dieser Stelle keine konkreten Vorgaben gemacht werden. Hierfür kommen auf der Senderseite möglicherweise noch weitere Parameter hinzu, die in der Konfiguration abgespeichert werden müssen.

Es wird vorgeschlagen die Konfiguration der Synchronisationsbeziehung in einem eigenen Teilmodell zur führenden Verwaltungsschale zu hinterlegen, wobei das Teilmodell für die Empfänger der Synchronisationsnachrichten nicht zugreifbar sein darf. Die Initiative zur Konfiguration geht zwar vom Empfänger

der Synchronisationsnachrichten aus, aber die Konfiguration kommt durch Interaktion, durch den Austausch von Nachrichten, zu Stande und nicht durch direktes Schreiben des Empfängers in das Konfigurations-Teilmodell.

Als Interaktionsprotokoll zur Konfiguration der Synchronisationsbeziehung bietet sich das in VDI/VDE 2193-2 [46] beschriebene Ausschreibungsverfahren an. In diesem Zusammenhang kann man das Senden der Synchronisationsnachrichten als Dienstleistung (Service) ansehen, welche von der führenden Verwaltungsschale angeboten wird. Man kann die sendende Verwaltungsschale als Service Provider ansehen. Die empfangende Verwaltungsschale gibt mit der Initiative zum Konfigurieren der Synchronisationsbeziehung den Auftrag zum Senden der Nachrichten und hat somit die Rolle des *Service Requesters* (Auftraggeber).

Das Interaktionsprotokoll Ausschreibungsverfahren startet mit der Ausschreibungsnachricht vom Service Requester zum Service Provider. Inhalt der Ausschreibungsnachricht sind die oben aufgezählten Punkte.

Kann der Service Provider die Parameter der Synchronisationsbeziehung akzeptieren, antwortet er mit einem Angebot, welches die Parameter der Ausschreibung bestätigt. Ist die Einrichtung der Synchronisation nicht möglich, weil der Empfänger z.B. nicht Zugriff auf alle relevanten Teilmodelle hat oder die Zeitintervalle außerhalb des zulässigen Bereichs liegen, lehnt der Service Provider die Ausschreibung ab oder antwortet mit einem Angebot, das aus Sicht des Service Providers funktionierende Konfigurationsparameter enthält.

Der Service Requester wiederum hat die Möglichkeit das Angebot mit den ggf. geänderten Konfigurationsparametern mit einer entsprechenden Nachricht zu akzeptieren oder abzulehnen.

Der Service Provider bestätigt die Angebotsannahme mit einer Nachricht und ab diesem Zeitpunkt ist die Synchronisationsbeziehung konfiguriert und die Synchronisationsnachrichten werden an den neu hinzugefügten Empfänger (Service Requester) gesendet.

Die erste Nachricht beinhaltet das komplette Teilmodell, um den Service Requester mit einem initialen Datenstand zu versorgen. So ist ausgeschlossen, dass Änderungen zwischen dem Einrichten der Synchronisationsbeziehung und der ersten gesendeten Änderung für den Service Requester verloren gehen.

Die Besonderheit des Ausschreibungsverfahrens in diesem Kontext ist die Tatsache, dass nur ein Service Provider angefragt wird. Nur die Verwaltungsschale, die den SPoT darstellt, kann diesen Service erbringen.

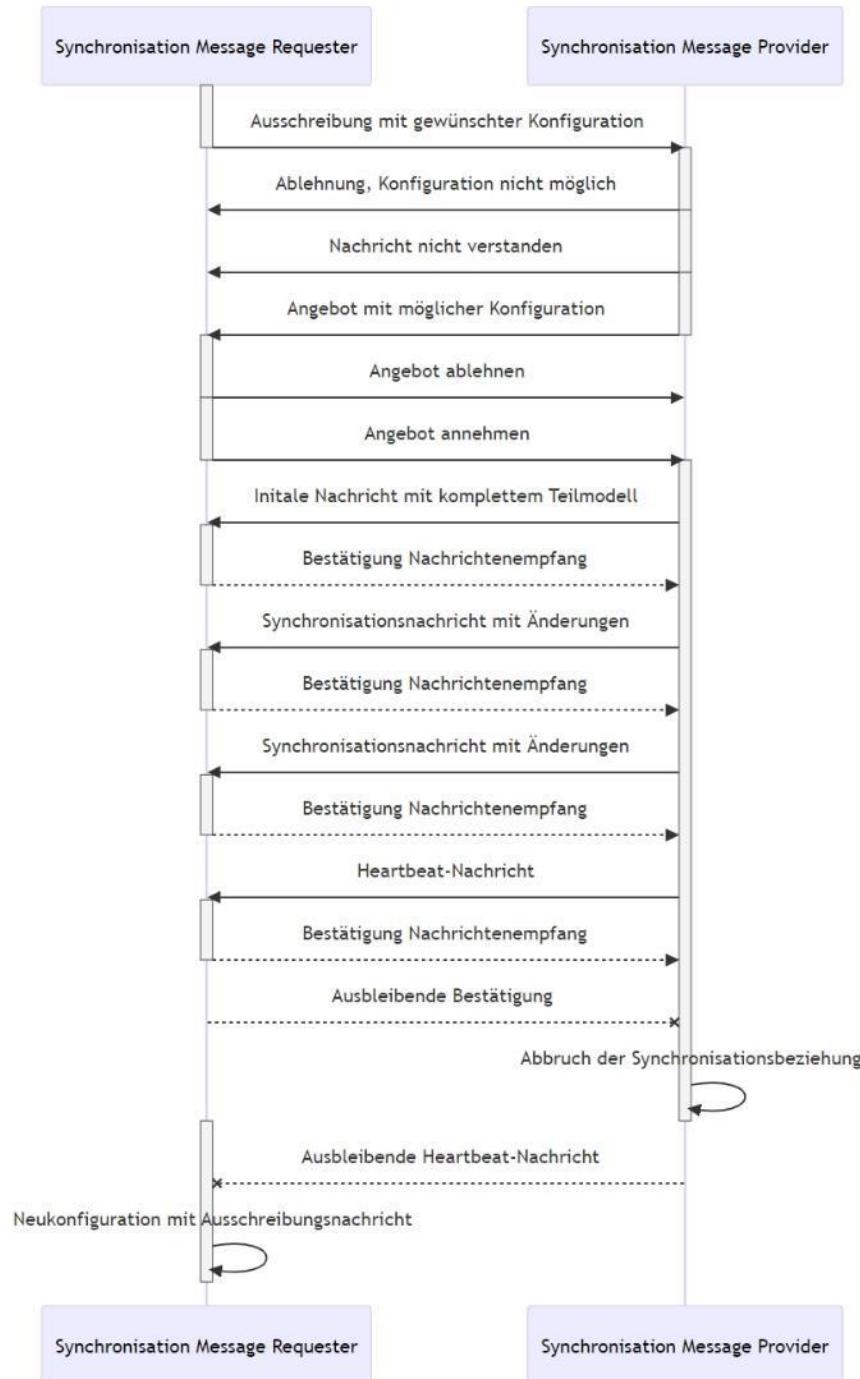


Abbildung : Interaktionsprotokoll zur VWS-Synchronisation

12.6.3 Senden von I4.0-Nachrichten zur Synchronisation

Im TP 6 von VWS4LS wurde als Mechanismus zum Austausch von I4.0-Nachrichten das Konzept der "Operation" des Verwaltungsschalen-Metamodells vorgeschlagen [72]. Zur Implementierung eines solchen Nachrichtenaustausch wurde ein Teilmodell namens „Message Participant“ erarbeitet, welches eine entsprechende Operation „newMessage“ definiert. Die Operation wird vom Sender der Nachricht aufgerufen, wobei die Operation im Teilmodell „Message Participant“ der empfangenden Verwaltungsschale ausgeführt wird.

Dieses Teilmodell und die Vorgehensweise wurden zunächst für den Nachrichtenaustausch im Interaktionsprotokoll „Ausschreibungsverfahren“ konzipiert. Da das Ausschreibungsverfahren zur Konfiguration der Synchronisationsbeziehung eingesetzt werden soll, kann vorausgesetzt werden, dass sowohl

Service Requester als auch Service Provider das Teilmodell "Message Participant" implementieren. So mit ist es naheliegend dieses Teilmodell mit der entsprechenden Operation zum Austausch von I4.0-Nachrichten auch für die Synchronisationsnachrichten zu nutzen.

Ein erster Ansatz für ein Interaktionsprotokoll ist, dass bei einer Änderung in einem Teilmodell auf Seite des Synchronisation Service Provider überprüft wird, ob für das Teilmodell Synchronisationsbeziehungen konfiguriert sind. Falls ja, wird für jede konfigurierte Synchronisationsbeziehung eine Nachricht generiert.

Die Granularität bzw. der Umfang der Änderungen, die innerhalb einer Synchronisationsnachricht gesendet werden, orientiert sich an Vorgängen, die zu Wertänderungen führen. Werden bspw. mit einer PUT- oder PATCH-Operation mehrere Teilmodellelemente gleichzeitig geändert, sind alle Wertänderungen verpflichtend innerhalb einer Synchronisationsnachricht zu übermitteln, um auf Seite des Service Requesters die gleichen Bedingungen in Bezug auf Datenkonsistenz zu schaffen wie auf Seite des Service Providers gegeben. Um dies zu gewährleisten, muss auch die Nachricht als Ganzes auf die Service Requester VWS angewendet werden. Wird ein einzelnes Datenelement geändert, so wird im allgemeinen Fall direkt eine Nachricht generiert mit dieser einzelnen Wertänderung. Im besonderen Fall, bei eingestelltem Mindestintervall, wird diese Nachricht aber zunächst gepuffert und ggf. mit weiteren Nachrichten, die innerhalb dieses Intervalls erzeugt werden, zusammengefasst.

Einen Mechanismus zur Gewährleistung von Konsistenz und Transaktionssicherheit über mehrere Teilmodelle hinweg sieht das Interaktionsprotokoll nicht vor. Ein solcher Mechanismus ist auch bei der lokalen VWS über die API-Zugriffe nicht gegeben. Um einen solchen Mechanismus zu realisieren sind Maßnahmen außerhalb des Interaktionsprotokolls zu treffen. Vorstellbar ist z.B. ein Datenelement, das einen konsistenten bzw. inkonsistenten Zustand anzeigt. Will man mehrere Teilmodelle ändern, ist dieser Zustand auf "inkonsistent" zu setzen, die Änderungen auf die betreffenden Teilmodelle anzuwenden und erst dann wieder den Zustand auf "konsistent" zu setzen. Vor den Zugriffen auf die VWS ist immer der Zustand zu prüfen. Um den Mechanismus auch im Rahmen der Synchronisation anwendbar zu machen, sind die Nachrichten in exakt der Reihenfolge zu erstellen, in der die lokalen API-Aufrufe stattfinden und die Nachrichten sind auch unbedingt in der Reihenfolge ihrer Erstellung zuzustellen.

Die Nachricht wird wie beschrieben per Operationsaufruf zugestellt. Dafür führt der Service Provider für die in der Konfiguration hinterlegte VWS-Id einen Discovery-Prozess bei dem ebenfalls konfigurierten VWS-Repository durch und ermittelte das "Message Participant"-Teilmodell. Per Rückgabewert des Operationsaufrufs wird mitgeteilt, ob die Zustellung erfolgreich war. War die Zustellung der Nachricht nicht erfolgreich, z.B. wegen temporär unterbrochener Netzwerkverbindung, puffert der Service Provider die Nachricht und wiederholt den Zustellversuch zu einem späteren Zeitpunkt. Gegebenenfalls ist die Konfiguration der Synchronisationsbeziehung um eine max. Anzahl an gepufferten Nachrichten, eine Höchstdauer für die Pufferung, oder eine Zeitintervall für die Wiederholungsversuche zu erweitern.

12.6.4 Inhalte der I4.0-Nachrichten zur Synchronisation

Der Aufbau der Synchronisationsnachricht folgt VDI/VDE 2193-1 [49] und teilt sich in Nachrichtenrahmen und Interaktionselementen mit den Nutzdaten zur Übermittlung der Wertänderungen.

Elemente des Nachrichtenrahmens:

| Nachrichten-element | Beschreibung | Verwendung |
|---------------------|--|---------------|
| Typ | Zweck der Nachricht: Synchronisation, Heartbeat | Verpflichtend |
| Sender | Absender einer Nachricht (VWS-ID) | Verpflichtend |
| Empfänger | Empfänger einer Nachricht (VWS-ID) | Optional |
| Konversations-ID | ID der Synchronisationsbeziehung, festgelegt während der Konfiguration (Verhandlungsprozess) | Verpflichtend |

| | | |
|------------------------|---|---------------|
| Nachrichten-ID | ID einer Nachricht, um z.B. eine doppelte Verarbeitung auf Empfängerseite zu verhindern. | Verpflichtend |
| Als-Antwort-Auf | Ausdruck, der die Ursprungs-nachricht referenziert, wird in diesem Interaktionsprotokoll bisher nicht benötigt. | Nicht genutzt |
| Antworten-bis | Zeitpunkt bis zu dem die Antwort vorliegen muss, wird in diesem Interaktionsprotokoll bisher nicht benötigt. | Nicht genutzt |
| Semantisches Protokoll | Identifikation des semantischen Protokolls auf das sich der Zweck bezieht, hier Synchronisation | Verpflichtend |
| Rolle | Aufgabe des Senders der Nachricht im semantischen Protokoll, hier Synchronisation Message Provider | Optional |

Im Bereich der Interaktionsmodelle wird zum einen die Teilmodellreferenz des geänderten Teilmodells angegeben und zum anderen als Datenelement eine Serialisierung der geänderten Teilmodellelemente. Optional kann in einem weiteren Datenelement eine Serialisierung der geänderten Teilmodellelemente mit ihren Werten vor der Änderung übertragen werden. Da es erforderlich sein kann, die vorherigen Werte je Service Requester zu speichern, z.B. in Verbindung mit Nutzung der minimalen Intervalle zwischen zwei Nachrichten, kann diese Speicherung zu Ressourcenproblemen beim Service Provider kommen. Beispiel sind begrenzte Hardwareressourcen oder eine hohe Anzahl an Service Requestern die mit Synchronisationsnachrichten zu versorgen sind. Aus diesem Grund wird das Datenelement mit den Vorgängerwerten als optional vorgeschlagen und wäre im Konfigurationsprozess zu vereinbaren.

12.6.5 Verarbeiten von I4.0-Nachrichten zur Synchronisation

Die Zustellung der Synchronisationsnachricht erfolgt, wie oben beschrieben, über den Aufruf der entsprechenden Operation im Teilmodell „Message Participant“ und nicht etwa, in dem das betroffene Element durch den *Synchronisation Message Provider* direkt im Zielteilmodell des *Synchronisation Message Requesters* geändert wird.

Mit der Implementierung dieser Operation besteht auf Seite des *Synchronisation Message Requesters* ein hoher Freiheitsgrad, wie mit der empfangenen Nachricht umgegangen wird. Wie eingangs beschrieben, werden verschiedene Einsatzgebiete des Interaktionsprotokolls Synchronisation gesehen. Im angesprochenen Anwendungsfall Caching, könnte die Funktion der Operation lediglich darin bestehen, in einem Cache mit 1:1 Kopien der SPoT-VWS den Cache ungültig zu setzen, damit beim nächsten Zugriff auf den Cache wieder ein Zugriff auf die SPoT-VWS erfolgt und ein aktualisiertes Cache-Objekt generiert wird. Der Algorithmus der Operation benötigt dazu lediglich die VWS-Identifikation aus dem Rahmen der I4.0-Nachricht und die Interaktionselemente der Nachricht würde der Algorithmus nicht auswerten. Mit der VWS-Identifikation, also der VWS-Id im Sender-Datenfeld des Nachrichtenrahmens, wird das Cache-Objekt, welches zu dieser VWS-ID gehört, gesucht und aus dem Cache entfernt.

