

Leitfaden

System- und Servicearchitekturen

Nächste Überprüfung am 31. März 2025

I. ÄNDERUNGSVERZEICHNIS

| Datum | Version | Vorgang | Autor |
|------------|---------|--|-----------------------------|
| 29.07.2020 | 0.10 | Ersterstellung | Systemarchitekt IT-SysBw |
| 14.08.2020 | 0.20 | Erarbeitung Kapitel 1 | Systemarchitekt IT-SysBw |
| 31.08.2020 | 0.30 | Erarbeitung Kapitel 2 | Systemarchitekt IT-SysBw |
| 04.09.2020 | 0.40 | Redaktionelle Änderungen an Kapitel 1 und 2 | Systemarchitekt IT-SysBw |
| 15.09.2020 | 0.50 | Einpfelegen der Modellierungskonventionen in Kapitel 2 | Systemarchitekt IT-SysBw |
| 22.09.2020 | 0.60 | Review Kapitel 2.2 und 2.3 | Systemarchitekt IT-SysBw |
| 09.10.2020 | 0.70 | Einpfelegen der Beispielmodelle in Kapitel 2 | Systemarchitekt IT-SysBw |
| 12.10.2020 | 0.80 | Einpfelegen der Übersichtsmodelle je CPM Phase | Systemarchitekt IT-SysBw |
| 16.10.2020 | 0.90 | Review und Korrektur aller Kapiteln | Systemarchitekt IT-SysBw |
| 03.11.2020 | 1.00 | Finalisierung aller Kapitel | Systemarchitekt IT-SysBw |
| 26.11.2020 | 1.10 | Überführung von TaggedValues mit RefGUID-Listen in Relationen (A2, A8, L2, Lr, Cr, Sr, P3) | Systemarchitekt IT-SysBw |
| 07.12.2020 | 1.20 | Anpassung Kapitel 1, Entfernung Abkürzungsverzeichnis; Anpassung Gültigkeit | Systemarchitekt IT-SysBw |
| 14.01.2021 | 1.30 | Ergänzung Lr in allen Phasen; Ergänzung externe Abhängigkeiten im P2 in allen Phasen; Ergänzung R8 in Nutzungsphase; Löschung S8 in Realisierungsphase; Änderungen in der Modellierung vom A3, A6, Ar und S8; Umbenennung R2 & R8; Umstrukturierung des Kapitel 2.3, 2.4 und 2.5 | Systemarchitekt IT-SysBw |
| 17.03.2021 | 1.40 | Redaktionelle Änderungen in Kapitel 2; Änderung Diagrammbenennung S1, S2, S3, S4, S7, S8, Sr | Systemarchitekt IT-SysBw |
| 23.07.2021 | 1.41 | Gültigkeitsdatum geändert | BAAINBw D1 |
| 25.10.2021 | 1.42 | Gültigkeitsdatum geändert | BAAINBw D1 |
| 06.12.2021 | 1.5 | Änderung A2, Cr, P2, Sr; Änderung Grafik Überblick der Elemente in Kapitel 2.3, 2.4 und 2.5; Gültigkeitsdatum geändert; Einstufung des Dokuments geändert | BAAINBw D1 |
| 08.02.2022 | 1.51 | Änderung Kapitel 1.5.3 und 1.5.4; Änderung A1; Änderung P2 | BAAINBw D1.1 |
| 15.06.2022 | 1.52 | Gültigkeitsdatum geändert | BAAINBw D1.1 |
| 20.12.2022 | 1.53 | Änderung Kontaktdaten Op.Architekt IT-SysBw; Löschung A8, Ar, S8, Sr; Vollständige Überarbeitung aller Viewpoints Ergänzung von Package-Strukturen | BAAINBw D1.1 |

| | | | |
|------------|------|---|--------------|
| 24.04.2023 | 1.54 | Gültigkeitsdatum geändert | BAAINBw D1.1 |
| 11.12.2023 | 2.00 | Straffung, Neustrukturierung, Anpassung | BAAINBw D1.1 |

II. INHALTSVERZEICHNIS

| | | |
|--------|---|----|
| I. | Änderungsverzeichnis..... | 2 |
| II. | Inhaltsverzeichnis | 4 |
| III. | Abbildungsverzeichnis | 8 |
| IV. | Tabellenverzeichnis | 11 |
| 1 | Vorbemerkungen..... | 12 |
| 1.1 | Hinweise zum Verständnis des Dokumentes | 12 |
| 1.2 | Hinweise zur Formatierung | 12 |
| 1.3 | Hinweise zu den Beispielen | 12 |
| 1.4 | Hinweise zur Modellierung..... | 13 |
| 1.5 | Notation..... | 13 |
| 1.6 | Geltungsbereich | 13 |
| 2 | Anwendungsfall System- und Servicearchitekturen im CPM | 14 |
| 2.1 | Übersicht | 14 |
| 2.2 | Aufbau der Kapitel zum Anwendungsfall System- und Servicearchitekturen im CPM | 15 |
| 2.3 | System- und Servicearchitekturen in der Analysephase Teil 2 | 16 |
| 2.3.1 | Übergreifende Package-Struktur der System- und Servicearchitektur | 18 |
| 2.3.2 | Package-Struktur der Systemarchitektur | 19 |
| 2.3.3 | A1 – Meta-Data Definitions..... | 20 |
| 2.3.4 | A2 – Stakeholder | 22 |
| 2.3.5 | A2 – Concerns..... | 23 |
| 2.3.6 | A2 – Viewpoints..... | 24 |
| 2.3.7 | A2 – Architecture Products | 25 |
| 2.3.8 | A3 – Architecture Correspondence | 26 |
| 2.3.9 | A6 – Architecture Versions..... | 27 |
| 2.3.10 | Lr – Lines of Development..... | 28 |
| 2.3.11 | Cr – Capability Roadmap (optionaler Viewpoint)..... | 30 |
| 2.3.12 | P1 – Resource Types..... | 31 |
| 2.3.13 | P2 – Resource Structure | 32 |
| 2.3.14 | P2 – Application Hosting | 33 |
| 2.3.15 | P2 – External Dependencies..... | 34 |
| 2.3.16 | P2 – Node Realization (optionaler Viewpoint)..... | 36 |
| 2.3.17 | P3 – Resource Interface..... | 37 |
| 2.3.18 | P7 – Data Elements..... | 38 |
| 2.3.19 | P7 – Data Model | 39 |
| 2.3.20 | P7 – InformationElement Realization (optionaler Viewpoint)..... | 40 |

| | | |
|--------|--|----|
| 2.3.21 | P8 – Resource Constraints (optional Viewpoint) | 41 |
| 2.3.22 | P8 – Resource Constraint Realization (optional Viewpoint) | 43 |
| 2.3.23 | Package-Struktur der Servicebeschreibung | 44 |
| 2.3.24 | S1 – Service Description | 45 |
| 2.3.25 | S1 – Service Taxonomy | 47 |
| 2.3.26 | S1 – Service Varianten (optional Viewpoint) | 48 |
| 2.3.27 | C1-S1 – Capability to Service Mapping | 49 |
| 2.3.28 | S2 – Service Structure | 50 |
| 2.3.29 | S2 – External Service Dependencies | 51 |
| 2.3.30 | S2 – Service Warranties | 52 |
| 2.3.31 | S3 – Service Interfaces | 54 |
| 2.3.32 | S4 – Service Functions | 56 |
| 2.3.33 | S7 – Service Interface Parameters | 58 |
| 2.3.34 | Sr – Service Change | 59 |
| 2.3.35 | P1 – Projektbezug | 60 |
| 2.3.36 | Package-Struktur der Anforderungen | 61 |
| 2.3.37 | R2 – Requirement Catalogue | 62 |
| 2.3.38 | R3 – Requirement Dependencies | 64 |
| 2.3.39 | R7 – Requirement Derivation | 65 |
| 2.3.40 | Auswertediagramme | 67 |
| 2.4 | System- und Servicearchitekturen in der Realisierungsphase | 68 |
| 2.4.1 | Package-Struktur der System- und Servicearchitektur | 69 |
| 2.4.2 | A1 – Meta-Data Definitions | 69 |
| 2.4.3 | A2 – Architecture Products | 70 |
| 2.4.4 | A3 – Architecture Correspondence | 72 |
| 2.4.5 | A6 – Architecture Versions | 73 |
| 2.4.6 | Lr – Lines of Development | 74 |
| 2.4.7 | Cr – Capability Roadmap (optional Viewpoint) | 75 |
| 2.4.8 | P1 – Resource Types | 76 |
| 2.4.9 | P2 – Resource Structure | 77 |
| 2.4.10 | P3 – Resource Interfaces | 79 |
| 2.4.11 | P7 – Data Model | 80 |
| 2.4.12 | S1 – Service Description | 82 |
| 2.4.13 | S1 – Service Taxonomy | 83 |
| 2.4.14 | S1 – Service Varianten (optional Viewpoint) | 84 |
| 2.4.15 | C1-S1 – Capability to Service Mapping | 85 |
| 2.4.16 | S2 – Service Structure | 86 |
| 2.4.17 | S2 – External Service Dependencies | 87 |

| | | |
|--------|---|-----|
| 2.4.18 | S2 – Service Warranties..... | 88 |
| 2.4.19 | S3 – Service Interfaces..... | 89 |
| 2.4.20 | S4 – Service Functions | 90 |
| 2.4.21 | S7 – Service Interface Parameters..... | 91 |
| 2.4.22 | Sr – Service Change | 92 |
| 2.4.23 | P1 – Projektbezug..... | 93 |
| 2.4.24 | R8 – Requirement Fulfilment (optionaler Viewpoint)..... | 94 |
| 2.4.25 | Rr – Requirement Realization..... | 95 |
| 2.4.26 | Auswertediagramme | 96 |
| 2.5 | System- und Servicearchitekturen in der Nutzungsphase | 97 |
| 2.5.1 | A1 – Meta–Data Definitions | 98 |
| 2.5.2 | A2 – Architecture Products | 99 |
| 2.5.3 | A3 – Architecture Correspondence | 102 |
| 2.5.4 | A6 – Architecture Versions..... | 103 |
| 2.5.5 | Lr – Lines of Development..... | 105 |
| 2.5.6 | Cr – Capability Roadmap (optionaler Viewpoint)..... | 106 |
| 2.5.7 | P1 – Resource Types..... | 107 |
| 2.5.8 | P2 – Resource Structure | 108 |
| 2.5.9 | P3 – Resource Interfaces | 111 |
| 2.5.10 | P7 – Data Model | 112 |
| 2.5.11 | P8 – Resource Constraints..... | 114 |
| 2.5.12 | S1 – Service Description | 115 |
| 2.5.13 | S1 – Service Taxonomy | 116 |
| 2.5.14 | S1 – Service Varianten (optionaler Viewpoint) | 117 |
| 2.5.15 | C1-S1 – Capability to Service Mapping..... | 118 |
| 2.5.16 | S2 – Service Structure..... | 119 |
| 2.5.17 | S2 – External Service Dependencies..... | 120 |
| 2.5.18 | S2 – Service Warranties..... | 121 |
| 2.5.19 | S3 – Service Interfaces..... | 122 |
| 2.5.20 | S4 – Service Functions | 123 |
| 2.5.21 | S7 – Service Interface Parameters..... | 124 |
| 2.5.22 | Sr – Service Change | 125 |
| 2.5.23 | P1 – Projektbezug..... | 126 |
| 2.5.24 | R2 – Requirement Catalogue..... | 127 |
| 2.5.25 | R3 – Requirement Dependencies | 128 |
| 2.5.26 | R7 – Requirement Derivation | 129 |
| 2.5.27 | R8 – Requirement Fulfilment (optionaler Viewpoint)..... | 130 |
| 2.5.28 | Rr – Requirement Realization..... | 131 |

| | | |
|--------|----------------------------------|-----|
| 2.5.29 | Auswertediagramme | 132 |
| 3 | Anlagen | 133 |
| 3.1 | Template Initialisierung..... | 133 |
| 3.2 | Struktur Forderungskatalog | 133 |
| 4 | Änderungsprozess | 134 |

III. ABBILDUNGSVERZEICHNIS

| | |
|--|----|
| Abbildung 1: Beispiel „Name des Viewpoint“ | 15 |
| Abbildung 2: Überblick der Elemente in System- und Servicearchitekturen in der AP2 | 17 |
| Abbildung 3: Übergreifende Package-Struktur | 18 |
| Abbildung 4: System Package-Struktur | 19 |
| Abbildung 5: Beispiel A1 : MASC : Architectural Description | 21 |
| Abbildung 6: Beispiel A2 : MASC : Stakeholder | 22 |
| Abbildung 7: Beispiel A2 : MASC : Concerns | 23 |
| Abbildung 8: Beispiel A2 : MASC : Viewpoints | 24 |
| Abbildung 9: Beispiel A2 : MASC : Architecture Products | 25 |
| Abbildung 10: Beispiel A3 : MASC : Architecture Correspondence..... | 26 |
| Abbildung 11: Beispiel A6 : MASC : Architecture Versions | 27 |
| Abbildung 12: Beispiel Lr : MASC : Lines of Development | 29 |
| Abbildung 13: Beispiel Cr : MASC : Capability Roadmap..... | 30 |
| Abbildung 14: Beispiel P1 : MASC : Resource Types | 31 |
| Abbildung 15: Beispiel P2 : MASC : Resource Structure..... | 32 |
| Abbildung 16: Beispiel P2 : MASC : Application Hosting..... | 33 |
| Abbildung 17: Beispiel P2 : MASC : External Dependencies | 35 |
| Abbildung 18: Beispiel P2 : MASC : Node Realization | 36 |
| Abbildung 19: Beispiel P3 : MASC : Resource Interfaces..... | 37 |
| Abbildung 20: Beispiel P7 : MASC : Data Elements | 38 |
| Abbildung 21: Beispiel P7 : MASC : Data Model..... | 39 |
| Abbildung 22: Beispiel P7 : MASC : InformationElement Realization | 40 |
| Abbildung 23: Beispiel P8 : MASC : Resource Constraints | 42 |
| Abbildung 24: Beispiel P8 : MASC : Resource Constraint Realization | 43 |
| Abbildung 25: Service Package-Struktur | 44 |
| Abbildung 26: Beispiel S1 : MASC : Service Description (Spacecraft Information Service Bw)..... | 46 |
| Abbildung 27: Beispiel S1 : MASC : Service Taxonomy (Spacecraft Information Service Bw) | 47 |
| Abbildung 28: Beispiel S1 : MASC : Service Varianten (Spacecraft Information Service Bw) | 48 |
| Abbildung 29: Beispiel C1-S1 : MASC : Capability to Service Mapping (Spacecraft Information Service Bw)..... | 49 |
| Abbildung 30: Beispiel S2 : MASC : Service Structure (Spacecraft Information Service Bw) | 50 |
| Abbildung 31: Beispiel S2 : MASC : External Service Dependencies (Spacecraft Information Service Bw) | 51 |
| Abbildung 32: Beispiel S2 : MASC : Service Warranties (Spacecraft Information Service Bw) | 53 |
| Abbildung 33: Beispiel S2 : MASC : Service Interface (Spacecraft Information Service Bw)..... | 53 |
| Abbildung 34: Beispiel S3 : MASC : Service Interfaces (Spacecraft Information Service Bw) | 55 |
| Abbildung 35: Beispiel S4 : MASC : Service Functions (Spacecraft Information Service Bw)..... | 57 |
| Abbildung 36: Beispiel S4 : MASC : Service Functions - optional (Spacecraft Information Service Bw)..... | 57 |
| Abbildung 37: Beispiel S7 : MASC : Service Interface Parameters (Spacecraft Information Service Bw) | 58 |
| Abbildung 38: Beispiel Sr : MASC : Service Change (Spacecraft Information Service Bw)..... | 59 |
| Abbildung 39: Beispiel P1 : MASC : Serviceprovision | 60 |
| Abbildung 40: Anforderungen Package-Struktur | 61 |
| Abbildung 41: Beispiel R2 : MASC: Requirement Catalogue | 63 |

| | |
|--|-----|
| Abbildung 42: Beispiel R3 : MASC : Requirement Dependencies | 64 |
| Abbildung 43: Beispiel R7 : MASC : Requirement Derivation | 66 |
| Abbildung 45: Beispiel P2 : MASC : Auswertediagramm Lösungsvorschläge | 67 |
| Abbildung 46: Überblick der Elemente in System- und Servicearchitekturen in der RP..... | 68 |
| Abbildung 47: Beispiel A1 : MASC : Architectural Description..... | 69 |
| Abbildung 48: Beispiel A2 : MASC : Concerns | 70 |
| Abbildung 49: Beispiel A2 : MASC : Stakeholder | 70 |
| Abbildung 50: Beispiel A2 : MASC : Viewpoints | 71 |
| Abbildung 51: Beispiel A2 : MASC : Architecture Products..... | 71 |
| Abbildung 52: Beispiel A3 : MASC : Architecture Correspondence..... | 72 |
| Abbildung 53: Beispiel A6 : MASC : Architecture Versions | 73 |
| Abbildung 54: Beispiel Lr : MASC : Lines of Development | 74 |
| Abbildung 55: Beispiel Cr : MASC : Capability Roadmap..... | 75 |
| Abbildung 56: Beispiel P1 : MASC : Resource Types | 76 |
| Abbildung 57: Beispiel P2 : MASC : Resource Structure..... | 77 |
| Abbildung 58: Beispiel P2 : MASC : Application Hosting..... | 78 |
| Abbildung 59: Beispiel P2 : MASC : External Dependencies | 78 |
| Abbildung 60: Beispiel P2 : MASC : Node Realization | 78 |
| Abbildung 61: Beispiel P3 : MASC : Resource Interfaces..... | 79 |
| Abbildung 62: Beispiel P7 : MASC : Data Elements | 80 |
| Abbildung 63: Beispiel P7 : MASC : Data Model..... | 80 |
| Abbildung 64: Beispiel P7 : MASC : InformationElement Realization | 81 |
| Abbildung 65: Beispiel S1 : MASC : Service Description (Spacecraft Information Service Bw)..... | 82 |
| Abbildung 66: Beispiel S1 : MASC : Service Taxonomy (Spacecraft Information Service Bw) | 83 |
| Abbildung 67: Beispiel S1 : MASC : Service Varianten (Spacecraft Information Service Bw) | 84 |
| Abbildung 68: Beispiel C1-S1 : MASC : Capability to Service Mapping (Spacecraft Information Service Bw)..... | 85 |
| Abbildung 69: Beispiel S2 : MASC : Service Structure (Spacecraft Information Service Bw) | 86 |
| Abbildung 70: Beispiel S2 : MASC : External Service Dependencies (Spacecraft Information Service Bw) | 87 |
| Abbildung 71: Beispiel S2 : MASC : Service Warranties (Spacecraft Information Service Bw) | 88 |
| Abbildung 72: Beispiel S2 : MASC : Service Interface (Spacecraft Information Service Bw)..... | 88 |
| Abbildung 73: Beispiel S3 : MASC : Service Interfaces (Spacecraft Information Service Bw) | 89 |
| Abbildung 74: Beispiel S4 : MASC : Service Functions (Spacecraft Information Service Bw)..... | 90 |
| Abbildung 75: Beispiel S4 : MASC : Service Functions - optional (Spacecraft Information Service Bw)..... | 90 |
| Abbildung 76: Beispiel S7 : MASC : Service Interface Parameters (Spacecraft Information Service Bw) | 91 |
| Abbildung 77: Beispiel Sr : MASC : Service Change (Spacecraft Information Service Bw)..... | 92 |
| Abbildung 78: Beispiel P1 : MASC : Serviceprovision | 93 |
| Abbildung 44: Beispiel R8 : MASC : Requirement Fulfilment..... | 94 |
| Abbildung 79: Beispiel Rr : MASC : Requirement Realization | 95 |
| Abbildung 80: Überblick der Elemente in System- und Servicearchitekturen in der NP | 97 |
| Abbildung 81: Beispiel A1 : MASC : Architectural Description..... | 98 |
| Abbildung 82: Beispiel A2 : MASC : Concerns | 100 |
| Abbildung 83: Beispiel A2 : MASC : Stakeholder | 100 |

| | |
|---|-----|
| Abbildung 84: Beispiel A2 : MASC : Viewpoints | 101 |
| Abbildung 85: Beispiel A2 : MASC : Architecture Products | 101 |
| Abbildung 86: Beispiel A3 : MASC : Architecture Correspondence..... | 102 |
| Abbildung 87: Beispiel A6 : MASC : Architecture Versions | 104 |
| Abbildung 88: Beispiel Lr : MASC : Lines of Development | 105 |
| Abbildung 89: Beispiel Cr : MASC : Capability Roadmap..... | 106 |
| Abbildung 90: Beispiel P1 : MASC : Resource Types | 107 |
| Abbildung 91: Beispiel P2 : MASC : Resource Structure..... | 109 |
| Abbildung 92: Beispiel P2 : MASC : Application Hosting..... | 109 |
| Abbildung 93: Beispiel P2 : MASC : External Dependencies | 110 |
| Abbildung 94: Beispiel P2 : MASC : Node Realization | 110 |
| Abbildung 95: Beispiel P3 : MASC : Resource Interfaces..... | 111 |
| Abbildung 96: Beispiel P7 : MASC : Data Elements | 112 |
| Abbildung 97: Beispiel P7 : MASC : Data Model..... | 112 |
| Abbildung 98: Beispiel P7 : MASC : InformationElement Realization | 113 |
| Abbildung 99: Beispiel P8 : MASC : Resource Constraints | 114 |
| Abbildung 100: Beispiel P8 : MASC : Resource Constraint Realization | 114 |
| Abbildung 101: Beispiel S1 : MASC : Service Description (Spacecraft Information Service Bw)..... | 115 |
| Abbildung 102: Beispiel S1 : MASC : Service Taxonomy (Spacecraft Information Service Bw) | 116 |
| Abbildung 103: Beispiel S1 : MASC : Service Varianten (Spacecraft Information Service Bw) | 117 |
| Abbildung 104: Beispiel C1-S1 : MASC : Capability to Service Mapping (Spacecraft Information Service Bw)..... | 118 |
| Abbildung 105: Beispiel S2 : MASC : Service Structure (Spacecraft Information Service Bw) | 119 |
| Abbildung 106: Beispiel S2 : MASC : External Service Dependencies (Spacecraft Information Service Bw) | 120 |
| Abbildung 107: Beispiel S2 : MASC : Service Warranties (Spacecraft Information Service Bw) | 121 |
| Abbildung 108: Beispiel S2 : MASC : Service Interface (Spacecraft Information Service Bw)..... | 121 |
| Abbildung 109: Beispiel S3 : MASC : Service Interfaces (Spacecraft Information Service Bw) | 122 |
| Abbildung 110: Beispiel S4 : MASC : Service Functions (Spacecraft Information Service Bw)..... | 123 |
| Abbildung 111: Beispiel S4 : MASC : Service Functions - optional (Spacecraft Information Service Bw) | 123 |
| Abbildung 112: Beispiel S7 : MASC : Service Interface Parameters (Spacecraft Information Service Bw) | 124 |
| Abbildung 113: Beispiel Sr : MASC : Service Change (Spacecraft Information Service Bw)..... | 125 |
| Abbildung 114: Beispiel P1 : MASC : Serviceprovision | 126 |
| Abbildung 115: Beispiel R2 : MASC: Requirement Catalogue | 127 |
| Abbildung 116: Beispiel R3 : MASC : Requirement Dependencies | 128 |
| Abbildung 117: Beispiel R7 : MASC : Requirement Derivation | 129 |
| Abbildung 118: Beispiel R8 : MASC : Requirement Fulfilment..... | 130 |
| Abbildung 119: Beispiel Rr : MASC : Requirement Realization | 131 |

IV. TABELLENVERZEICHNIS

| | |
|---|----|
| Tabelle 1: Zuordnung der Viewpoints zu den CPM Phasen | 14 |
|---|----|

VORBEMERKUNGEN

Hinweise zum Verständnis des Dokumentes

Dieses Dokument beschreibt die Modellierung von Architekturen nach den Vorgaben des NAFv4-ADMBw ab der Analysephase Teil 2 (AP2) des CPM.

Die Grundlage der Nutzung von Architekturen nach NAFv4-ADMBw ist im Dokument „Grundlagen Methode Architektur gemäß NATO Architecture Framework Version 4“ dargestellt. Dieses Dokument enthält auch Erläuterungen zum Aufbau des NAFv4-ADMBw, den einzelnen Viewpoints sowie grundsätzlicher Zusammenhänge.

Das Vorgehen zur Modellierung mit dem Modellierungswerkzeug SPARX Enterprise Architect wird im Dokument „Dokumentation Verwendung des ADMBw in Sparx Enterprise Architect“ beschrieben.

Weitergehende Informationen und eine Liste der aktuell gültigen Dokumente zur Anwendung der Methode Architektur werden im WikiBw auf dem [Portal Architekturen](#) und im [Internet](#) bereitgestellt.

Weiterführende Informationen zur Weiterentwicklung der Methode werden vorrangig auf der Plattform Management von Ergebnissen und Erkenntnissen ([MErk](#)) bereitgestellt.¹

Hinweise zur Formatierung

Beispiele zur Verdeutlichung der Inhalte werden in Grün gesetzt und mit einem gleichfarbigen Rahmen versehen.

Hinweise, die unbedingt zu beachten sind, werden mit einem schwarzen Rahmen versehen.

Verlinkungen werden im Dokument, wie hier die Verlinkung zur [E-Mail-Adresse des Operationellen Architekten Bundeswehr](#), blau gesetzt und unterstrichen.

Hinweise zu den Beispielen

Bei den in diesem Dokument aufgeführten Beispielen handelt es sich um Beispiele zur Verdeutlichung methodischer Zusammenhänge. Diese Beispiele orientieren sich inhaltlich an der Realität, sind aber weder validiert, noch verifiziert. Aus Gründen der Veranschaulichung sind diese Beispiele methodisch auf die Darstellung des zu verdeutlichenden Zusammenhanges reduziert. Inhalte, die für das Verständnis des zugrundeliegenden Problems nicht unbedingt notwendig sind, wurden unter Umständen weggelassen. In einer realen Architektur müssen solche Inhalte jedoch dargestellt werden. In einigen Diagrammen sind verwendete Elemente nur eingeblendet, um den Modellierern bei der Modellierung zu helfen.

Eine Verwendung der in diesem Dokument aufgeführten Beispiele ist, außer zu Zwecken der Aus- und Weiterbildung, untersagt.

¹ Das MErk - Portal ist zugangsbeschränkt. Zugang kann für Angehörige des GB BMVg über PlgABw IV 1 (2) beantragt werden. https://arbber.dokm.bundeswehr.org/sites/PlgABw_90/default.aspx

Das verwendete Beispiel ist als „Running Example“ konzipiert und wird in allen Dokumenten und Ausbildungen zum Thema ADMBw verwendet. Es ist ein fiktives Projekt, bei welchem ein bewaffnetes Mehrzweckraumschiff (MASC: Multipurpose Armed Spacecraft) für die Dimension Weltraum die Forderungen der Bedarfsträger am besten erfüllt. Es kann sowohl zur Aufklärung, als auch zur Wirkung eingesetzt werden.

Hinweise zur Modellierung

Die Arbeit mit Architekturen bleibt erfahrungsgemäß vom Personalwechsel nicht verschont. Daher ist es ungemein wichtig, den Elementen Beschreibungen im "Notes-Feld" hinzuzufügen, wie in der Dokumentation "Verwendung des ADMBw im SPARX Enterprise Architect" (Gelbes Dokument) im Unterkapitel 3.1.2 gefordert.

Notation

Bei der Darstellung der Elemente und Relation wird die auf dem Unified Architecture Framework (UAF) Domain Metamodell (DMM) basierende Notation ADMBw angewendet. Das UAF DMM basiert wiederum auf der Unified Modeling Language (UML) und der Business Process Model and Notation (BPMN).

Einzelheiten zur Notation, zu Elementen und Relationen sowie zur Verwendung abstrakter Elemente sind im Dokument „Grundlagen Methode Architektur gemäß NATO Architecture Framework Version 4“ enthalten.

Geltungsbereich

Dieses Dokument ist für den Anteil der nach NAFv4 modellierten Architekturen im gesamten Geschäftsbereich BMVg sowie für die Auftragnehmer die Architekturen für diesen erstellen, verbindlich.

ANWENDUNGSFALL SYSTEM- UND SERVICEARCHITEKTUREN IM CPM

Übersicht

Nachfolgende Tabelle zeigt einen Überblick welche Viewpoints in den unterschiedlichen Phasen des CPM erstellt bzw. fortgeschrieben werden.

| Viewpoint | Analysephase Teil 2 | Realisierungsphase | Nutzungsphase |
|---------------------------------------|---------------------|--------------------|---------------|
| A1 - Meta-Data Definitions | | | |
| A2 - Stakeholder | | | |
| A2 - Concerns | | | |
| A2 - Viewpoints | | | |
| A2 - Architecture Products | | | |
| A3 - Architecture Correspondence | | | |
| A6 - Architecture Versions | | | |
| Lr - Lines of Development | | | |
| Cr - Capability Roadmap | | | |
| P1 - Resource Types | | | |
| P2 - Resource Structure | | | |
| P2 - Application Hosting | | | |
| P2 - External Dependencies | | | |
| P2 - Node Realization | | | |
| P3 - Resource Interfaces | | | |
| P7 - Data Elements | | | |
| P7 - Data Model | | | |
| P7 - InformationElement Realization | | | |
| P8 - Resource Constraints | | | |
| P8 - Resource Constraint Realization | | | |
| S1 - Service Description | | | |
| S1 - Service Taxonomy | | | |
| S1 - Service Varianten | | | |
| C1-S1 - Capability to Service Mapping | | | |
| S2 - Service Structure | | | |
| S2 - External Service Dependencies | | | |
| S2 - Service Warranties | | | |
| S3 - Service Interfaces | | | |
| S4 - Service Functions | | | |
| S7 - Service Interface Parameters | | | |
| P1 - Projektbezug | | | |
| R2 - Requirement Catalogue | | | |
| R3 - Requirement Dependencies | | | |
| R7 - Requirement Derivation | | | |
| R8 - Requirement Fulfilment | | | |
| Rr - Requirement Realization | | | |

Legende: ■ Ersterstellung; ■ Optional; ■ Überarbeitung/Ersterstellung.

Tabelle 1: Zuordnung der Viewpoints zu den CPM Phasen

Aufbau der Kapitel zum Anwendungsfall System- und Servicearchitekturen im CPM

Der Aufbau der Unterkapitel der Kapitel 2.3 bis 2.5 ist immer gleich strukturiert und soll Ihnen helfen sich in diesem Dokument bestmöglich zu Recht zu finden. Aus diesem Grund werden im Folgenden die Bestandteile der Unterkapitel kurz dargestellt und ihre Bedeutung erläutert:

Name des Viewpoint

Im ersten Absatz wird beschrieben, welche Informationsbedarfe gem. NAFv4@ADMBw mit dem betrachteten Viewpoint gedeckt werden können.

Der zweite Absatz dient der Darstellung, welche Informationen der Viewpoint im betrachteten Anwendungsfall liefern soll.