Ein komplexerer Caching-Algorithmus könnte die serialisierten Teilmodellelemente, die in der Nachricht enthalten sind, dazu nutzen, um die Cache-Objekte direkt zu aktualisieren.

Das Anwendungsgebiet ist nicht auf Caching beschränkt. Die diskutierten Anwendungsfälle aus Sicht der fachlichen Domäne Leitungssatz zeigen u.a., dass es sinnvoll ist bei den abhängigen, abgeleiteten Verwaltungsschalen eine Normierung der Daten vorzunehmen. Beispielsweise die Umrechnung von Grad Fahrenheit in Grad Celsius oder AWG-Leitungsquerschnitte in das metrische System. Aber auch komplexere Geschäftslogiken mit Fallunterscheidungen bis hin zum Durchlauf langlaufender Workflows sind denkbar. Gerade bei den langlaufenden Algorithmen taucht wieder die Problematik der Datenkonsistenz auf und die Wichtigkeit der Verarbeitungsreihenfolge. Sollte in diesem Fall ein Puffern der eingehenden Synchronisationsnachrichten notwendig sein, so ist diese auf Seite des *Synchronisation Message Requesters* zu implementieren. Das Interaktionsprotokoll sieht vor, dass der Nachrichteneingang durch den Rückgabewert der Operation bestätigt wird. Diese Bestätigung ist aber rein auf die Datenübertragung zu beziehen und nicht auf die erfolgreiche Verarbeitung der empfangenen Nachricht. Es macht keinen Sinn, einen Fehlerstatus des Operationsaufrufs zurückzumelden, weil die Nachrichtenverarbeitung nicht abgeschlossen ist oder die Inhalte aus fachlicher Sicht, also aus Sicht der Geschäfts-

logik, nicht akzeptiert werden. An dieser Stelle sei auf das Änderungsmanagement (Kapitel 12.6) verwiesen, welches sich mit der fachlichen Abstimmung von Änderungen beschäftigt und auf der technischen Übertragung von Wertänderungen, wie sie hier beschrieben wurde, aufsetzt.

12.6.6 Umgang mit Fehlerzuständen

Wie oben beschrieben stellt der Operationsaufruf die Schnittstelle zwischen *Synchronisation Message Provider* und *Synchronisation Message Requester* dar. Der *Synchronisation Message Provider* ist dafür verantwortlich, dass der Operationsaufruf zuverlässig und in der korrekten Reihenfolge ausgeführt wird. Erst wenn der Operationsaufruf der vorhergehenden Nachricht mit erfolgreichem Rückgabestatus abgeschlossen wurde, darf der *Synchronisation Message Provider* die Nachricht als zugestellt betrachten und den Operationsaufruf für die folgenden Nachricht ausführen. Wenn ein Operationsaufruf technisch im Augenblick scheitert oder der *Synchronisation Message Requester* den Operationsaufruf mit einem Fehlerstatus quittiert, ist diese Nachricht zwischenzuspeichern und der Operationsaufruf ist, nach einem vorab vereinbarten Zeitintervall, zu wiederholen. Die Anzahl der Wiederholungen des Zustellversuchs, das Zeitintervall zwischen den Zustellversuchen, und die Obergrenze für Zeidauer und Nachrichtenzahl für die Zwischenspeicherung auf Providerseite sind Bestandteil der Konfiguration der Synchronisationsbeziehung. Sollte einer der Parameter überschritten werden, so werden sowohl die Zustellversuche der betreffenden Nachricht als auch die weitere Synchronisationsbeziehung durch den *Synchronisation Message Provider* beendet. Der *Synchronisation Message Provider* informiert den *Synchronisation Message Requester* durch den einmaligen Zustellversuch einer Nachricht vom Typ "Canceled". Bei diesem Nachrichtentyp ist die Angabe der Konversation-Id verpflichtend, damit der *Synchronisation Message Requester* erkennen kann, welche Synchronisationsbeziehung abgebrochen wurde, falls er mehrere Konfigurationen beim gleichen Provider eingerichtet hat.

Grundsätzlich ist nach einem erfolgreichen Operationsaufruf die Verantwortung für die weitere Verarbeitung der Nachricht auf den Requester übergegangen. Sollte eine Pufferung der Nachrichten für die folgende Verarbeitung notwendig sein, so ist diese auf Seite des Requesters zu implementieren.

Sollte innerhalb eines konfigurierten Zeitintervalls keine Nachricht vom Provider gesendet worden sein, so erzeugt der Provider eine Heartbeat-Nachricht mit dem entsprechenden Typ im Nachrichtenrahmen und der Konversations-Id, welche der Konfiguration der Synchronisationsbeziehung entspricht. Ziel dieser Nachrichten ist es, dem Requester eine Möglichkeit zu bieten, das Funktionieren der Synchronisationsbeziehung zu überwachen und eine Annahme über die Aktualität der Daten zu treffen, die ihm vorliegen.

Sollte innerhalb des konfigurierten Zeitintervalls weder eine Synchronisationsnachricht noch eine Heartbeat-Nachricht eingehen, muss der Requester davon ausgehen, dass die Synchronisationsbeziehung abgebrochen ist. Soll aus seiner Sicht die Synchronisation fortgesetzt werden, so ist - wie bei einem neuen Einrichten einer Synchronisationsbeziehung - der entsprechende Verhandlungsprozess durch eine Ausschreibungsrichtung zu starten. An dieser Stelle ist die bestehende Konversations-Id der abgebrochenen Synchronisationsbeziehung weiterzuverwenden.

Mit dem erfolgreichen Durchlaufen des Ausschreibungsverfahrens ist auch eine Re-Initialisierung des Datenbestandes durch komplettes Abrufen aller relevanten Teilmödelle durchzuführen. Dies ist im Verantwortungsbereich des Requesters.

Das gleiche Vorgehen, wie beim Ausbleiben der Heartbeat-Nachricht, wendet der Requester bei Eingang einer Nachricht vom Typ "Canceled" an.

Das beschriebene Vorgehen zum Umgang mit Fehlern ist bewusst einfach gehalten, in der Hoffnung, dass dadurch möglichst viele VWS-Implementierungen das Interaktionsprotokoll zur Synchronisation unterstützen. Im Ergebnis entspricht das im Wesentlichen dem Quality of Service Level 2 ("Exactly once") des MQTT -Protokolls. Dies ist den Diskussionen der fachlichen Anforderung an Synchronisierung geschuldet. Sollte sich später zeigen, dass es andere Anforderungen gibt, sei an dieser Stelle empfohlen sich an Definitionen und Mechanismen zu orientieren, die im Bereich der Message Broker etabliert sind.

12.8 Änderungsmanagement

Um flexibel und effizient auf sich ändernde Anforderungen reagieren zu können, ist ein gut durchdachter Änderungsprozess entscheidend. Hierbei werden eine Reihe von grundlegenden Änderungsbedarfen unterschieden:

- **Technischer Änderungsbedarf:** Der technische Änderungsbedarf bezieht sich auf alle Änderungen an der Bezugskonfiguration, z.B. die Länge einer Leitung, der Austausch einer vollständigen Leitung oder die Integration einer neuen Fahrzeugfunktion.
- **Produktänderung:** Produktänderungen entstehen, wenn neue Technologien oder Komponenten eingeführt werden, wie der Austausch eines Steckers gegen einen anderen.
- **Kostenoptimierung:** Änderung wegen Kostenoptimierung zielen darauf ab, die wirtschaftliche Effizienz zu erhöhen und Kosten zu reduzieren.
- **Prozessoptimierung:** Verbesserung von Fertigungsabläufen zur Steigerung der Produktivität, bspw. durch den Einsatz anderer Komponenten oder Verfahren, die den Prozess vereinfachen. Hierfür sind i.d.R. Änderungen am Produkt notwendig (z.B. zusätzliche Halter).
- **Lieferengpässe:** wenn Materialien oder Komponenten nicht rechtzeitig verfügbar sind und alternative Lösungen oder Lieferanten gefunden werden müssen.
- **Regularien:** Änderungsbedarf, für die Anpassung an aktuelle gesetzliche Vorgaben und Standards, erfordern.

Gemeinsam mit den Projektpartnern, die über das Domänenwissen des Leitungssatzes verfügen, sollte ein Ablauf für das Änderungsmanagement erstellt werden. Nach gemeinsamer Auffassung findet das Änderungsmanagement ausschließlich auf Typ-Ebene im Entwicklungsprozess statt. Aus dem Ablauf sollten wiederkehrende Interaktionen erkannt und generalisiert werden, sodass am Ende ein flexibler, aus Bausteinen zusammengesetzter Ablauf, entsteht.

Das Konzept soll die Abläufe durch proaktive Verwaltungsschalen realisieren. Dazu sind neben einer VWS für das Asset spezielle VWS für das Änderungsmanagement vorgesehen. Hierbei war zu entscheiden, ob es eine zentrale VWS zur Steuerung des Änderungsmanagement geben sollte, oder ob mehrere kleinere Workflows in verteilten VWS verankert werden sollten, die dann automatisiert bei Änderungen gestartet werden.

Die Definition des Ablaufs sollte Fragen klären, wie:

- Wer initiiert eine Änderung?
- Wer genehmigt eine Änderung?
- Welche Rollen sind im Änderungsmanagementprozess beteiligt?

Durch zusätzliche Definitionen von Interaktionsabläufen im nicht genehmigten Fall oder im Fehlerfall sollen alle möglichen Fälle abgefangen werden.

Zudem sollten für das Änderungsmanagement aktuell bestehende Prozesse in der Domäne analysiert werden. Sofern bereits jetzt ein großer manueller Aufwand vorhanden ist, könnten Konzepte wie GIT/SVN zumindest einen zeitlichen Vorteil mit sich bringen. Automatisierte Änderungsverwaltung oder auch weitere Konzepte sollten entsprechend der Anforderungen aus der Domäne betrachtet und (auch in ökonomischer Hinsicht) evaluiert werden.

Ähnlich zur Versionierung in Kapitel 12.4 werden während des Änderungsprozesses manuelle Arbeitsschritte notwendig sein, wie z.B. das Anstoßen eines solchen Prozesses oder die Analyse der technischen Umsetzbarkeit der gewünschten Änderung.

12.8.1 Ablauf

Die eingangs aufgeführten konkreten Änderungsbedarfe haben keinen Einfluss auf den grundsätzlichen Ablauf des Änderungsprozesses.

Der Änderungsprozess beginnt immer mit der Identifikation einer Änderung, gefolgt von einer Auswirkungsanalyse und einer Prüfung der technischen Machbarkeit. Anschließend erfolgt die Entscheidung, ob und wie die Änderung umgesetzt wird. Zuletzt wird die Änderung dokumentiert.

Je nachdem, ob der Änderungsbedarf durch den OEM oder von einem Tier festgestellt wurde, ergeben sich daraus unterschiedliche Ablaufszenarien. Diese werden nachfolgend näher erläutert und basieren auf einer Masterarbeit¹²³, die im Zusammenhang mit dem Projekt VWS4LS durchgeführt wurde.

12.8.1.1 Szenario 1: Änderungsbedarf seitens des OEM

In diesem Szenario stellt der OEM einen Änderungsbedarf fest und startet den Änderungsprozess. Dieser ist in 10 Einzelschritte aufgegliedert (Abbildung 12-37):

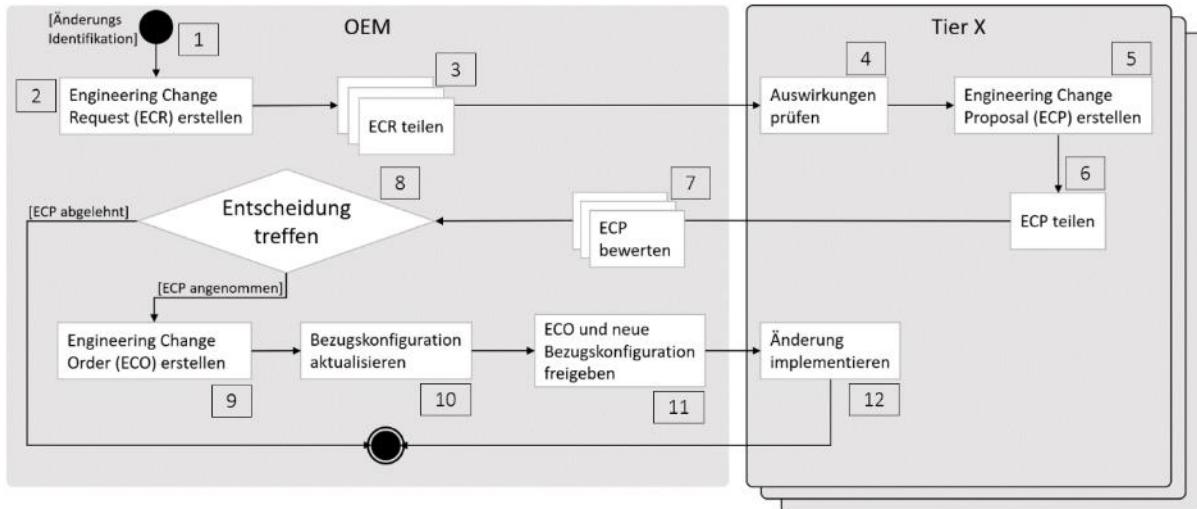


Abbildung 12-37: Änderungsmanagement: Änderungsbedarf vom OEM.

1. Änderungsidentifikation:

- Zunächst stellt der OEM einen Änderungsbedarf fest und löst damit den Änderungsprozess aus.

2. Engineering Change Request (ECR) erstellen:

- Der OEM erstellt in der Rolle der „Requesters“ einen ECR.
- Der erste Schritt zur Erstellung erfolgt in der Regel in informellen Formaten, wie Email, handschriftliche Notizen oder per mündlicher Überlieferung.
- Anschließend werden daraus alle notwendigen Informationen extrahiert und in das ECR Teilmödell (Siehe 12.8.2.1) überführt.
- In die Erstellung des ECR werden Fachprojektleiter und Entwicklungingenieure mit einbezogen.

3. ECR teilen:

- Eine Kopie des ECR wird mit Tier X geteilt. (Die Bezeichnung „Tier X“ wird hier als Sammelbegriff für alle Zuliefererstufen verwendet, also Tier 1-, Tier 2-, Tier n-Zulieferer und Konfektionäre).
- Wenn ein OEM mehrere Zulieferer für dieselben Teile beauftragt hat, erhält jeder betroffene Zulieferer eine Kopie des ECR.

4. Auswirkungen prüfen:

¹²³ Omar Abdelmageed, „Konzept zur Umsetzung eines Konfigurationsmanagements mit Hilfe der Verwaltungsschale“, Masterarbeit, Otto-von-Guericke-Universität Magdeburg, Fakultät für Elektrotechnik und Informationstechnik, November 2023.

- Tier X prüft die Auswirkungen der vorgeschlagenen Änderung. Dies kann bei einfachen Änderungen teilautomatisiert geschehen. Für komplexere Änderungsanfragen werden Mitarbeiter in den Prozess involviert, die die Auswirkungen der Änderungen analysieren.
 - Für Komplexe Änderungsanfragen wird ein internes Change Control Board (CCB) wird auf Seiten des Zulieferers durchlaufen
 - Betroffene Fachabteilungen werden identifiziert.
 - Eingangsinformationen aus ECR werden in Tier X-spezifische Formate übersetzt (z.B. KBL).
 - Kaufmännische Prozesse werden abgedeckt.
 - Ausarbeitung wird in Zusammenarbeit der betroffenen Fachabteilungen erstellt.
5. Engineering Change Proposal (ECP) erstellen:
- Basierend auf der Auswirkungsanalyse erstellt Tier X das Teilmodell Engineering Change Proposal (ECP) (Siehe 12.8.2.2)
 - Dieser enthält das ein Angebot, zu welchen Konditionen der Zulieferer die Angefragte Änderung umsetzen kann.
6. ECP teilen:
- Der ECP wird durch den Vertrieb des Zulieferers zur Bewertung mit dem OEM geteilt.
 - Der ECP verweist auf den initialen ECR, damit dieser zugeordnet werden kann
7. ECP bewerten:
- Einkauf, Fachabteilungen und Requester des OEM bewerten den ECP.
 - Der Requester hat dabei die Rolle des Geldgebers und vertritt Interessen auf Gesamtfahrzeugebene
8. Entscheidung treffen:
- Der OEM trifft eine Entscheidung, ob die Änderung angenommen oder abgelehnt wird.
 - Wenn das ECP abgelehnt wird, endet der Prozess hier.
 - Wenn das ECP angenommen wird, geht es weiter mit Schritt 9.
9. Engineering Change Order (ECO) erstellen:
- Der das Entwicklungsteam des OEM erstellt eine Engineering Change Order (ECO) (Siehe 12.8.2.3).
 - Dieser enthält die technische Beschreibung der Änderung, eine neue Bezugskonfiguration und den Einsatzzeitpunkt.
 - Die ECO verweist auf den zugehörigen ECP, damit diese zugeordnet werden kann.
10. Bezugskonfiguration aktualisieren:
- Der OEM aktualisiert die Bezugskonfiguration basierend auf dem ECO.
11. ECO und neue Bezugskonfiguration freigeben:
- Der ECO und die neue Bezugskonfiguration werden durch den OEM freigegeben.
12. Änderung implementieren:
- Tier X implementiert die Änderung.

12.8.1.2 Szenario 2: Änderungsbedarf seitens des Tier X

In diesem Szenario stellt der Tier X einen Änderungsbedarf fest und löst den Änderungsprozess aus. Dieser ist in 9 Einzelschritte mit diversen Entscheidungswegen untergliedert (Abbildung 12-38):

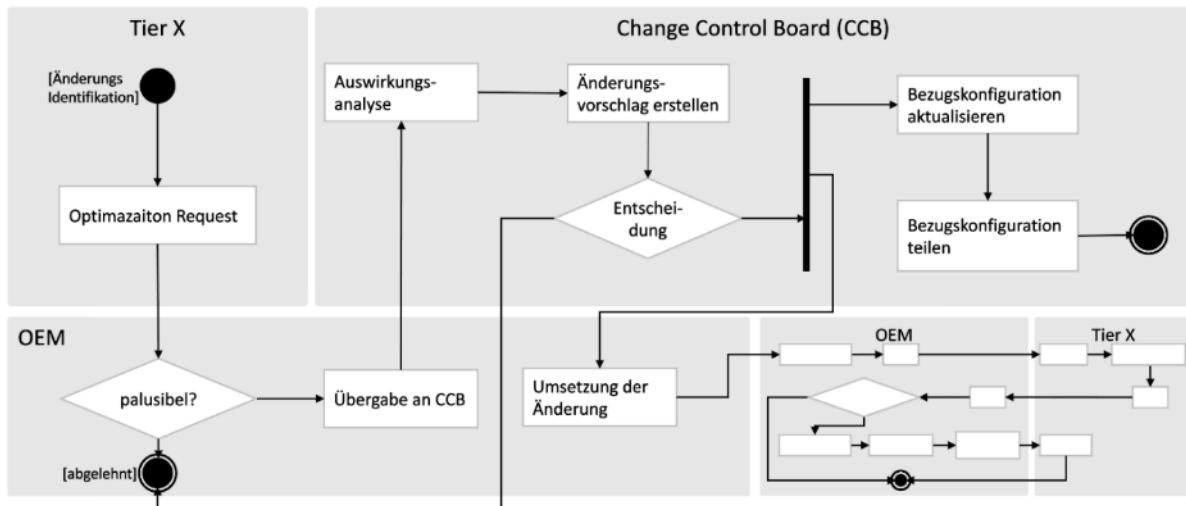


Abbildung 12-38: Änderungsmanagement: Änderungsbedarf vom Tier X

1. Änderungsidentifikation:

- Der Prozess beginnt mit der Identifikation einer notwendigen Änderung durch Tier X.

2. Optimization Request:

- Tier X erstellt einen Optimierungsantrag.

3. Prüfung auf Plausibilität:

- Der OEM prüft, ob der Optimierungsantrag plausibel ist.
- Wenn der Antrag plausibel ist, wird er an ein Change Control Board, bestehend aus allen betroffenen Parteien, übergeben (3.1). Hier können sowohl Vertreter des OEM als auch des Konfektionärs beteiligt sein.

Wenn der Antrag nicht plausibel ist, wird er abgelehnt (3.2) und der Prozess endet hier.

4. Auswirkungsanalyse:

- Wenn der Antrag plausibel ist, führt Tier X eine Auswirkungsanalyse durch. Dies kann entweder teilautomatisiert oder manuell geschehen, wie es im vorhergehenden Kapitel beschrieben wurde.

5. Änderungsvorschlag erstellen:

- Tier X erstellt basierend auf der Analyse einen Änderungsvorschlag und übergibt ihn an das Change Control Board (CCB).

6. Entscheidung:

- Das CCB trifft eine Entscheidung über den Änderungsvorschlag.
- Wenn die Entscheidung positiv ausfällt, werden die nächsten Schritte 7 und 8 parallel zu 9 ausgeführt
- Wenn die Entscheidung negativ ausfällt, wird der Antrag abgelehnt (3.2) und der Prozess endet hier.

7. Bezugskonfiguration aktualisieren:

- Die Bezugskonfiguration wird in der VWS des Konfigurationsobjektes aktualisiert.

8. Konfigurationsobjekt teilen:

- Die aktualisierte VWS des Konfigurationsobjektes wird mit allen relevanten Parteien geteilt.

9. Umsetzung der Änderung:

- Bei Zustimmung wird der Prozess „OEM gibt Änderungen vor“ angestoßen, wie er im vorhergehenden Kapitel beschrieben ist. (Hier in Abbildung 12-38 dargestellt, als stilisierte Version des Prozesses.)

12.8.2 Änderungsmanagement in der Verwaltungsschale

Um das Änderungsmanagement in der Verwaltungsschale abzubilden, wird zunächst eine VWS für das Konfigurationsobjekt erzeugt (Siehe Abbildung 12-39: ÄM_Konfigurationsobjekt).

). Dieses dient als Ankerpunkt für die Teilmodelle *EngineeringChangeRequests*, *EngineeringChangeProposals*, *EngineeringChangeOrders* und *Bezugskonfigurationen*.

Dateiname: Änderungsmanagement.aasx

| |
|--|
| AAS "ÄM_Konfigurationsobjekt" |
| ▷ SM "EngineeringChangeRequests" [https://example.com/ids/sm/5145_0101_9032_2782] |
| ▷ SM "EngineeringChangeProposals" [https://example.com/ids/sm/0245_0101_9032_3254] |
| ▷ SM "EngineeringChangeOrders" [https://example.com/ids/sm/2245_0101_9032_6948] |
| ▷ SM "Bezugskonfigurationen" [https://example.com/ids/sm/4245_0101_9032_8279] |

Abbildung 12-39: ÄM_Konfigurationsobjekt).

12.8.2.1 Teilmodell EngineeringChangeRequests

Das Teilmodell *EngineeringChangeRequests* (

Abbildung 12-40) enthält eine oder mehrere *SubmodelElementCollection*, die jeweils eine Änderungsanfrage repräsentieren und eine Reihe von Eigenschaften enthält:

| |
|---|
| AAS "ÄM_Konfigurationsobjekt" [..] of [NotApplicable] |
| Asset AssetInformation |
| ▷ SM "EngineeringChangeRequests" [https://example.com/ids/sm/5145_0101_9032_2782] |
| ▷ SMC "ECR_0" (8 elements) @{Multiplicity=OneToMany} |
| Prop "Status_ECR" = abgeschlossen @{Multiplicity=One} @{FormChoices=in Bearbeitung; angenommen; abgelehnt;} |
| Prop "Name_Antragsteller_ECR" @{Multiplicity=One} |
| Prop "Zeitpunkt" = 30.1.2024 @{Multiplicity=One} |
| Prop "Beschreibung_des_Änderungsbedarf" = Leitung 2 muss gegen gewistete Leitung aufgrund einer Funktionsänderung geändere... @{Multiplicity=One} |
| Ref "Derzeitige_Bezugskonfiguration" => [Submodel, https://example.com/ids/sm/4245_0101_9032_8279], [SubmodelElementCol... @{Multiplicity=One}] |
| File "File" @{Multiplicity=OneToMany} |
| Ref "zugehöriger_ECP" => [Submodel, https://example.com/ids/sm/0245_0101_9032_3254], [SubmodelElementCol... @{Multiplicity=One}] |
| Prop "Begründung_zur_Ablehnung" @{Multiplicity=One} |

Abbildung 12-40: VWS-Änderungsmanagement: Teilmodell EngineeringChangeRequests

- Status_ECR*: beschreibt den aktuellen Bearbeitungsstatus des ECR. Initial wird der Status "in Bearbeitung" vergeben. Für den weiteren Verlauf stehen "angenommen" oder "abgelehnt" zur Verfügung.
- Name_Antragsteller_ECR*: ist die verantwortliche Person für den ECR.
- Zeitpunkt*: spiegelt den Eingangszeitpunkt des ECR wieder.
- Beschreibung_des_Änderungsbedarf*: formlose Beschreibung der Änderung.
- Derzeitige_Bezugskonfiguration*: verweis auf die Bezugskonfiguration, die abgeändert werden soll.
- File*: eine oder mehrere ergänzende Dateien. Beispiele: CAD-Datei der vorgesehenen Änderung, Änderungsanfrage als PDF.
- zugehöriger_ECP*: verweist auf den zugehörigen ECP, wenn der ECR angenommen wurde.
- Begründung_zur_Ablehnung*: enthält den Grund für die Ablehnung, wenn der ECR abgelehnt wurde.

12.8.2.2 Teilmodell EngineeringChangeProposals

Das Teilmodell *ECP* (Abbildung 12-41) enthält eine oder mehrere SubmodelElementCollection(s), die jeweils einen Änderungsvorschlag repräsentieren und eine Reihe von Eigenschaften enthält:

| |
|--|
| AAS "ÄM_Konfigurationsobjekt" [] of [NotApplicable] |
| Asset AssetInformation |
| SM "EngineeringChangeProposals" [https://example.com/ids/sm/0245_0101_9032_3254] |
| SMC "ECP" (10 elements) @{Multiplicity=OneToMany} |
| Prop "Status_ECP" @{Multiplicity=One} @{FormChoices=in Bearbeitung; angenommen; abgelehnt;} |
| Ref "zugehöriger_ECR" @{Multiplicity=One} |
| Prop "Name_Antragsteller" @{Multiplicity=One} |
| Prop "Zeitpunkt" = 31.1.2024 @{Multiplicity=One} |
| Prop "vorgeschlagene Änderung" = Tausch der leitung zwei gegen eine getwistete Leitung @{Multiplicity=One} |
| SMC "Auswirkungsanalyse" (1 elements) @{Multiplicity=One} |
| Prop "Kriterium_Bericht" @{Multiplicity=OneToMany} |
| SMC "Liste_betroffene_Komponente" (1 elements) @{Multiplicity=One} |
| Ent "Komponente" @{Multiplicity=OneToMany} |
| SMC "Liste_Ressourcenbedarf" (1 elements) @{Multiplicity=One} |
| Prop "Ressource" @{Multiplicity=OneToMany} |
| File "Dokumente" @{Multiplicity=OneToMany} |
| Ref "zugehörige_ECO" = [GlobalReference, 1]Submodel, https://example.com/ids/sm/2245_0101_9032_6948]... @{Multiplicity=One} |
| SMC "ECP_abgelehnt" (1 elements) @{Multiplicity=One} |
| SMC "ECP" (10 elements) @{Multiplicity=OneToMany} |
| Prop "Status_ECP" @{Multiplicity=One} @{FormChoices=in Bearbeitung; angenommen; abgelehnt;} |
| Ref "zugehöriger_ECR" @{Multiplicity=One} |
| Prop "Name_Antragsteller" @{Multiplicity=One} |
| Prop "Zeitpunkt" = 31.1.2024 @{Multiplicity=One} |
| Prop "vorgeschlagene Änderung" @{Multiplicity=One} |
| SMC "Auswirkungsanalyse" (1 elements) @{Multiplicity=One} |
| SMC "Liste_betroffene_Komponente" (1 elements) @{Multiplicity=One} |
| SMC "Liste_Ressourcenbedarf" (1 elements) @{Multiplicity=One} |
| File "Dokumente" @{Multiplicity=OneToMany} |
| Prop "Begründung_zur_Ablehnung" @{Multiplicity=One} |

Abbildung 12-41: VWS-Änderungsmanagement: Teilmodell EngineeringChangeProposals

- *Status_ECP*: beschreibt den aktuellen Bearbeitungsstatus des ECP. Initial wird der Status "in Bearbeitung" vergeben. Für den weiteren Verlauf stehen "angenommen" oder "abgelehnt" zur Verfügung.
- *zugehöriger_ECR*: verweist auf den zugehörigen Engineering Change Request (ECR), welcher diesen Vorschlag initiiert hat.
- *Name_Antragsteller*: ist für den Initiator des ECP vorgesehen.
- *Zeitpunkt*: spiegelt den Zeitpunkt der Erstellung des ECP wider.
- *vorgeschlagene Änderung*: beschreibt die vorgeschlagene Änderung.
- *Auswirkungsanalyse*: enthält eine Analyse der Auswirkungen des Änderungsvorschlags, die in dem internen Change Control Board (CCB) der Entwicklungspartner durchgeführt wird.
- *Liste_betroffene_Komponente*: listet die von der vorgeschlagenen Änderung betroffenen Komponenten auf.
- *Liste_Ressourcenbedarf*: spezifiziert den Ressourcenbedarf für die vorgeschlagene Änderung.
- *Dokumente*: kann ergänzende Dateien enthalten, wie z.B. Pläne oder technische Spezifikationen.
- *Begründung_zur_Ablehnung*: enthält den Ablehnungsgrund, falls der ECP abgelehnt wurde.