Alle nachfolgenden Absätze beinhalten ergänzende Informationen und dienen der Steigerung des allgemeinen Verständnisses.

| Modellierungskonventionen | | |
|--|--|--|
| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
| xx-MK1 <i>(Diese Spalte vergibt zu jeder Modellierungskonvention eine fortlaufende ID.)</i> | Zur Darstellung bestimmter Informationen einer Architektur ist ein Diagramm vom Typ „Name des Viewpoint“ mit dem Namen „Nummer des Viewpoint : Projektname : Name des Viewpoint“ im Package „Nummer des Viewpoint“ angelegt. <i>(Die Spalte Modellierungskonvention beinhaltet die bei der Erstellung eines Architekturmodells zu berücksichtigenden Modellierungskonventionen und dient der Qualitätssicherung der Modelle.)</i> | xx-A <i>(Mittels dieser Spalte wird eine Verlinkung zur Dokumentation „Verwendung des ADMBW im Sparx Enterprise Architect“ hergestellt. In der benannten Dokumentation ist der jeweilige Modellierungsschritt detailliert beschrieben.)</i> |

Abbildung 1: Beispiel „Name des Viewpoint“

Beispielmodell
 An dieser Stelle wird ein Auszug der Architektur zum Multipurpose Armed Spacecraft (MASC) dargestellt, der den Informationsbedarf des jeweiligen Viewpoints deckt.

Erläuterung:

Nachdem das Beispielmodell grafisch dargestellt wurde, folgt hier eine textuelle Erläuterung der Inhalte des Modells.

System- und Servicearchitekturen in der Analysephase Teil 2

Die Inhalte einer Architektur werden in sogenannten Sichten (Viewpoints) dargestellt. In diesem Kapitel werden die Sichten des ADMBw angesprochen, die für die Architekturerstellung in Analysephase Teil 2 (AP2) von Relevanz sein können. Aufgrund der hohen Formalisierung und Standardisierung von Architekturen werden bestimmte Inhalte immer auf derselben Sicht dargestellt. Die Auswahl der Sichten in diesem Dokument ist als „Best Practice“ zu verstehen und greift den generellen Analyse- und Informationsbedarf in der Analysephase 2 auf. In Einzelfällen können bestimmte Sichten entfallen oder zusätzliche Sichten müssen modelliert werden.

Die jeweiligen Sichten sind im Dokument „Grundlagen der Methode Architektur gem. NAFv4“ erklärt. Das Vorgehen zur Modellierung ist in der Dokumentation „Verwendung ADMBw im SPARX EA“ beschrieben.

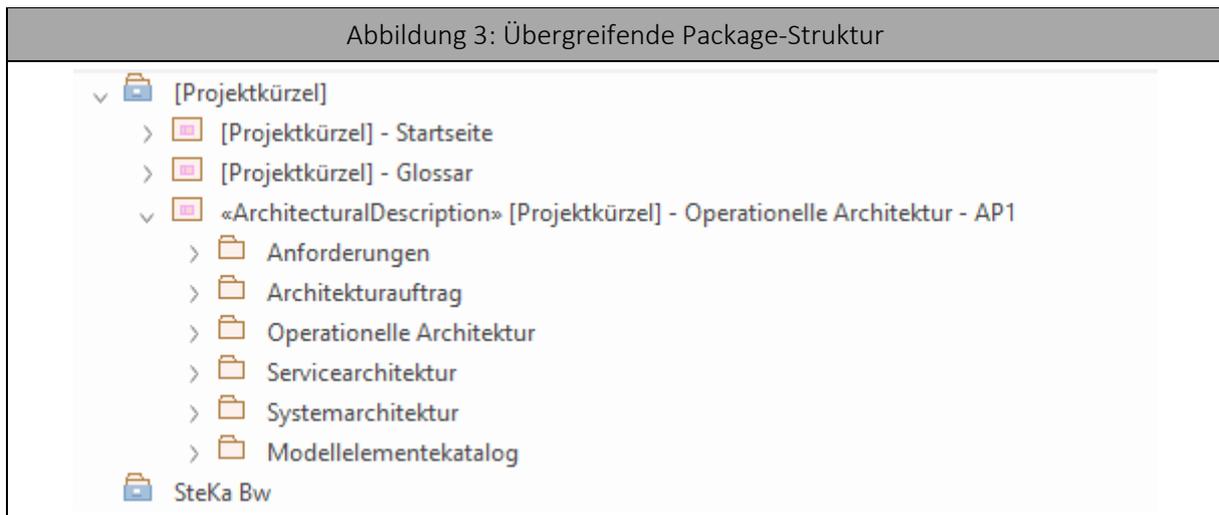
Im Sinne der vereinfachten Darstellung sind in der folgenden Abbildung die wesentlichen Elemente und Relationen eines Architekturmodells bzw. einer Architektur für die Analysephase Teil 2 dargestellt. Die Abbildung beschränkt sich auf die wichtigsten Elemente und ist nicht vollständig. Die detaillierte Darstellung ist der Dokumentation des ADMBw zu entnehmen. Ein detaillierter Einblick in das MetaModell des ADMBw kann über den in Anlage 3.2 aufgeführten Link (MetaModelViewer) abgerufen werden.

Übergreifende Package-Struktur der System- und Servicearchitektur

Bei der Modellierung von Systemen, Services und Anforderungen ist die in Abbildung 3 abgebildete Package-Struktur vorgegeben. Die Package-Struktur bildet die Basis für die automatisierte Weiterverarbeitung der Informationen und muss eingehalten werden. In den folgenden Abschnitten werden die Bereiche *Anforderungen*, *Systemarchitektur* und *Servicearchitektur* im Detail betrachtet.

Hinweis: Für jeden Lösungsvorschlag ist eine eigene Package-Struktur in der hier dargestellten Form zu erstellen.

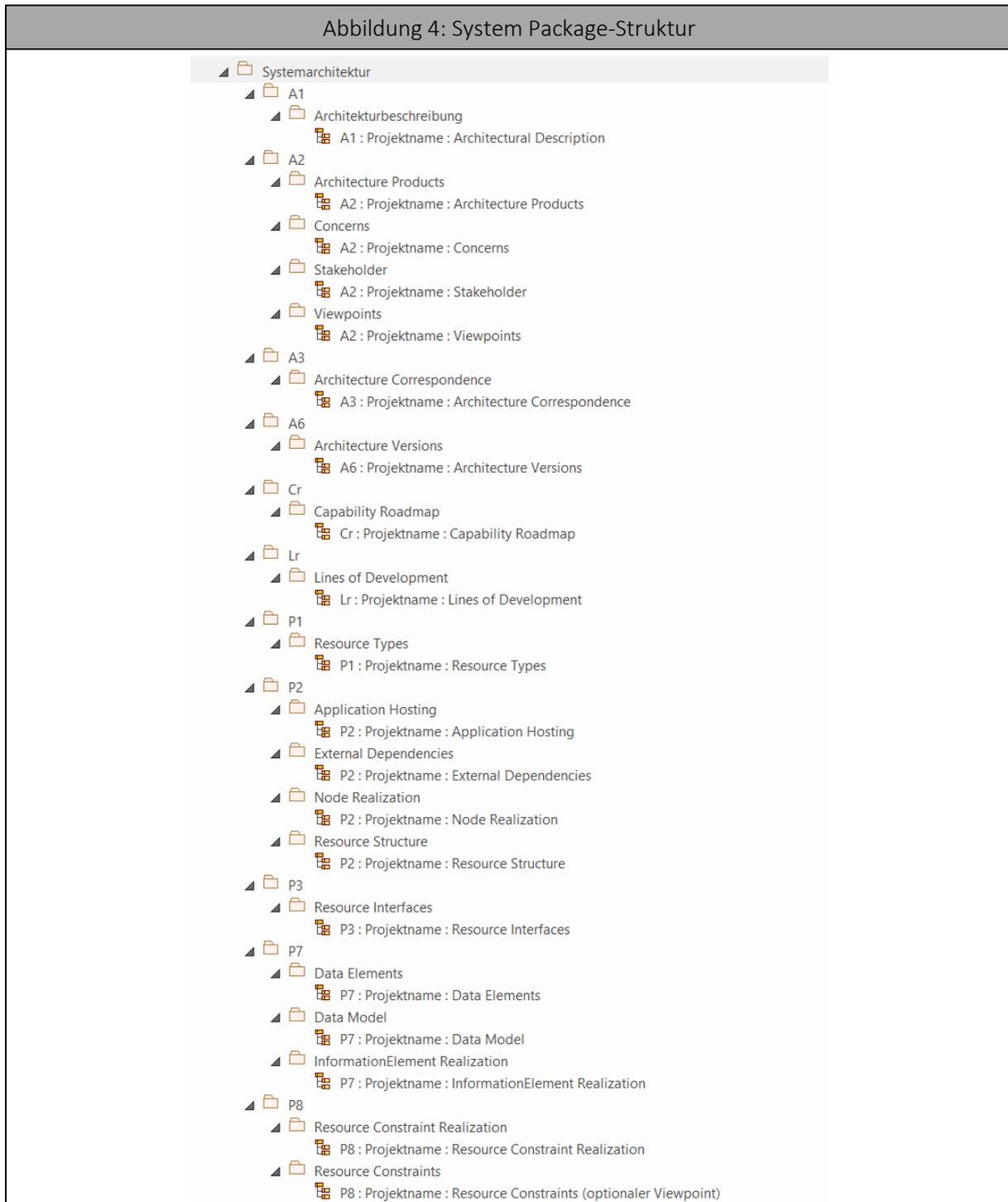
Abbildung 3: Übergreifende Package-Struktur



Package-Struktur der Systemarchitektur

Im Rahmen der Systemmodellierung ist die in Abbildung 4 abgebildete Package-Struktur vorgegeben. Die Package-Struktur bildet die Basis für die automatisierte Weiterverarbeitung der Informationen und muss eingehalten werden. Im folgenden Abschnitt werden die einzelnen Viewpoints dargestellt. Sofern Informationen auf mehrere Diagramme vom selben Viewpoint verteilt werden, sind diese in die Package-Struktur an der entsprechenden Stelle einzuordnen.

Abbildung 4: System Package-Struktur



A1 – Meta-Data Definitions

Der *A1 – Meta Data Definitions* befasst sich mit den Metadaten einer Architektur, welche über die gesamte Architektur hindurch verwendet werden.

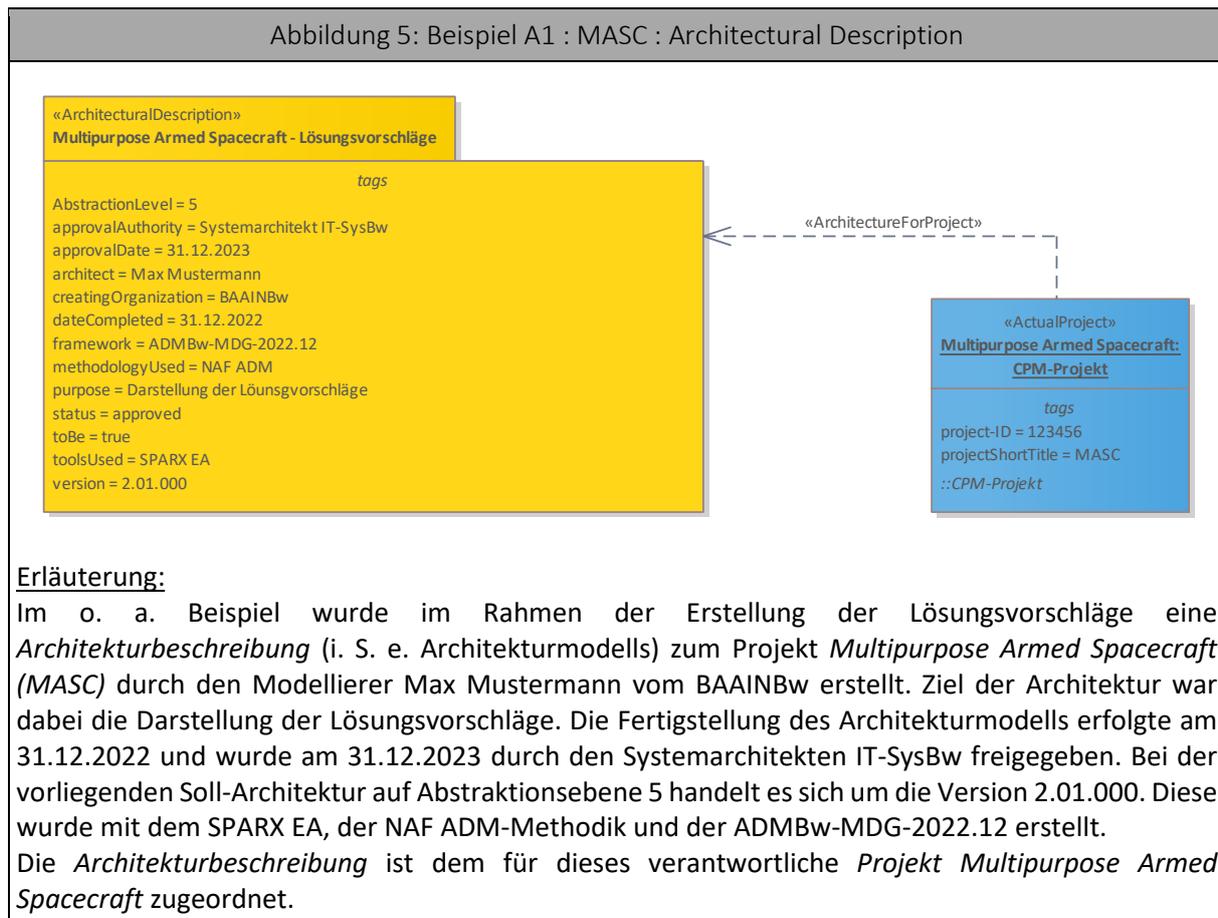
Unter Metadaten werden hier genau festgelegte Daten verstanden, die eine Identifizierung der Architektur und eine inhaltliche bzw. methodische Klassifizierung ermöglichen. Der A1 dient in der Architektur dazu, weitere, noch nicht erfasste Informationen zur Architektur zusammenzufassen.

Neben den originären Inhalten des A1 beinhaltet der A1 im Kontext des CPM immer auch die Inhalte des A4 - Methodology Used sowie A5 - Architecture Status.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| A1-MK1 | Die Informationen werden auf dem Diagramm <i>A1 : Projektname : Architectural Description</i> im Package <i>A1/Architekturbeschreibung</i> erfasst. | A1-A |
| A1-MK2 | Die Architekturbeschreibung (ARCHITECTURALDESCRIPTION) mit dem Namen „ <i>Projektname - Art der Architektur</i> “ ist angelegt. | A1-B |
| A1-MK3 | <p>Die nachfolgenden TaggedValues der Architekturbeschreibung sind zu befüllen:</p> <ul style="list-style-type: none"> • Abstraktionsebene (<i>AbstractionLevel</i>) • Mitprüfende Organisation (<i>approvalAuthority</i>) • Mitprüfungsdatum (<i>approvalDate</i>) • Architekt (<i>architect</i>) • Organisation des Architekten (<i>creatingOrganization</i>) • Abschlussdatum (<i>dateCompleted</i>) • Genutztes Framework (<i>framework</i>) • Genutzte Methode (<i>methodologyUsed</i>) • Zweck (<i>purpose</i>) • Status (<i>status</i>) • Art der Architektur [Ist/Soll] (<i>toBe</i>) • Genutztes Tool (<i>toolsUsed</i>) • Version (<i>version</i>) <p>Hinweis: Bei jedem CPM-Phasenwechsel ist ein neues Architekturmodell und somit ein neuer A1 - Meta Data Definitions sowie eine neue Architekturbeschreibung (ARCHITECTURALDESCRIPTION) anzulegen. Das Versionierungsschema besteht dabei aus drei Bestandteilen:</p> <ul style="list-style-type: none"> • <u>x</u>.xx.xxx: <ul style="list-style-type: none"> ➤ 0. = Voranalysephase ➤ 1. = Analysephase Teil 1 ➤ 2. = Analysephase Teil 2 ➤ 3. = Realisierungsphase ➤ 4. = Nutzungsphase • x.<u>xx</u>.xxx: größere Anpassungen | A1-C |

| | | |
|--------|---|----------------|
| | <ul style="list-style-type: none"> x.xx.xxx: kleinere Anpassungen Beispielsweise steht 2.01.000 für ein Architekturmodell der Analysephase Teil 2, in welchem eine größere Anpassung vorgenommen wurde. | |
| A1-MK4 | Alle projektbezogenen Abkürzungen, Akronyme und Begriffe sind im Glossar des Architekturmodells hinterlegt. | A1-D |
| A1-MK5 | Es ist das für die Architektur verantwortliche Projekt anzulegen und mit der Architekturbeschreibung zu verbinden. | A1-Ba A1-Bc |

Abbildung 5: Beispiel A1 : MASC : Architectural Description



Erläuterung:

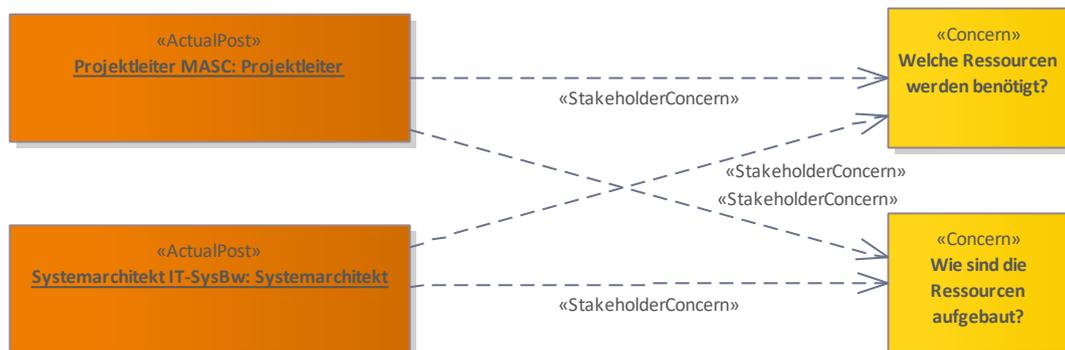
Im o. a. Beispiel wurde im Rahmen der Erstellung der Lösungsvorschläge eine *Architekturbeschreibung* (i. S. e. Architekturmodells) zum Projekt *Multipurpose Armed Spacecraft (MASC)* durch den Modellierer Max Mustermann vom BAAINBw erstellt. Ziel der Architektur war dabei die Darstellung der Lösungsvorschläge. Die Fertigstellung des Architekturmodells erfolgte am 31.12.2022 und wurde am 31.12.2023 durch den Systemarchitekten IT-SysBw freigegeben. Bei der vorliegenden Soll-Architektur auf Abstraktionsebene 5 handelt es sich um die Version 2.01.000. Diese wurde mit dem SPARX EA, der NAF ADM-Methodik und der ADMBw-MDG-2022.12 erstellt. Die *Architekturbeschreibung* ist dem für dieses verantwortliche *Projekt Multipurpose Armed Spacecraft* zugeordnet.

A2 – Stakeholder

Der *A2 – Stakeholder* erfasst die Stakeholder der Architektur und deren Informationsbedarf.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| A2-MK1 | Die Informationen werden auf dem Diagramm <i>A2 : Projektname : Stakeholder</i> im Package <i>A2/ Stakeholder</i> erfasst. | A2-A |
| A2-MK2 | Die relevanten Stakeholder sind angelegt oder aus dem Katalog auf das Diagramm gezogen. | A2-C |
| A2-MK3 | Die Informationsbedarfe der Stakeholder sind angelegt oder aus dem Katalog auf das Diagramm gezogen. | A2-D |
| A2-MK4 | Der Informationsbedarf ist dem Stakeholder zuzuordnen. | A2-E |

Abbildung 6: Beispiel A2 : MASC : Stakeholder

**Erläuterung:**

Die im Beispiel betrachteten Stakeholder des MASC sind der *Systemarchitekt IT-SysBw* sowie der *Projektleiter MASC*. Sowohl der *Projektleiter MASC* als auch der *Systemarchitekt IT-SysBw* möchte wissen, welche Ressourcen benötigt werden und wie diese aufgebaut sind.

A2 – Concerns

Der *A2 – Concerns* erfasst alle Informationsbedarfe in einer Übersicht.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| A2-MK1 | Die Informationen werden auf dem Diagramm <i>A2 : Projektname : Concerns</i> im Package <i>A2/Concerns</i> erfasst. | A2-A |
| A2-MK3 | Die Informationsbedarfe der Stakeholder sind angelegt oder aus dem Katalog auf das Diagramm gezogen. | A2-D |

Abbildung 7: Beispiel A2 : MASC : Concerns

Erläuterung:

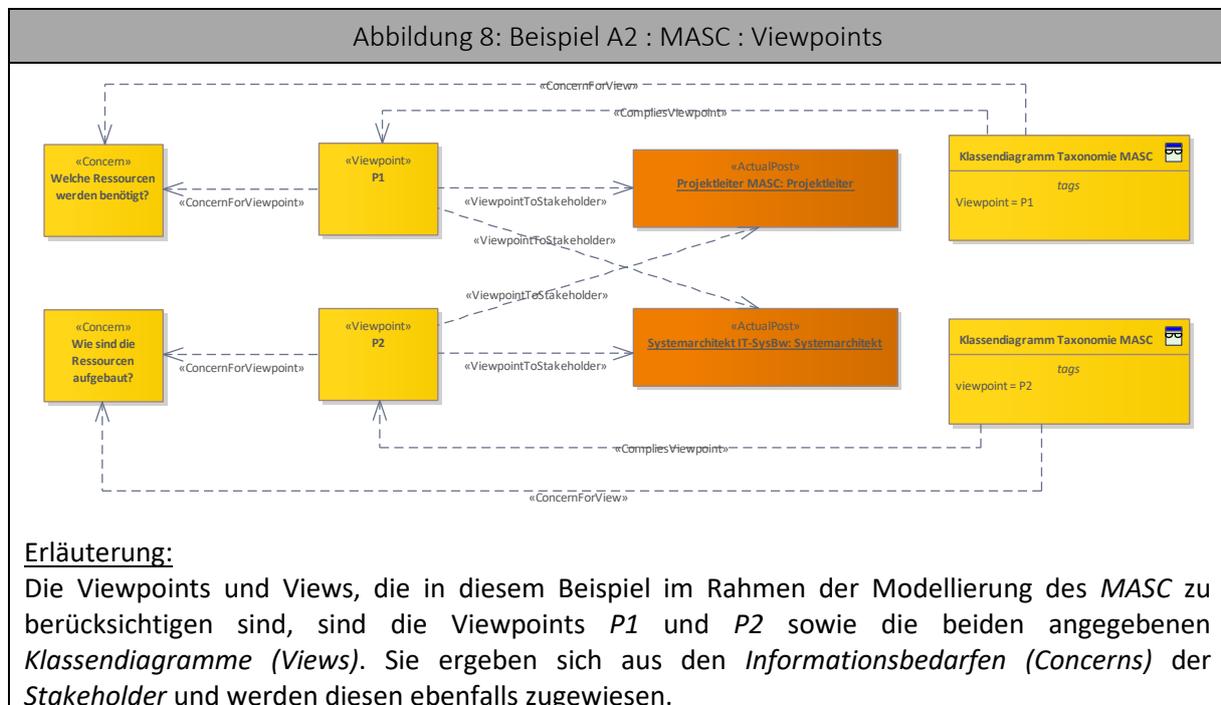
Die im Beispiel betrachteten Stakeholder des MASC möchten gerne wissen, welche Ressourcen für die Umsetzung benötigt werden und wie diese aufgebaut sind.

A2 – Viewpoints

Der *A2 – Viewpoints* erfasst die von den Stakeholdern benötigten Viewpoints und Views (auf Basis deren Informationsbedarfe).

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|-----------------------|
| A2-MK1 | Die Informationen werden auf dem Diagramm <i>A2 : Projektname : Viewpoints</i> im Package <i>A2/Viewpoints</i> erfasst. | A2-A |
| A2-MK2 | Die relevanten Stakeholder sind angelegt oder aus dem Katalog auf das Diagramm gezogen. | A2-C |
| A2-MK3 | Die Informationsbedarfe der Stakeholder sind angelegt oder aus dem Katalog auf das Diagramm gezogen. | A2-D |
| A2-MK4 | Die relevanten Viewpoint sind angelegt oder aus dem Katalog auf das Diagramm gezogen. | A2-F |
| A2-MK5 | Die Viewpoints sind den Informationsbedarfen und Stakeholdern zuzuordnen. | A2-G A2-H |
| A2-MK6 | <i>Hinweis: Dieser Schritt ist erst sinnvoll durchführbar, wenn die Views im Rahmen der Architekturarbeit erstellt sind.</i> Die View angelegt oder aus dem Katalog auf das Diagramm gezogen und den Viewpoint sowie Concerns zuzuordnen. | A2-I A2-J A2-Ja |

Abbildung 8: Beispiel A2 : MASC : Viewpoints

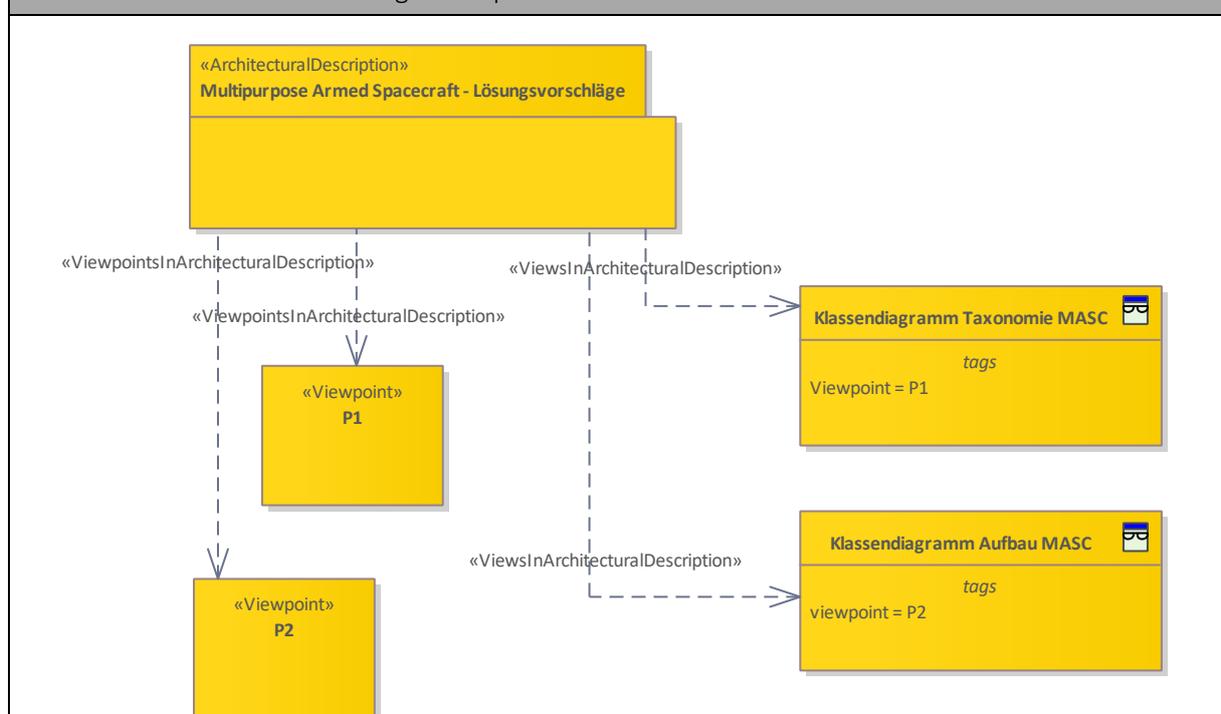


A2 – Architecture Products

Der *A2 – Architecture Products* erfasst die von den Stakeholdern benötigten Viewpoints und Views (auf Basis deren Informationsbedarfe).

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| A2-MK1 | Die Informationen werden auf dem Diagramm <i>A2 : Projektname : Architecture Products</i> im Package <i>A2/Architecture Products</i> erfasst. | A2-A |
| A2-MK2 | Die relevanten Viewpoints sind aus dem Katalog auf das Diagramm gezogen. | A2-F |
| A2-MK3 | Die relevanten Views sind aus dem Katalog auf das Diagramm gezogen. | A2-I |
| A2-MK4 | Die Architekturbeschreibung ist aus dem Katalog auf das Diagramm gezogen. | A2-B |
| A2-MK5 | <i>Hinweis: Dieser Schritt ist für die Views erst sinnvoll durchführbar, wenn die Views im Rahmen der Architekturarbeit erstellt sind.</i> Die View und Viewpoints sind der Architekturbeschreibung zuzuordnen. | A2-K A2-L |

Abbildung 9: Beispiel A2 : MASC : Architecture Products



Erläuterung:

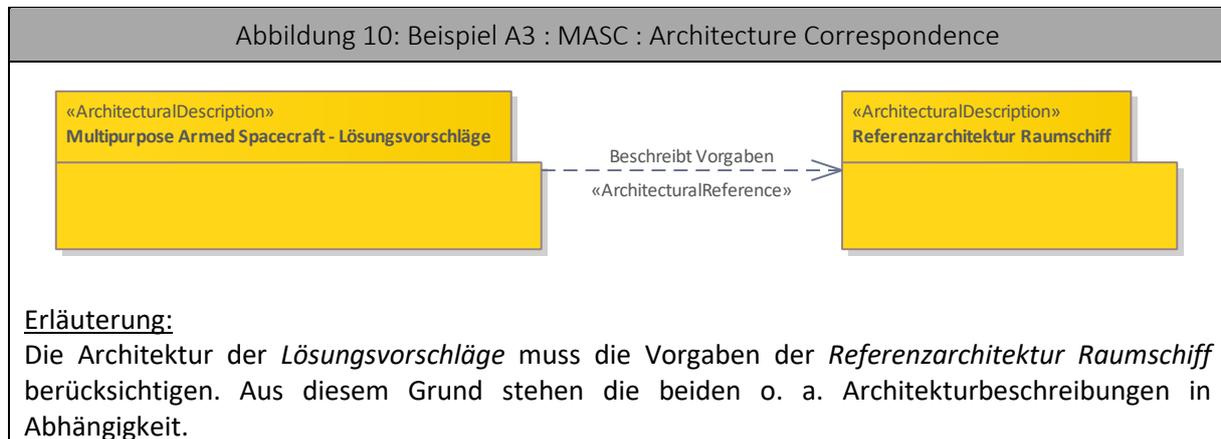
Zum Abschluss der Modellierung des Viewpoints A2 zu den Lösungsvorschlägen werden die zu modellierenden Viewpoints (P1 und P2) und Views (Klassendiagramm Aufbau MASC und Klassendiagramm Taxonomie MASC) zur Architekturbeschreibung des MASC hinzugefügt.