12.8.2.3 Teilmodell EngineeringChangeOrders

Das Teilmodell *ECO* (Abbildung 12-42) enthält eine oder mehrere SubmodelElementCollection(s), die jeweils einen Änderungsauftrag repräsentieren und eine Reihe von Eigenschaften enthält:

| | | |
|--------------|---|--|
| AAS | "ÄM_Konfigurationsobjekt" | □ of [NotApplicable] |
| Asset | AssetInformation | |
| SM | "EngineeringChangeOrders" | [https://example.com/ids/sm/2245_0101_9032_6948] |
| SMC | "ECO" | (7 elements) @{Multiplicity=OneToMany} |
| Prop | "Name_Auftragsteller" | |
| Ref | "zugehöriger_ECP" | ⇒ [Submodel, https://example.com/ids/sm/0245_0101_9032_3254] [SubmodelElementCol... @{Multiplicity=One}] |
| Prop | "ErstellungsZeitpunkt" | = 31.1.2024 |
| Prop | "Beschreibung_der_Änderung" | = Tausch der Leitung zwei gegen eine getwistete Leitung @{Multiplicity=One} |
| SMC | "Schritte_zur_Änderungsimplementierung" | (1 elements) @{Multiplicity=One} |
| Prop | "Implementierungsschritt" | @{Multiplicity=OneToMany} |
| File | "Dokumente" | @{Multiplicity=OneToMany} |
| Ref | "neue_Bezugskonfiguration" | @{Multiplicity=One} |

Abbildung 12-42: VWS-Änderungsmanagement: Teilmodell EngineeringChangeOrders

- *Name_Auftragsteller*: gibt den Namen des Auftragstellers an.
- *zugehöriger_ECP*: verweist auf den zugehörigen ECP, welcher diesen Auftrag initiiert hat.
- *Erstellungszeitpunkt*: gibt den Zeitpunkt der Erstellung des Änderungsauftrags an.
- *Beschreibung_der_Änderung*: beschreibt die durchzuführende Änderung.
- *Schritte_zur_Änderungsimplementierung*: beschreibt mit einzelnen Properties die notwendigen Schritte zur Implementierung der Änderung durch einen Freitext.
- *Implementierungsbericht*: enthält einen oder mehrere Berichte oder Dokumente zur Umsetzung der Änderung.
- *neue_Bezugskonfiguration*: verweist auf die neue Konfiguration, die nach Umsetzung der Änderung gültig ist.

12.8.2.4 Teilmodell Bezugskonfigurationen

Das Teilmodell *Bezugskonfigurationen* (Abbildung 12-43) enthält eine oder mehrere SubmodelElementCollection(s), die jeweils eine Bezugskonfiguration repräsentieren und eine Reihe von Eigenschaften enthält. Eine Bezugskonfiguration ist der Datenstamm (VEC, KBL, etc.), auf den eine Änderung angewendet wird.

| | | |
|--------------|--------------------------------|---|
| AAS | "ÄM_Konfigurationsobjekt" | □ of [NotApplicable] |
| Asset | AssetInformation | |
| SM | "Bezugskonfigurationen" | [https://example.com/ids/sm/4245_0101_9032_8279] |
| SMC | "Bezugskonfiguration_Version" | (3 elements) @{Multiplicity=OneToMany} |
| Prop | "Datum_Freigabe" | = ABC15789 @{Multiplicity=One} |
| Prop | "Grund_für_die_Aktualisierung" | = Funktionsänderung, welche eine weitere Leitung benötigt @{Multiplicity=One} |
| File | "Dokumente" | @{Multiplicity=OneToMany} |

Abbildung 12-43: VWS-Änderungsmanagement: Teilmodell Bezugskonfiguration

- *Datum_Freigabe*: enthält das Datum, ab wann die Bezugskonfiguration vom OEM freigegeben ist.
- *Grund_für_die_Aktualisierung*: beschreibt den Grund für die Aktualisierung in menschenlesbarer Form.
- *Dokumente*: enthält ein oder mehrere Dokumente, die den Datenstamm für die Bezugskonfiguration repräsentieren.

12.10 Rückverfolgbarkeit

Das Thema Rückverfolgbarkeit fokussiert sich darauf, Informationen über die gesamte Wertschöpfungskette hinweg im VWS-Ökosystem auffindbar zu machen. Dazu wurden zwei Szenarien betrachtet:

12.10.1 Szenario 1: Rückverfolgbarkeit über mehrere Tier-Stufen

Der OEM oder der Komponentenlieferant informiert den Kabelkonfektionär über eine fehlerhafte Charge. Der OEM verlangt eine Liste der betroffenen Leitungssätze, in denen Komponenten dieser Charge verbaut wurden.

12.10.1.1 Anforderungen

Unter Berücksichtigung einer modularen Verwaltungsschalen-Infrastruktur (vgl. Kapitel 12.2), ist eine eindeutige Identifizierung der am Herstellungsprozess beteiligten Assets über die gesamte Wertschöpfungskette unverzichtbar.

Schon bei der Bestellung des OEM beim Kabelkonfektionär muss eine eindeutige Identifizierung des Assets (Fahrzeug / Leitungssatz) erfolgen. Dieselbe Anforderung gilt bei der Beschaffung von Komponenten, die der Komponentenlieferant bereitstellt und beim Kabelkonfektionär verbaut werden (vgl. Abbildung 12-44).

12.10.1.2 Lösungsweg

Für eine lückenlose Rückverfolgbarkeit müssen die Bestellinformationen eindeutig identifizierbar sein. Hierzu gehören unter anderem:

- Fahrzeugidentifikationsnummer (FIN)
- Artikelnummer oder Konfiguration des Leitungssatzes oder der Komponente
- Auftragsnummer

Sobald das Produkt des Lieferanten an den Besteller übergeht, müssen die Lieferinformationen eindeutig identifizierbar sein. Hierzu gehören unter anderem:

- FIN
- Datum und Uhrzeit des Warenausgangs
- Chargen- bzw. Seriennummer des Erzeugnisses

Diese Informationen müssen beim jeweiligen Lieferanten und Besteller in der zugehörigen Verwaltungsschale des Assets hinterlegt werden. Der Kabelkonfektionär muss für jeden Leitungssatz jede verbaute Komponente eindeutig identifizieren und in seiner Verwaltungsschale dokumentieren.

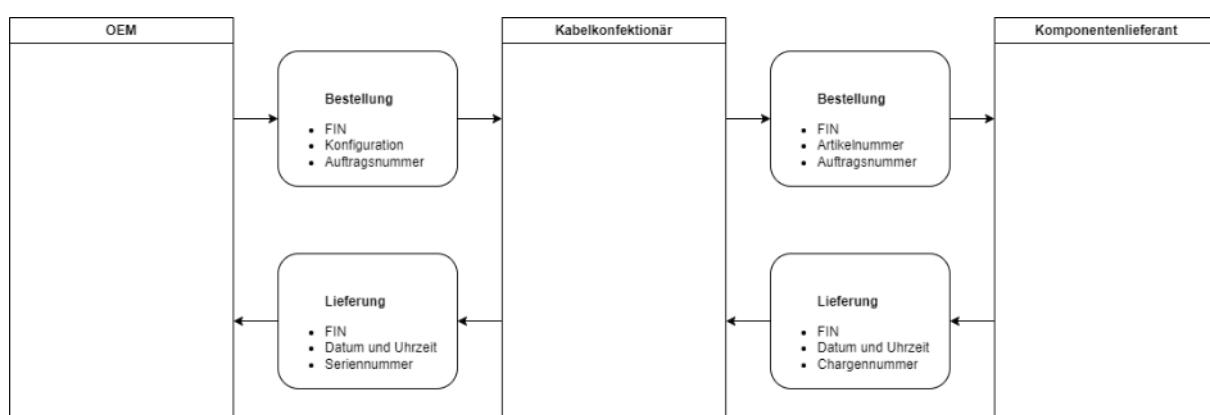


Abbildung 12-44: Rückverfolgbarkeit über mehrere Tier-Stufen

Mit diesem Ansatz wird das Szenario1 erfüllt:

Anhand der Chargennummer des Komponentenlieferanten kann der Kabelkonfektionär in seinen Verwaltungsschalen nachvollziehen, in welchen Leitungssätzen die Komponenten verbaut wurden, und übermittelt dem OEM die zugehörigen Bestell- bzw. Lieferinformationen (insbesondere die FIN).

12.10.2 Szenario 2: Rückverfolgbarkeit beim Kabelkonfektionär

Der OEM reklamiert Leitungssätze, bei denen Schraubverbindungen Feldausfälle verursacht haben und fordert den Kabelkonfektionär dazu auf, den Schraubfall gemäß Spezifikation nachzuweisen.

12.10.2.1 Anforderungen

Ausgehend von der Bestellung des OEM bzw. eigenen Produktionserwartungen muss der Kabelkonfektionär eine lückenlose Rückverfolgbarkeit aller hergestellten Leitungssätze garantieren. Hierzu müssen die Chargennummern der Komponentenlieferanten sowie die Prozessdaten der Ressourcen dokumentiert und mit den Produktions-, Bestell- und Lieferdaten verknüpft werden.

12.10.2.2 Lösungsweg

Die folgende Abbildung illustriert den Zusammenhang ausgehend von der Bestellung, die in einen Produktionsauftrag überführt wird, bis zur Lieferung.

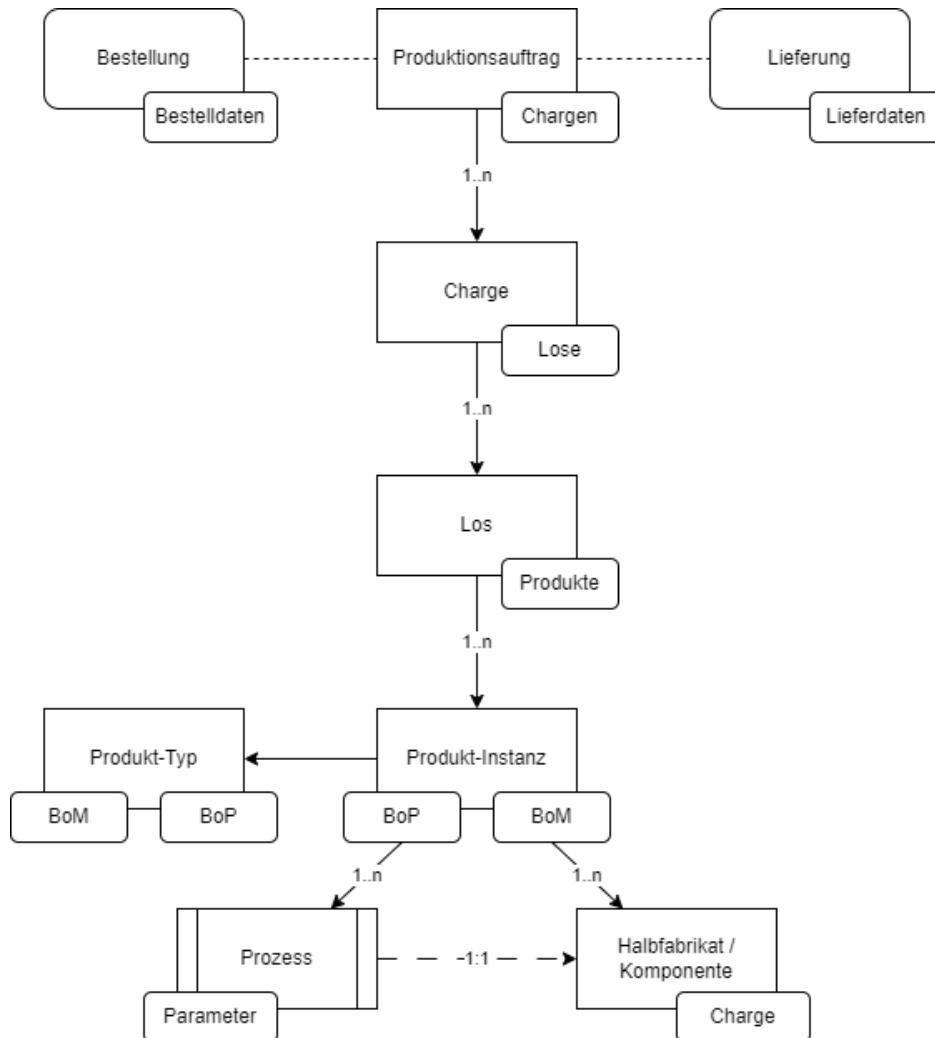


Abbildung 12-45: Rückverfolgbarkeit beim Kabelkonfektionär

Der Produktionsauftrag verweist auf eine oder mehrere Chargen, die wiederum auf ein oder mehrere Lose verweisen. Jedes Los verweist wiederum auf eine oder mehrere Produkt-Instanzen (z.B. Leitungssatz oder Komponente). Diese Aufgliederung entspricht den heutigen Produktionsbedingungen.

Die Produkt-Instanz beinhaltet Informationen über die durchgeführten Produktionsprozesse und wird von einem Produkt-Typ abgeleitet, der wiederum die geplanten Prozessschritte (BoP) sowie die dafür benötigten Komponenten und Halbfabrikate definiert (BoM).

In der BoP im Produkt-Typ werden die einzelnen Prozessschritte mit ihren jeweiligen Soll-Parametern definiert. In der Produkt-Instanz werden in der BoP die Prozessergebnisse dokumentiert und mit den Chargeninformationen der Komponenten und Halbfabrikate verknüpft.

Mit diesem Ansatz wird das Szenario 2 erfüllt:

Anhand der vom OEM bereitgestellten FIN kann der Kabelkonfektionär die zugehörigen Produktionsaufträge und somit die Produkt-Instanzen ermitteln. In den Produkt-Instanzen ermittelt der Kabelkonfektionär dann den geforderten Produktionsprozess für die Schraubverbindung und kann anhand der dokumentierten Prozessdaten den geforderten Nachweis erbringen.

12.10.3 Datenvorhaltung

Um den Anforderungen für die Rückverfolgbarkeit und den damit einhergehenden Anforderungen über die Detailtiefe der Informationen gerecht zu werden, ist es notwendig viele Informationen über lange Zeiträume vorzuhalten. Die hierdurch entstehenden Datenmengen müssen begrenzt werden, um die Kosten für die Datenhaltung beherrschbar zu machen.

Neben den Kosten für die Datenhaltung müssen auch gesetzliche und vertragliche Anforderungen beachtet und der Nutzen der Daten bewertet werden.

Für diesen Zweck wurde ein eigenes Teilmodell für die Datenvorhaltung definiert (Abbildung 12-46) und als „IDTA 02056: Data Retention Policies“ [25] standardisiert. Dieses Teilmodell beschreibt die Anforderungen an eine Datenvorhaltung und lässt sich stufenweise ergänzen. So ist es beispielsweise möglich, gesetzliche Anforderungen zentral, und vertragliche Anforderungen produktspezifisch zu definieren. Auf das jeweilige Produkt werden dann alle Anforderungen angewendet.