A3 – Architecture Correspondence

Der *A3 – Architecture Correspondence* erfasst die Abhängigkeiten zu anderen Architekturen.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| A3-MK1 | Die Informationen werden auf dem Diagramm <i>A3 : Projektname : Architecture Correspondence</i> im Package <i>A3/Architecture Correspondence</i> erfasst. | A3-A |
| A3-MK2 | Die relevanten Architekturbeschreibungen sind aus dem Katalog auf das Diagramm gezogen. | A3-B |
| A3-MK3 | Abhängigkeiten zwischen Architekturbeschreibungen sind angelegt. <i>Hinweis: Jede Abhängigkeit ist im Namens-Feld mit einer Beschreibung zu versehen.</i> | A3-C |

Abbildung 10: Beispiel A3 : MASC : Architecture Correspondence

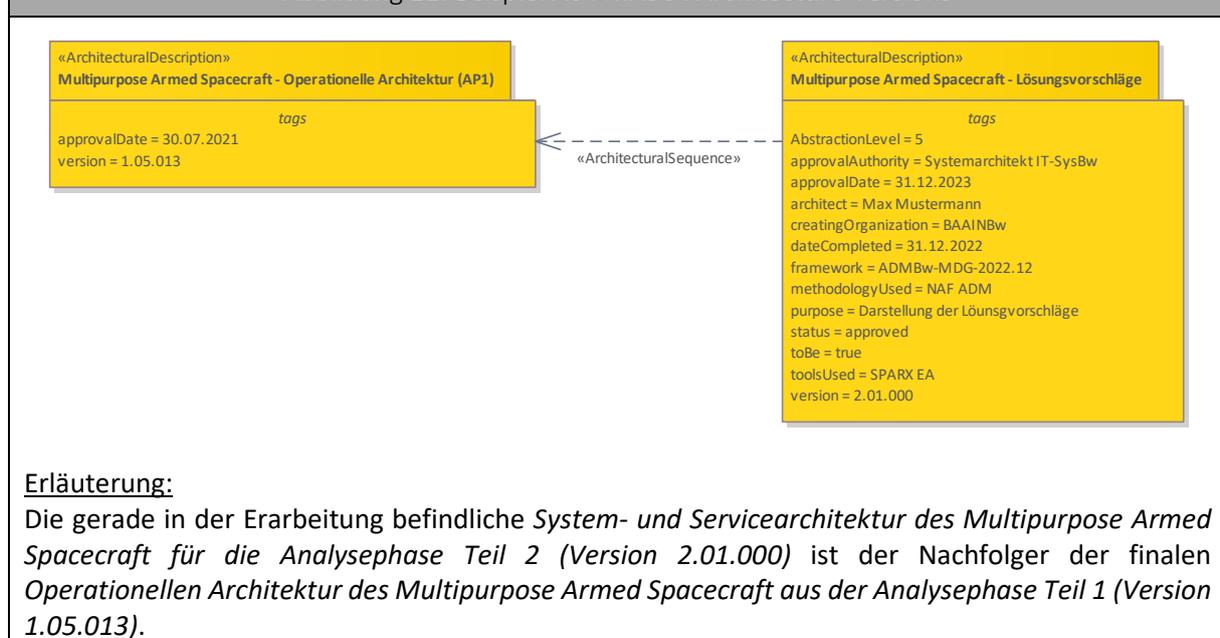


A6 – Architecture Versions

Der *A6 – Architecture Versions* erfasst die Versionsgeschichte einer Architektur.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| A6-MK1 | Die Informationen werden auf dem Diagramm <i>A6 : Projektname : Architecture Versions</i> im Package <i>A6/Architecture Versions</i> erfasst. | A6-A |
| A6-MK2 | Die relevanten Architekturbeschreibungen sind aus dem Katalog auf das Diagramm gezogen. | A6-B |
| A6-MK3 | Die Architekturbeschreibungen sind entsprechend der zeitlichen Abfolge in Beziehung gesetzt. <i>Hinweis: Die Version der Architekturbeschreibung wird im TaggedValue version erfasst.</i> | A6-C |

Abbildung 11: Beispiel A6 : MASC : Architecture Versions

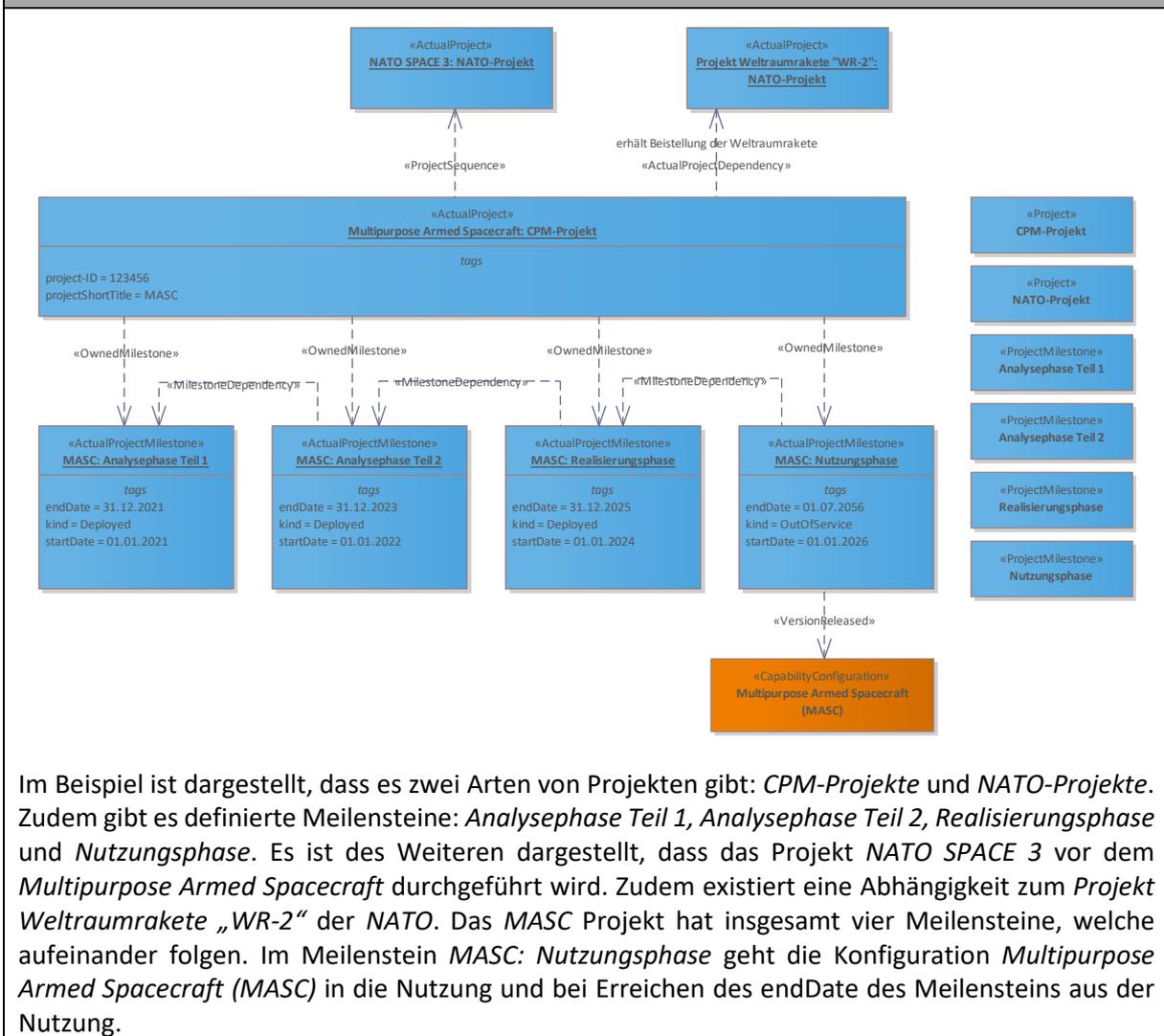


Lr – Lines of Development

Der *Lr – Lines of Development* behandelt Projektabhängigkeiten. Dazu werden die relevanten Programme und Projekte und ihre jeweiligen Meilensteine sowie die darüber realisierten Ressourcen dargestellt.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|-------------------------------|
| Lr-MK1 | Die Informationen werden auf dem Diagramm <i>Lr : Projektname : Lines of Development</i> im Package <i>Lr/ Lines of Development</i> erfasst. | Lr-A |
| Lr-MK2 | Alle relevanten Projekte sind anzulegen. <i>Hinweis:</i> Im Namensfeld wird der Projektname erfasst. Zusätzlich sind die TaggedValues Project-ID (Projekt-ID aus der Projekt-DB/IT-U FM) und ProjectShortTitle (Kurzname des Projektes) zu pflegen. | Lr-B |
| Lr-MK3 | Die relevanten zeitlichen Zusammenhänge und Abhängigkeiten zwischen den Projekten sind angeben. <i>Hinweis:</i> Das betrachtete Projekt ist von einem anderen Projekt abhängig, wenn die Ergebnisse oder Ressourcen des anderen Projekts Voraussetzung für einen erfolgreichen Verlauf des betrachteten Projekts sind. Ist eine Abhängigkeit gegeben, dann werden durch eine zeitliche Verzögerung des anderen Projekts die Ergebnisse des betrachteten Projekts ebenfalls zeitlich verzögert. | Lr-H Lr-I |
| Lr-MK4 | Der Aufbau des eigenen Projekts ist über Meilensteine dargestellt. Abhängigkeiten zwischen den Meilensteinen sind abgebildet. | Lr-Ba Lr-C Lr-D Lr-G |
| Lr-MK5 | Jede Ressource die zu einem bestimmten Meilenstein in Nutzung geht , ist mit dem entsprechenden Meilenstein verknüpft. <i>Hinweis:</i> TaggedValue startDate und endDate am entsprechenden Meilenstein sind zu pflegen, so dass der Nutzungsbeginn jeder Ressource erkennbar ist. | Lr-E Lr-F Lr-Ca |
| Lr-MK6 | Jede Ressource die zu einem bestimmten Meilenstein aus der Nutzung geht , ist mit dem entsprechenden Meilenstein verknüpft. <i>Hinweis:</i> Die TaggedValue startDate und endDate am entsprechenden Meilenstein sind zu pflegen, so dass das Nutzungsende jeder Ressource erkennbar ist. | Lr-E Lr-F Lr-Ca |

Abbildung 12: Beispiel Lr : MASC : Lines of Development



Im Beispiel ist dargestellt, dass es zwei Arten von Projekten gibt: *CPM-Projekte* und *NATO-Projekte*. Zudem gibt es definierte Meilensteine: *Analysephase Teil 1*, *Analysephase Teil 2*, *Realisierungsphase* und *Nutzungsphase*. Es ist des Weiteren dargestellt, dass das Projekt *NATO SPACE 3* vor dem *Multipurpose Armed Spacecraft* durchgeführt wird. Zudem existiert eine Abhängigkeit zum *Projekt Weltraumrakete „WR-2“* der *NATO*. Das *MASC* Projekt hat insgesamt vier Meilensteine, welche aufeinander folgen. Im Meilenstein *MASC: Nutzungsphase* geht die Konfiguration *Multipurpose Armed Spacecraft (MASC)* in die Nutzung und bei Erreichen des *endDate* des Meilensteins aus der Nutzung.

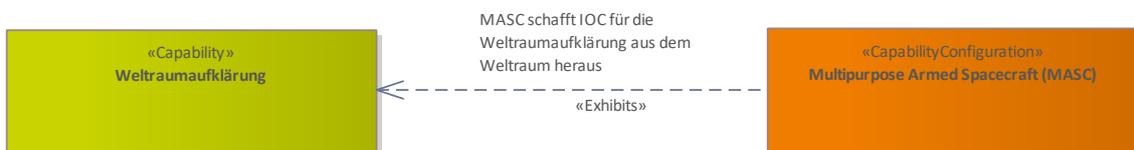
Cr – Capability Roadmap (optionaler Viewpoint)

Hinweis: Viewpoint ist nur zu erstellen, sofern keine Operationelle Architektur vorliegt.

Der *Cr – Capability Roadmap* behandelt die Darstellung der Ressourcen, die zur Bereitstellung einer Fähigkeit herangezogen werden.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| Cr-MK1 | Die Informationen werden auf dem Diagramm <i>Cr : Projektname : Capability Roadmap</i> im Package <i>Cr/Capability Roadmap</i> erfasst. | A6-A |
| Cr-MK2 | Die relevanten Ressourcen sind aus dem Katalog auf das Diagramm gezogen. | Cr-C |
| Cr-MK3 | Die relevanten Fähigkeiten sind aus dem Katalog auf das Diagramm gezogen. <i>Hinweis: Fähigkeiten werden nicht eigenständig durch ein Projekt angelegt. Im Rahmen des Startpaketes wird ein Katalog von Fähigkeiten (gegliedert in Funktionale Bausteine, NATO Targets und Fähigkeitsforderungen) durch den Operationellen Architekten der Bw bereitgestellt. Sollte der Katalog nicht vorliegen, kann dieser über den Operationellen Architekten bezogen werden.)</i> | Cr-B |
| Cr-MK4 | Die Ressourcen sind den Fähigkeiten zugeordnet. <i>Hinweis: Die Auswirkungen auf die Fähigkeitsentwicklung sind im Namen der Relation zwischen Ressource und Fähigkeit vermerkt. Informationen zu den für eine Ressource relevanten Fähigkeiten sind in der FFF zu finden.</i> | Cr-D |

Abbildung 13: Beispiel Cr : MASC : Capability Roadmap



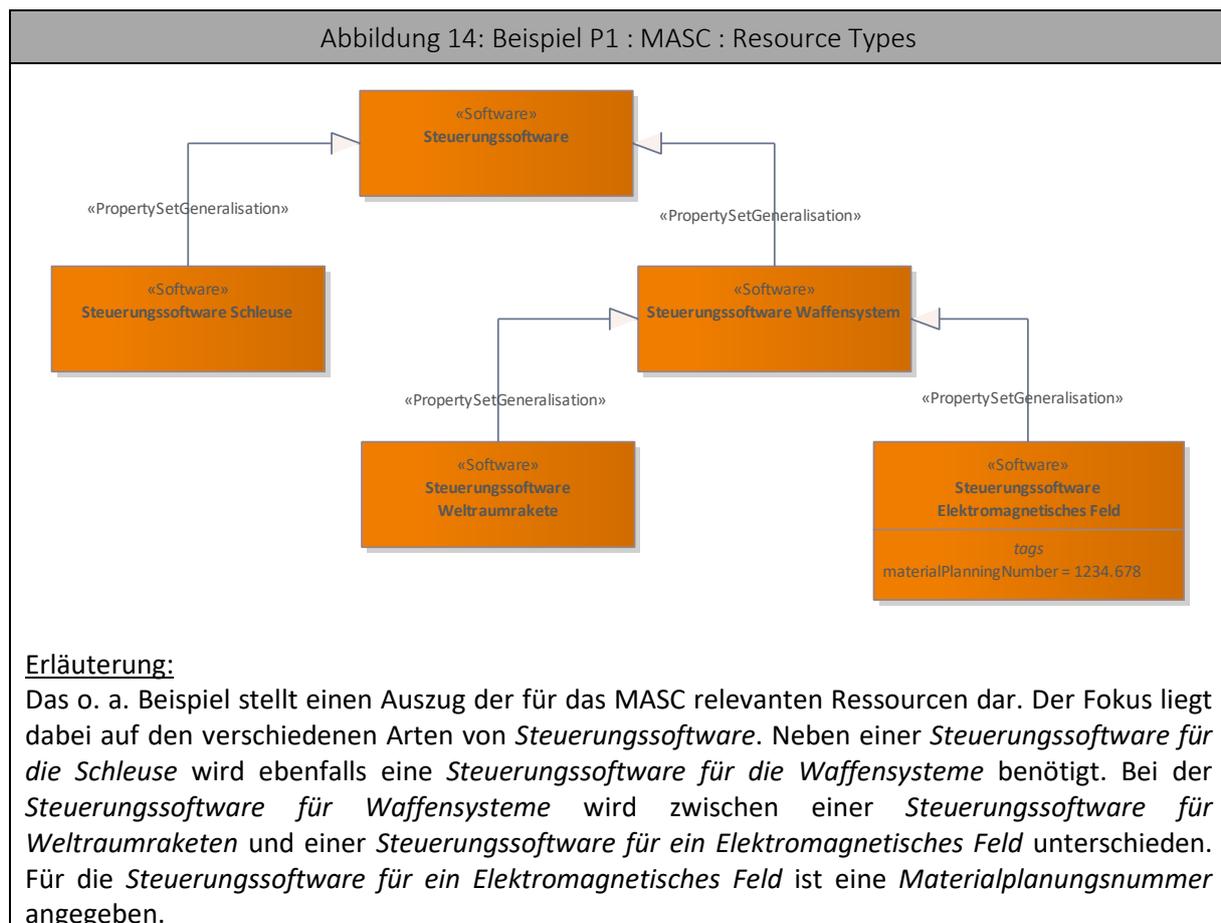
Im Beispiel ist dargestellt, dass und wie – nämlich als Anfangsbefähigung für die Fähigkeit – die Konfiguration MASC zur Fähigkeit Weltraumaufklärung beiträgt.

P1 – Resource Types

Der *P1 – Resource Types* behandelt die Spezifikation der Ressourcen sowie deren Einordnung in eine Taxonomie.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| P1-MK1 | Die Informationen werden auf dem Diagramm <i>P1 : Projektname : Resource Types</i> im Package <i>P1/Resource Types</i> erfasst. | P1-A |
| P1-MK2 | Jede relevante Ressource ist angelegt oder aus dem Katalog auf das Diagramm gezogen. | P1-B |
| P1-MK3 | Sofern vorhanden, ist die Materialplanungsnummer aus SASPF (TaggedValue materialPlanningNumber) an allen Elementen mit den Stereotypen CAPABILITYCONFIGURATION, RESOURCEARCHITECTURE, RESOURCEARTIFACT und SOFTWARE zu pflegen. | P1-C |
| P1-MK5 | Alle Ressourcen sind hierarchisch in eine Taxonomie einsortiert. Dabei wird immer die nächsthöhere Taxonomie-Ebene auf dem Diagramm angezeigt. | P1-D |

Abbildung 14: Beispiel P1 : MASC : Resource Types

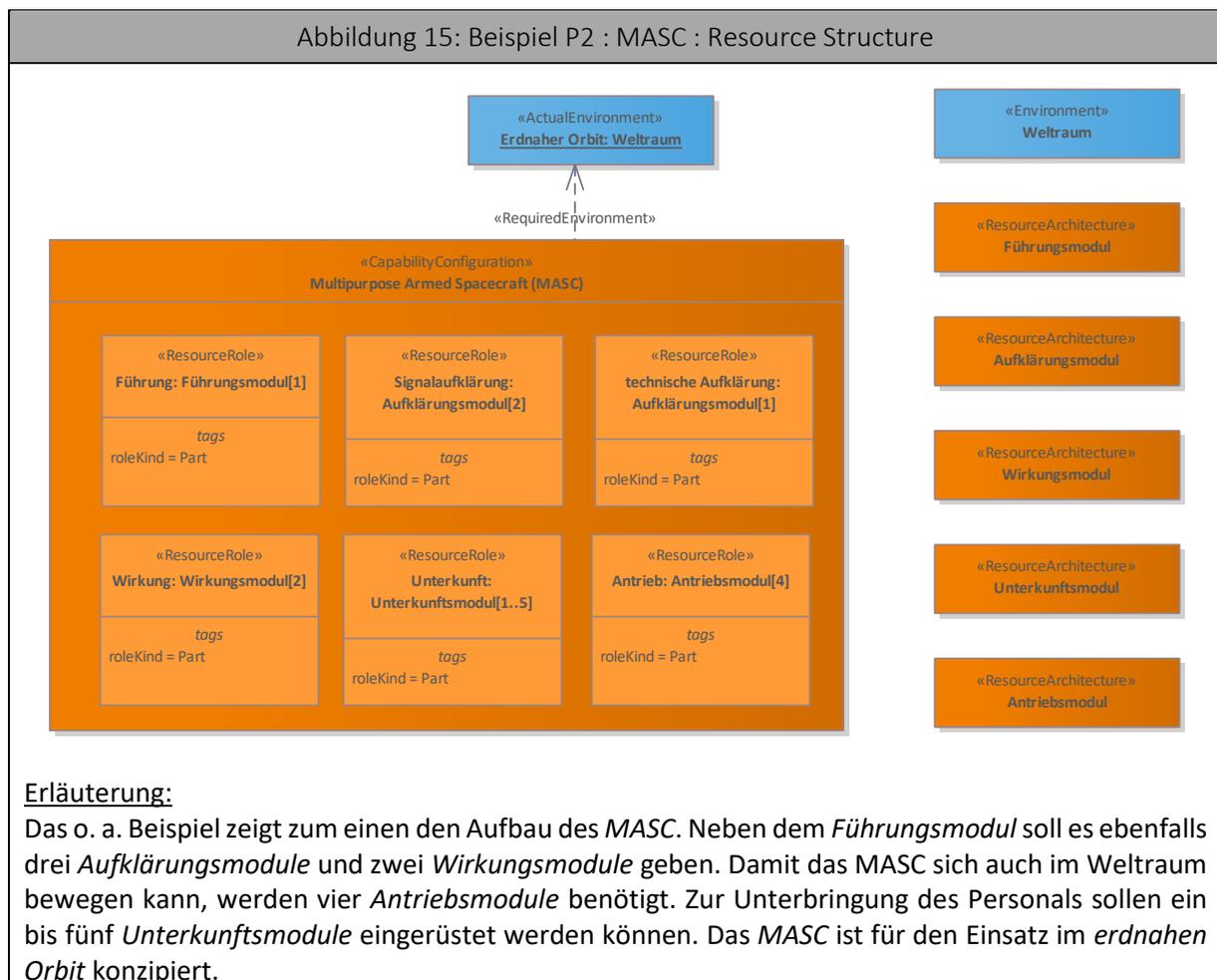


P2 – Resource Structure

Der *P2 – Resource Structure* behandelt die Zusammensetzung einer Ressource.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------------------|
| P2-MK1 | Die Informationen werden auf dem Diagramm <i>P2 : Projektname : Resource Structure</i> im Package <i>P2/ Resource Structure</i> erfasst. | P2-A |
| P2-MK2 | Jede relevante Ressource ist aus dem Katalog auf das Diagramm gezogen. | P2-B |
| P2-MK3 | Alle Komponenten der Ressourcen sind angelegt. <i>Hinweis: Beim Anlegen der Komponenten ist der TaggedValue „rolekind“ zu befüllen.</i> | P2-C P2-D |
| P2-MK4 | Bei Bedarf sind die einzelnen Komponenten mit Orts- und Umweltinformationen zu versehen. | P2-Da P2-Db P2-Dc P2-Dd |

Abbildung 15: Beispiel P2 : MASC : Resource Structure

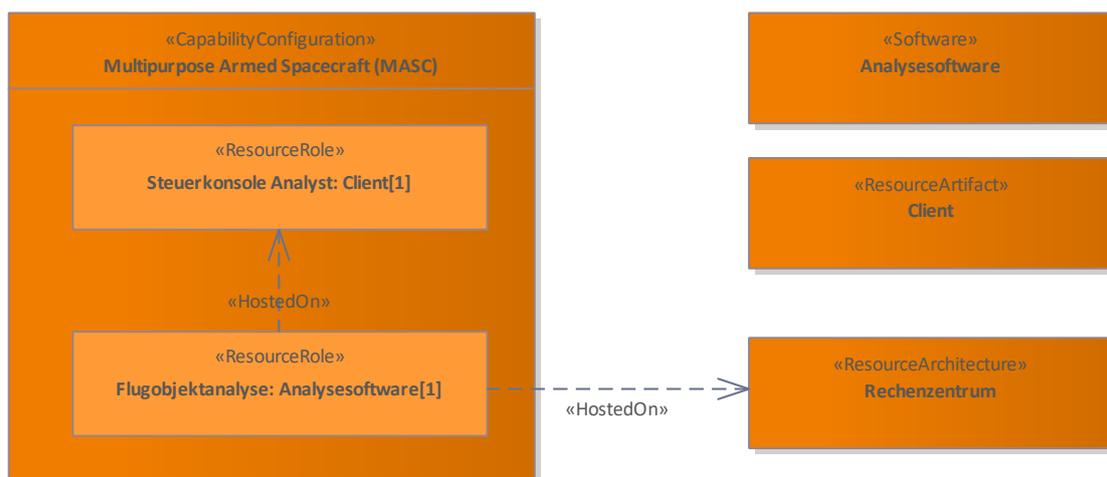


P2 – Application Hosting

Der *P2 – Application Hosting* beschreibt die Zuordnung von Anwendungen und genutzter Plattform.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| P2-MK5 | Die Informationen werden auf dem Diagramm <i>P2 : Projektname : Application Hosting</i> im Package <i>P2/ Application Hosting</i> erfasst. | P2-A |
| P2-MK6 | Alle zu beschreibenden Applikationen werden aus dem Katalog auf das Diagramm gezogen. | P2-B P2-C |
| P2-MK7 | Alle benötigten Infrastrukturelemente werden neu angelegt oder aus dem Modellelementekatalog auf das Diagramm gezogen | P2-B |
| P2-MK8 | Alle Applikationen werden den Infrastrukturkomponenten zugewiesen, auf denen diese betrieben werden. | P2-P |

Abbildung 16: Beispiel P2 : MASC : Application Hosting



Erläuterung:

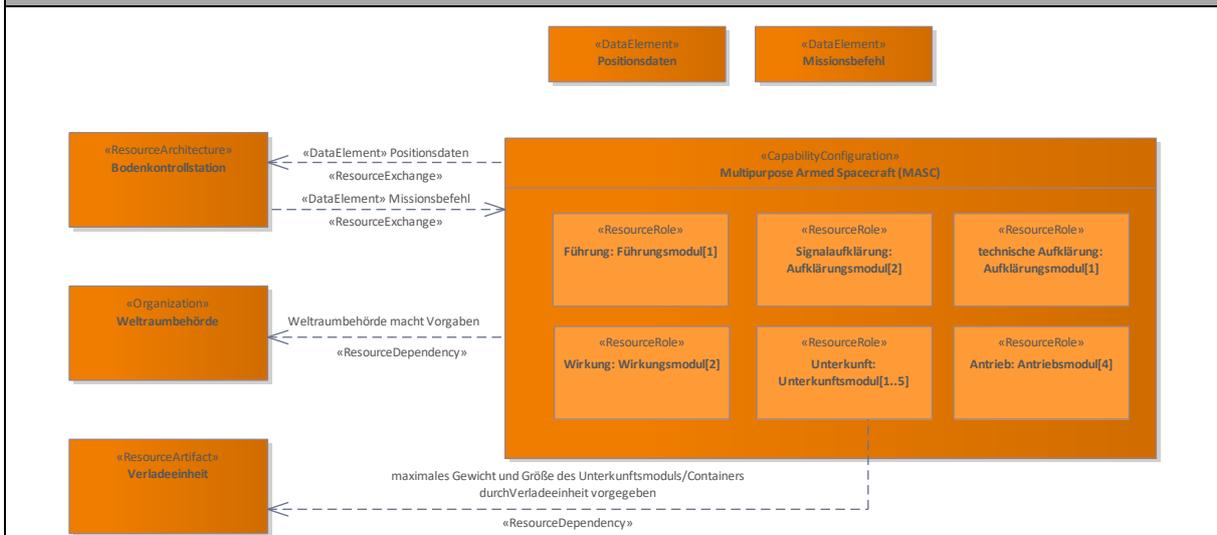
Das o. a. Beispiel zeigt die *Analysesoftware* als Bestandteil des *MASC*. Diese wird auf dem *Client* und im *Rechenzentrum* betrieben.

P2 – External Dependencies

Der *P2 – External Dependencies* beschreibt die Abhängigkeiten der zu betrachtenden Ressource.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| P2-MK9 | Die Informationen werden auf dem Diagramm <i>P2 : Projektname : External Dependencies</i> im Package <i>P2/ External Dependencies</i> erfasst. | P2-A |
| P2-MK10 | Jede relevante Ressource und deren Komponenten ist aus dem Katalog auf das Diagramm gezogen. | P2-B P2-C |
| P2-MK11 | Alle externen Datenflüsse sind angelegt. <i>Hinweis: Jeder Datenfluss muss mindestens ein Datenelement transportieren. Hierbei sind bidirektionale Verbindungen unzulässig.</i> | P2-E P2-F |
| P2-MK12 | Alle externen Abhängigkeiten sind angelegt. <i>Hinweis: Jede Abhängigkeit muss eine Beschreibung besitzen, die die Art der Abhängigkeit angibt. Eine Systemabhängigkeit zu einem anderen System liegt dann vor, wenn das betrachtete System ohne das andere System nicht oder nur sehr eingeschränkt nutzbar ist. Eine organisatorische Abhängigkeit zu einer Organisation liegt dann vor, wenn die Organisation Vorgaben macht oder die Tätigkeiten der Organisation für eine Nutzbarkeit des Systems Voraussetzung sind. Hierbei sind bidirektionale Verbindungen unzulässig.</i> | P2-K |
| P2-MK13 | Wenn ein System direkt von einem Service abhängig ist, dann ist eine ResourceToServiceDependency zu modellieren. | |

Abbildung 17: Beispiel P2 : MASC : External Dependencies



Erläuterung:

Im o. a. Beispiel wird zum einen dargestellt, dass das MASC einen bidirektionalen Datenaustausch mit der Bodenkontrollstation durchführt. Zum anderen gibt es zwei dargestellte Abhängigkeiten. Auf der einen Seite hängen die Größe und das Gewicht des Unterkunftsmoduls von der Verladeeinheit am Boden ab, die das Unterkunftsmodul am Boden in das Raumschiff verlädt. Auf der anderen Seite muss das MASC die Vorgaben der Weltraumbehörde berücksichtigen.

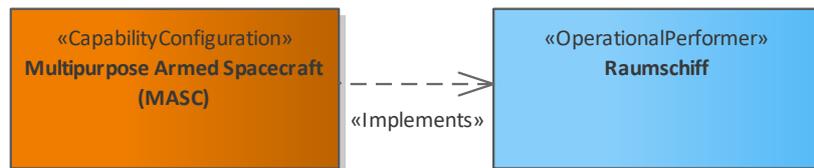
P2 – Node Realization (optionaler Viewpoint)

Hinweis: Viewpoint ist nur zu erstellen, sofern eine Operationelle Architektur vorliegt.

Der *P2 – Node Realization* beschreibt die Zuordnung von Ressourcen zu den durch diese realisierten Aufgabenträgern der Operationellen Architektur.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| P2-MK14 | Die Informationen werden auf dem Diagramm <i>P2 : Projektname : Node Realization</i> im Package <i>P2/ Node Realization</i> erfasst. | P2-A |
| P2-MK15 | Jede relevante Ressource ist aus dem Katalog auf das Diagramm gezogen. | P2-B |
| P2-MK16 | Jeder relevante Aufgabenträger ist aus dem Katalog auf das Diagramm gezogen. | P2-G |
| P2-MK17 | Alle relevanten Ressourcen sind den Aufgabenträgern zugeordnet. | P2-H |

Abbildung 18: Beispiel P2 : MASC : Node Realization



Erläuterung:

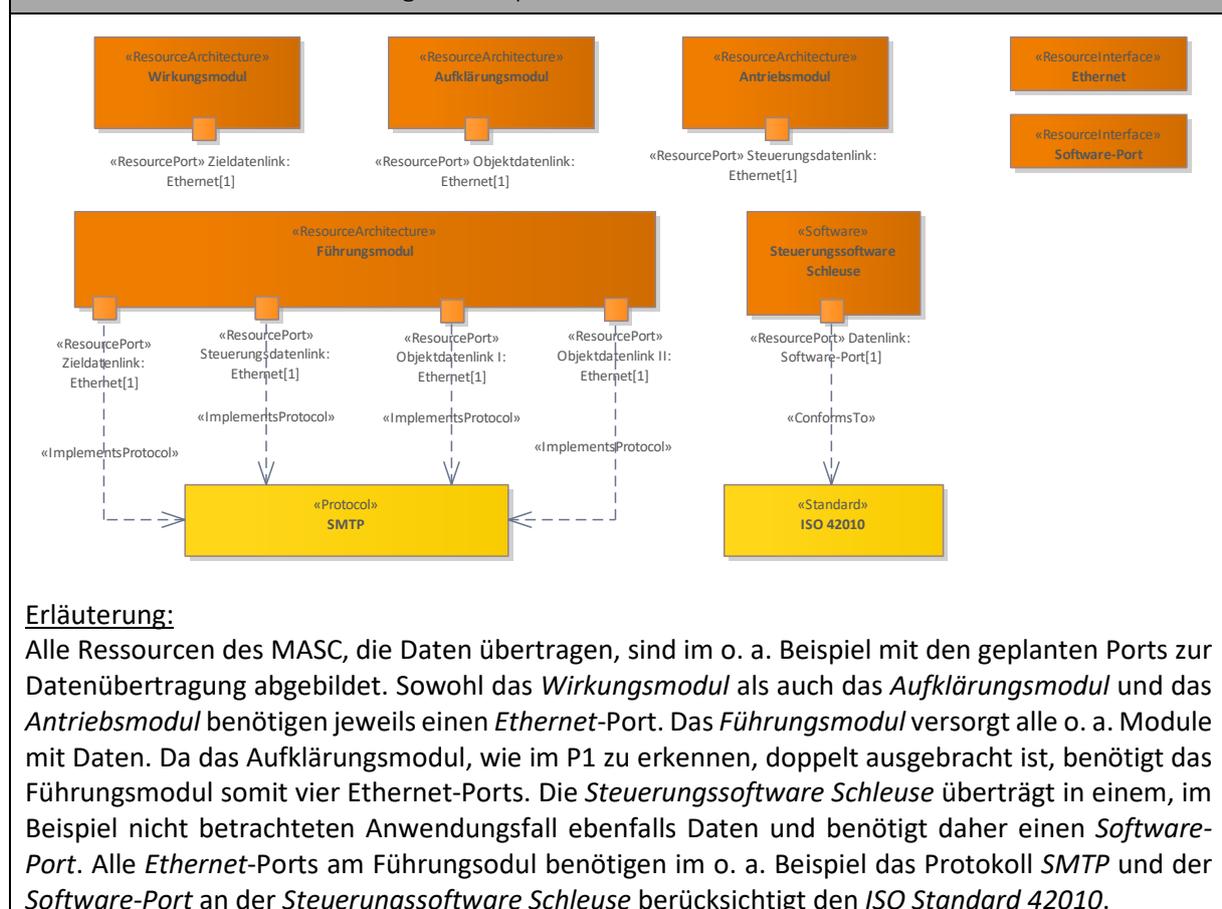
Im o. a. Beispiel wird dargestellt, dass das *MASC* dazu dient, den in der FFF beschriebenen, logischen Rollen- und Aufgabenträger *Raumschiff* umzusetzen.

P3 – Resource Interface

Der *P3 – Resource Interfaces* beschreibt die Systemschnittstellen.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|--------------------------------------|
| P3-MK1 | Die Informationen werden auf dem Diagramm <i>P3 : Projektname : Resource Interfaces</i> im Package <i>P3/Resource Interfaces</i> erfasst. | P3-A |
| P3-MK2 | Jede relevante Ressource ist aus dem Katalog auf das Diagramm gezogen. | P3-B |
| P3-MK3 | Alle genutzten Schnittstellen sind der Ressource zugewiesen. Jede Schnittstelle ist mit einem Standard oder Protokoll spezifiziert | P3-D P3-G P3-H P3-I P3-J |

Abbildung 19: Beispiel P3 : MASC : Resource Interfaces

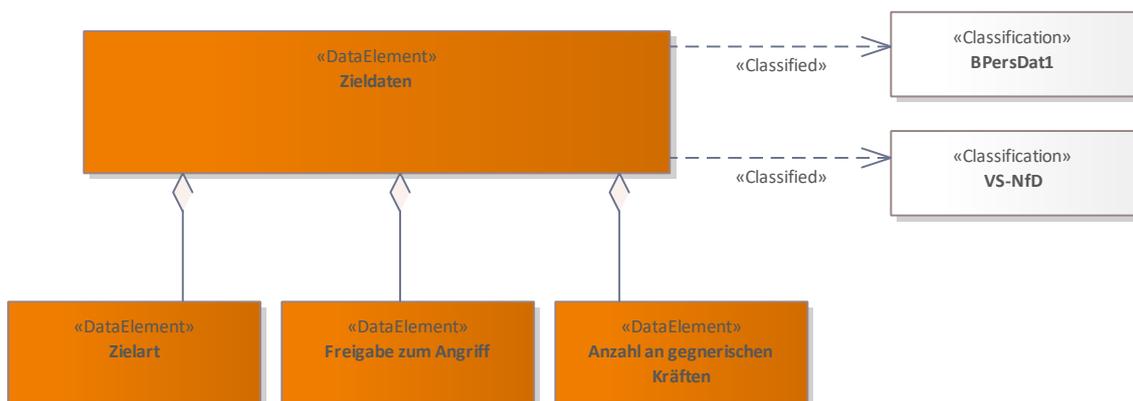


P7 – Data Elements

Der *P7 – Data Elements* behandelt die Struktur der Datenelemente, die in der Architektur untereinander ausgetauscht werden.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|-------------------------------|
| P7-MK1 | Die Informationen werden auf dem Diagramm <i>P7 : Projektname : Data Elements</i> im Package <i>P7/Data Elements</i> erfasst. | P7-A |
| P7-MK2 | Alle relevanten Datenelemente sind angelegt. <i>Hinweis: An allen Datenelementen ist der Geheimhaltungsgrad sowie der Schutzbereich abzubilden. Die CLASSIFICATION-Elemente sind aus dem StEka zu verwenden.</i> | P7-B P7-Ba P7-I P7-J |
| P7-MK4 | Alle Datenelemente sind je nach Art der Zusammenhänge mittels der Konnektoren AGGREGATION, ASSOCIATION, COMPOSITION und GENERALIZATION in Beziehung gesetzt. | P7-C |

Abbildung 20: Beispiel P7 : MASC : Data Elements



Erläuterung:

Der o. a. View stellt dar, dass sich das Datenelement *Zieldaten* aus drei anderen Datenelementen zusammensetzt. Zu diesen zählen die *Zielart*, die *Anzahl an generischen Kräften* und die *Freigabe zum Angriff*. Für die *Zieldaten* sind der *Geheimhaltungsgrad (VS-NfD)* und der *Schutzbereich (BPersDat)* angegeben.

P7 – Data Model

Der *P7 – Data Model* ordnet alle Datenelemente der Architektur in ein Datenmodell ein.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| P2-MK1 | Die Informationen werden auf dem Diagramm <i>P7 : Projektname : Data Model</i> im Package <i>P7/Data Model</i> erfasst. | P2-A |
| P2-MK2 | Das Datenmodell ist auf dem Diagramm angelegt. <i>Hinweis: Das TaggedValue kind ist auf Physical gesetzt und eingeblendet.</i> | P7-D P7-E |
| P2-MK4 | Alle relevanten Datenelemente sind in das Datenmodell eingeordnet und werden auf dem Diagramm eingeblendet. | P7-F |

Abbildung 21: Beispiel P7 : MASC : Data Model



Erläuterung:

Alle im Rahmen des Lösungsvorschlages zum MASC benannten, zu übertragenden Daten werden in diesem View zum physischen *Datenmodell des Lösungsvorschlages* hinzugefügt. Neben den *Zieldaten* gehören ebenfalls *Objektdaten* und *Steuerungsdaten* zum dargestellten Datenmodell.

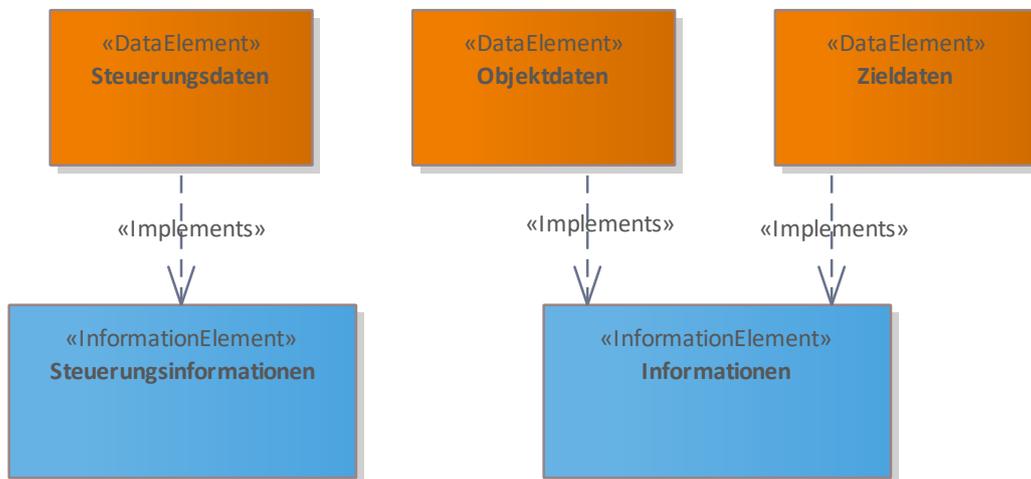
P7 – InformationElement Realization (optional Viewpoint)

Hinweis: Viewpoint ist nur zu erstellen, sofern eine Operationelle Architektur vorliegt.

Der *P7 – InformationElement Realization* ordnet die Datenelemente den durch diese realisierten Informationselemente der Operationellen Architektur zu.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| P2-MK1 | Die Informationen werden auf dem Diagramm <i>P2 : Projektname : InformationElement Realization</i> im Package <i>P2/InformationElement Realization</i> erfasst. | P2-A |
| P7-MK2 | Jedes relevante Datenelement ist aus dem Katalog auf das Diagramm gezogen. | P7-B |
| P7-MK3 | Jedes relevante Informationselement ist aus dem Katalog auf das Diagramm gezogen. | P7-G |
| P2-MK4 | Alle Datenelemente sind auf die durch diese realisierten Informationselemente abgebildet. | P2-H |

Abbildung 22: Beispiel P7 : MASC : InformationElement Realization



Erläuterung:

Wie schon im P2 - Resource Structure - Operational Realization geht es auch in diesem View um die geplante Umsetzung von Elementen, die in der FFF beschrieben sind. Dieses Mal liegt der Fokus jedoch nicht auf Rollen- und Aufgabenträgern und Ressourcen, sondern auf Informations- und Datenelementen. Die in der FFF beschriebenen *Steuerungsinformationen* sollen durch *Steuerungsdaten* umgesetzt werden. Die *Objektdaten* und *Zieldaten* dienen der Realisierung der in der FFF benannten *Informationen*.

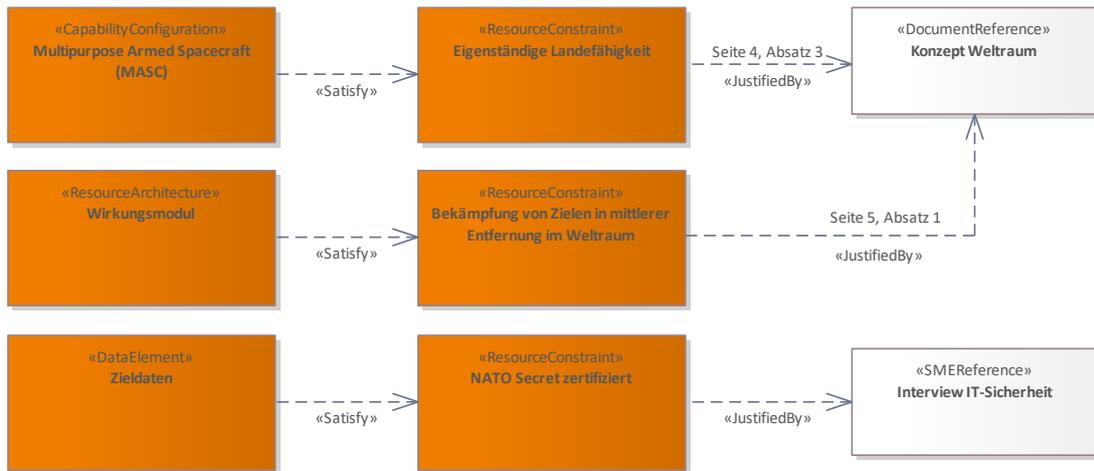
P8 – Resource Constraints (optional Viewpoint)

Hinweis: Implementierungsvorgaben können in der gesamten Physical Specification Row modelliert werden. Zweckmäßiger Weise werden Implementierungsvorgaben in die den Viewpoints erfasst, deren Inhalt sie betreffen. So ist eine Implementierungsvorgabe zu einer Schnittstelle im P3 sinnvoll verortet. Der P8 dient der Zusammenfassung aller Implementierungsvorgaben an zentraler Stelle in der Architektur, ist jedoch nicht zwingend zu erstellen.

Der *P8 – Resource Constraints* stellt die Implementierungsvorgaben für die Ressourcen dar. Zu den Implementierungsvorgaben zählen die Nutzung bereits vorhandener Ressourcen, die Herstellung von Interoperabilität zu anderen Ressourcen oder Einschränkungen beim Einsatz bestimmter Ressourcen.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| P8-MK1 | Die Informationen werden auf dem Diagramm <i>P8 : Projektname : Resource Constraints (optional Viewpoint)</i> im Package <i>P8/ Resource Constraints</i> erfasst. | P2-A |
| P8-MK2 | Jede relevante Implementierungsvorgabe ist aus dem Katalog auf das Diagramm gezogen. | P7-B |
| P8-MK3 | Alle Elemente, auf die sich die Implementierungsvorgaben beziehen sind aus dem Katalog auf das Diagramm gezogen und mit der jeweiligen Implementierungsvorgabe in Beziehung gesetzt. | P8-C P8-D |
| P8-MK6 | Für jede Implementierungsvorgabe ist die zugehörige Referenz aus dem Katalog auf das Diagramm gezogen und auf die Implementierungsvorgabe abgebildet. | P8-G P8-H |

Abbildung 23: Beispiel P8 : MASC : Resource Constraints

**Erläuterung:**

Der o. a. View stellt dar, dass sich aus dem Dokument *Konzept Weltraum* zwei Vorgaben und Rahmenbedingungen ableiten. Zum einen muss gem. Seite 4, Absatz 3 das MASC in der Lage sein *eigenständig zu landen* und zum anderen muss das *Wirkungsmodul* gem. Seite 5, Absatz 1 fähig sein, *Ziele in mittlerer Entfernung im Weltraum zu bekämpfen*. Zusätzlich zum Genannten geht aus einem *Interview zum Thema IT-Sicherheit* hervor, dass sämtliche *Zieldaten NATO SECRET zertifiziert* sein müssen.

P8 – Resource Constraint Realization (optional Viewpoint)

Hinweis: Viewpoint ist nur zu erstellen, sofern eine Operationelle Architektur vorliegt.

Hinweis: Implementierungsvorgaben können in der gesamten Physical Specification Row modelliert werden. Der P8 dient der Zusammenfassung aller Implementierungsvorgaben an zentraler Stelle in der Architektur, ist jedoch nicht zwingend zu erstellen.

Der *P8 – Resource Constraint Realization* stellt die Ableitung von Implementierungsvorgaben für die Ressourcen aus strategischen, operationellen oder servicerelevanten Vorgaben dar.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| P8-MK1 | Die Informationen werden auf dem Diagramm <i>P8 : Projektname : Resource Constraint Realization (optional Viewpoint)</i> im Package <i>P8/Resource Constraint Realization</i> erfasst. | P2-A |
| P8-MK2 | Jede relevante Implementierungsvorgabe ist aus dem Katalog auf das Diagramm gezogen. | P7-B |
| P8-MK3 | Jede benötigte strategische, operationelle oder servicerelevante Vorgabe ist aus dem Katalog auf das Diagramm gezogen. | P8-E |
| P8-MK6 | Alle Implementierungsvorgaben sind einer strategischen, operationellen oder servicerelevanten Vorgabe zugeordnet. | P8-F |

Abbildung 24: Beispiel P8 : MASC : Resource Constraint Realization

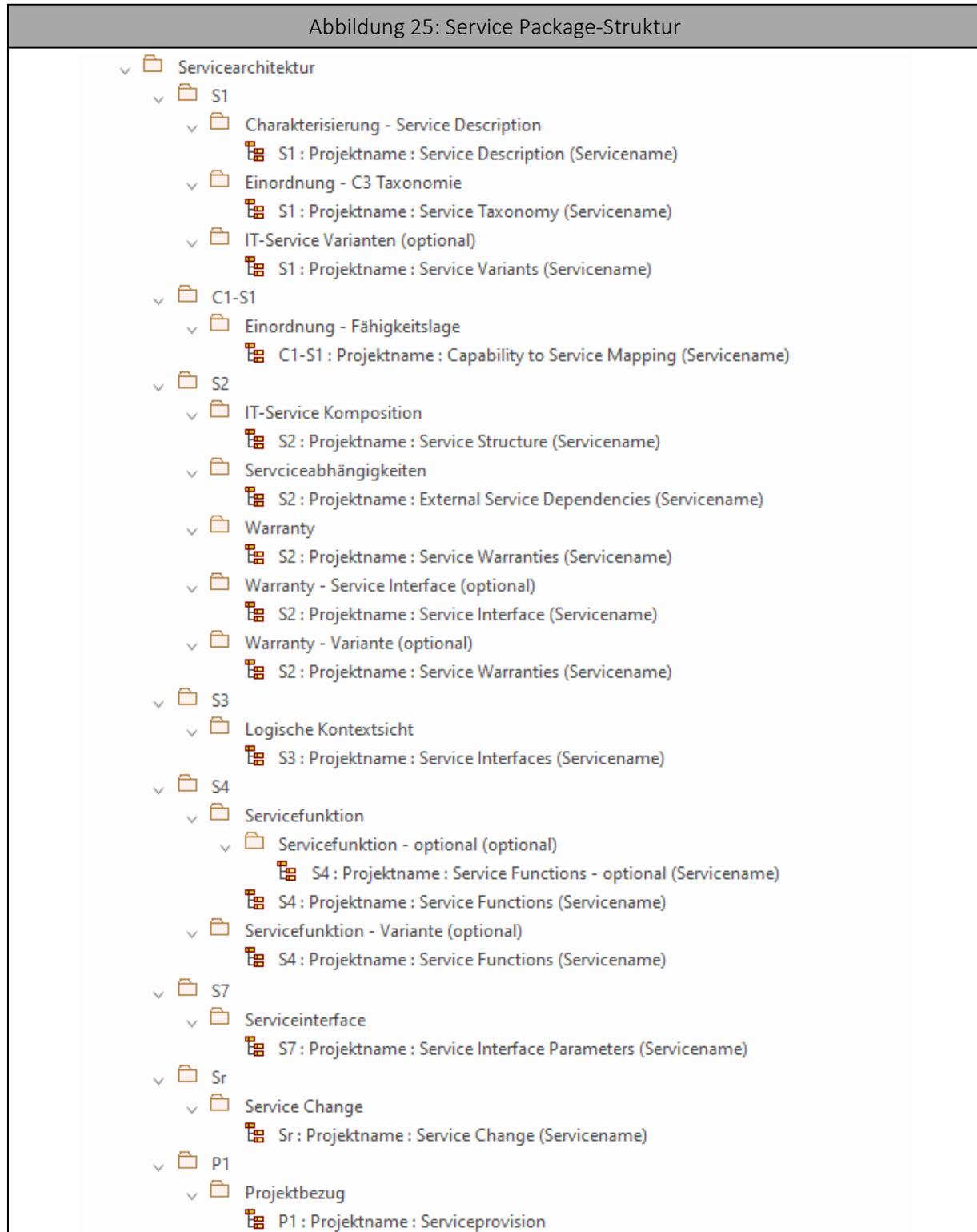


Erläuterung:

Die Vorgabe, dass das MASC *Ziele in mittlerer Entfernung im Weltraum bekämpfen* können muss, setzt die operationelle Vorgabe, dass das MASC *Ziele im Weltraum bekämpfen* können muss, um.

Package-Struktur der Servicebeschreibung

Im Rahmen der Servicemodellierung ist die in Abbildung 25 abgebildete Package-Struktur vorgegeben. Die Package-Struktur bildet die Basis für die automatisierte Weiterverarbeitung der Informationen und muss eingehalten werden. Im folgenden Abschnitt werden die einzelnen Viewpoints dargestellt.



S1 – Service Description

Der *S1 – Service Description* dient der Erfassung der Servicebeschreibung und der Einordnung des Service in eine der drei Ebenen des Portfolios.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| S1-MK1 | Die Informationen werden auf dem Diagramm <i>S1 : Projektname : Service Description (Servicename)</i> im Package <i>S1/ Charakterisierung - Service Description</i> erfasst. | S1-A |
| S1-MK2 | <p>Der zu beschreibende Service wird aus dem Standardelementekatalog auf das Diagramm gezogen oder neu angelegt. Der Service ist mit einem Namen und einer Beschreibung zu versehen. Anschließend ist die Beschreibung des Service auf dem Diagramm einzublenden.</p> <p>Hinweis Servicename: Der Name eines Service enthält mitunter ein Suffix. Welches Suffix dabei zu verwenden ist, hängt von der Ebene des Service im Portfolio ab. Es wird zwischen drei Ebenen unterschieden:</p> <ul style="list-style-type: none"> • ServiceSpecifications der IT-Serviceebene (vorher: IT-Services der Ebene 1) erhalten das Suffix “Service Bw” (z. B. Platform Service Bw). • ServiceSpecifications der IT-Produktebene (vorher: IT-Services der Ebene 2) erhalten das Suffix “Service” (z. B. Platform Core Service). • ServiceSpecifications der Technikebene (vorher: IT-Services der Ebene 3) erhalten kein Suffix (z. B. Platform Common Core). • IT-Service Module erhalten das Präfix “IT-Service Modul” (z. B. IT-Service Modul GMN 5) <p>Hinweis Servicebeschreibung: Es ist die wesentliche Leistung des IT-Service/IT-Service Modul/IT-Service Package aus Nutzersicht in wenigen Sätzen zusammenfassend zu beschreiben (Kurzbeschreibung).</p> | S1-B |
| S1-MK3 | <p>Wenn es sich um einen neu erstellten Service handelt, an dem die folgenden TaggedValues noch nicht angelegt sind, sind diese mit dem Skript „NAFv4-ADMBw/AddTaggedValueToService“ hinzuzufügen und entsprechend zu befüllen.</p> <p>Folgende TaggedValues sind zu pflegen:</p> <ul style="list-style-type: none"> - ITSD_IT-ServKatBw - ITSD_IT-SVEB - ITSD_Projektreferat - ITSD_ServiceOwner - ITSD_ServiceProvider - ITSD_ServiceStatus - ITSD_ServiceTyp - ITSD_ServiceEbene | |

Abbildung 26: Beispiel S1 : MASC : Service Description (Spacecraft Information Service Bw)

Erläuterung:

Das Beispiel stellt den *Spacecraft Information Service Bw* inkl. Beschreibung dar. Es handelt sich um einen Service der Ebene 1 (am Begriff *Service Bw* am Ende im Namen zu erkennen).

S1 – Service Taxonomy

Der *S1 – Service Taxonomy* ordnet den Service in die C3-Taxonomie ein.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| S1-MK1 | Die Informationen werden auf dem Diagramm <i>S1 : Projektname : Service Taxonomy (Servicename)</i> im Package <i>S1/ Einordnung - C3 Taxonomie</i> erfasst. | S1-A |
| S1-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. Zusätzlich sind die relevanten Services (Servicekategorien der C3-Taxonomie, denen der zu beschreibende Service zugeordnet werden soll) aus dem Standardelementekatalog auf das Diagramm zu ziehen. | S1-B |
| S1-MK3 | Der zu beschreibende Service wird nun über den Konnektor SERVICECLASSIFICATION auf die Servicekategorie abgebildet. | S1-D |

Abbildung 27: Beispiel S1 : MASC : Service Taxonomy (Spacecraft Information Service Bw)



Erläuterung:

Das o. a. Beispiel stellt einen Auszug der für das MASC relevanten Services dar. Der Fokus liegt dabei auf dem *Spacecraft Information Service Bw*. Dieser ist zum einen in der Taxonomie der Servicekategorie *Information Management Services* zugeordnet (sowohl Service als auch Servicekategorie werden mit dem Stereotyp SERVICE SPECIFICATION modelliert).

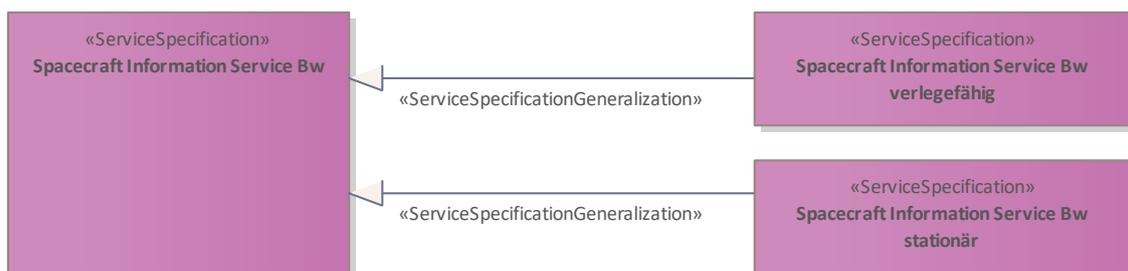
S1 – Service Varianten (optionaler Viewpoint)

Der *S1 – Service Varianten* erfasst die verschiedenen Varianten eines Service.

Hinweis: Der Viewpoint ist nur zu erstellen sofern Service Varianten vorhanden sind.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| S1-MK1 | Die Informationen werden auf dem Diagramm <i>S1 : Projektname : Service Variants (Servicename)</i> im Package <i>S1/ IT-Service Varianten</i> erfasst. | S1-A |
| S1-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. <i>Hinweis Servicevariantenname:</i> <i>Der Name einer Servicevariante enthält den Servicename, zu welchem die Servicevariante gehört. Zusätzlich wird eine Variantenspezifische Ergänzung beigefügt (z. B. Spacecraft Information Service Bw verlegefähig).</i> | S1-B |
| S1-MK3 | Sofern der Service über eine oder mehrere Servicevarianten verfügt, sind diese anzulegen und mit einem Namen und einer Beschreibung zu versehen. | S1-B |
| S1-MK4 | Alle Servicevarianten werden nun über den Konnektor SERVICESPECIFICATIONGENERALIZATION auf den zugehörigen Service abgebildet. | S1-D |

Abbildung 28: Beispiel S1 : MASC : Service Varianten (Spacecraft Information Service Bw)



Erläuterung:

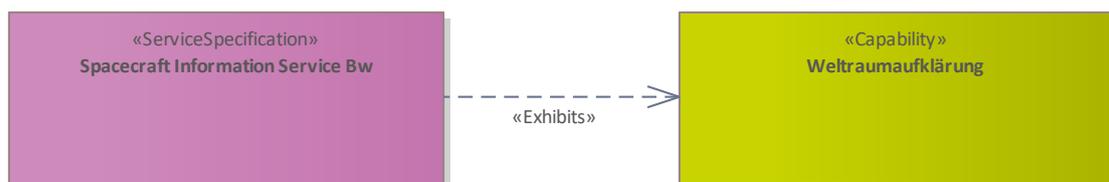
Das o. a. Beispiel stellt einen Auszug der für das MASC relevanten Services dar. Der Fokus liegt dabei auf dem *Spacecraft Information Service Bw*. Der *Spacecraft Information Service Bw* wird in den zwei Varianten *Spacecraft Information Service Bw verlegefähig* & *Spacecraft Information Service Bw stationär* ausgebracht (sowohl Service als auch Servicevariante werden mit dem Stereotyp SERVICESPECIFICATION modelliert).

C1-S1 – Capability to Service Mapping

Der *C1-S1 – Capability to Service Mapping* dient der Einordnung des Services in die Fähigkeitsslage.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| C1-S1-MK1 | Die Informationen werden auf dem Diagramm <i>C1-S1 : Projektname : Capability to Service Mapping (Servicename)</i> im Package <i>C1-S1/ Einordnung - Fähigkeitsslage</i> erfasst. | C1-S1-A |
| C1-S1-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. Zusätzlich sind die relevanten Fähigkeiten aus dem Standardelementekatalog auf das Diagramm zu ziehen. <i>Hinweis:</i> Fähigkeiten werden nicht eigenständig durch ein Projekt angelegt. Im Rahmen des Startpaketes wird ein Katalog von Fähigkeiten (gegliedert in Funktionale Bausteine, NATO Targets und Fähigkeitsforderungen) durch den Auftraggeber bereitgestellt. Sollte der Katalog nicht vorliegen, kann dieser über den Auftraggeber von BAAINBwStabDigitalisierungD1@bundeswehr.org bezogen werden.) | C1-S1-B C1-S1-C |
| C1-S1-MK5 | Der zu beschreibende Service wird über den Konnektor EXHIBITS der Fähigkeit zugeordnet. | C1-S1-D |

Abbildung 29: Beispiel C1-S1 : MASC : Capability to Service Mapping (Spacecraft Information Service Bw)



Erläuterung:

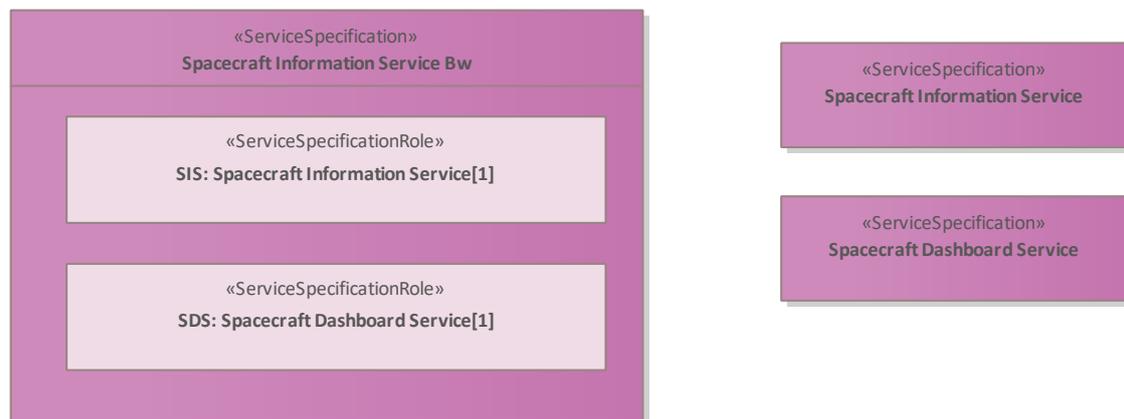
Im o. a. Beispiel leistet der Service *Spacecraft Information Service Bw* einen Beitrag zur Fähigkeit *Weltraumaufklärung*.