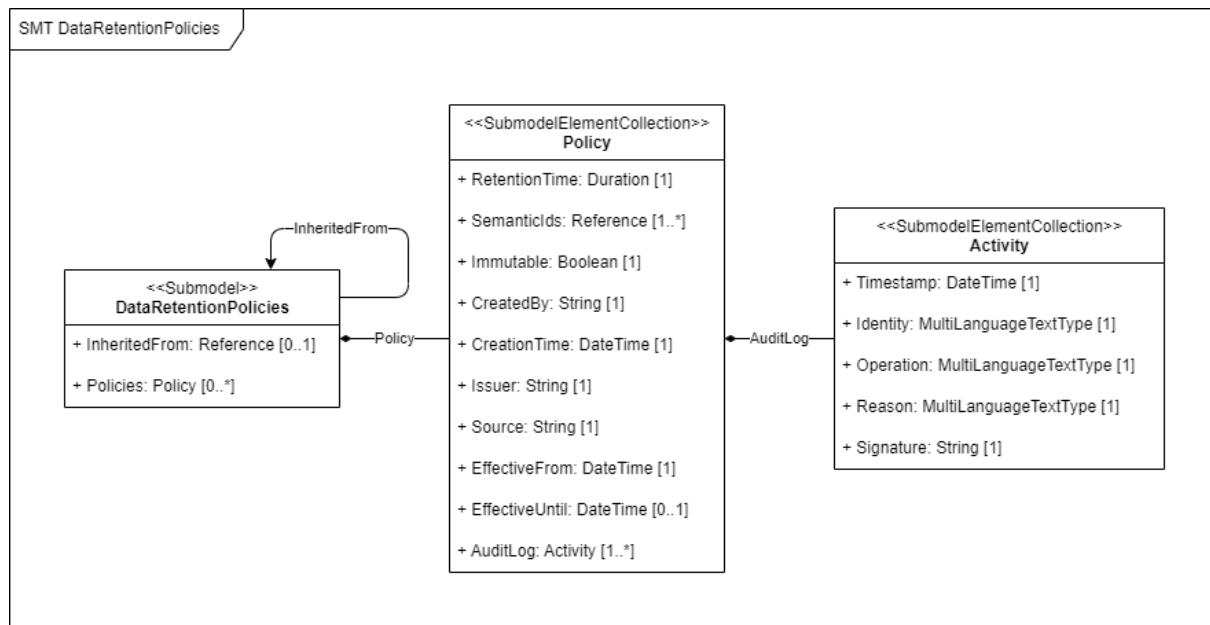


Abbildung 12-46: Teilmodell "Data Retention Policies"

Das Teilmodell wird auf Typ-Verwaltungsschalen definiert und verweist mittels semantischer IDs („SemanticIds“) auf die betroffenen Informationselemente. Anhand der „RetentionTime“ wird festgelegt, wie lange das Informationselement vorgehalten werden muss.

Die zusätzlichen Metadaten „Source“ und „Issuer“ verweisen auf die Quelle der Regel. Beispielsweise auf eine gesetzliche, vertragliche, oder interne Regel.

Ob und wann eine Regel angewendet wird, ist über die Parameter „EffectiveFrom“ und „EffectiveUntil“ festgelegt. Somit können Regeln ab einem bestimmten Zeitpunkt und für einen bestimmten Zeitraum angewendet werden.

Da Regeln mittels der „InheritedFrom“-Beziehung überschrieben werden können – beispielsweise wegen einer vertraglichen Regelung länger aufbewahrt werden sollen – besteht auch die Möglichkeit, eine

Regel mit „Immutable“ als unveränderlich zu definieren. Als „Immutable“ deklarierte Regeln dürfen auf keinen Fall überschrieben werden. Dazu gehören beispielsweise gesetzliche Regelungen.

Weil Regeln sich ändern können, kann anhand der Autordaten „Creator“, „CreationTime“, sowie „AuditLog“ genau nachvollzogen werden, wer bzw. wann eine Regel hinzugefügt bzw. geändert hat bzw. wurde.

Die Regeln müssen durch ein automatisches System angewendet und ausgewertet werden. Im Entstehungszeitpunkt eines Informationselements muss dieses automatische System die anzuwendenden Regeln auswerten und das Informationselement annotieren. Hierbei werden zwei Informationen hinterlegt:

- Die Quelle der Regel „PolicySource“.
- Der frühestmögliche Zeitpunkt für eine Löschung „KeepUntil“.

Damit die Daten auch gelöscht werden, muss das automatische System die Verwaltungsschalen regelmäßig überprüfen und fällige Daten löschen. Hierbei sind hierarchische Strukturen zu beachten, wobei ein tieferliegendes Element später gelöscht werden sollte als ein höherliegendes Element. Beispielsweise soll ein Teilmodell nach einem Jahr, ein Element innerhalb des Teilmodells aber erst nach zwei Jahren gelöscht werden. In diesem Fall darf das Teilmodell ebenfalls erst nach zwei Jahren gelöscht werden.

12.11 Fazit

Die umfassende Betrachtung der zentralen Architekturthemen – **Modularisierung, Versionierung, Verlinkung, Synchronisation, Änderungsmanagement** und **Rückverfolgbarkeit** – im Kontext des Leitungssatzes zeigt, wie entscheidend eine durchdachte, integrierte Strategie für das Managen von Verwaltungsschalen über den gesamten Lebenszyklus hinweg ist. Die Ergebnisse sind im Fokus des Projektes VWS4LS entstanden, in dem die Wertschöpfung des Leitungssatzes im Vordergrund steht. Dennoch wurde versucht, einen generalistischen Ansatz zu finden und die Ergebnisse für jegliche Industrie anwendbar zu machen.

Modularisierung und **Verlinkung** bilden die technische Grundlage für die Umsetzung einer verteilten und kollaborativen VWS-Lösung. Die Modularisierung erlaubt es, verschiedene Sichtweisen, mit unterschiedlichem Informationsgehalt der in der Wertkette angesiedelten Stakeholder, zu ermöglichen. Der Komponentenhersteller hat in diesem Szenario bspw. eine detailliertere Sicht auf die in der VWS enthaltenen Informationen (bspw. Qualitäts- und Produktionsdaten), die nicht für den Konfektionär oder den OEM bestimmt sind. Die *specificAssetId* spielt hierbei eine zentrale Rolle, da sie für die Verlinkung zwischen den VWS verschiedener Stakeholder verwendet wird. Diese ID ermöglicht das Auffinden der VWS in den Registries der Wertschöpfungspartner. Allerdings ist die Änderung der *specificAssetId* nur mit erheblichem Aufwand möglich, da sie den Kern der Verlinkung darstellt. Die **dezentrale Architektur** ist besonders wichtig, um die Vertraulichkeit der Daten zu gewährleisten und Akzeptanzprobleme aufgrund eines zentralen Administrators zu vermeiden. Die Implementierung eines Discovery-Konzeptes für den Datenraum ist dabei erforderlich, um die verteilte Datenverwaltung zu optimieren und den sicheren Austausch der Daten zwischen den Beteiligten zu gewährleisten.

Versionierung und die Definition des SPoT sind ebenfalls grundlegende Konzepte für eine dezentrale VWS-Lösung. Die Versionierung stellt sicher, dass alle Änderungen nachvollzogen und korrekt implementiert werden. Der SPoT sorgt dafür, dass es eine eindeutige, zentrale Quelle für alle relevanten Daten gibt, auch wenn diese dezentral verwaltet werden. In der Versionierung werden Arbeitsstände der VWS-Struktur festgehalten, um auf einem konsistenten Datenstand der VWS gemeinsam mit allen in der Wertschöpfungskette beteiligten Partnern zu arbeiten. Eine Grundlage dazu bietet das Änderungsmanagement, welches als Ergebnis eines Änderungsprozesses eine neue Version hervorbringt.

Änderungsmanagement muss spezielle Herausforderungen adressieren, die sich aus der gleichzeitigen Bearbeitung von VWSen durch verschiedene Unternehmen ergeben. Merge-Konflikte sind potenzielle Probleme, die durch den Abgleich von VWSen unterschiedlicher Arbeitsstände entstehen können. Eine mögliche Lösung besteht darin, dass Änderungen, die nicht vom aktuellen Besitzer der VWS vorgenommen wurden, in einer „Erweiterungs-VWS“ münden, die die aktuelle Version referenziert und eine

eigene Sicht für das Unternehmen darstellt. Der Einsatz von Mechanismen wie Merge- oder Pull-Requests, ähnlich denen bei GIT, kann dabei helfen, Konsens zwischen den Partnern zu finden und Datenkonflikte effizient zu lösen.

Rückverfolgbarkeit ist von zentraler Bedeutung für die Nachverfolgung von Komponenten und Prozessen über die gesamte Wertschöpfungskette hinweg. Zwei Szenarien verdeutlichen, wie Verwaltungsschalen zur lückenlosen Identifizierung von Komponenten und zur Dokumentation der Spezifikationskonformität beitragen können. Durch die Verwendung von VWSen wird die Rückverfolgbarkeit optimiert und die Qualitätssicherung entlang der Lieferkette unterstützt.

Synchronisation gemäß VDI/VDE 2193-1 bietet ein strukturiertes Verfahren für die Verwaltung von Datenänderungen zwischen VWSen. Die Definition der Nachrichtenstruktur und der Fehlerbehandlung gewährleistet eine konsistente und zuverlässige Datenübertragung. So wird gewährleistet, dass alle Partner entlang der Wertkette auf die jeweils aktuellen Daten zugreifen können. Das Synchronisationsprotokoll ermöglicht verschiedene Verarbeitungsarten und bietet eine solide Grundlage für zukünftige Anpassungen.

Zusammenfassend lässt sich sagen, dass die Modularisierung und Verlinkung die technische Basis für die Verwaltung des Leitungssatzes legen, während Versionierung, Änderungsmanagement und Rückverfolgbarkeit wesentliche Komponenten für eine präzise, flexible und nachvollziehbare Prozesskontrolle darstellen. Die Integration dieser Konzepte in eine dezentrale Architektur ist entscheidend, um sowohl die Datenkonsistenz zu gewährleisten als auch die Akzeptanz bei den Stakeholdern zu sichern. Trotz der gut strukturierten Ansätze sind weitere Detailarbeiten und Feinabstimmungen über das Projekt VWS4LS hinaus erforderlich, um die Implementierung der VWS zu vervollständigen und zukünftige Anforderungen auch in anderen Domänen zu erfüllen.

Literaturverzeichnis

- [1] Plattform Industrie 4.0, „Digitale Ökosysteme global gestalten,“ Berlin, 2019.
- [2] B. Ramis-Ferrer, B. Ahmad, A. Lobov, R. Harrison und J. Martinez-Lastra, „Product, process and resource model coupling for knowledge-driven assembly automation,“ 2016. [Online]. Available: <https://doi.org/10.1515/auto-2015-0073>.
- [3] H. Walsh und A. Kunz, „Product, Process, Resource – an Integrated Modeling Approach for Production Engineering and Industrialized Construction,“ Project Production Institute , 2023. [Online]. Available: https://www.researchgate.net/publication/379534390_Product_Process_Resource_-an_Integrated_Modeling_Approach_for_Production_Engineering_and_Industrialized_Construction.
- [4] J. Pfrommer, M. Schleipen und J. Beyerer, „PPRS: Production skills and their relation to product, process, and resource,“ IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA 2013), 2013. [Online]. Available: <https://doi.org/10.1109/ETFA.2013.6648114>.
- [5] Prostep ivip, „Harness Description List (KBL),“ prostep ivip, 26 Jun 2022. [Online]. Available: <https://ecad-wiki.prostep.org/specifications/kbl/>.
- [6] Prostep ivip, „Vehicle Electric Container (VEC),“ prostep ivip, 8 Jan 2024. [Online]. Available: <https://ecad-wiki.prostep.org/specifications/vec/v210/>.
- [7] OPC Foundation, „OPC UA Online Reference - Released Specifications,“ 2024. [Online]. Available: <https://reference.opcfoundation.org/>.
- [8] ECLASS e.V., „ECLASS-Standard,“ [Online]. Available: <https://eclass.eu/eclass-standard/content-suche/search>.
- [9] Platform Industrie 4.0, „RAMI 4.0: Ein Referenzarchitekturmodell als Kommunikationsgrundlage in der Industrie 4.0,“ 11 04 2022. [Online]. Available: <https://www.dke.de/de/arbeitsfelder/industry/rami40>.
- [10 „prostep ivip,“ prostep ivip e.V., [Online]. Available: <https://www.prostep.org/>.
]
- [11 „Verband der Automobilindustrie (VDA),“ [Online]. Available: <https://www.vda.de/de>.
]
- [12 J. Becker, „Whitepaper KBL vs. VEC - Similarities and differences - briefly and concisely summarized,“ 16 December 2022. [Online]. Available: <https://ecad-wiki.prostep.org/post/kbl-vs-vec/>.
- [13 „DIN 72036:2024-06 Straßenfahrzeuge - Automatisierung der Leistungssatzfertigung,“ DIN-Normenausschuss Auto und Mobilität, 06 2024. [Online]. Available: <https://dx.doi.org/10.31030/3521962>.
- [14 IEC, „IEC 61360-4 - IEC/SC 3D - Common Data Dictionary,“ [Online]. Available: <https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TreeFrameset?OpenFrameSet>.

- [15 „DIN SPEC 91345:2016-04 - Referenzarchitekturmodell Industrie 4.0 (RAMI4.0),“ 2016. [Online].
] Available: <https://dx.doi.org/10.31030/2436156>.
- [16 KEBA, „OPC UA – der zentrale Standard für Industrie 4.0 im Überblick,“ [Online]. Available:
] <https://www.keba.com/de/news/industrial-automation/ueberblick-opc-ua-zentraler-standard-industrie-4-0>.
- [17 OPC Foundation, „OPC 40001-1: Machinery Basic Building Blocks,“ [Online]. Available:
] <https://reference.opcfoundation.org/Machinery/v103/docs/>.
- [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 Prostep ivip, „VEC Release Notes - Version 2.1.0,“ prostep ivip, 08 01 2024. [Online]. Available:
] <https://ecad-wiki.prostep.org/specifications/vec/v210/release-notes/>.
- [20 OPC Foundation, „OPC 40001-3: Machinery Job Mgmt,“ OPC Foundation, [Online]. Available:
] <https://reference.opcfoundation.org/Machinery/Jobs/v100/docs/>.
- [21 OPC Foundation, „OPC 40001-101: Machinery Result Transfer,“ [Online]. Available:
] <https://reference.opcfoundation.org/Machinery/Result/v100/docs/>.
- [22 OPC Foundation, „OPC 10031-4: ISA-95-4 Job Control,“ [Online]. Available:
] <https://reference.opcfoundation.org/ISA95JOBCONTROL/v200/docs/>.
- [23 OPC Foundation, „OPC UA Nodesets,“ [Online]. Available: <https://github.com/OPCFoundation/UA-Nodeset>.
- [24 „VWS4LS-Github,“ ARENA2036 e.V., [Online]. Available: <https://github.com/VWS4LS>.
]
- [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.
- [26 Industrial Digital Twin Association e.V., „IDTA 02031-1-0 Bill of Process (WiP),“ [Online]. Available:
] <https://industrialdigitaltwin.org/content-hub/teilmodelle>.
- [27 Industrial Digital Twin Association e.V., „Registrierte IDTA Submodelle,“ [Online]. Available:
] <https://industrialdigitaltwin.org/en/content-hub/submodels>.
- [28 Industrial Digital Twin Association e.V., „IDTA 02006-2-0 Digital Nameplate for Industrial
] Equipment,“ [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Digital%20nameplate/2/0>.
- [29 Industrial Digital Twin Association e.V., „IDTA 02020-1-0 Capability Description,“ [Online].
] Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/development/Capability/1/0>.
- [30 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>.