S2 – Service Structure

Der *S2 – Service Structure* beschreibt die Zusammensetzung des Service aus dritten Services.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| S2-MK1 | Die Informationen werden auf dem Diagramm <i>S2: Projektname : Service Structure (Servicename)</i> im Package <i>S2/ IT-Service Komposition</i> erfasst. | S2-A |
| S2-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. | S2-B |
| S2-MK3 | Es sind nun alle (Teil-)Service des zu beschreibenden Service zu erfassen. <i>Hinweis Teilservice:</i> <i>Das Konzept des Teilservice ist dann zu nutzen, wenn der Gesamtservice ohne den Teilservice nicht deployed werden kann, da ein zentraler Bestandteil fehlt. Der Teilservice muss als Serviceinstanz im Gesamtservice ausgebracht werden. Dabei können nur Services einer tieferen Ebene verwendet werden.</i> <i>Services der Ebene 1 dürfen nur aus Services der Ebene 2 komponiert werden.</i> <i>IT-Service Module dürfen nur aus Ebene 1 Services komponiert werden.</i> | S2-C |

Abbildung 30: Beispiel S2 : MASC : Service Structure (Spacecraft Information Service Bw)



Erläuterung:

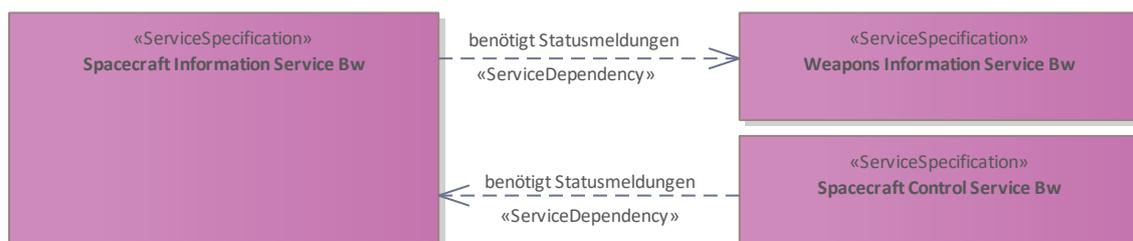
Der im o. a. Beispiel ausgewählte *Spacecraft Information Service Bw* setzt sich aus den beiden Teilservices, dem *Spacecraft Information Service* und dem *Spacecraft Dashboard Service*, zusammen.

S2 – External Service Dependencies

Der *S2 – External Service Dependencies* stellt die Abhängigkeiten des Service dar.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| S2-MK1 | Die Informationen werden auf dem Diagramm <i>S2 : Projektname : External Service Dependencies (Servicename)</i> im Package <i>S2/Serviceabhängigkeiten</i> erfasst. | S2-A |
| S2-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. Zusätzlich sind alle Services, zu denen eine Abhängigkeit besteht, auf dem Diagramm abzulegen. | S2-B S2-D |
| S2-MK3 | Ausgehend vom zu beschreibenden Service sind nun die Abhängigkeiten angelegt. Jede Abhängigkeit ist mit einer Beschreibung im Namensfeld des Konnektors zu versehen. Hinweis Direktionalität: <i>Bei bidirektionalen Abhängigkeiten müssen zwei unidirektionale Konnektoren modelliert werden.</i> Hinweis Serviceabhängigkeit: <i>Das Konzept der Serviceabhängigkeit ist dann zu nutzen, wenn der Gesamtservice ohne die hier aufgeführte Abhängigkeit nur mit Einschränkungen nutzbar ist. Es werden nur Services der Ebene 1 betrachtet.</i> | S2-E |

Abbildung 31: Beispiel S2 : MASC : External Service Dependencies (Spacecraft Information Service Bw)



Erläuterung:

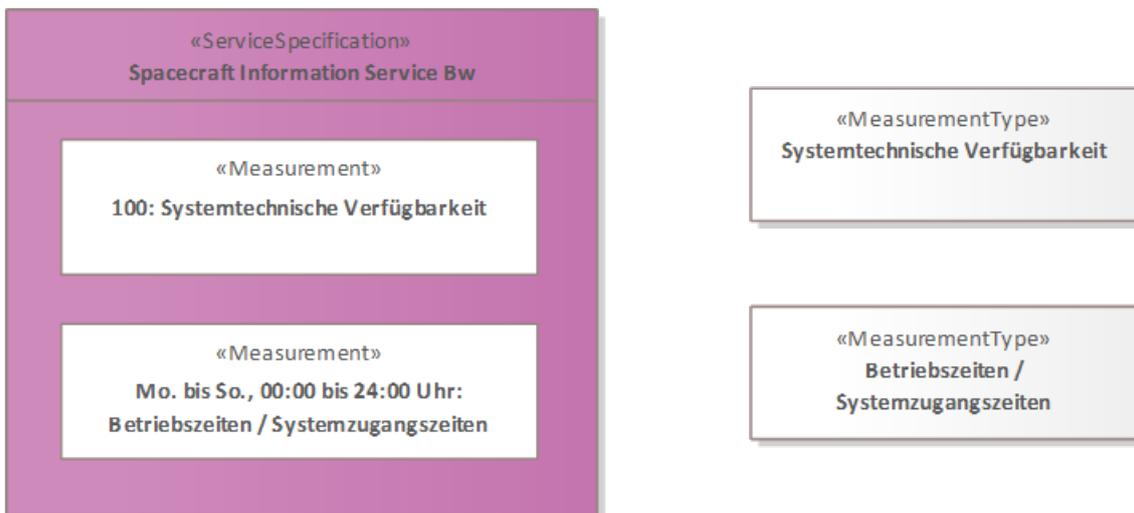
Im o. a. Beispiel sind einige Abhängigkeiten des *Spacecraft Information Service Bw* aufgezeigt. Der Service ist abhängig vom *Weapons Information Service Bw* und vom *Spacecraft Control Service Bw*.

S2 – Service Warranties

Der *S2 – Service Warranties* wird die Warranty des Service erfasst.

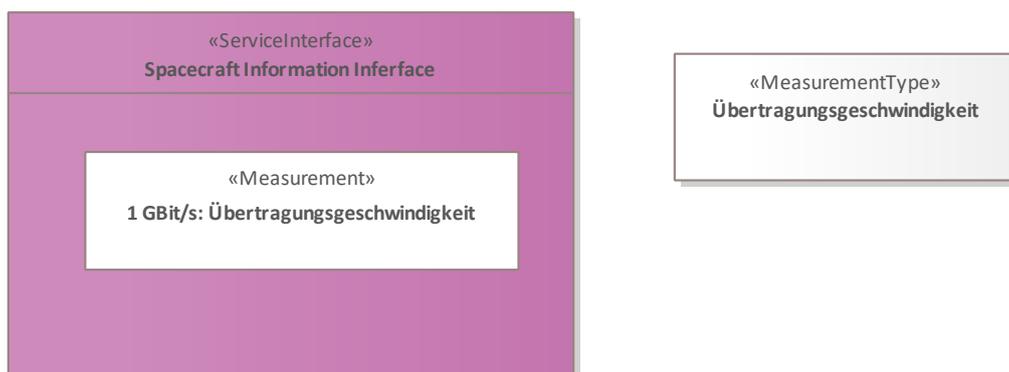
| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| S2-MK1 | Die Informationen werden auf dem Diagramm <i>S2: Projektname : Service Warranties (Servicename)</i> im Package <i>S2/ Warranty</i> erfasst. | S2-A |
| S2-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. | S2-B |
| S2-MK3 | <p>Im Folgenden sind nun sämtliche Serviceattribute (Warranty) am Service anzulegen. Dazu sind die MEASUREMENTTYPE aus dem Standardelementekatalog zu nutzen oder neue Elemente anzulegen.</p> <p>Am Measurement ist das TaggedValue „ITSD_ServiceClass“ zu pflegen. Akzeptierte Werte sind</p> <ul style="list-style-type: none"> - Serviceklasse 1 - Serviceklasse 2 - Serviceklasse 3 <p>Hinweis Anlegen neuer MEASUREMENTTYPE: Durch den IT-Service Designer werden standardisierte MeasurementTypes vorgegeben. Sollte ein neuer MEASUREMENTTYPE angelegt werden müssen, so ist das TaggedValue ITSD_Group zu pflegen.</p> | S2-F S2-G |
| | <p>Hinweis Servicevarianten: Sollte der Service über Servicevarianten verfügen sind für diese ebenfalls die Attribute zu erfassen. Die Informationen werden auf dem Diagramm <i>S2: Projektname : Service Warranties (Servicename)</i> im Package <i>S2/ Warranty - Variante (optional)</i> erfasst.</p> <p>Hinweis ServiceInterface: Sollte der Service über ServiceInterfaces verfügen und es notwendig sein an diesem Attribute zu erfassen, dann sind die Informationen auf dem Diagramm <i>S2: Projektname : ServiceInterface (Servicename)</i> im Package <i>S2/ Warranty - ServiceInterface (optional)</i> darzustellen.</p> | |

Abbildung 32: Beispiel S2 : MASC : Service Warranties (Spacecraft Information Service Bw)

Erläuterung:

Im o. a. Beispiel sind einige Eigenschaften des *Spacecraft Information Service Bw* aufgezeigt. Dem Service sind die Eigenschaften *Systemtechnische Verfügbarkeit* mit einem Wert von *100* und *Betriebszeiten / Systemzugangszeiten* von *Mo. Bis So., 00:00 bis 24:00 Uhr* zugeordnet.

Abbildung 33: Beispiel S2 : MASC : Service Interface (Spacecraft Information Service Bw)

Erläuterung:

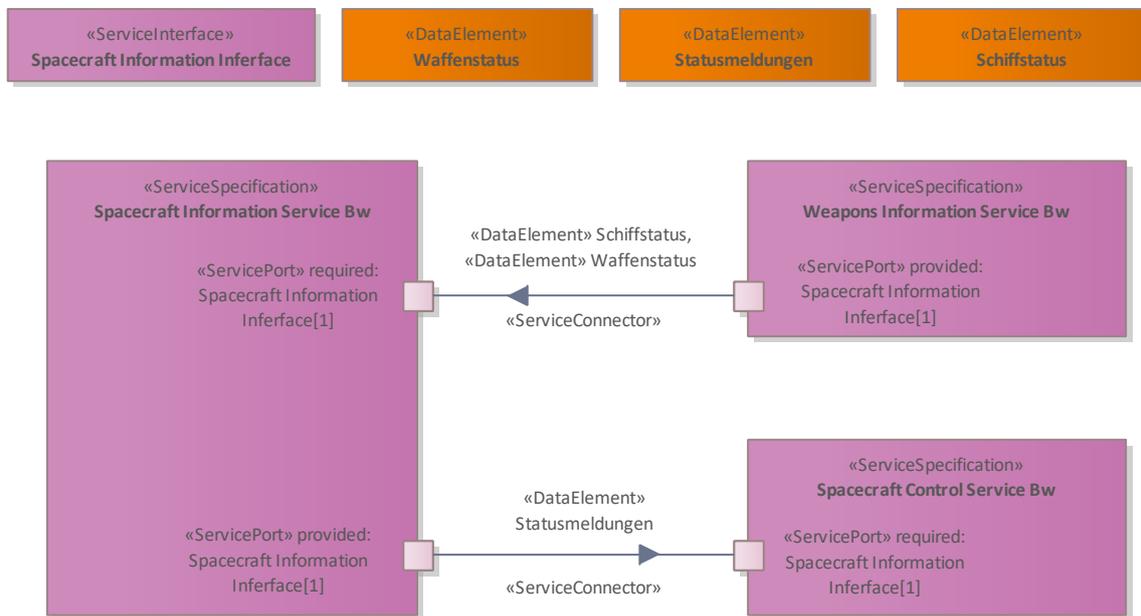
Im o. a. Beispiel sind einige Eigenschaften des *Spacecraft Information Interface* aufgezeigt. Dem Interface ist die Eigenschaft *Übertragungsgeschwindigkeit* mit einem Wert von *1 Gbit/s* zugeordnet.

S3 – Service Interfaces

Der *S3 – Service Interfaces* bildet die Schnittstellen des Service und die Austauschbeziehungen zu anderen Services ab.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| S3-MK1 | Die Informationen werden auf dem Diagramm <i>S3 : Projektname : Service Interfaces (Servicename)</i> im Package <i>S3/ Logische Kontextsicht</i> erfasst. | S3-A |
| S3-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. | S3-B |
| S3-MK4 | Die Schnittstellen des Service werden angelegt. Hinweis Benamung: <i>Schnittstellen, die der Service bereitstellt werden mit „provided“ bezeichnet.</i> <i>Schnittstellen, die der Service nutzt, werden mit „required“ bezeichnet.</i> Hinweis Multiplicity: <i>Einer Schnittstelle wird immer eine Multiplicity von 1 zugewiesen.</i> | S3-C |
| S3-MK5 | Jeder Service, der mit dem zu beschreibenden Service Daten austauscht, wird auf das Diagramm gezogen. Anschließend werden alle relevanten Schnittstellen eingeblendet oder neu angelegt. | S3-F |
| S3-MK6 | Es werden alle Austauschbeziehungen zwischen den Serviceschnittstellen angelegt. Zusätzlich werden jeder Austauschbeziehung die übertragenen Daten zugewiesen. Hinweis Austauschbeziehungen: <i>Es sind die Austauschbeziehungen zwischen den Services der Ebene 1 zu modellieren. Pro InformationFlow ist nur ein DataElement zuzuordnen.</i> | S3-G S3-H |

Abbildung 34: Beispiel S3 : MASC : Service Interfaces (Spacecraft Information Service Bw)



Erläuterung:

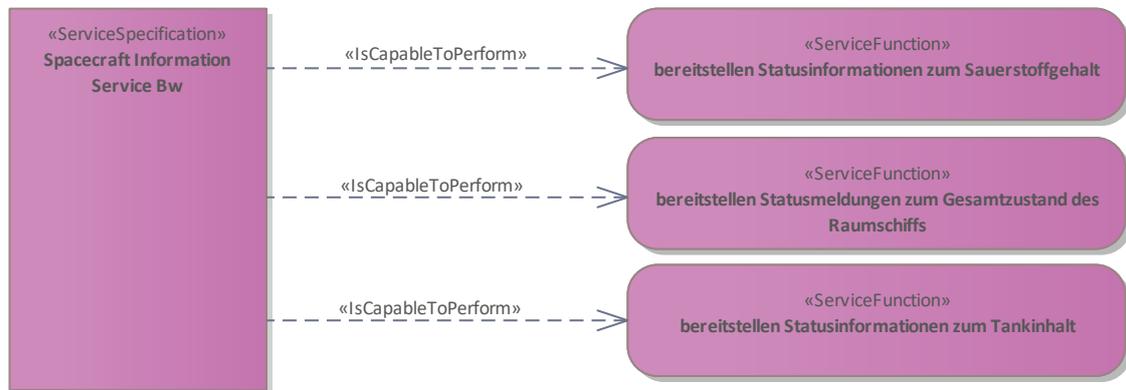
Im o. a. Auszug des Architekturmodells zum MASC empfängt der *Spacecraft Information Service Bw* den *Schiffstatus* sowie *Waffenstatus* vom *Weapons Information Service Bw* und sendet *Statusmeldungen* an den *Spacecraft Control Service Bw*.

S4 – Service Functions

Der *S4 – Service Functions* dient der Darstellung der Funktionen, die vom Service zur Verfügung gestellt werden.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| S4-MK1 | Die Informationen werden auf dem Diagramm <i>S4: Projektname : Service Functions (Servicename)</i> im Package <i>S4/ Servicefunktion</i> erfasst. | S4-A |
| S4-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. | S4-B |
| S4-MK3 | Jede Servicefunktion des zu beschreibenden Service wird aus dem Katalog auf das Diagramm gezogen oder neu angelegt. An der ServiceFunction ist das TaggedValue „ITSD_UTILITYUSAGE“ zu pflegen. <i>Hinweis optionale Servicefunktionen:</i> <i>Sollte der Service über optionale Servicefunktionen verfügen, sind diese im Diagramm S4: Projektname : Service Functions – optional (Servicename) im Package S4/ Servicefunktion - optional (optional) zu erfassen. Am Konnektor IsCapableToPerform wird im Namensfeld „optional“ vermerkt.</i> | S4-C |
| S4-MK4 | Jede Servicefunktion wird dem Service zugeordnet. | S4-E |
| | <i>Hinweis Servicevarianten:</i> <i>Sollte der Service über Servicevarianten verfügen sind für diese ebenfalls die Servicefunktionen zu erfassen. Die Informationen werden auf dem Diagramm S4: Projektname : Service Functions (Servicename) im Package S4/ Servicefunktion - Variante (optional) erfasst.</i> | |

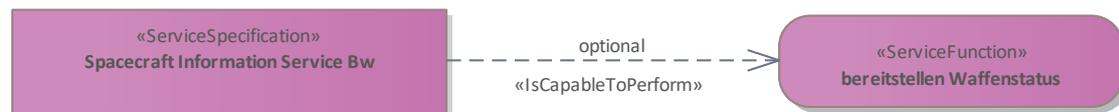
Abbildung 35: Beispiel S4 : MASC : Service Functions (Spacecraft Information Service Bw)



Erläuterung:

Der *Spacecraft Information Service Bw* stellt drei verschiedene Servicefunktionen zur Verfügung. Zu diesen zählen neben dem *Bereitstellen von Statusinformationen zum Sauerstoffgehalt*, das *Bereitstellen von Statusmeldungen zum Gesamtzustand des Raumschiffs* sowie das *Bereitstellen von Statusinformationen zum Tankinhalt*.

Abbildung 36: Beispiel S4 : MASC : Service Functions - optional (Spacecraft Information Service Bw)



Erläuterung:

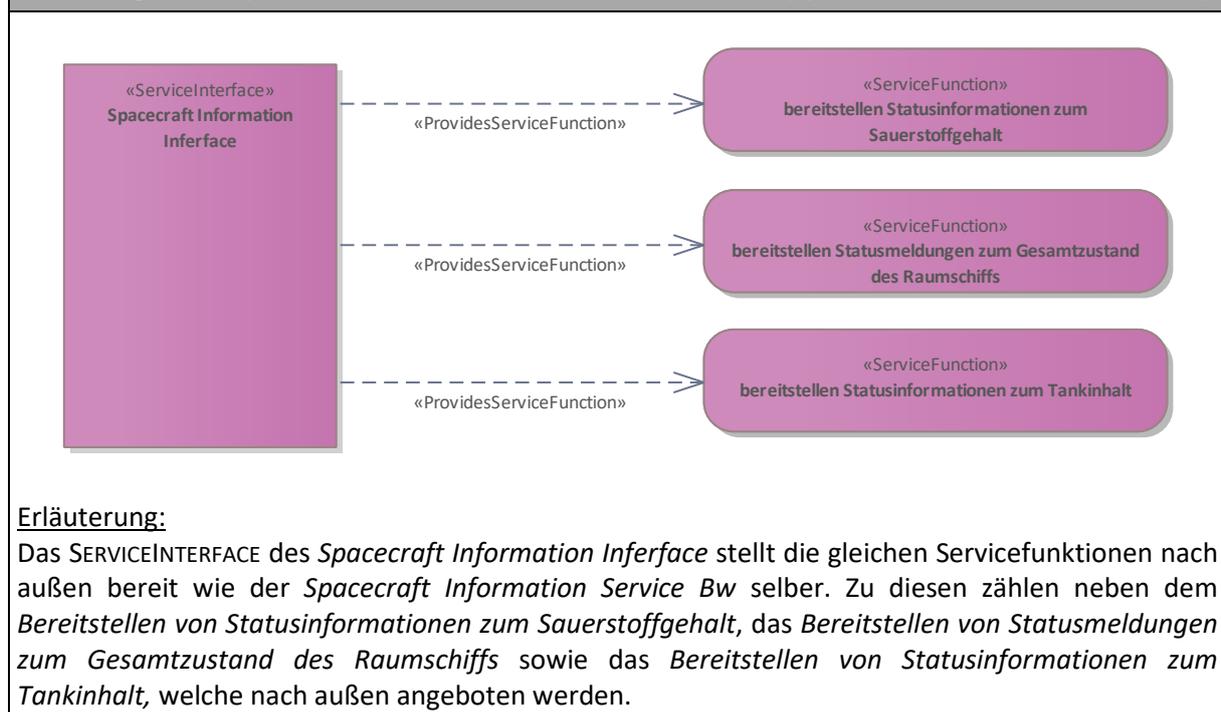
Dem *Spacecraft Information Service Bw* ist die *optionale* Servicefunktion *bereitstellen Waffenstatus* zugeordnet.

S7 – Service Interface Parameters

Der *S7 – Service Interface Parameters* beschreibt, welche Servicefunktionen eine Serviceschnittstelle nach außen anbietet.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| S7-MK1 | Die Informationen werden auf dem Diagramm <i>S7 : Projektname : Service Interface Parameters (Servicename)</i> im Package <i>S7/Serviceinterface</i> erfasst. | S7-A |
| S7-MK2 | Jede relevante Serviceschnittstelle wird aus dem Katalog auf das Diagramm gezogen. | S7-B |
| S7-MK3 | Jede relevante Servicefunktion wird aus dem Katalog auf das Diagramm gezogen. | S7-C |
| S7-MK4 | Jede Servicefunktion wird mit der bereitstellenden Serviceschnittstelle verbunden. | S7-D |

Abbildung 37: Beispiel S7 : MASC : Service Interface Parameters (Spacecraft Information Service Bw)



Sr – Service Change

Der *Sr – Service Change* beschreibt, an welchen Services im Rahmen der Servicedokumentation ein Änderungsbedarf besteht.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| Sr-MK1 | Die Informationen werden auf dem Diagramm <i>Sr : Projektname : Service Change (Servicename)</i> im Package <i>Sr/ Service Change</i> erfasst. | Sr-A |
| Sr-MK2 | Jeder relevante Service, an dem ein Änderungsbedarf besteht, wird aus dem Katalog auf das Diagramm gezogen. | Sr-E |
| Sr-MK3 | Für jeden Änderungsbedarf wird ein Element vom Typ „Change“ aus der Toolbox das Diagramm gezogen. Es sind der Name und die Beschreibung des Changes zu hinterlegen. | |
| Sr-MK4 | Jeder Change wird mit einem Konnektor vom Typ „Trace“ mit dem jeweiligen Service verbunden. | |

Abbildung 38: Beispiel Sr : MASC : Service Change (Spacecraft Information Service Bw)



Erläuterung:

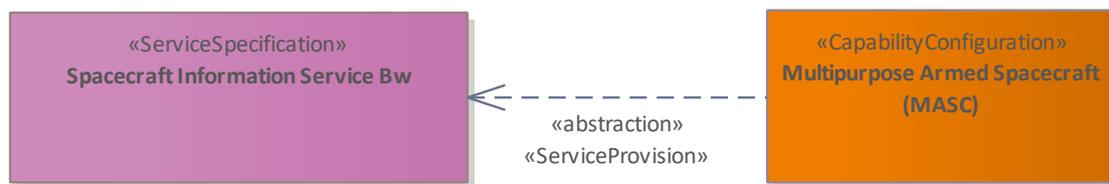
Am *Spacecraft Information Service Bw* soll eine Anpassung der Schnittstellen erfolgen.

P1 – Projektbezug

Der *P1 – Projektbezug* erfasst die einem Service zugeordneten Ressourcen.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| P1-MK1 | Die Informationen werden auf dem Diagramm <i>P1 : Projektname : Serviceprovision</i> im Package <i>P1/Projektbezug</i> erfasst. | P1-A |
| P1-MK2 | Der zu beschreibende Service wird aus dem Katalog auf das Diagramm gezogen. | P1-E |
| P1-MK3 | Die zuzuordnenden Ressourcen werden aus dem Katalog auf das Diagramm gezogen. | P1-D |
| P1-MK4 | Der Service wird mit den bereitstellenden Ressourcen verbunden. | P1-F |

Abbildung 39: Beispiel P1 : MASC : Serviceprovision

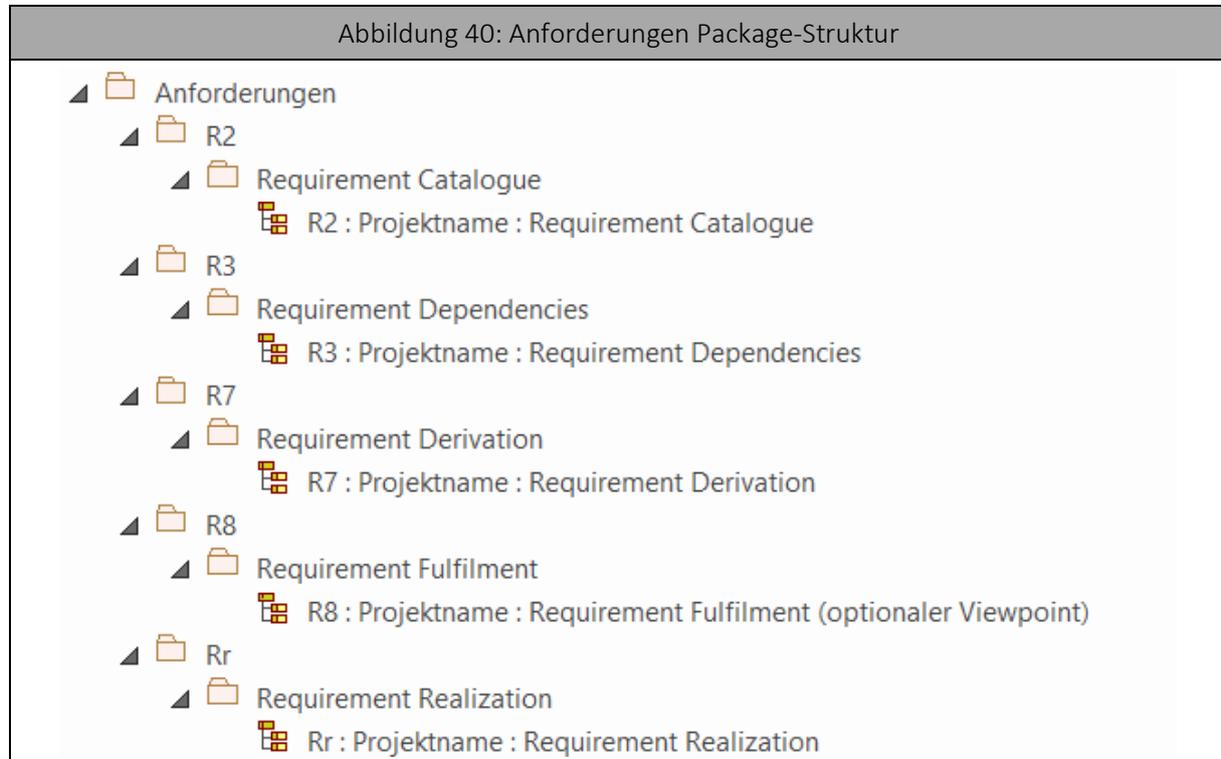
**Erläuterung:**

Im o. a. Beispiel wird dargestellt, dass das MASC den Service *Spacecraft Information Service Bw* bereitstellt.

Package-Struktur der Anforderungen

Für die Modellierung von Anforderungen ist die in *Abbildung 39* abgebildete Package-Struktur vorgegeben. Die Package-Struktur bildet die Basis für die automatisierte Weiterverarbeitung der Informationen und muss eingehalten werden. Im folgenden Abschnitt werden die einzelnen Viewpoints dargestellt.

Abbildung 40: Anforderungen Package-Struktur



R2 – Requirement Catalogue

Der *R2 – Requirement Catalogue* behandelt die Struktur der Forderungen. Dies geschieht, indem die Forderungen in einem Katalog in Kategorien eingeteilt werden.

Hinweis: Der R2 aus der Analyse Phase 1 wird fortgeschrieben.

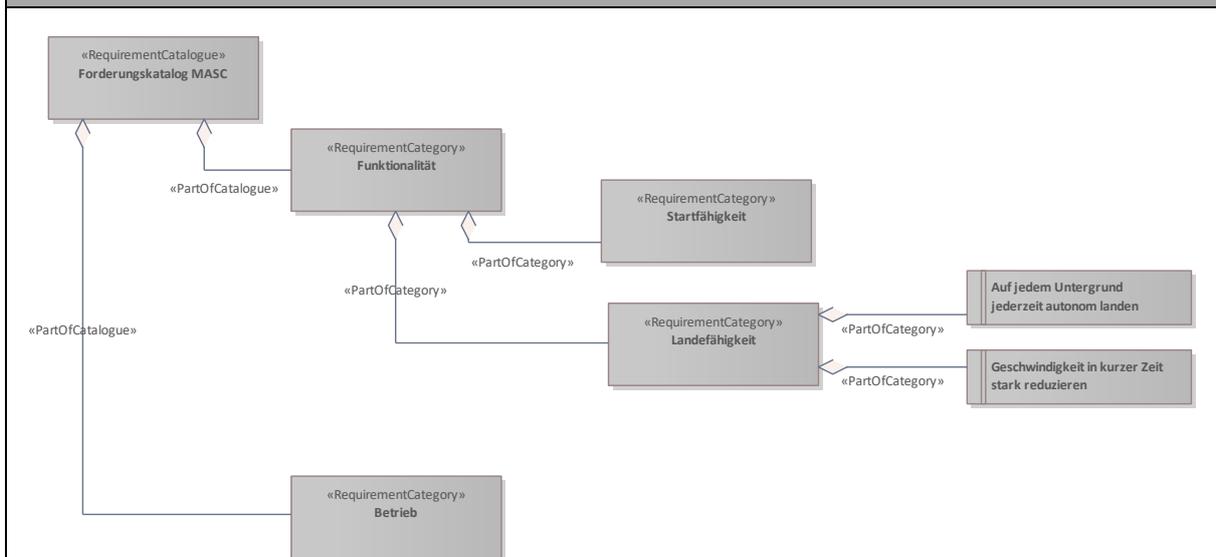
Der R2 dient in der Architektur für die Analysephase Teil 2 dazu, die ergänzten oder präzisierten Forderungen in Kategorien einzuordnen und so inhaltlich zusammengehörende Forderungen für unterschiedliche Systemelemente zusammenzufassen. Die Strukturierung ist teilweise durch die Regelung für den priorisierten Forderungskatalog vorgegeben, kann jedoch individuell erweitert werden.

Die vorgegebene Struktur kann in Anlage 3.3 heruntergeladen und automatisiert in die Architektur eingebunden werden. Der R2 entspricht dann sowohl inhaltlich und als auch in seiner Struktur dem Priorisierten Forderungskatalog gemäß A1-1500/3-7000.

Sofern der R2 bereits in der Analysephase Teil 1 erstellt wurde, ist dieser in der Analysephase Teil 2 nicht neu anzulegen, sondern fortzuschreiben.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| R2-MK1 | Die Informationen werden auf dem Diagramm <i>R2 : Projektname : Requirement Catalogue</i> im Package <i>R2/ Requirement Catalogue</i> erfasst. | R2-A |
| R2-MK2 | Das Element für den Anforderungskatalog sowie dessen Kategorien sind aus dem Katalog auf das Diagramm gezogen. Die Kategorien sind dem Anforderungskatalog zu zugeordnet. <i>Hinweis: Es dürfen keine neuen Kategorien angelegt werden, wenn in der Analyse Phase 1 bereits Kategorien angelegt wurden.</i> | R2-B R2-C R2-D |
| R3-MK3 | Alle Anforderungen sind neu angelegt oder aus dem Katalog auf das Diagramm gezogen und einer Kategorie zugeordnet. | R2-F R2-G |

Abbildung 41: Beispiel R2 : MASC: Requirement Catalogue

**Erläuterung:**

Der o. a. Auszug des Forderungskatalog MASC besteht aus der Überkategorien *Funktionalität* und *Betrieb*. Die Kategorie *Funktionalität* untergliedert sich in die Kategorien *Startfähigkeit* und *Landefähigkeit*. Der Kategorie *Landefähigkeit* sind dabei die beiden Forderungen *Auf jedem Untergrund jederzeit autonom landen* und *Geschwindigkeit in kurzer Zeit stark reduzieren* zugeordnet.