- [31 „ETL-Prozess,“ [Online]. Available: <https://de.wikipedia.org/wiki/ETL-Prozess>.
]
- [32 Prostep ivip, „ECAD-Wiki,“ 2023. [Online]. Available: <https://ecad-wiki.prostep.org/specifications/vec/v202/component-characteristics/geometric-properties-of-connector-housings-definitions/>. [Zugriff am 01.6.2023].
- [33 OPC Foundation, „OPC 10000-210: Industrial automation - Relative Spatial Location,“ 2023. [Online]. Available: <https://reference.opcfoundation.org/RSL/v100/docs/>.
- [34 ECLASS e.V., „Technical Specification Conceptual Data Model,“ 2020.
]
- [35 C. Diedrich, A. Belyaev, R. Blumenfeld, J. Bock, S. Grimm, J. Hermann, T. Klausmann, A. Köcher, M. Maurmaier, K. Meixner, J. Peschke, M. Schleipen, S. Schmitt, B. Schnebel, G. Stephan, M. Volkmann, A. Wannagat, K. Watson, M. Winter und P. Zimmermann, „Information Model for Capabilities, Skills & Services,“ Plattform Industrie 4.0, Berlin, 2022.
- [36 Plattform Industrie 4.0, „Capabilities, Skills, Services,“ 11.2022. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/CapabilitiesSkillsServices.pdf?__blob=publicationFile&v=1.
- [37 ISO/IEC, „ISO/IEC 19510:2013: Business Process Model and Notation (BPMN),“ [Online]. Available: <https://www.iso.org/standard/62652.html>.
- [38 Plattform Industrie 4.0, „Verwaltungsschale in der Praxis,“ 2021. [Online]. Available: https://industrialdigitaltwin.org/wp-content/uploads/2021/09/08_verwaltungsschale_in_der_praxis_de_2020.pdf.
- [39 Plattform Industrie 4.0, „Beziehungen zwischen I4.0-Komponenten – Verbundkomponenten und intelligente Produktion,“ 2017. [Online]. Available: <https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/beziehungen-i40-komponenten.pdf>.
- [40 Industrial Digital Twin Association e.V., „IDTA 02005 Provision of Simulation Models,“ [Online]. Available: https://github.com/admin-shell-io/submodel-templates/tree/main/published/Provision_of_Simulation_Models/1/0.
- [41 Industrial Digital Twin Association e.V., „IDTA 02020-1-0 Capability Description (WiP),“ [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/development/Capability/1/0>.
- [42 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>.
- [43 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>.
- [44 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>.

- [45] Industrial Digital Twin Association e.V., „IDTA 02010-1-0 Service Request Notification,“ 2023.
] [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published/Service%20Request%20Notification/1/0>.
- [46] VDI/VDE, VDI/VDE 2193 Blatt 2 - Sprache für I4.0-Komponenten - Interaktionsprotokoll für Ausschreibungsverfahren, Bd. VDI/VDE Richtlinien, 2020.
- [47] Plattform Industrie 4.0, „Vertrauensinfrastrukturen,“ 03 2021. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Vertrauensinfrastrukturen.pdf?__blob=publicationFile&v=1.
- [48] Industrial Digital Twin Association e.V., „IDTA 01002-3-0-2: Specification of the Asset Administration Shell Part 2: Application Programming Interfaces,“ June 2024. [Online]. Available: https://industrialdigitaltwin.org/content-hub/aasspecifications/idta_01002-3-0-2_application_programming_interfaces.
- [49] VDI/VDE, VDI/VDE 2193 Blatt 1 - Sprache für I4.0-Komponenten - Struktur von Nachrichten, Bd. VDI/VDE Richtlinien, V. M. u. Automatisierungstechnik, Hrsg., 2020, p. 36.
- [50] M. Mosley und M. Brackett, „The DAMA guide to the data management body of knowledge (DAMA-DMBOK guide),“ Bradley Beach, N.J., 2010.
- [51] M. Fleckenstein und L. Fellows, Modern Data Strategy, S. Cham, Hrsg., Cham: Springer International Publishing, 2018, p. 263.
- [52] A. M. Hosseini, T. Sauter und W. Kastner, „Towards Adding Safety and Security Properties to the Industry 4.0 Asset Administration Shell,“ in *17th IEEE International Conference on Factory Communication Systems (WFCS)*, Linz, Austria, 2021.
- [53] ISO/IEC, „ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements“.
- [54] A. B. H. T. L. W. C. D. Andre Bröring, „Secure Asset Administration Shell exchange with Distributed Ledger Technology,“ in *Plattform Industrie 4.0*, 2021.
- [55] A. Broring, M. Ehrlich, L. Wisniewski, H. Trsek und S. Heiss, „Towards an Asset Administration Shell Integrity Verification Scheme,“ in *IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Stuttgart, 2022.
- [56] A. M. Hosseini, T. Sauter und W. Kastner, „A Safety and Security Reference Architecture for Asset Administration Shell Design,“ in *IEEE 18th International Conference on Factory Communication Systems (WFCS)*, Pavia, Italy, 2022.
- [57] ISO, „ISO/IEC 27002:2022-02: Information security, cybersecurity and privacy protection - Information security controls,“ 2022. [Online]. Available: www.iso.org/standard/75652.html.
- [58] „OPC UA Online Reference - Released Specifications,“ 2024. [Online]. Available: <https://reference.opcfoundation.org/>.
- [59] 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.

- [60] K. Fraczek und M. Plechawska-Wojcik, „Comparative Analysis of Relational and Non-relational Databases in the Context of Performance in Web Applications,“ in *Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation*, 2017.
- [61] M. Angos-Mediavilla, M. Gorenzweig, L. Beil, C. Kosel, A. Pomp, M. Freund und T. Meisen, „Advancing Industry 4.0: Integrating Data Governance into Asset Administration Shell for Enhanced Interoperability,“ in *International Conference on Enterprise Information Systems*, 2024.
- [62] M. Angos-Mediavilla, M. Gorenzweig, L. Beil, C. Kosel, A. Pomp, M. Freund und T. Meisen, „Enabling Value Chain Interoperability: Integrating and evaluating robust Data Governance aspects into the Asset Administration Shell,“ in *Springer Lecture Notes in Computer Science*, 2025.
- [63] „Monetization of Data at the Example of Manufacturing Machines,“ Springer, Berlin, Heidelberg, 2023.
- [64] R. Klostermeier, S. Haag und A. Benlian, „Digitale Zwillinge – Eine explorative Fallstudie zur Untersuchung von Geschäftsmodellen,“ Bd. HMD Praxis der Wirtschaftsinformatik, Nr. 55, 4 2018.
- [65] 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>.
- [66] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller und K. Scarfone, „Guide to Attribute Based Access Control (ABAC) Definition and Considerations,“ National Institute of Standards and Technology Special Publication, Gaithersburg, 2014.
- [67] 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>.
- [68] 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>.
- [69] Industrial Digital Twin Association e.V., „IDTA 02003-1-2: Generic Frame for Technical Data for Industrial Equipment in Manufacturing,“ [Online]. Available: https://github.com/admin-shell-io/submodel-templates/tree/main/published/Technical_Data/1/2.
- [70] 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>.
- [71] M. Stolze, „Realizing automated production planning via proactive AAS and Business Process Models,“ in *ETFA*, Padua, 2024.
- [72] M. Stolze, „Entwurf komplexer Automatisierungssysteme: Konzepte zur Realisierung proaktiver Verwaltungsschalen und deren Kommunikation,“ in *EKA*, 2024.
- [73] „Single Point of Truth,“ [Online]. Available: https://de.wikipedia.org/wiki/Single_Point_of_Truth.
- [74] Industrial Digital Twin Association e.V., „Decentralized Registries: Taxonomy of decentralized registries and an architectural overview,“ June 2023. [Online]. Available:

https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2023/06/Decentralized-Registries-Taxonomy-of-decentralized-registries-and-an-architectural-overview_.pdf.

- [75] A. Bröring, M. Ehrlich, L. Wisniewski, H. Trsek und S. Heiss, „An Asset Administration Shell Version Control to Enforce Integrity Protection,“ 2022. [Online]. Available: <https://doi.org/10.25644/a4ws-9a49>.
- [76] VDI 2860, „Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole,“ Beuth Verlag, Berlin, 1990.
- [77] M. Fechter, C. Seeber und S. Chen, „Integrated Process Planning and Resource Allocation for Collaborative Robot Workplace Design,“ 0, 2018.
- [78] Plattform Industrie 4.0, „Interoperability at Runtime - Exchanging Information via Application Programming Interfaces,“ Plattform Industrie 4.0, Berlin, 2021.
- [79] M. Redeker, S. Volgmann, F. Pethig und J. Kalhoff, „Towards Data Sovereignty of Asset Administration Shells across Value Added Chains,“ in *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020.

Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1-1: Matrixstruktur der Teilprojekte in VWS4LS..... | 10 |
| Abbildung 1-2: Projektteilnehmer VWS4LS (Fettgedruckt → TP-Leitung)..... | 11 |
| Abbildung 1-3 Initiale (rot) und angepasste Planung (grün) der Teilprojekte und Meilensteine | 13 |
| Abbildung 1-4: Teilprojekttreffen 2022 | 16 |
| Abbildung 1-5 Teilprojekttreffen 2023 | 17 |
| Abbildung 1-6 Teilprojekttreffen 2024 | 17 |
| Abbildung 1-7: Leitbild Plattform Industrie 4.0. (Quelle: Plattform Industrie 4.0, 2019) | 18 |
| Abbildung 2-1: Arbeitsdokument Anforderungssammlung (Auszug) | 21 |
| Abbildung 2-2: PPR-Modell | 23 |
| Abbildung 2-3: KBL & VEC | 25 |
| Abbildung 2-4 Vergleich Scope VEC & KBL [12] | 25 |
| Abbildung 2-5: ECLASS-Klassifizierungen für KFZ-Bordnetze [8]..... | 28 |
| Abbildung 2-6: ECLASS-Klassifizierungen für Produktionsmaschinen [8]..... | 28 |
| Abbildung 2-7: OPC UA Technologie (Quelle: VDMA) | 29 |
| Abbildung 2-8: Leitungssatzarchitektur | 31 |
| Abbildung 2-9: WireHarness Data Domains..... | 33 |
| Abbildung 2-10: Zusammenhänge im Produktionsdatensatz [18]..... | 34 |
| Abbildung 2-11: Nutzung des VEC-Modells | 35 |
| Abbildung 2-12: Beispiel VEC Terminal Datamodell (Quelle: VEC Recommendation V2.1)..... | 36 |
| Abbildung 2-13: Feature Requests an den VEC | 37 |
| Abbildung 2-14: Relevante OPC UA Spezifikationen..... | 37 |
| Abbildung 2-15: MachineldentificationType. (Quelle: OPC UA for Machinery [17])..... | 38 |
| Abbildung 2-16: MachineryItemState. (Quelle: OPC UA for Machinery [17]) | 39 |
| Abbildung 2-17: Relevante OPC UA Informationsmodelle [17] [20] [21] [22] | 40 |
| Abbildung 2-18: OPC40570: Wire Harness Manufacturing..... | 43 |
| Abbildung 2-19: VWS4LS Datenmodell | 45 |
| Abbildung 2-20: Im Rahmen von VWS4LS erstellte neue Standards | 46 |
| Abbildung 3-1: Kollaborationsbeziehungen in der Leitungssatz-Entwicklung..... | 48 |
| Abbildung 3-2: Eingangs- und Ausgangsdaten im Entwicklungsprozess im Überblick | 49 |
| Abbildung 3-3: Anfrage beim Komponentenhersteller stellen | 50 |
| Abbildung 3-4: Auskunft auf Anfrage..... | 50 |
| Abbildung 3-5: 2. Anfrage mit Detail-Anforderungen..... | 51 |
| Abbildung 3-6: Beispiel für zentralen Ansatz..... | 52 |
| Abbildung 3-7: Freigaben und Änderungs-Management beim dezentralen Ansatz | 53 |
| Abbildung 3-8: SPoT-Bezeichnung beim dezentralen Einsatz..... | 53 |
| Abbildung 3-9: Beispiel für einen Stecker, der an zwei verschiedenen Standorten verwendet wird | 54 |
| Abbildung 3-10: Freigaben und Änderungs-Management | 54 |
| Abbildung 3-11: Überblick Teilprozesse der Leitungssatzentwicklung | 55 |
| Abbildung 3-12: Referenzprozess der Leitungssatz-Entwicklung | 56 |
| Abbildung 3-13: Workflow Entwicklung Leitungssatz | 57 |
| Abbildung 3-14: Assets und ihre VWS im Entwicklungsprozess des Leitungssatzes..... | 60 |
| Abbildung 3-15: Ausschnitt aus der Anforderungsanalyse für VWS und deren Teilmodelle | 61 |
| Abbildung 3-16: VWS-Schnittstellen im Entwicklungsprozess zwischen OEM und Konfektionär | 62 |
| Abbildung 3-17: Beispieleitungssatz aus dem PPR-Workshop | 63 |
| Abbildung 3-18: SVG-Darstellung des Leitungssatzes | 64 |
| Abbildung 3-19: WIRE und BOM mit dargestellter Modularisierung (V1, V2 und V3) | 64 |
| Abbildung 3-20: MLK 1,2 Terminal mit/ohne ELA (KOSTAL Kontakt Systeme GmbH & Co. KG) | 65 |
| Abbildung 3-21: Auszug aus einer Typ-Verwaltungsschale Terminal | 66 |
| Abbildung 3-22: Äquivalenzbeziehungen per „specificAssetId“ | 67 |
| Abbildung 3-23: OEM-Verwaltungsschale für den Gesamtleitungssatz | 68 |
| Abbildung 3-24: Verknüpfung Komponenten-Verwendung mit Komponenten-Typ per GlobalAssetId . | 68 |
| Abbildung 3-25: Verwaltungsschale für ein Halbfabrikat "Geschnittene Leitung" | 69 |
| <i>Abbildung 4-1: Phasenmodell für die Anforderungsspezifikation.....</i> | 71 |
| <i>Abbildung 4-2: High-Level Architektur.....</i> | 85 |
| <i>Abbildung 4-3: Use Case Predictive Maintenance.....</i> | 86 |

| | |
|---|-----|
| Abbildung 4-4: Pay-per-Use: Maschinenhersteller greift auf Ressource AAS bei Kunde zu | 87 |
| Abbildung 4-5: OEE-Berechnung: MES greift auf Ressource AAS zu | 88 |
| Abbildung 4-6: Palette mit zu bestückenden Gehäusen | 89 |
| Abbildung 4-7: Gehäuse | 90 |
| Abbildung 4-8: Kontaktbeispiel 1 (Y-Achse zeigt in Richtung Primärverriegelung) | 91 |
| Abbildung 4-9: Definition des Nullpunkts für das Terminal und gegenüber dem Gehäuse | 91 |
| Abbildung 4-10: Stiftkontakt | 91 |
| Abbildung 4-11: Kontaktbeispiele mit zwei Verriegelungen | 92 |
| Abbildung 4-12: VEC-Definition für Steckrichtung und Achsensystem (Source Prostep) | 92 |
| Abbildung 4-13: Stiftgehäuse mit Pinbild | 94 |
| Abbildung 4-14: Buchsengehäuse mit Pinbild | 94 |
| Abbildung 4-15: Produktionsprozesse beim Konfektionär | 96 |
| Abbildung 4-16: PPR-Modell | 97 |
| Abbildung 4-17: Initiale Prozessliste | 98 |
| Abbildung 4-18: UML-Klassendiagramm für Produktionsprozesse | 98 |
| Abbildung 4-19: Allgemeingültige Klassen | 99 |
| Abbildung 4-20: ECLASS Datenmodell und Detaillierung des Properties (Quelle: ECLASS [34]) | 100 |
| Abbildung 4-21 Entwicklung der Darstellung zu den Prozessen | 101 |
| Abbildung 4-22 Erweiterte Properties zu den Prozessparametern | 101 |
| Abbildung 4-23 Einheiten mit den dazugehörigen ECLASS IRDIs | 102 |
| Abbildung 4-24 Common-Bereich der finalen Prozessliste | 102 |
| Abbildung 4-25: VWS ProductType | 104 |
| Abbildung 4-26: SMC Product Sequence | 104 |
| Abbildung 4-27: Nominalprozess Cut | 105 |
| Abbildung 4-28: SMC PPR, ProcessBom | 105 |
| Abbildung 4-29: Required Capabilities | 106 |
| Abbildung 4-30: VWS ProductionOrder | 106 |
| Abbildung 4-31: IDTA 2031 Bill of Process | 107 |
| Abbildung 4-32: CSS-Modell [36] | 108 |
| Abbildung 4-33: Beispielprodukt zur Ableitung der Required Capabilities | 109 |
| Abbildung 4-34: Prozesssequenz | 111 |
| Abbildung 4-35: Beispiel der Sicherungsbox | 112 |
| Abbildung 4-36: Simple Prozessequenz zum Schraubvorgang an zwei Positionen | 113 |
| Abbildung 4-37: Prozessequenz am Beispiel Sicherungsbox | 113 |
| Abbildung 4-38: Darstellung des Prozessbeispiels als BPMN | 114 |
| Abbildung 4-39: Prozessequenz am Beispiel Sicherungsbox inkl. Ressourcenzuteilung | 114 |
| Abbildung 4-40: Darstellung eines Prozessablaufs als BPMN | 115 |
| Abbildung 4-41: Exemplarische Darstellung des Data Mapping Services mit VWS als Zielsystem ... | 116 |
| Abbildung 4-42: AAS Resource mit AAS und SM | 118 |
| Abbildung 4-43: AAS und SM und SMC InformationModelDescriptionSetJSONSchema | 118 |
| Abbildung 4-44: AAS und SM und SMC InformationModelDescriptionSetAASElement | 119 |
| Abbildung 4-45: Keine Datensenkenzuordnung im Transformer | 121 |
| Abbildung 4-46: Mit Datensenkenzuordnung | 122 |
| Abbildung 4-47: Mit Datensenkenzuordnung im Transformer | 122 |
| Abbildung 4-48: Mögliche Datenmapping-Integration | 122 |
| Abbildung 4-49: Übersicht der Produktionsschritte in der Leistungssatzproduktion | 123 |
| Abbildung 4-50: Typ- und Instanzen-AAS | 124 |
| Abbildung 4-51: Baugruppe einer geschnittenen Einzelleitung mit Kontaktteilen inklusive ELA | 125 |
| Abbildung 4-52: Struktur eines Produktionsauftrags. (Eigene Darstellung) | 126 |
| Abbildung 4-53: Submodel ProductionOrder | 127 |
| Abbildung 4-54: AAS & Submodel Batch | 127 |
| Abbildung 4-55: AAS Lot / Submodel Lot & Executed Processes | 127 |
| Abbildung 4-56: AAS & Submodel Lot | 128 |
| Abbildung 4-57: Überwachung des Schraubprozesses mit parallel abgearbeiteten Aufgaben | 129 |
| Abbildung 4-58: Architektur der Prozesssteuerung | 130 |
| Abbildung 4-59: ProcessFactory Sequenzdiagramm | 131 |
| Abbildung 4-60: WorkerManager Sequenzdiagramm | 131 |

| | |
|---|-----|
| Abbildung 4-61: Testszenario Validation Service | 132 |
| Abbildung 4-62: Repräsentation des Kabelherstellungsprozesses | 133 |
| Abbildung 4-63: Repräsentation des Kabelherstellungsprozesses in der VWS | 134 |
| Abbildung 4-66: Architektur für die Konzeptvalidierung | 135 |
| Abbildung 4-65: Darstellung des Produktionsauftrags in BaSyx | 136 |
| Abbildung 4-66: Darstellung des Batchs in BaSyx | 136 |
| Abbildung 4-67: Darstellung des Lots in BaSyx | 136 |
| Abbildung 4-68: Darstellung des Lots zur Kabelproduktion | 137 |
| Abbildung 4-69: Darstellung des Lots zur Silikonproduktion | 137 |
| <i>Abbildung 4-70: Repräsentation des Kabelherstellungsprozesses</i> | 138 |
| Abbildung 4-71: Repräsentation des PPS-Fertigungsauftrages | 139 |
| Abbildung 4-72: Mapping der Daten eines beispielhaften Fertigungsauftrags in die VWS | 140 |
| <i>Abbildung 4-73: Repräsentation des Steckers (Terminal) der Verwaltungsschale</i> | 140 |
| Abbildung 5-1 Maße der Packtasche eines Leitungssatzes | 148 |
| Abbildung 5-2 Maße der Karosserie (S-Klasse) | 149 |
| <i>Abbildung 5-3 Montagezelle für das Einbringen des Leitungssatzes</i> | 150 |
| Abbildung 5-4 Greifer zum Greifen der Transporttaschen | 151 |
| Abbildung 5-5: Hauptkoordinatensystem der Montagezelle | 151 |
| Abbildung 5-6: Idee: Fördergreifer als mögliche Lösung | 155 |
| Abbildung 5-7: Kabeldichtungen (Quelle: https://www.woco-kkw.com) | 156 |
| Abbildung 5-8: Dichtungselement mit axialer Einbaurichtung | 157 |
| Abbildung 5-9: Radiale Einbaurichtung | 158 |
| Abbildung 5-10: Clip mit Bolzenaufnahme | 159 |
| Abbildung 5-11: Kantenbefestiger | 159 |
| Abbildung 5-12: Abbildung eines Kabelkanals mit Hilfselementen (Quelle: Dräxlmaier) | 161 |
| Abbildung 5-13: Konstruktionsmerkmale von Steckverbindungen | 162 |
| Abbildung 5-14: Steckverbindung mit Rastnase | 162 |
| Abbildung 5-15: Mini-Fakra Steckverbinder System | 164 |
| Abbildung 5-16: Funktionsweise CPA-Vorrichtung | 164 |
| Abbildung 6-1: Ergebnis der Workshop-Reihe | 168 |
| Abbildung 6-2: Unterschiedliche Arten von Verwaltungsschalen als Teil der VBK „Produkt“ | 177 |
| Abbildung 6-3: Umsetzung hierarchischer Beziehungen mit Hilfe des BOM-Teilmodells | 192 |
| Abbildung 6-4: Konzept der "Specific Asset ID" aus dem Metamodell der VWS (s. Abschnitt 5.3.4). | 193 |
| Abbildung 6-5: Exemplarische Umsetzung des Sachnummern-Mapping mittels „specificAssetId“ | 194 |
| Abbildung 6-6: Teilmodell "Freigabeliste" | 195 |
| Abbildung 6-7: Beispiel-Ressourcen 2 – Wezag UP 150 (links) und Komax Sigma 688 (rechts) | 196 |
| Abbildung 6-8: Übersicht möglicher Fähigkeiten Komax Sigma 688 | 197 |
| Abbildung 6-9: Einordnung in das PPR-Modell | 198 |
| Abbildung 6-10: Modellierung von Werkzeug-Freigaben am Beispiel Crimp-Kontakt | 199 |
| Abbildung 6-11: Abbildung der aktuellen Maschinenstruktur mit Hilfe des BOM-Teilmodells | 199 |
| Abbildung 6-12: Abbildung potenzieller Maschinenstrukturen mit Hilfe des BOM-Teilmodells | 200 |
| Abbildung 6-13: Modellierung potenzieller Maschinenstrukturen per "specificAssetId" | 200 |
| Abbildung 6-14: Prüfmodul mit Gegenstecker (Komax Testing) | 202 |
| Abbildung 6-15: „specificAssetId“ einer pneumatischen Komponente im Prüfmodul | 202 |
| Abbildung 6-16: Verwaltungsschale einer pneumatischen Antriebseinheit mit Bill of Material | 203 |
| Abbildung 6-17: Netzwerk aus FMU-Modellen für eine pneumatische Antriebseinheit (rechts) | 204 |
| Abbildung 6-18: Eingesetzte VIBN-Tools | 205 |
| Abbildung 6-19: Simulationsmodelle in der Verwaltungsschale des Prüfmoduls | 205 |
| Abbildung 6-20: Ausschnitt der erstellten Verwaltungsschalen | 206 |
| Abbildung 6-21: Exemplarische Zeichnung Leitungssatz inkl. Komponentenbezeichnungen | 207 |
| Abbildung 6-22: Exemplarische OEM-Komponentenspezifikation inkl. Sachnummern | 209 |
| Abbildung 6-23 Capability-Matching am Beispiel eines Crimp-Prozesses | 210 |
| Abbildung 6-24: Exemplarisches Teilmodell "RequiredCapabilities" | 212 |
| Abbildung 6-25: Exemplarisches Teilmodell "OfferedCapabilities" | 213 |
| Abbildung 6-26: Konzept zur Modellierung zusammengesetzter Fähigkeiten | 214 |
| Abbildung 6-27: Bedingung "RequiresToolCondition" | 215 |
| Abbildung 6-28: Beschreibung von Werkzeugtypen per "toolType"-Extension | 215 |

| | |
|--|-----|
| Abbildung 6-29: Zusammenhang zwischen Fähigkeit (Capability) und Fertigkeit (Skill) | 217 |
| Abbildung 6-30: Prototypische Ergänzung des AID-Teilmodells [42] für OPC UA | 218 |
| Abbildung 6-31: Konzept Use Case "Predictive Maintenance" | 224 |
| Abbildung 6-32: Zusammenhang der verschiedenen Teilmodelle | 226 |
| Abbildung 6-33: NodeRed-Flow zur Auswertung des Teilmodells "Predictive Maintenance" | 227 |
| Abbildung 7-1: Verhandlungsszenario 1 „Order Driven Production“ | 230 |
| Abbildung 7-2: Nutzung der I4.0-Nachricht für Synchronisation von Änderungen an Teilmodellen ... | 233 |
| Abbildung 7-3: Horizontale Kommunikation zwischen zwei Verwaltungsschalen vom Typ 3 | 234 |
| Abbildung 7-4: Ausschreibungsverfahren zwischen Auftraggeber und Auftragnehmer..... | 235 |
| Abbildung 7-5: Beispielimplementierung für einen Algorithmus..... | 235 |
| Abbildung 7-6: Start des Verhandlungsprozesses über die AAS-Weboberfläche | 237 |
| Abbildung 7-7: Prozessmodelle in der Flowable-Weboberfläche..... | 238 |
| Abbildung 7-8: NodeRed-Flow "startBiddingProcess" | 239 |
| Abbildung 7-9: Darstellung der Workflow-Instanz mit Variablen in Flowable..... | 240 |
| Abbildung 7-10: Zeit- und Signalereignis im Workflow des Service Requesters | 241 |
| Abbildung 7-11: Unter-Workflow für eingehende Angebots-Nachricht..... | 241 |
| Abbildung 7-12: Mapping von I4.0-Nachricht zu BPMN-Ereignissen in NodeRed..... | 242 |
| Abbildung 7-13: Erweiterter Fähigkeitenabgleich..... | 244 |
| Abbildung 7-14 VWS-Teilmodelle für Service-Requester und Service-Provider- | 251 |
| Abbildung 7-15: Architektur für die VWS Typ 3 ("Proaktive VWS")..... | 253 |
| Abbildung 8-1:Überblick über den systematischen Prozess der Literaturrecherche | 263 |
| Abbildung 8-2: Überblick über die Superklasse „UserInformation“ | 265 |
| Abbildung 8-3:Überblick über die Superklasse „Authorizable“..... | 266 |
| Abbildung 8-4: Übersicht über die Integration der identifizierten Data-Governance-Aspekte in das bestehende Metamodell der Verwaltungsschale (Erweiterungen fettgedruckt)..... | 267 |
| Abbildung 8-5: High-Level Architektur unternehmensübergreifender Datenaustauschs mittels IDS.. | 268 |
| Abbildung 8-6: Übermittlung von non-operational data | 268 |
| Abbildung 8-7: Übermittlung von operational data | 269 |
| Abbildung 8-8: Konzept VWS-Security | 269 |
| Abbildung 8-9: Definition des Kabelbaums seitens des OEM..... | 270 |
| Abbildung 8-10: Framework zur Datenmonetarisierung..... | 271 |
| Abbildung 8-11: Einfluss digitaler Zwillinge auf den Business Model Canvas [64] | 272 |
| Abbildung 8-12: Ablauf des automatisierten Angebotsverhandlungsprozesses | 274 |
| Abbildung 8-13: Demonstrator-Architektur mit Catena-X und BaSyx | 275 |
| Abbildung 9-1: Vereinfachte Darstellung des Leitungssatzes | 277 |
| Abbildung 9-2: Beispielhafte Darstellung der Wertschöpfungskette eines Leitungssatzes | 278 |
| Abbildung 9-3: Prozessdiagramm Use Case | 280 |
| Abbildung 9-4: Benötigte Services eines Data Providers..... | 282 |
| Abbildung 9-5: ARENA-X-VWS4LS-Architektur | 284 |
| Abbildung 9-6: Detail-Architektur | 285 |
| Abbildung 9-7: Erzeugte Security-Dateien zur Registrierung in nginx | 288 |
| Abbildung 9-8: Identity Provider | 293 |
| Abbildung 9-9: Anlegen User..... | 294 |
| Abbildung 9-10: Anpassung Sub-Identity-Provider | 301 |
| Abbildung 9-11: Anpassung Redirect-URLs des Portal-Users..... | 301 |
| Abbildung 9-12: Hinzufügen Rollen | 302 |
| Abbildung 9-13: BaSyx-Integration in Use Case | 308 |
| Abbildung 9-14: Grafische Darstellung des Aspektmodells PartTypeInformation.ttl..... | 310 |
| Abbildung 9-15: Eigenschaften eines Submodells als JSON | 311 |
| Abbildung 9-16: Tractus-X Digital Twin Registry - Asset Administration Shell Domain Model | 312 |
| Abbildung 9-17: Die Anforderungen des OEM als AAS mit Submodel | 313 |
| Abbildung 9-18: Zusammenbau Leitungssatz mit Stückliste | 314 |
| Abbildung 9-19: AAS-Security (Quelle: https://v3.admin-shell.io.com/) | 316 |
| Abbildung 9-20: Grundsätzliche Interaktion zwischen Konnektoren und Verwaltungsschalen | 316 |
| Abbildung 9-21: Dataflow RBAC in BaSyx | 317 |
| Abbildung 9-22: Zugriffskontrolle – Konzept für BaSyx..... | 317 |
| Abbildung 9-23: Data Provisioning | 319 |