R3 – Requirement Dependencies

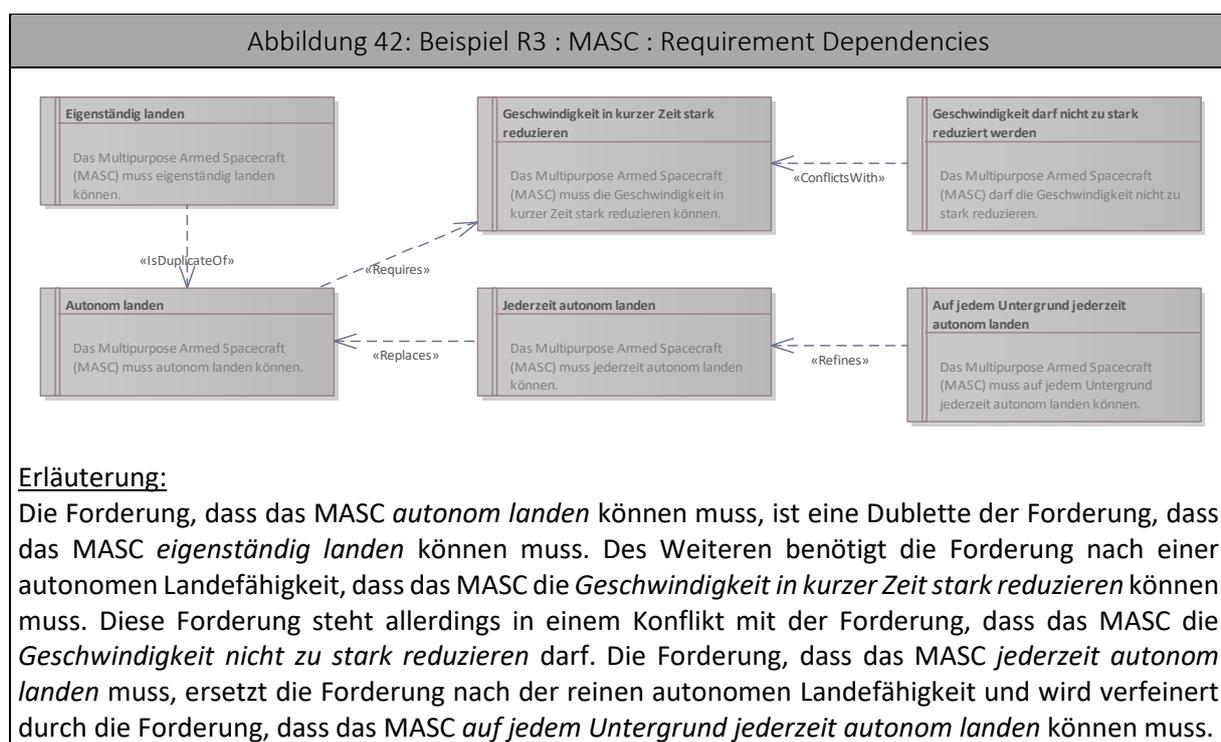
Der *R3 – Requirement Dependencies* behandelt die Beziehungen zwischen Forderungen.

Der R3 dient in der Architektur für die Analysephase Teil 2 dazu Abhängigkeiten, Konflikte, Dubletten und Verfeinerungen zwischen Forderungen darzustellen. Er ist damit ein wesentliches Hilfsmittel zur Forderungsanalyse.

Abhängigkeiten zwischen den Forderungen sind vollständig beizubehalten, so dass sich aus den Abhängigkeiten ergebende Konsequenzen nachvollziehbar sind. Konflikte zwischen Forderungen sind zum Ende der Forderungserhebung aufzulösen. Dubletten können, insofern die Forderung für mehrere Systemelemente benötigt werden, erhalten bleiben.

Sofern der R3 bereits in der Analysephase Teil 1 erstellt wurde, ist dieser in der Analysephase Teil 2 nicht neu anzulegen, sondern fortzuschreiben. Existiert in der Analysephase Teil 2 noch kein R3, so ist der Umfang dieses Viewpoints mit dem begleitenden Referat festzulegen.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| R3-MK1 | Die Informationen werden auf dem Diagramm <i>R3 : Projektname : Requirement Dependencies</i> im Package <i>R3/Requirement Dependencies</i> erfasst. | R3-A |
| R3-MK2 | Alle relevanten Anforderungen sind aus dem Katalog auf das Diagramm gezogen. | R3-B |
| R3-MK3 | Sämtliche Abhängigkeiten sind zwischen den Anforderungen erfasst. | R3-C |



R7 – Requirement Derivation

Der *R7 – Requirement Derivation* behandelt die Ableitung der Forderungen aus operationellen Vorgaben und Rahmenbedingungen und den dazugehörigen Elementen.

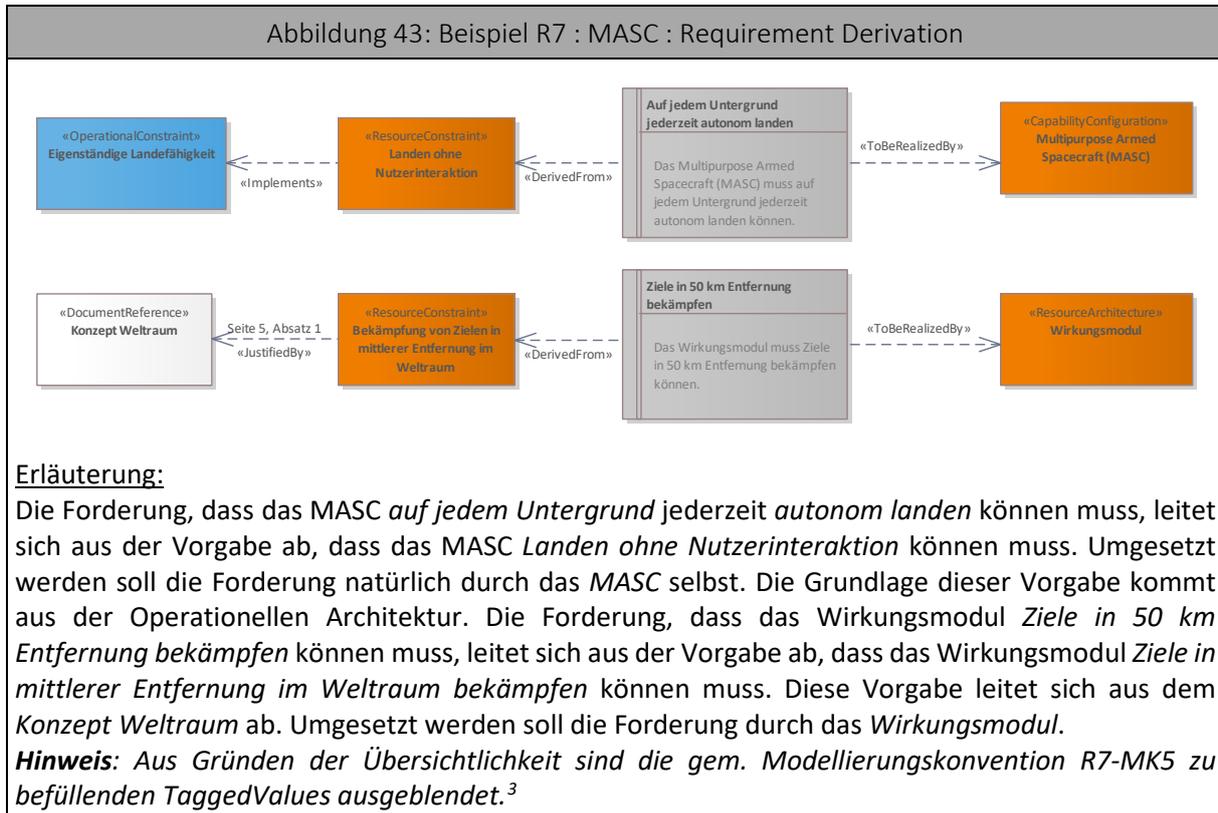
Der R7 dient in der Architektur für die Analysephase Teil 2 primär dazu, die in der Analysephase Teil 1 erhobene Forderungslage zu präzisieren und fortzuschreiben. Des Weiteren wird im R7 in der Analysephase Teil 2 die Zuordnung der Forderungen zu den geplanten, realisierenden Ressourcen getroffen.

Sofern der R7 bereits in der Analysephase Teil 1 erstellt wurde, ist dieser in der Analysephase Teil 2 nicht neu anzulegen, sondern fortzuschreiben. Existiert in der Analysephase Teil 2 noch kein R7, so ist der Umfang dieses Viewpoints mit dem begleitenden Referat festzulegen.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| R7-MK1 | Die Informationen werden auf dem Diagramm <i>R7 : Projektname : Requirement Derivation</i> im Package <i>R7/Requirement Derivation</i> erfasst. | R3-A |
| R7-MK2 | Alle Implementierungsvorgaben die Anforderungsbegründen sind, sind aus dem Katalog auf das Diagramm gezogen. | R7-B |
| R7-MK4 | <p>Alle relevanten Anforderungen sind aus dem Katalog auf das Diagramm gezogen.</p> <p><i>Hinweis: Die folgenden TaggedValue sind zupflegen. Diese ist nur dann manuell notwendig, wenn der SMA nicht genutzt wird.</i></p> <ul style="list-style-type: none"> • <i>AG_ID</i> • <i>Akteur</i> • <i>Aktivität</i> • <i>Anforderungsart</i> • <i>Ansprechpartner</i> • <i>Hinweis</i> • <i>Object und Ergänzungen</i> • <i>Operative Bewertung</i> • <i>Prozesswort</i> • <i>QS_Status</i> • <i>Qualität</i> • <i>Randbedingung</i> • <i>Singular</i> • <i>Status</i> • <i>Subjekt</i> • <i>Titelsperre</i> • <i>Verbindlichkeit</i> • <i>Zu</i> | R7-C |

| | | |
|--------|--|------|
| R7-MK5 | Alle Anforderungen sind auf die entsprechenden Implementierungsvorgaben abgebildet. <i>Hinweis: Jede Implementierungsvorgabe muss einer Referenz zugeordnet sein, die auf dem Diagramm einblendbar ist.</i> | R7-E |
| R6-MK8 | Alle Elemente, die dazu vorgesehen sind, die Anforderung zu realisieren, sind auf Diagramm eingeblendet. | R7-F |

Abbildung 43: Beispiel R7 : MASC : Requirement Derivation



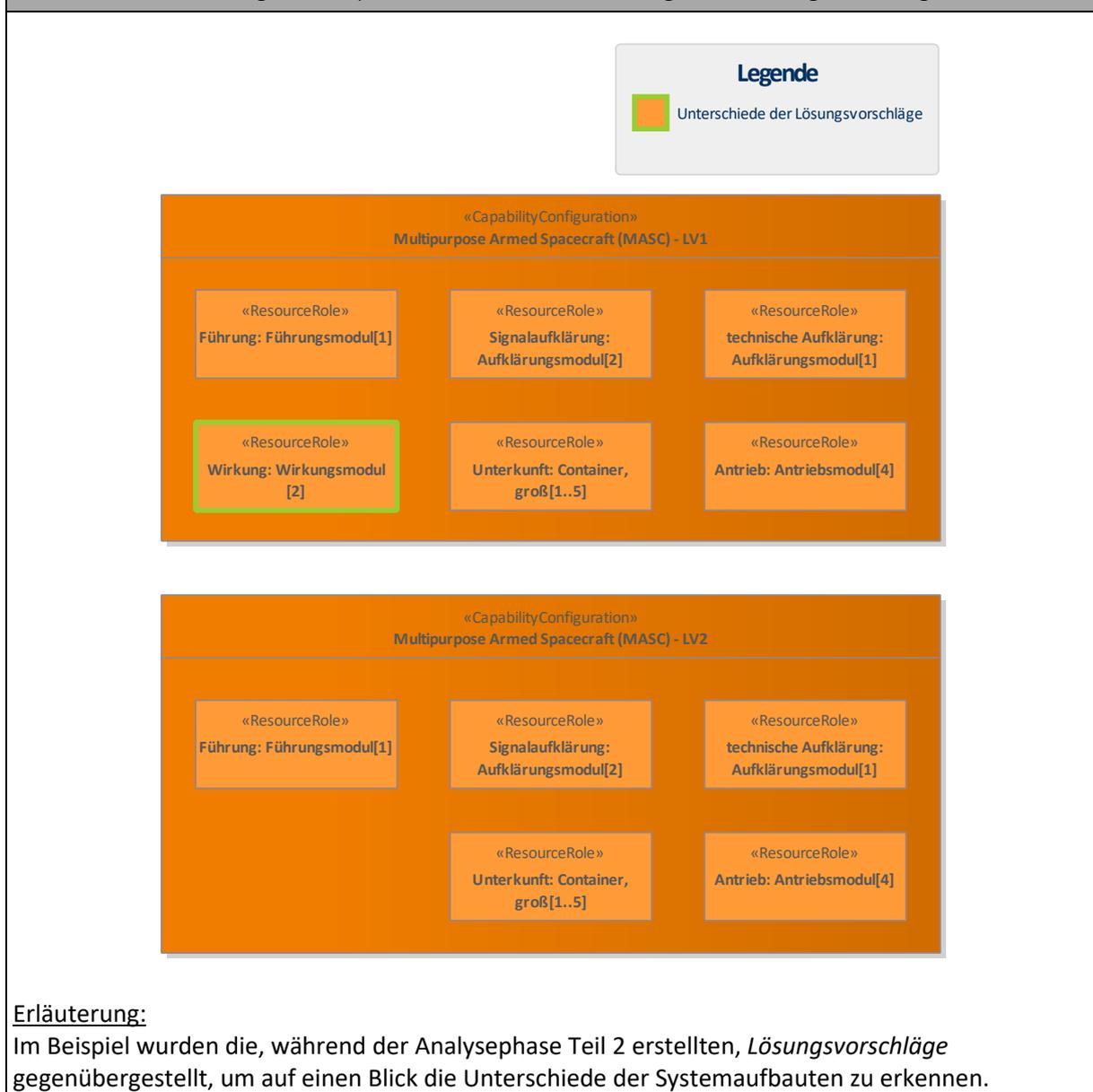
³ Ein Beispiel zur Befüllung der TaggedValues finden Sie in der Dokumentation „Verwendung des ADMBw im SPARX Enterprise Architect“ in Kapitel 4.5.3.

Auswertediagramme

Die sogenannten *Auswertediagramme* (im NAF Kontext auch als FFPV, fit-for-purpose-Views, bezeichnet) dienen dazu, einen ganz bestimmten und nur für das Projekt bestimmten Analysebedarf zu decken. Hierbei wird jeweils ein spezifischer Inhalt in einem gesonderten View dargestellt.

Auf einem Auswertediagramm dürfen keine neuen Inhalte erzeugt oder die Konventionen des Metamodells gebrochen werden. Es fasst lediglich bereits existierende Inhalte zusammen. Bei einem Auswertediagramm werden häufig Zusammenhänge zwischen Inhalten auf einem Auswertediagramm dargestellt, die ansonsten auf unterschiedlichen Views der Architektur verteilt wären. Existiert kein ADMBw-Viewpoint, welcher geeignet ist, die gewünschten Inhalte darzustellen, so ist ein UML Klassendiagramm als View zu verwenden.

Abbildung 44: Beispiel P2 : MASC : Auswertediagramm Lösungsvorschläge



System- und Servicearchitekturen in der Realisierungsphase

In diesem Kapitel werden die Sichten des ADMBw angesprochen, die für die Architekturerstellung in der Realisierungsphase (RP) von Relevanz sein können. Die Auswahl der Sichten in diesem Dokument ist als „Best Practice“ zu verstehen und greift den generellen Analyse- und Informationsbedarf in der Realisierungsphase auf. In Einzelfällen können bestimmte Sichten entfallen oder zusätzliche Sichten müssen modelliert werden.

Im Sinne der vereinfachten Darstellung sind in der folgenden Abbildung die wesentlichen Elemente und Relationen eines Architekturmodells bzw. einer Architektur für die Realisierungsphase dargestellt. Die Abbildung beschränkt sich auf die wichtigsten Elemente und ist nicht vollständig. Die detaillierte Darstellung ist der Dokumentation des ADMBw zu entnehmen. Ein detaillierter Einblick in das MetaModell des ADMBw kann über den in Anlage 0 aufgeführten Link (MetaModelViewer) abgerufen werden.

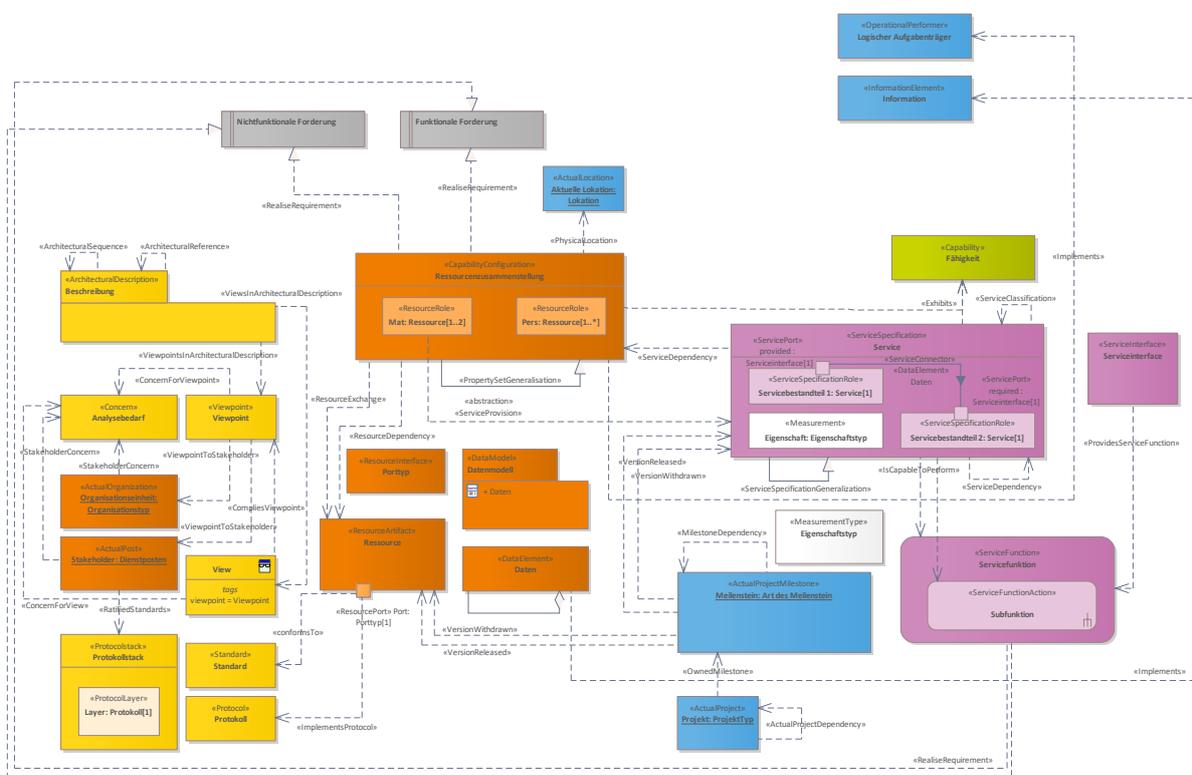


Abbildung 45: Überblick der Elemente in System- und Servicearchitekturen in der RP

Hinweis: Bei der Architekturerstellung ist es wichtig zu verstehen, dass es nicht darum geht Diagramme und Bilder zu erstellen, sondern ein Modell zu generieren. Die Viewpoints stehen nie alleine, da diese immer nur einen Ausschnitt aus einem Gesamtmodell repräsentieren. Die Zusammenhänge der Sichten ergeben sich über die Inhalte, die dargestellt werden. In einer rein theoretischen Betrachtung ist es möglich das gesamte Modell auf einer Sicht darzustellen und anschließend in handhabbare und überschaubare Diagramme aufzuteilen.

Package-Struktur der System- und Servicearchitektur

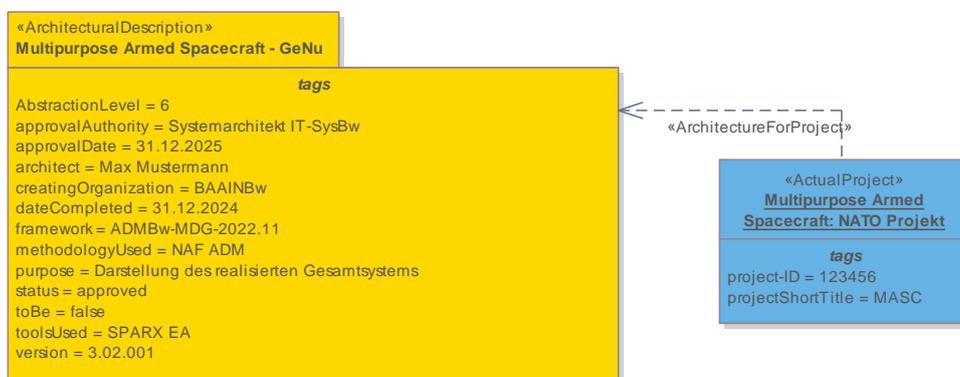
Die zu nutzende Packagestruktur entspricht den Vorgaben der Analysphase 2 in den Abschnitten 2.3.2, 2.3.23 und 2.3.36 .

A1 – Meta-Data Definitions

Der *A1 – Meta Data Definitions* befasst sich mit den Metadaten einer Architektur, welche über die gesamte Architektur hindurch verwendet werden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.3.

Abbildung 46: Beispiel A1 : MASC : Architectural Description



Erläuterung:

Im o. a. Beispiel wurde im Rahmen der Erstellung der Genehmigung zur Nutzung (GeNu) eine *Architekturbeschreibung* (i. S. e. Architekturmodells) zum Projekt *Multipurpose Armed Spacecraft (MASC)* durch den Modellierer Max Mustermann vom BAAINBw erstellt. Ziel der Architektur war dabei die Darstellung des realisierten Gesamtsystems. Die Fertigstellung des Architekturmodells erfolgte am 31.12.2024 und wurde am 31.12.2025 durch den Systemarchitekten IT-SysBw freigegeben. Bei der vorliegenden Ist-Architektur auf Abstraktionsebene 6 handelt es sich um die Version 3.02.001. Diese wurde mit dem SPARX EA, der NAF ADM-Methodik und der ADMBw-MDG-2022.11 erstellt.

Die *Architekturbeschreibung* ist dem für diese verantwortlichen *Projekt Multipurpose Armed Spacecraft* zugeordnet.

A2 – Architecture Products

Der *A2 – Architecture Products* beschreibt die Architekturprodukte, die in einer Architektur vorhanden sind.

Der A2-Viewpoint dient in der Architektur für die Realisierungsphase dazu, die Stakeholder der Architektur und deren Anliegen bzw. deren Analyse- bzw. Informationsbedarf zu definieren. Daraus werden alle weiteren notwendigen Viewpoints für die Architektur abgeleitet.

Der A2 ermöglicht eine Bewertung, inwieweit der Analyse-/Informationsbedarf durch entsprechende Diagramme abgedeckt wird.

Unter Architekturprodukten werden neben den erstellten Diagrammen (Views) auch der Analysebedarf selbst und die an der Modellierung beteiligten Personen verstanden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.4, 2.3.5, 2.3.6, 2.3.7

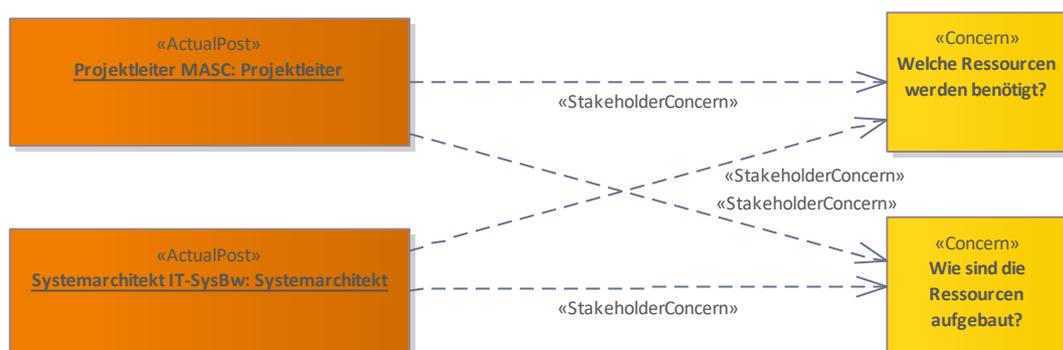
Abbildung 47: Beispiel A2 : MASC : Concerns



Erläuterung:

Die im Beispiel betrachteten Stakeholder des MASC möchten gerne wissen, welche Ressourcen benötigt werden und wie diese aufgebaut sind.

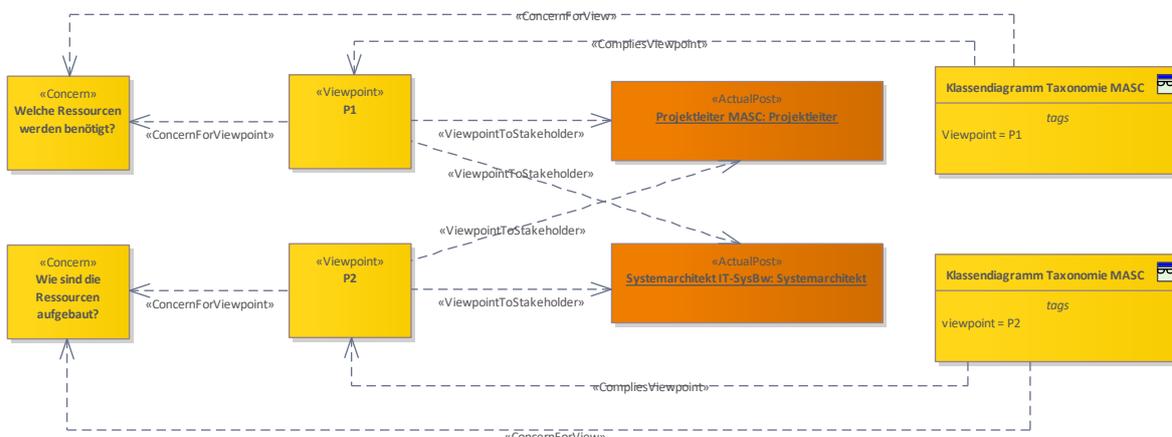
Abbildung 48: Beispiel A2 : MASC : Stakeholder



Erläuterung:

Die im Beispiel betrachteten Stakeholder des MASC sind der *Systemarchitekt IT-SysBw* sowie der *Projektleiter MASC*. Sowohl der *Projektleiter MASC* als auch der *Systemarchitekt IT-SysBw* möchte wissen, welche Ressourcen benötigt werden und wie diese aufgebaut sind.

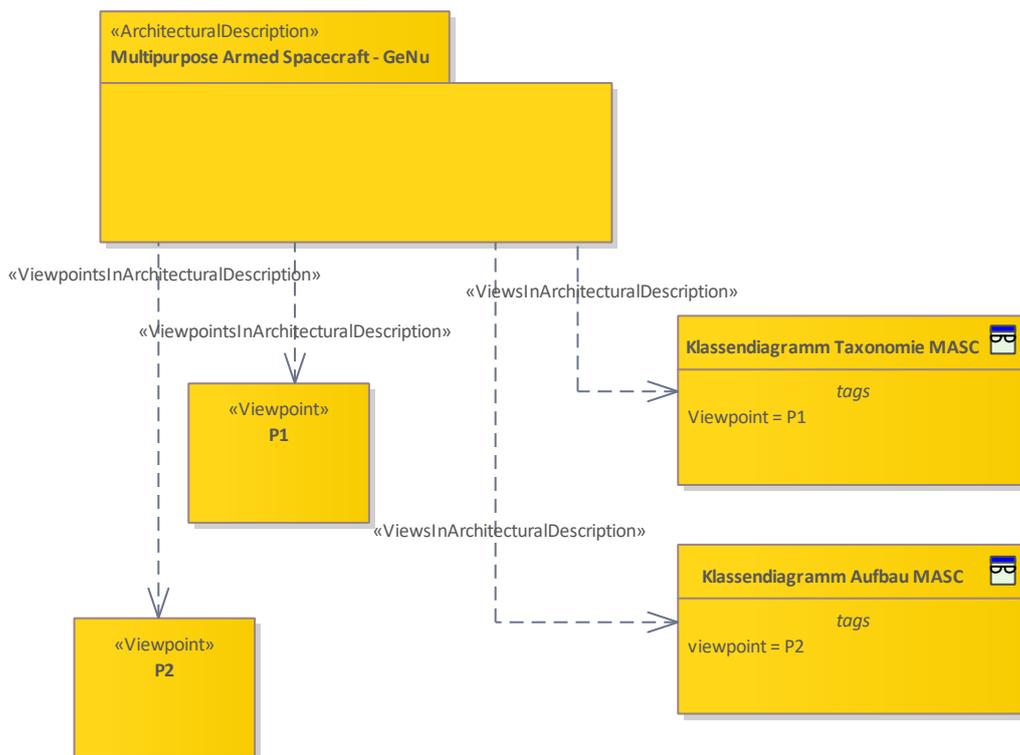
Abbildung 49: Beispiel A2 : MASC : Viewpoints



Erläuterung:

Die Viewpoints und Views, die in diesem Beispiel im Rahmen der Modellierung des MASC zu berücksichtigen sind, sind die Viewpoints *P1* und *P2* sowie die beiden angegebenen *Klassendiagramme*. Sie ergeben sich aus den Informationsbedarfen (Concerns) der Stakeholder und werden diesen ebenfalls zugewiesen.

Abbildung 50: Beispiel A2 : MASC : Architecture Products



Erläuterung:

Zum Abschluss der Modellierung des Viewpoints A2 zur GeNu werden die zu modellierenden Viewpoints (*P1* und *P2*) und Views (*Klassendiagramm Aufbau MASC* und *Klassendiagramm Taxonomie MASC*) zur Architekturbeschreibung des MASC hinzugefügt.

A3 – Architecture Correspondence

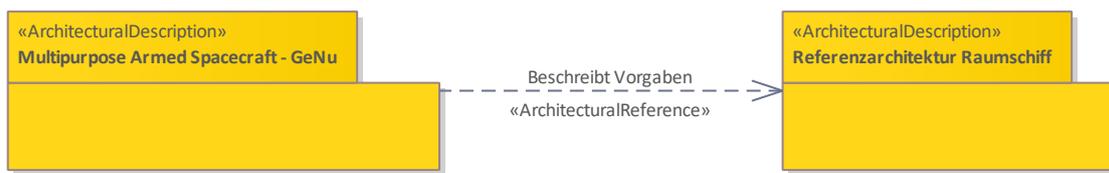
Der *A3 – Architecture Correspondence* befasst sich mit Abhängigkeiten zwischen Architekturen.

Der A3 dient in der Architektur für die Realisierungsphase dazu, die in der Analysephase Teil 1 und 2 beschriebenen Abhängigkeiten zwischen Architekturen bei Bedarf zu ergänzen und auf das Architekturmodell zur Realisierungsphase umzuhängen. Die Abhängigkeiten reichen wie schon in der Analysephase Teil 1 und 2 von der Nennung der entsprechenden Architekturen bis hin zur Wieder- und Weiterverwendung von Architekturartefakten.

Die Abhängigkeit zwischen Projekten ist kein Bestandteil des A3, sondern wird im Lr – Lines of Development dargestellt.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.8.

Abbildung 51: Beispiel A3 : MASC : Architecture Correspondence



Erläuterung:

Die Architektur *Multipurpose Armed Spacecraft - GeNu* in der Realisierungsphase muss die Vorgaben der *Referenzarchitektur Raumschiff* berücksichtigen. Aus diesem Grund stehen die beiden o. a. Architekturbeschreibungen in Abhängigkeit.

A6 – Architecture Versions

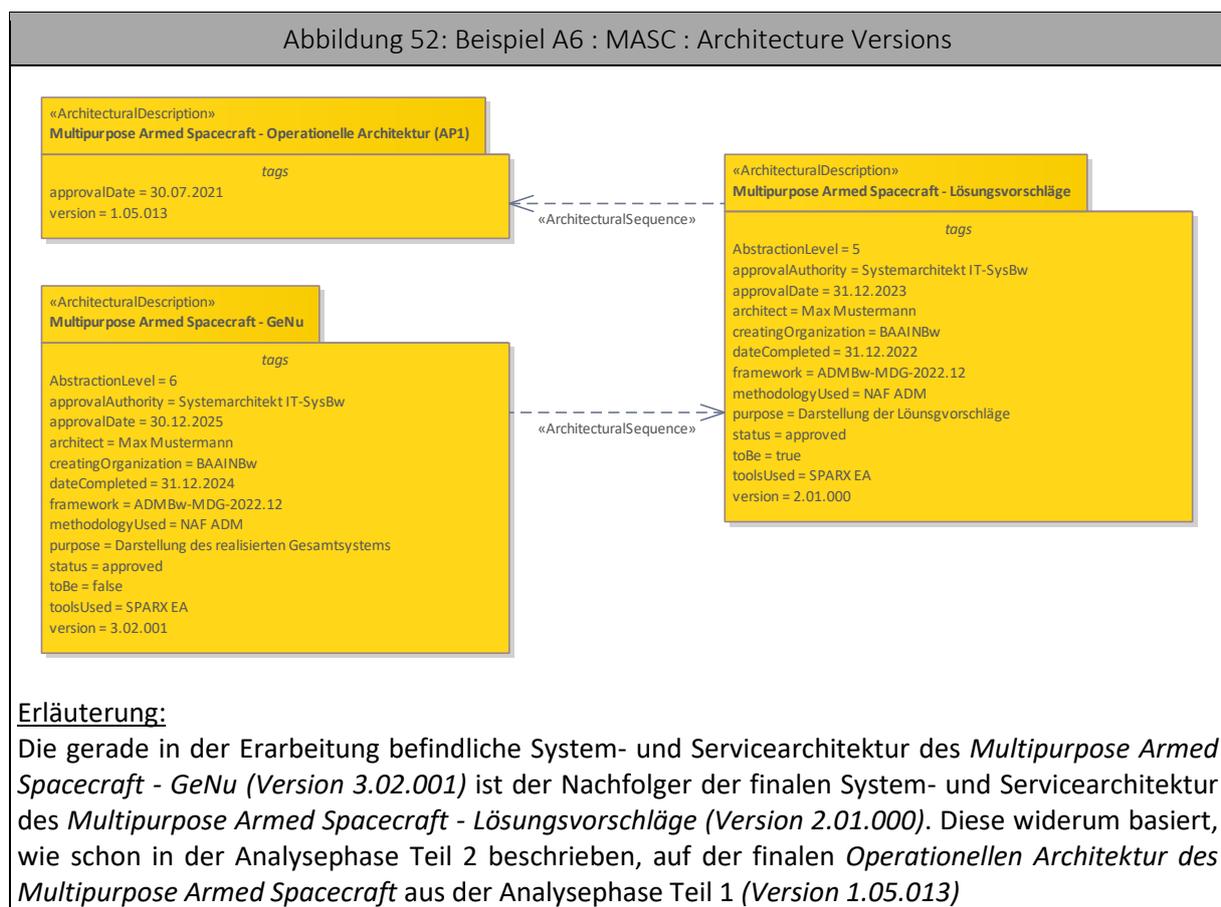
Der *A6 – Architecture Versions* befasst sich mit der Versionsgeschichte der Architektur, wobei relevante Metadaten genutzt werden.

Der A6 dient in der Architektur für die Realisierungsphase dazu, die Versionsgeschichte der betrachteten Architektur nachzuvollziehen. Ist die vorliegende Version die erste Architektur zum betrachteten Projekt, ist der A6 zu initial zu erstellen und bei jeder neuen Version der Architektur fortzuschreiben.

Wenn in der Realisierungsphase auf einer in der Analysephase Teil 2 erstellten Architektur aufgesetzt wird, ist eine neue Architekturbeschreibung für die Realisierungsphase im A1 anzulegen und die beiden Beschreibungen im A6 zu verknüpfen.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.9.

Abbildung 52: Beispiel A6 : MASC : Architecture Versions

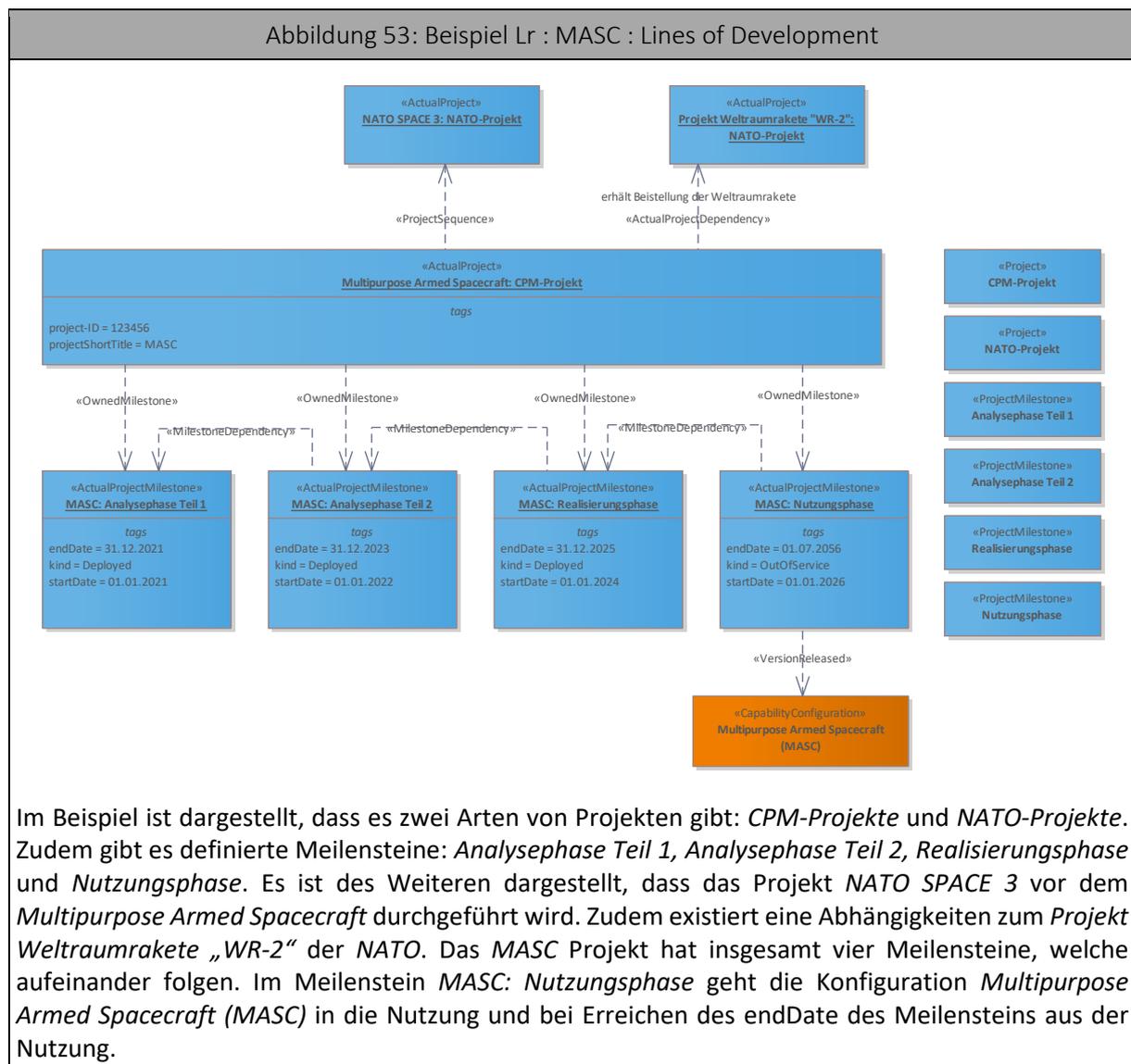


Lr – Lines of Development

Der *Lr – Lines of Development* behandelt die Identifizierung der Projektabhängigkeiten. Dazu werden die relevanten Programme und Projekte und ihre Abhängigkeiten dargestellt.

Der Lr dient in der Realisierungsphase wie schon in der Analysephase Teil 2 dazu, die wesentlichen Abhängigkeiten des betrachteten Projekts zu anderen Projekten abzubilden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich bei Bedarf fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.10.

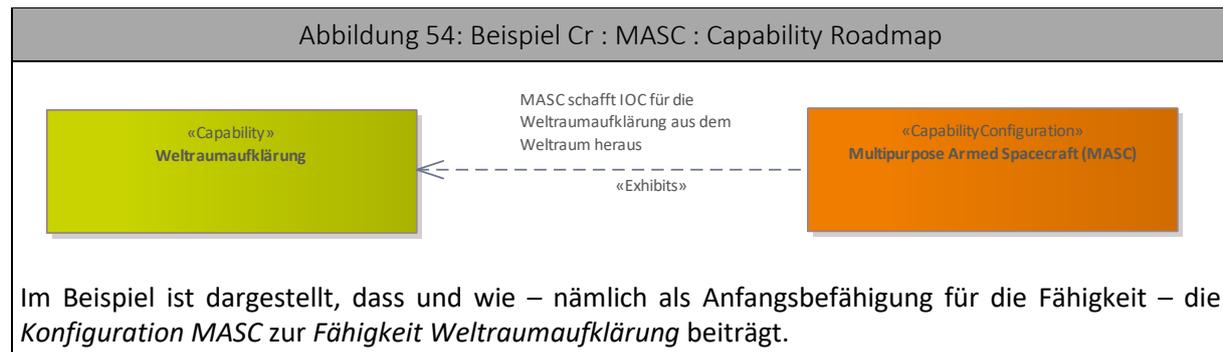


Cr – Capability Roadmap (optionalen Viewpoint)

Der *Cr – Capability Roadmap* behandelt die Darstellung der Ressourcen, die zur Bereitstellung einer Fähigkeit herangezogen werden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern nur bei Bedarf fortgeschrieben. Sollte es bis zur Realisierungsphase noch keinen Cr geben, so ist dieser neu zu modellieren. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.11

Abbildung 54: Beispiel Cr : MASC : Capability Roadmap

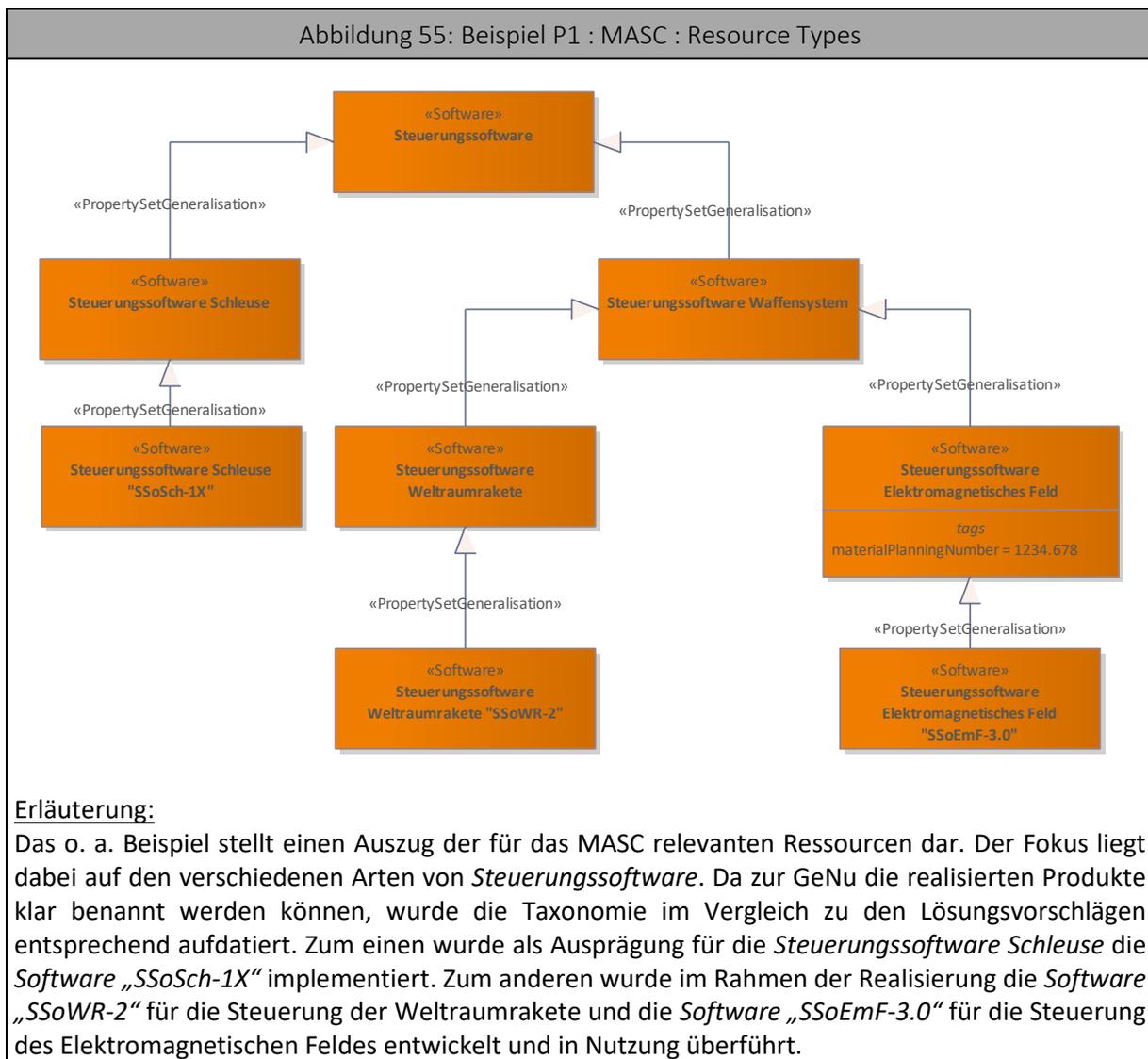


Im Beispiel ist dargestellt, dass und wie – nämlich als Anfangsbefähigung für die Fähigkeit – die *Konfiguration MASC* zur *Fähigkeit Weltraumaufklärung* beiträgt.

P1 – Resource Types

Der *P1 – Resource Types* behandelt die Spezifikation der Ressourcen sowie deren Einordnung in eine Taxonomie.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.12.



P2 – Resource Structure

Der *P2 – Resource Structure* behandelt die Zusammensetzung und Interaktion von Ressourcen.

Der P2 dient in der Architektur für die Realisierungsphase dazu, den konkreten internen Aufbau der realisierten Ressourcen sowie die konkrete Umsetzung der operationellen Anteile darzustellen. Die Darstellung des Aufbaus ist notwendig, um realisierten Funktionen einzelnen Systembestandteilen zuzuordnen und somit eine zielgerichtete Wiederverwendung von Ressourcen ermöglichen zu können. Zusätzlich zum Genannten werden im P2 ebenfalls die externen Abhängigkeiten und Datenaustauschbeziehungen abgebildet.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.13, 2.3.14, 2.3.15, 2.3.16.

Abbildung 56: Beispiel P2 : MASC : Resource Structure

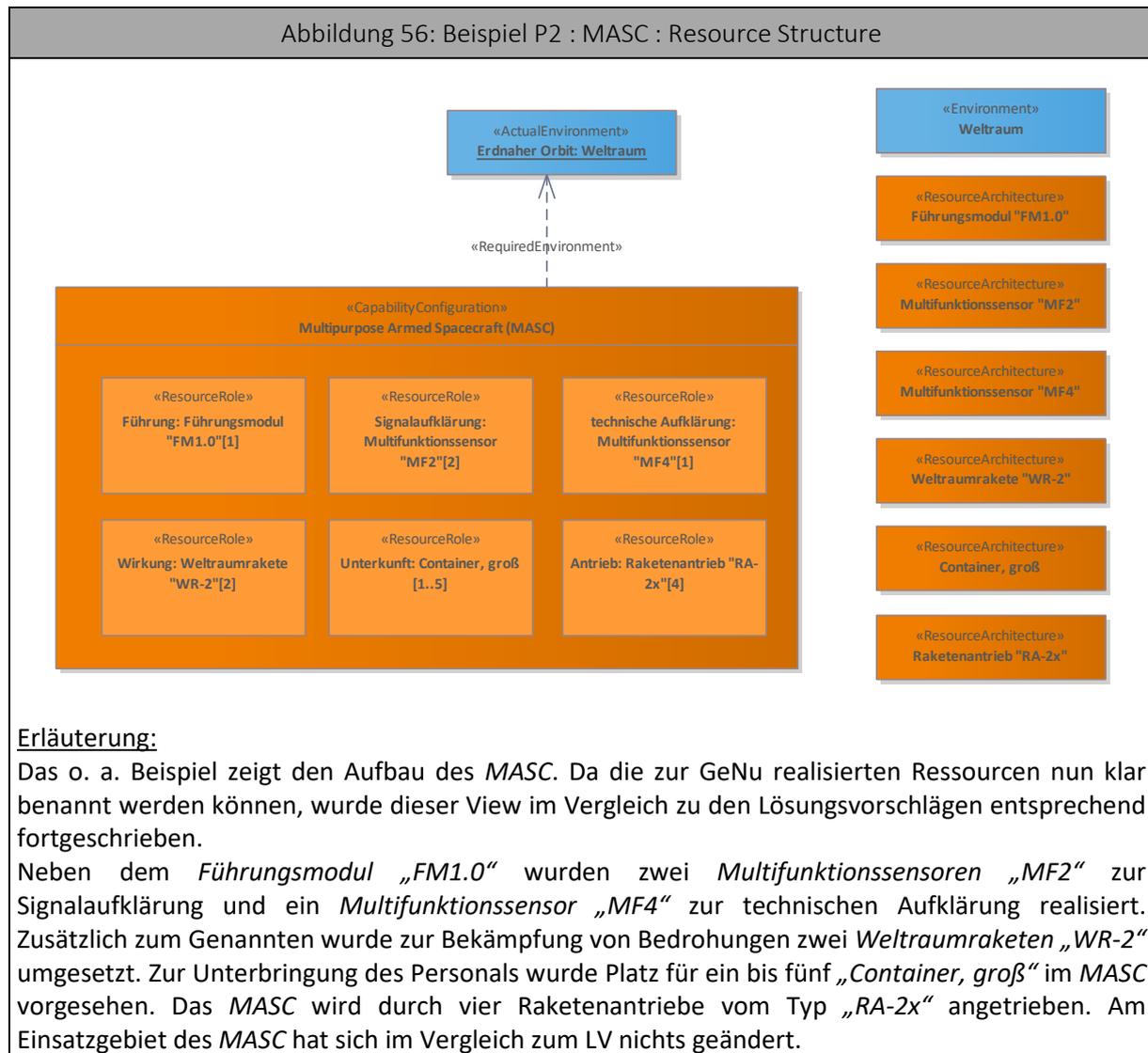
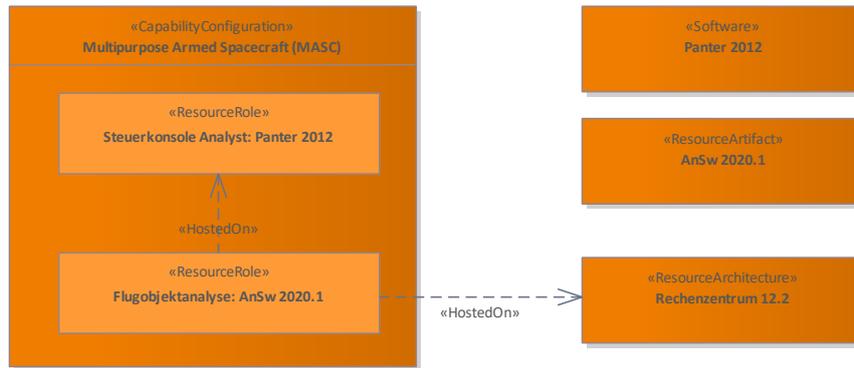
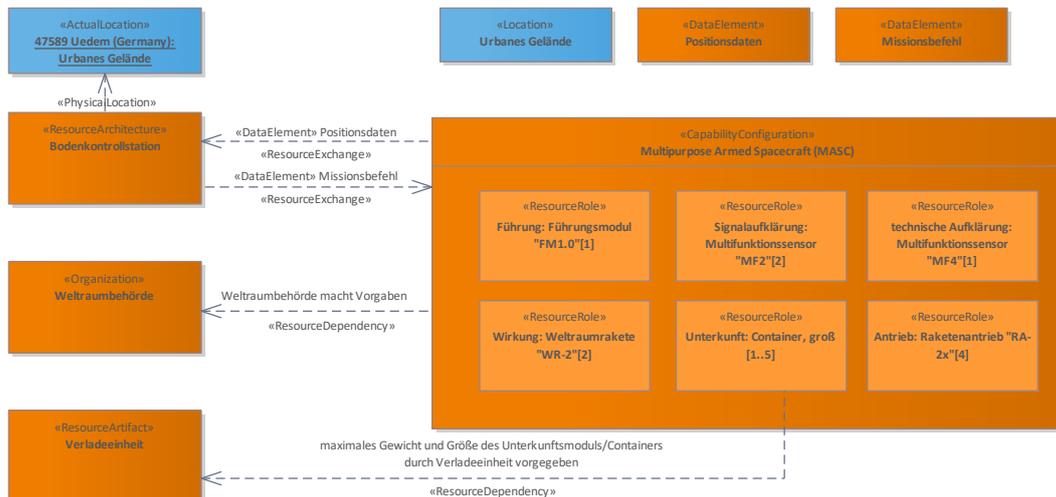


Abbildung 57: Beispiel P2 : MASC : Application Hosting

**Erläuterung:**

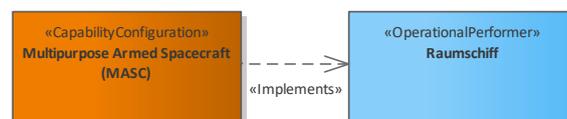
Das o. a. Beispiel zeigt die *AnSw 2020.1* als Bestandteil des *MASC*. Diese wird auf dem *Panter 2012* und im *Rechenzentrum 12.2* betrieben.

Abbildung 58: Beispiel P2 : MASC : External Dependencies

**Erläuterung:**

Im o. a. Beispiel wird zum einen dargestellt, dass das *MASC* auch weiterhin einen bidirektionalen Datenaustausch mit der *Bodenkontrollstation in 47589 Uedem* durchführt. Zum anderen gibt es zwei dargestellte Abhängigkeiten. Auf der einen Seite hängen die Größe und das Gewicht des *Unterkunftsmoduls* von der *Verladeeinheit* am Boden ab, die das Untermodul am Boden in das Raumschiff verlädt. Auf der anderen Seite muss das *MASC* die Vorgaben der *Weltraumbehörde* berücksichtigen. Im Vergleich zu den Abhängigkeiten der Analysephase Teil 2 hat sich hier keine Änderung ergeben.

Abbildung 59: Beispiel P2 : MASC : Node Realization

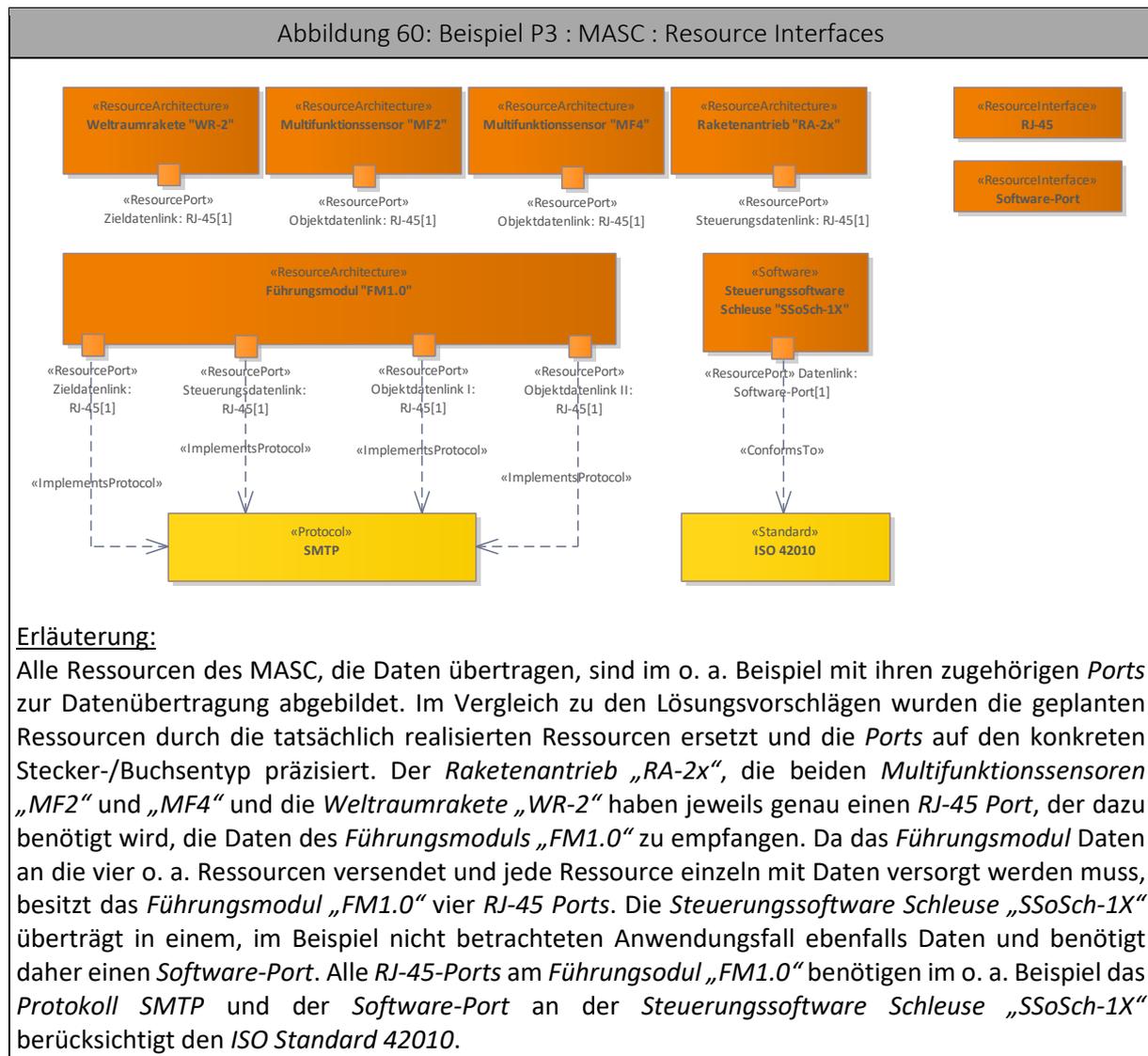
**Erläuterung:**

Im o. a. Beispiel wird dargestellt, dass das *MASC* dazu dient, den in der FFF beschriebenen, logischen Rollen- und Aufgabenträger *Raumschiff* umzusetzen. Im Vergleich zu den Lösungsvorschlägen hat sich hier keine Änderung ergeben.

P3 – Resource Interfaces

Der *P3 – Resource Interfaces* beschreibt die Systemschnittstellen.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.17.



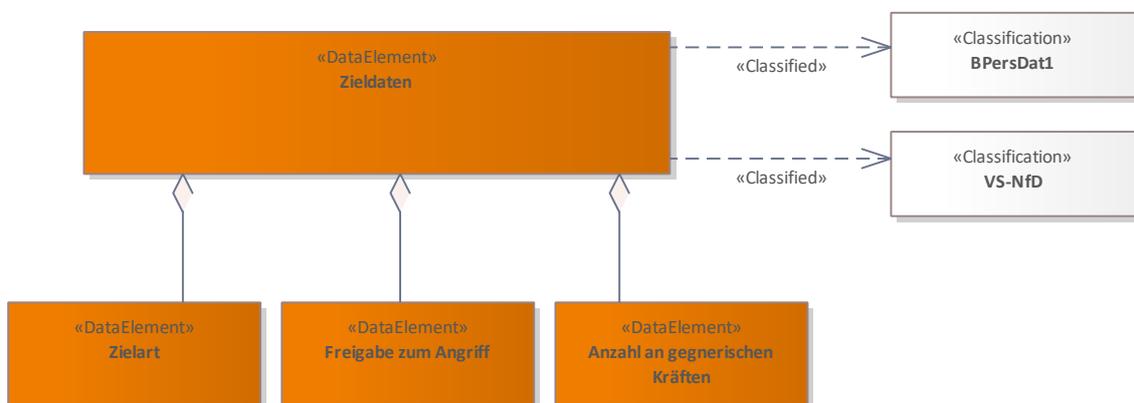
P7 – Data Model

Der *P7 – Data Model* behandelt die Struktur der Daten, die von den Ressourcentypen in der Architektur untereinander ausgetauscht werden.

Der P7 dient in der Architektur für die Realisierungsphase dazu, das in der Analysephase Teil 2 beschriebene Datenmodell bei Bedarf fortzuschreiben.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.18, 2.3.19, 2.3.20.

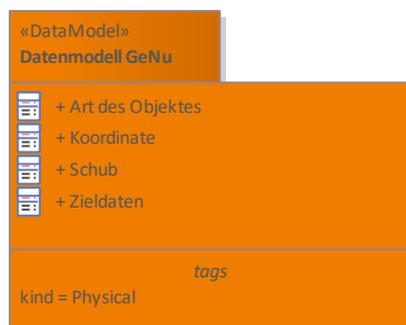
Abbildung 61: Beispiel P7 : MASC : Data Elements



Erläuterung:

Der o. a. View stellt dar, dass sich das Datenelement *Zieldaten* aus drei anderen Datenelementen zusammensetzt. Zu diesen zählen die *Zielart*, die *Anzahl an generischen Kräften* und die *Freigabe zum Angriff*. Für die *Zieldaten* sind der *Geheimhaltungsgrad (VS-NfD)* und der *Schutzbereich (BPersDat)* angegeben.

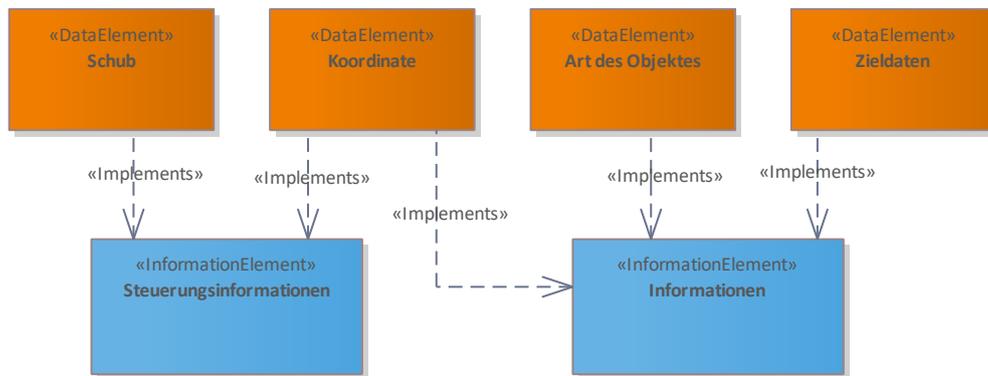
Abbildung 62: Beispiel P7 : MASC : Data Model



Erläuterung:

Alle im Rahmen der GeNu zum MASC benannten, zu übertragenden Daten werden in diesem View zum physischen *Datenmodell der GeNu* hinzugefügt. Neben den *Zieldaten* gehören nun durch die Präzisierung der Datenelemente in der Realisierungsphase die *Art des Objektes*, die *Koordinate* sowie die benötigte *Schubkraft* zum dargestellten Datenmodell.