| | |
|--|-----|
| Abbildung 9-24: Data Consumption | 323 |
| Abbildung 9-25: Darstellung „Bills of Material“ in BaSyx | 333 |
| Abbildung 9-26: AASX Package Explorer | 336 |
| Abbildung 9-27: Integration des Catena-X Submodels "PartTypeInformation" | 336 |
| Abbildung 9-28: BaSyx-Frontend zeigt ProductID in AAS MES_Info (Screenshot) | 337 |
| Abbildung 9-29: BaSyx-Frontend zeigt Maintenance_Counter von Machine_1 in AAS MES_Info | 337 |
| Abbildung 9-30: Interaktion Tractus-X und AAS via EDC | 340 |
| Abbildung 10-1: Zeitplan TP9 | 341 |
| Abbildung 10-2 Physische Leitungssatzpräsentation inkl. Produktverwaltungsschalen (Bildschirme) | 342 |
| Abbildung 10-3 Beispiel VWS-Terminal | 342 |
| Abbildung 10-4 Projektdemonstrator mit dem Fokus auf der Implementierung der VWS in ein Produktionsprozess | 343 |
| Abbildung 10-5 Bestandteile Demonstrator und Informationsfluss | 344 |
| Abbildung 10-6 Aufbau des Demonstrators | 345 |
| Abbildung 10-7 Schematischer Prozessablauf | 346 |
| Abbildung 10-8 Verhandlungsprozess zwischen Service Requester und Provider | 347 |
| Abbildung 11-1: Transferkontext | 349 |
| Abbildung 11-2: Transfer-X Lernmodule: Beiträge von VWS4LS | 350 |
| Abbildung 11-3 Darstellung der Konferenzveranstaltung vom 23.05.2024 mit Folien und Aufnahme | 351 |
| Abbildung 11-4: VWS4LS auf Github | 354 |
| Abbildung 12-1: Big Picture Workshop - Fragen | 356 |
| Abbildung 12-2: Zentrale VWS-Architektur | 357 |
| Abbildung 12-3: Dezentrale VWS-Architektur | 357 |
| Abbildung 12-4: Wechselnder SPoT im Lebenszyklus des Leitungssatzes | 358 |
| Abbildung 12-5: Network of Truth | 358 |
| Abbildung 12-6: Ermittlung der Assets und Nicht-Assets mit Stakeholderzuweisung | 360 |
| Abbildung 12-7: Modularisierung des Leitungssatzes | 361 |
| Abbildung 12-8: Übersicht über VWS der jeweiligen Stakeholder im Entwicklungsprozess | 362 |
| Abbildung 12-9: Übersicht über VWS der jeweiligen Stakeholder im Produktionsprozess | 362 |
| Abbildung 12-10: VWS und deren Verknüpfung für die Ressource | 363 |
| Abbildung 12-11: VWS und deren Verknüpfung für die Komponente im Entwicklungsprozess | 365 |
| Abbildung 12-12: VWS und deren Verknüpfung für die Komponente im Produktionsprozess | 366 |
| Abbildung 12-13: VWS und deren Verknüpfung für den Leitungssatz im Entwicklungsprozess | 367 |
| Abbildung 12-14: VWS und deren Verknüpfung für den Leitungssatz im Produktionsprozess | 367 |
| Abbildung 12-15: VWS und deren Verknüpfung für den Leitungssatz im Montageprozess | 368 |
| Abbildung 12-16: Typ-VWS des Leitungssatzes aus Sicht des OEM | 370 |
| Abbildung 12-17: Teilmodell "ProductSpecification" | 371 |
| Abbildung 12-18: Teilmodell "ProductBoM" | 371 |
| Abbildung 12-19: Teilmodell "ProductConfigurationBoM" | 372 |
| Abbildung 12-20: Typ-VWS des Leitungssatzes aus Sicht des Konfektionärs | 372 |
| Abbildung 12-21: Typ-VWS des Leitungssatzes aus Sicht des Konfektionärs | 374 |
| Abbildung 12-22: Teilmodell "ManufacturingBoM" | 375 |
| Abbildung 12-23: Beziehungen "ManufacturingConfigurationBoM" und "ManufacturingBoM" | 375 |
| Abbildung 12-24: Verlinkung mit dem Teilmodell "ProductBoM" | 376 |
| Abbildung 12-25: Verlinkung Typ/Instanz-VWS beim OEM | 377 |
| Abbildung 12-26: Typ- und Instanz-VWS für den Leitungssatz | 378 |
| Abbildung 12-27: VEC-Datei als File-Element in der Verwaltungsschale | 381 |
| Abbildung 12-28: Entity-Element einer Komponente | 381 |
| Abbildung 12-29: Relationship-Element mit FragmentReference in VEC-Datei | 382 |
| Abbildung 12-30: Entwicklungsversionen und korrespondierende Change-Sets in der Produktion .. | 385 |
| Abbildung 12-31: Vorgeschlagenes Versions-ID-Konzept | 385 |
| Abbildung 12-32: Beispiel versionierter Verwaltungsschalen (Teilmodell) | 386 |
| Abbildung 12-33: "Latest" default | 387 |
| Abbildung 12-34: Zugriff mit Discovery und Registry | 388 |
| Abbildung 12-35: Zugriff ohne Discovery | 388 |
| Abbildung 12-36: Zugriff ohne Registry | 389 |
| Abbildung 12-37: Änderungsmanagement: Änderungsbedarf vom OEM | 399 |

| | |
|---|-----|
| Abbildung 12-38: Änderungsmanagement: Änderungsbedarf vom Tier X | 401 |
| Abbildung 12-39: ÄM_Konfigurationsobjekt)..... | 402 |
| Abbildung 12-40: VWS-Änderungsmanagement: Teilmodell EngineeringChangeRequests | 402 |
| Abbildung 12-41: VWS-Änderungsmanagement: Teilmodell EngineeringChangeProposals | 403 |
| Abbildung 12-42: VWS-Änderungsmanagement: Teilmodell EngineeringChangeOrders..... | 404 |
| Abbildung 12-43: VWS-Änderungsmanagement: Teilmodell Bezugskonfiguration | 404 |
| Abbildung 12-44: Rückverfolgbarkeit über mehrere Tier-Stufen..... | 405 |
| Abbildung 12-45: Rückverfolgbarkeit beim Kabelkonfektionär..... | 406 |
| Abbildung 12-46: Teilmodell "Data Retention Policies" | 407 |

Tabellenverzeichnis

| | |
|--|-----|
| Tabelle 1-1 Anwendungsfälle bzw. Use Cases von VWS4LS | 11 |
| Tabelle 4-1 Vergleich zwischen initialer und finaler Prozessliste | 103 |
| <i>Tabelle 5-1 Bilder aus dem Onlinekatalog von Hellermann Tyton</i> | 159 |
| Tabelle 7-1: Aufbau der Synchronisationsnachrichten | 250 |
| Tabelle 7-2: Zusätzliche Merkmale für VWS4LS-I4.0-Nachrichten zu Ausschreibung und Angebot . | 252 |
| Tabelle 9-1: Benötigte Komponenten für Data Provisioning | 281 |
| Tabelle 9-2: Zusätzliche Services (Core- und Onboarding Services) | 283 |
| Tabelle 9-3: Liste der BPNLs..... | 292 |
| Tabelle 9-4: Seeding Data Authentifizierung | 293 |
| Tabelle 12-1 Architektonische Grundfragestellungen | 359 |

Abkürzungsverzeichnis

| | |
|------------------------------------|--|
| ABAC..... | <i>Attribute-Based-Acces-Control</i> |
| AMQP | <i>Advanced Message Queueing Protocol</i> |
| AP44, <i>Arbeitspaket</i> | |
| API..... | <i>Application Programming Interface, Programmierschnittstelle</i> |
| BoM..... | <i>Bill of Material</i> |
| BPMN | <i>Business Process Model and Notation</i> |
| BPN..... | Business Partner Number |
| CAD..... | <i>Computer Aided Design</i> |
| CCB | Change Control Board |
| <i>Configuration BoM</i> | <i>Configuration Bill of Material</i> |
| DSP | <i>Dataspace Protocol</i> |
| DTR..... | Digital Twin Registry |
| ECO..... | <i>Engineering Change Order</i> |
| ECP | <i>Engineering Change Proposal</i> |
| ECR | <i>Engineering Change Request</i> |
| EDC..... | <i>Eclipse Dataspace-Connector</i> |
| ERP | <i>Enterprise Resource Planning</i> |
| FIN | <i>Fahrzeugsidentifikationsnummer</i> |
| Fzg-Freigabezustand..... | <i>Fahrzeugfreigabezustand</i> |
| GVB..... | <i>Gesamtvorhabenbeschreibung</i> |
| HTTP | <i>Hypertext Transfer Protocol</i> |
| IDSA..... | <i>International Data Spaces Association</i> |
| IDTA..... | <i>Industrial Digital Twin Association</i> |
| KBL..... | <i>Kabelbaumliste</i> |
| KIT..... | <i>Keep It Together</i> |
| LS-Komponente | <i>Leitungssatz-Komponenten</i> |
| MQTT..... | <i>Message Queuing Telemetry Transport</i> |
| OEM..... | Original Equipment Manufacturer |
| OPC-UA | <i>Open Plattform Communications Unified Architecture</i> |
| PCF..... | <i>Product Carbon Footprint</i> |
| PDM..... | <i>Product Data Management</i> |
| PPR-Modell..... | <i>Produkt-Prozess-Ressource-Modell</i> |
| <i>Product BoM</i> | <i>Product Bill of Material</i> |
| RAMI 4.0 | Reference Architecture Model for Industry 4.0 |
| REST..... | <i>Representational State Transfer</i> |
| SDK | <i>Software Development Kit</i> |
| SPoT..... | <i>Single Point of Truth</i> |
| SPS..... | <i>Speicherprogrammierbare Steuerung</i> |
| STEP..... | <i>Standard for the Exchange of Product model data</i> |
| TP <i>Teilprojekt, Teilprojekt</i> | |
| VDMA | 33 |
| VEC | <i>Vehicle Electric Container</i> |
| VWS..... | <i>Verwaltungsschale</i> |

Autorenverzeichnis

An diesem Projekt haben mitgewirkt:

| | | |
|---|--|---|
|  | Name | Christian Kosel |
| | Organisation | ARENA2036 e.V. |
| | Funktion | Projekt Manager |
| | Über den Autor | |
|  | Name | |
| | Organisation | |
| | Funktion | |
| | Über den Autor | |
|  | Name | Bernd Kärcher |
| | Organisation | ARENA2036 e.V. |
| | Funktion | Projektoffice und Ergebniscontrolling |
| | Über den Autor | |
|  | Name | Markus Rentschler |
| | Organisation | ARENA2036 e.V. |
| | Funktion | Forschungskoordinator Digitale Interoperabilität |
| | <p>Markus Rentschler kam in der Schlussphase zum Projekt VWS4LS und hat sich vorwiegend in der Zusammenstellung und Aufbereitung der Dokumentation engagiert.</p> <p>Sein beruflicher Werdegang kann auf LinkedIn eingesehen werden.</p> | |
|  | Name | Dr. Alexander Salinas Segura |
| | Organisation | DRÄXLMAIER Group |
| | Funktion | Manager Digital Business |

| | | | | | | | | | |
|---|---|------|------------------|--------------|----------------------------|----------|---|-----------------------|--|
|  | <p>Über den Autor</p> <p>...</p> | | | | | | | | |
|  | <table border="1"> <tbody> <tr> <td>Name</td><td>Pascal Neuperger</td></tr> <tr> <td>Organisation</td><td>Komax Testing Germany GmbH</td></tr> <tr> <td>Funktion</td><td>Applikationstechniker & Produkt Support Manager</td></tr> </tbody> </table> <p>Pascal Neuperger ist ausgebildeter Mechatroniker und staatlich geprüfter Techniker in der Fachrichtung Maschinenbautechnik. Er ist seit 2005 bei Komax Testing Germany GmbH in Porta Westfalica tätig und arbeitet dort als Applikationstechniker für Prüfsysteme und Produkt Support Manager für Schraubtechnik. Seine Hauptaufgaben sind die Entwicklung von Prüfkonzepten, die Programmierung und Inbetriebnahme von Prüfsystemen sowie die Unterstützung von Kunden und Servicetechnikern bei technischen Fragen. Er ist spezialisiert auf Bussysteme, die Programmierung von Prüfsoftware in LabVIEW™ und C# und die Anbindung von Prüfsystemen an übergeordnete Systeme.</p> <p>Im Projekt VWS4LS trug er im Wesentlichen bei zu den Teilprojekten 1, 3 und 5. Er leitete die Architekturgruppen Verlinkung und Rückverfolgbarkeit.</p> | Name | Pascal Neuperger | Organisation | Komax Testing Germany GmbH | Funktion | Applikationstechniker & Produkt Support Manager | | |
| Name | Pascal Neuperger | | | | | | | | |
| Organisation | Komax Testing Germany GmbH | | | | | | | | |
| Funktion | Applikationstechniker & Produkt Support Manager | | | | | | | | |
| | <table border="1"> <tbody> <tr> <td>Name</td><td></td></tr> <tr> <td>Organisation</td><td></td></tr> <tr> <td>Funktion</td><td></td></tr> <tr> <td>Über den Autor</td><td></td></tr> </tbody> </table> | Name | | Organisation | | Funktion | | Über den Autor | |
| Name | | | | | | | | | |
| Organisation | | | | | | | | | |
| Funktion | | | | | | | | | |
| Über den Autor | | | | | | | | | |
|  | <table border="1"> <tbody> <tr> <td>Name</td><td>Mario Angos</td></tr> <tr> <td>Organisation</td><td>Coroplast Group</td></tr> <tr> <td>Funktion</td><td>Project Manager Digital Transformation</td></tr> </tbody> </table> | Name | Mario Angos | Organisation | Coroplast Group | Funktion | Project Manager Digital Transformation | | |
| Name | Mario Angos | | | | | | | | |
| Organisation | Coroplast Group | | | | | | | | |
| Funktion | Project Manager Digital Transformation | | | | | | | | |
| | <table border="1"> <tbody> <tr> <td>Name</td><td>Lena Beil</td></tr> <tr> <td>Organisation</td><td>Lisa Dräxlmaier GmbH</td></tr> </tbody> </table> | Name | Lena Beil | Organisation | Lisa Dräxlmaier GmbH | | | | |
| Name | Lena Beil | | | | | | | | |
| Organisation | Lisa Dräxlmaier GmbH | | | | | | | | |

| | | |
|--|--------------|--|
| | Funktion | Digital Business Specialist |
| Im Projekt VWS4LS leitete sie das Teilprojekt 8 (Data Storage Policy, Sicherheit, Anbindung an Catena-X) und wirkte zudem am Teilprojekt 7 mit. | | |
|  | Name | Carsten Weiß |
| | Organisation | WEZAG GmbH & Co. KG |
| | Funktion | Teamleiter Embedded Engineering |
| <p>Nach seinem Studium der Informations- und Kommunikationstechnik war er 15 Jahre lang als Entwickler für Embedded Messtechnik in einem Unternehmen aus dem Bereich der elektrischen Sicherheit tätig.</p> <p>Bei WEZAG leitet er seit 2021 das Embedded Engineering Team.</p> <p>Im Rahmen von VWS4LS war er festes Teammitglied in TP1 und TP8. Darüber hinaus trug er auch zur Entwicklung der Demonstratoren in TP9 sowie zu den Themen „Verlinkung“ und „Synchronisierung“ innerhalb des Architekturteams bei. Weitere fachliche Beiträge betrafen ferner die TPs 2, 3 und 7.</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> | | |
| | Name | |
| | Organisation | |
| | Funktion | |
| Über den Autor | | |
| | 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> | | |