Abbildung 63: Beispiel P7 : MASC : InformationElement Realization

**Erläuterung:**

Wie schon im P2 - Resource Structure - Operational Realization geht es auch in diesem View um die Umsetzung von Elementen, die in der FFF beschrieben sind. Dieses Mal liegt der Fokus jedoch nicht auf Rollen- und Aufgabenträgern und Ressourcen, sondern auf Informationselementen und Datenelementen. Die in der FFF beschriebenen *Steuerungsinformationen* werden durch die Präzisierung der Datenelemente in der Realisierungsphase nun durch die *Schubkraft* und die *Koordinate* umgesetzt werden. Die *Koordinate*, die *Art des Objektes* sowie die *Zieldaten* dienen der Realisierung der in der FFF benannten *Informationen*.

S1 – Service Description

Der *S1 – Service Description* dient der Erfassung der Servicebeschreibung und der Einordnung des Service in eine der drei Ebenen des Portfolios.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.24.

Abbildung 64: Beispiel S1 : MASC : Service Description (Spacecraft Information Service Bw)



Erläuterung:

Das Beispiel stellt den *Spacecraft Information Service Bw* inkl. Beschreibung dar. Es handelt sich um einen Service der Ebene 1 (am Begriff *Service Bw* am Ende im Namen zu erkennen).

S1 – Service Taxonomy

Der *S1 – Service Taxonomy* ordnet den Service in die C3-Taxonomie ein.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.25.

Abbildung 65: Beispiel S1 : MASC : Service Taxonomy (Spacecraft Information Service Bw)



Erläuterung:

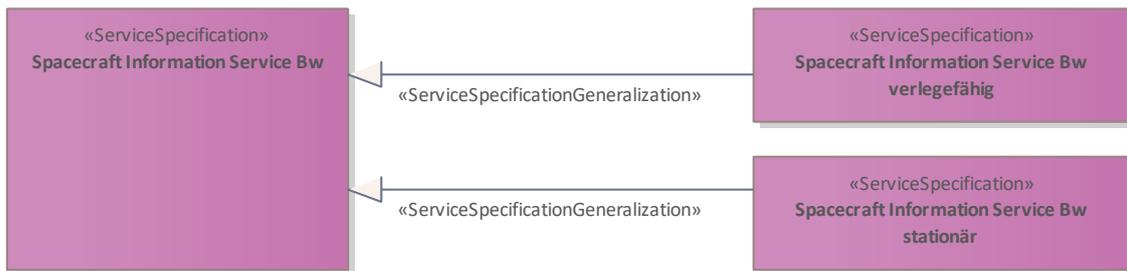
Das o. a. Beispiel stellt einen Auszug der für das MASC relevanten Services dar. Der Fokus liegt dabei auf dem *Spacecraft Information Service Bw*. Dieser ist zum einen in der Taxonomie der Servicekategorie *Information Management Services* zugeordnet (sowohl Service als auch Servicekategorie werden mit dem Stereotyp `SERVICE SPECIFICATION` modelliert).

S1 – Service Varianten (optionaler Viewpoint)

Der *S1 – Service Varianten* erfasst die verschiedenen Varianten eines Service.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.26.

Abbildung 66: Beispiel S1 : MASC : Service Varianten (Spacecraft Information Service Bw)



Erläuterung:

Das o. a. Beispiel stellt einen Auszug der für das MASC relevanten Services dar. Der Fokus liegt dabei auf dem *Spacecraft Information Service Bw*. Der *Spacecraft Information Service Bw* wird in den zwei Varianten *Spacecraft Information Service Bw verlegfähig* & *Spacecraft Information Service Bw stationär* ausgebracht (sowohl Service als auch Servicevariante werden mit dem Stereotyp SERVICE SPECIFICATION modelliert).

C1-S1 – Capability to Service Mapping

Der *C1-S1 – Capability to Service Mapping* dient der Einordnung des Services in die Fähigkeitsslage.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.27.

Abbildung 67: Beispiel C1-S1 : MASC : Capability to Service Mapping (Spacecraft Information Service Bw)



Erläuterung:

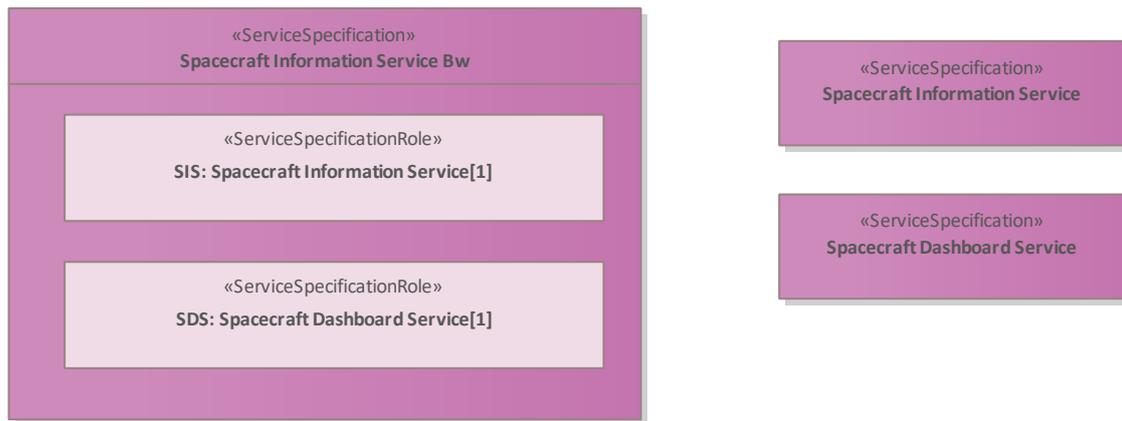
Im o. a. Beispiel leistet der Service *Spacecraft Information Service Bw* einen Beitrag zur Fähigkeit *Weltraumaufklärung*.

S2 – Service Structure

Der *S2 – Service Structure* beschreibt die Zusammensetzung des Service aus dritten Services.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.28.

Abbildung 68: Beispiel S2 : MASC : Service Structure (Spacecraft Information Service Bw)



Erläuterung:

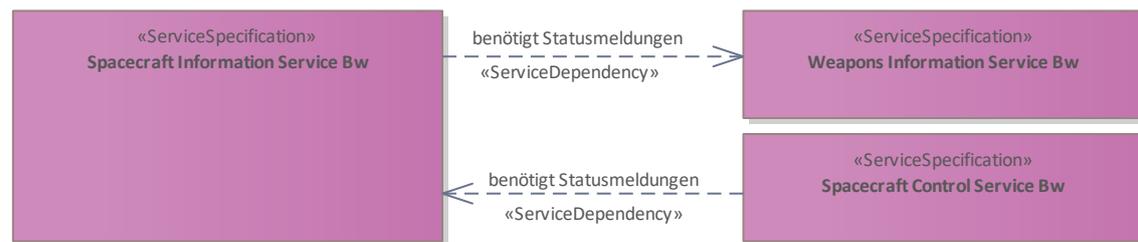
Der im o. a. Beispiel ausgewählte *Spacecraft Information Service Bw* setzt sich aus den beiden Teilservices, dem *Spacecraft Information Service* und dem *Spacecraft Dashboard Service*, zusammen.

S2 – External Service Dependencies

Der *S2 – External Service Dependencies* stellt die Abhängigkeiten des Service dar.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.29.

Abbildung 69: Beispiel S2 : MASC : External Service Dependencies (Spacecraft Information Service Bw)



Erläuterung:

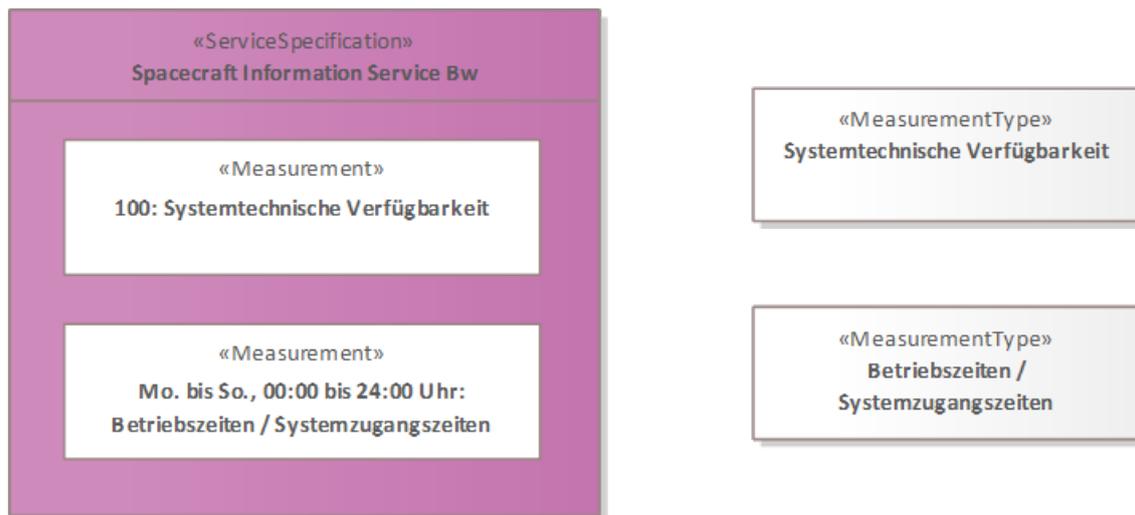
Im o. a. Beispiel sind einige Abhängigkeiten des *Spacecraft Information Service Bw* aufgezeigt. Der Service ist abhängig vom *Weapons Information Service Bw* und vom *Spacecraft Control Service Bw*.

S2 – Service Warranties

Der *S2 – Service Warranties* wird die Warranty des Service erfasst.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.30.

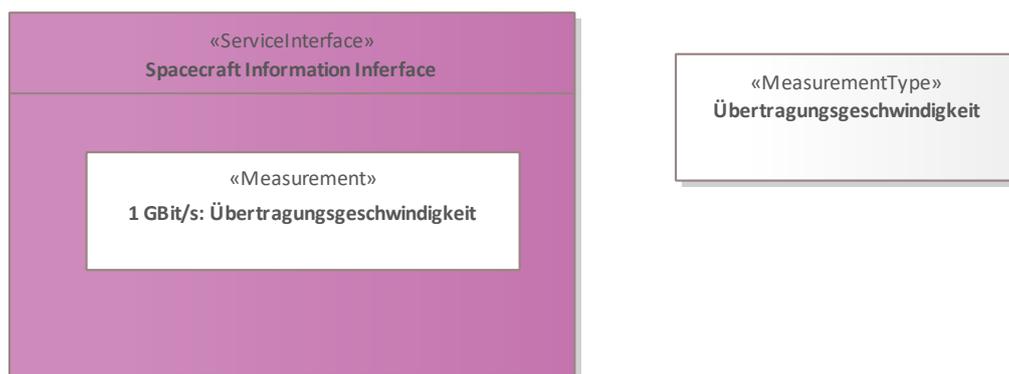
Abbildung 70: Beispiel S2 : MASC : Service Warranties (Spacecraft Information Service Bw)



Erläuterung:

Im o. a. Beispiel sind einige Eigenschaften des *Spacecraft Information Service Bw* aufgezeigt. Dem Service sind die Eigenschaften *Systemtechnische Verfügbarkeit* mit einem Wert von 100 und *Betriebszeiten / Systemzugangszeiten* von Mo. Bis So., 00:00 bis 24:00 Uhr zugeordnet.

Abbildung 71: Beispiel S2 : MASC : Service Interface (Spacecraft Information Service Bw)



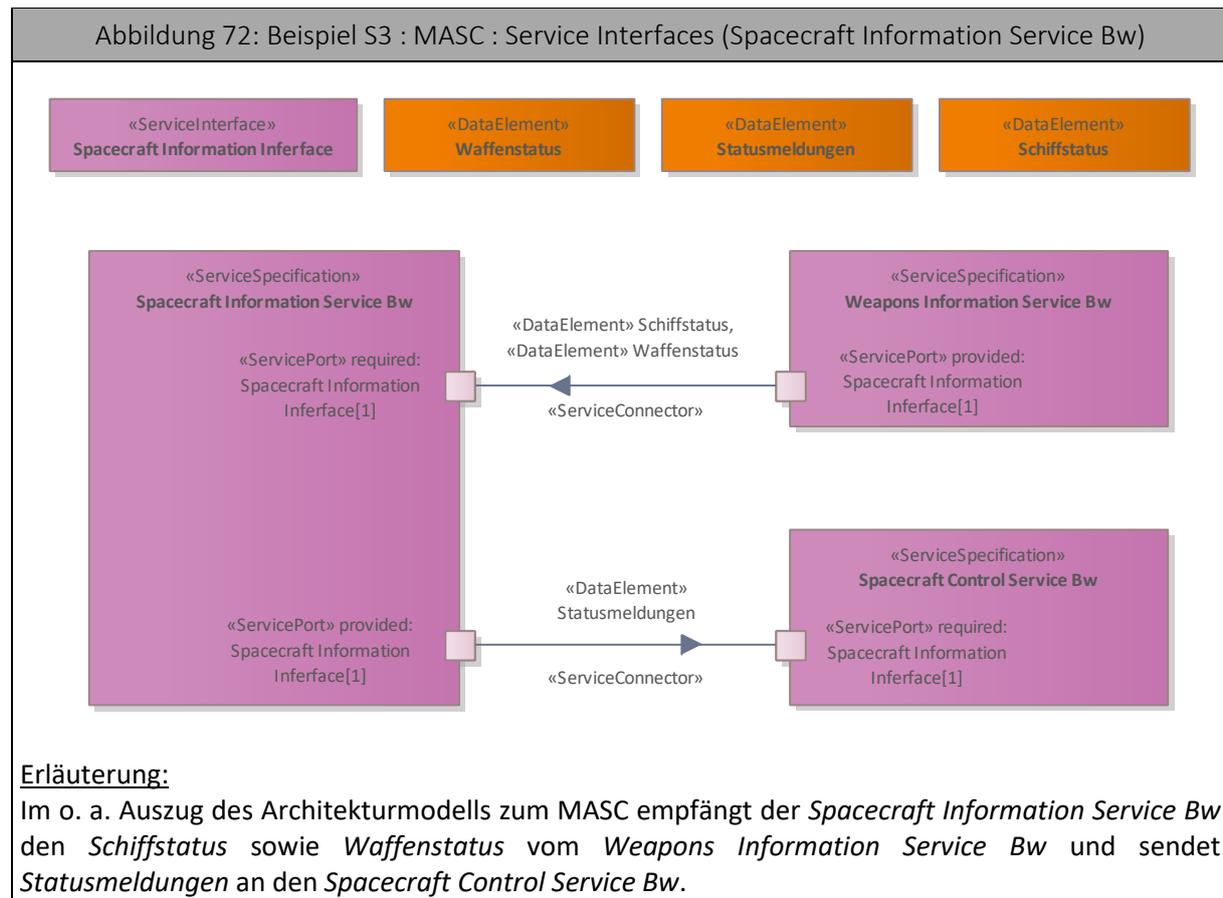
Erläuterung:

Im o. a. Beispiel sind einige Eigenschaften des *Spacecraft Information Interface* aufgezeigt. Dem Interface ist die Eigenschaft *Übertragungsgeschwindigkeit* mit einem Wert von 1 Gbit/s zugeordnet.

S3 – Service Interfaces

Der *S3 – Service Interfaces* bildet die Schnittstellen des Service und die Austauschbeziehungen zu anderen Services ab.

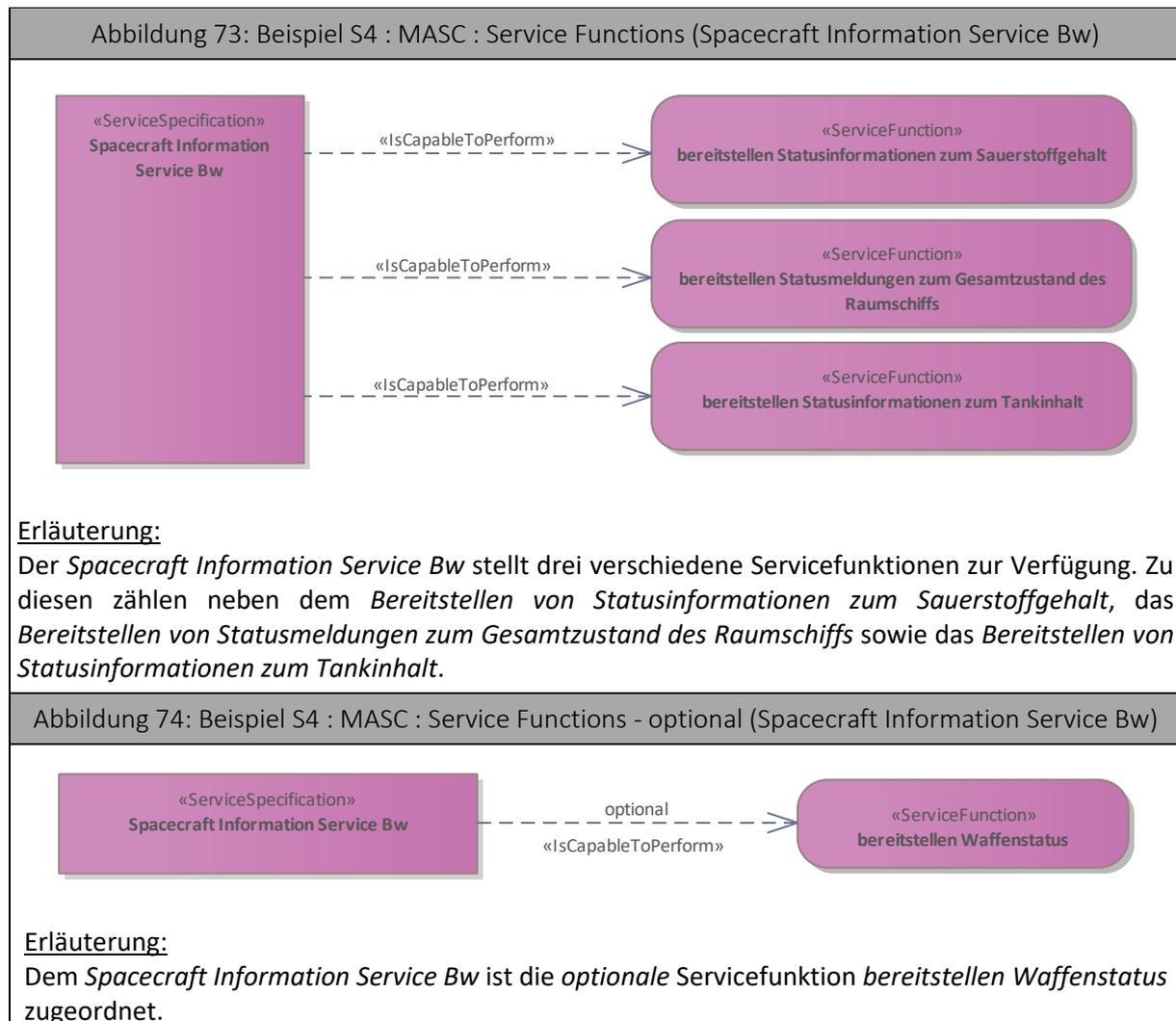
Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.31.



S4 – Service Functions

Der *S4 – Service Functions* dient der Darstellung der Funktionen, die vom Service zur Verfügung gestellt werden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.32.

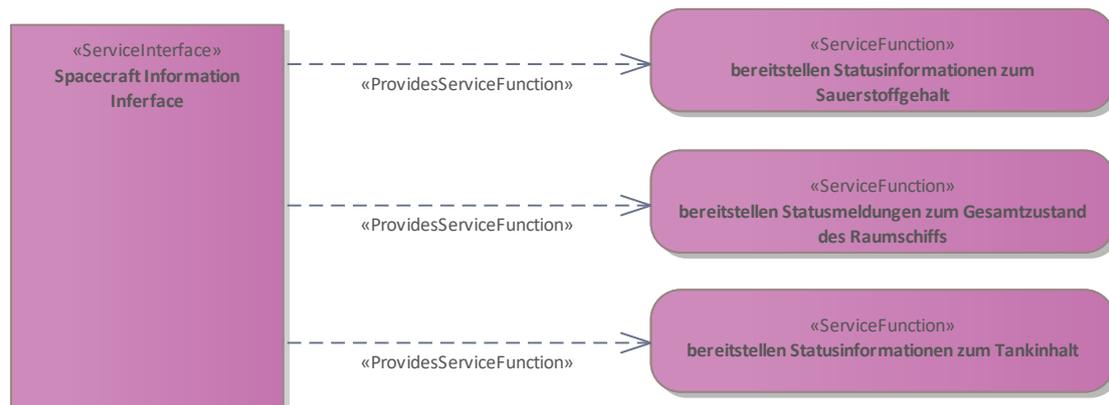


S7 – Service Interface Parameters

Der *S7 – Service Interface Parameters* beschreibt, welche Servicefunktionen eine Serviceschnittstelle nach außen anbietet.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.33.

Abbildung 75: Beispiel S7 : MASC : Service Interface Parameters (Spacecraft Information Service Bw)



Erläuterung:

Das SERVICEINTERFACE des *Spacecraft Information Interface* stellt die gleichen Servicefunktionen nach außen bereit wie der *Spacecraft Information Service Bw* selber. Zu diesen zählen neben dem *Bereitstellen von Statusinformationen zum Sauerstoffgehalt*, das *Bereitstellen von Statusmeldungen zum Gesamtzustand des Raumschiffs* sowie das *Bereitstellen von Statusinformationen zum Tankinhalt*, welche nach außen angeboten werden.

Sr – Service Change

Der *Sr – Service Change* beschreibt, an welchen Services im Rahmen der Servicedokumentation ein Änderungsbedarf besteht.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.34.

Abbildung 76: Beispiel Sr : MASC : Service Change (Spacecraft Information Service Bw)



Erläuterung:

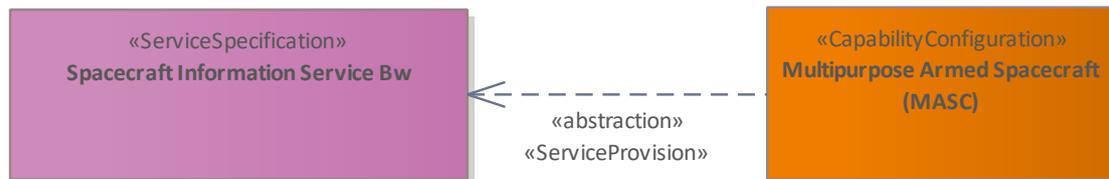
Am *Spacecraft Information Service Bw* soll eine Anpassung der Schnittstellen erfolgen.

P1 – Projektbezug

Der *P1 – Projektbezug* erfasst die einem Service zugeordneten Ressourcen.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.35.

Abbildung 77: Beispiel P1 : MASC : Serviceprovision



Erläuterung:

Im o. a. Beispiel wird dargestellt, dass das *MASC* den *Service Spacecraft Information Service Bw* bereitstellt.

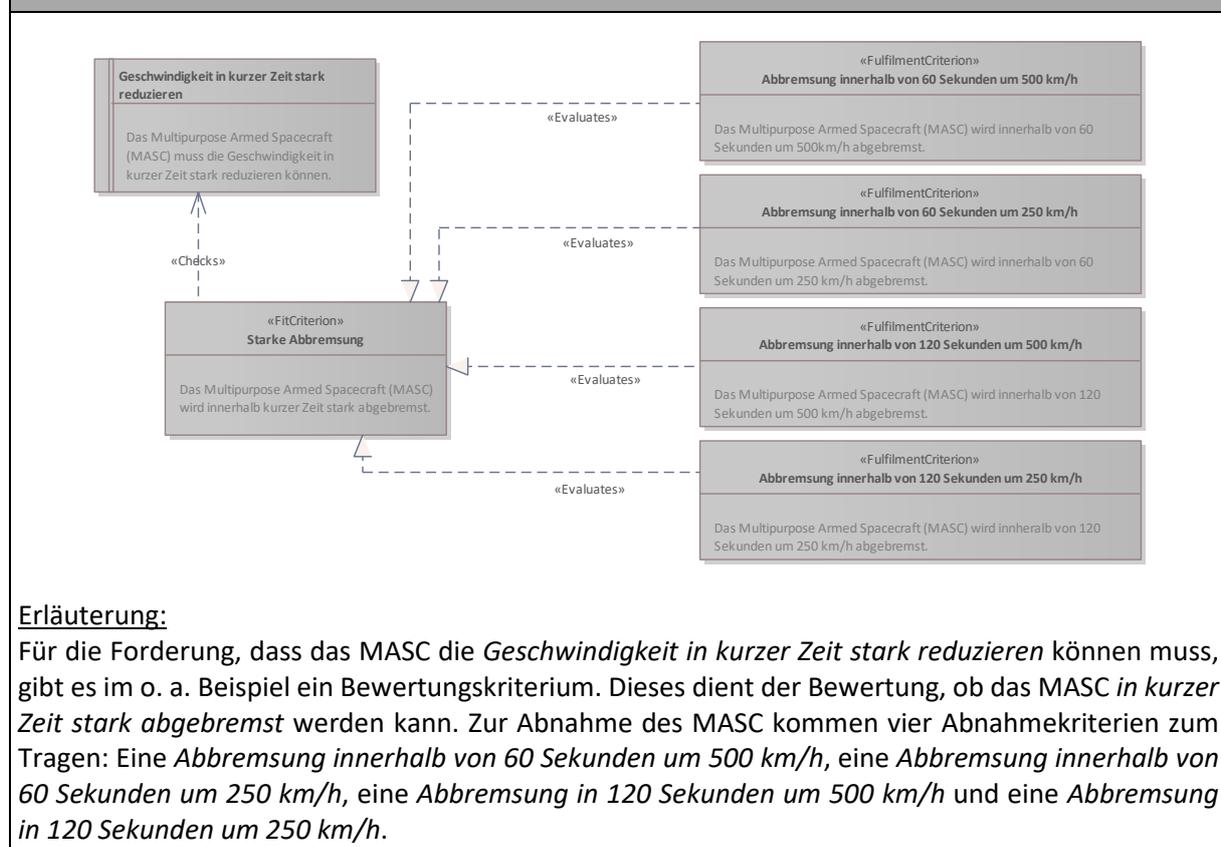
R8 – Requirement Fulfilment (optional Viewpoint)

Der *R8 – Requirement Fulfilment* behandelt die Abnahme- und Prüfkriterien für die Überprüfung der Forderungen.

Hinweis: Der Viewpoint dient der Verfeinerung der Anforderung. Dabei soll diese in für den Operateur verständliche, messbare Aussagen aufgegliedert werden. Sofern die Anforderung in einer messbaren bzw. überprüfbaren Art beschrieben ist, besteht keine Notwendigkeit für diesen Viewpoint.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|--|----------------------|
| R8-MK2 | Die Informationen werden auf dem Diagramm <i>R8 : Projektname : Requirement Fulfilment</i> im Package <i>R8/ Requirement Fulfilment</i> erfasst. | R8-A |
| R8-MK3 | Alle relevanten Anforderungen sind aus dem Katalog auf das Diagramm gezogen. | R8-B |
| R8-MK4 | Alle Bewertungskriterien sind angelegt oder aus dem Katalog auf das Diagramm gezogen und einer Anforderung zugewiesen | R8-C R8-D |
| R8-MK5 | Alle Abnahmekriterien sind angelegt oder aus dem Katalog auf das Diagramm gezogen und einem Bewertungskriterium zugewiesen. | R8-F |

Abbildung 78: Beispiel R8 : MASC : Requirement Fulfilment

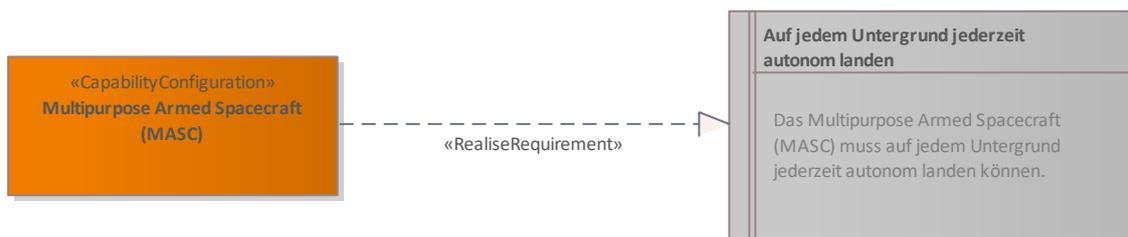


Rr – Requirement Realization

Der *Rr – Requirement Realization* hat das Ziel, die Realisierung von Forderungen darzustellen.

| Konventionsnummer | Modellierungskonvention | Modellierungsschritt |
|-------------------|---|----------------------|
| R3-MK1 | Die Informationen werden auf dem Diagramm <i>Rr : Projektname : Requirement Realization</i> im Package <i>Rr/Requirement Realization</i> erfasst. | R3-A |
| R3-MK2 | Alle relevanten Anforderungen sind aus dem Katalog auf das Diagramm gezogen. | R3-B |
| R3-MK3 | Alle relevanten anforderungsrealisierenden Elemente sind aus dem Katalog auf das Diagramm gezogen. | R3-C |
| R3-MK4 | Sämtliche Anforderungen sind den realisierenden Elementen zugewiesen. | R3-D |

Abbildung 79: Beispiel Rr : MASC : Requirement Realization



Erläuterung:

Im o. a. Beispiel wird auszugsweise dargestellt, durch welche Ressourcen die *Forderungen*, des in der Analysephase Teil 2 aufgestellten Forderungskatalogs, umgesetzt werden. So wird die *Forderung*, dass das *MASC auf jedem Untergrund jederzeit autonom landen* können muss, durch das *MASC selbst* erfüllt.

Auswertediagramme

Wie schon in der Analysephase Teil 2 kann es auch in der Realisierungsphase zielführend sein, Sichten zu definieren, die dazu dienen, einen ganz bestimmten und nur für das Projekt bestimmten Analysebedarf zu decken.

Alle relevanten Informationen zu den Auswertediagrammen finden Sie in Kapitel 2.3.41.

System- und Servicearchitekturen in der Nutzungsphase

In diesem Kapitel werden die Sichten des ADMBw angesprochen, die für die Architekturerstellung in der Nutzungsphase (NP) von Relevanz sein können. Die Auswahl der Sichten in diesem Dokument ist als „Best Practice“ zu verstehen und greift den generellen Analyse- und Informationsbedarf in der Nutzungsphase auf. In Einzelfällen können bestimmte Sichten entfallen oder zusätzliche Sichten müssen modelliert werden.

Im Sinne der vereinfachten Darstellung sind in der folgenden Abbildung die wesentlichen Elemente und Relationen eines Architekturmodells bzw. einer Architektur für die Nutzungsphase dargestellt. Die Abbildung beschränkt sich auf die wichtigsten Elemente und ist nicht vollständig. Die detaillierte Darstellung ist der Dokumentation des ADMBw zu entnehmen. Ein detaillierter Einblick in das MetaModell des ADMBw kann über den in Anlage 0 aufgeführten Link (MetaModelViewer) abgerufen werden.

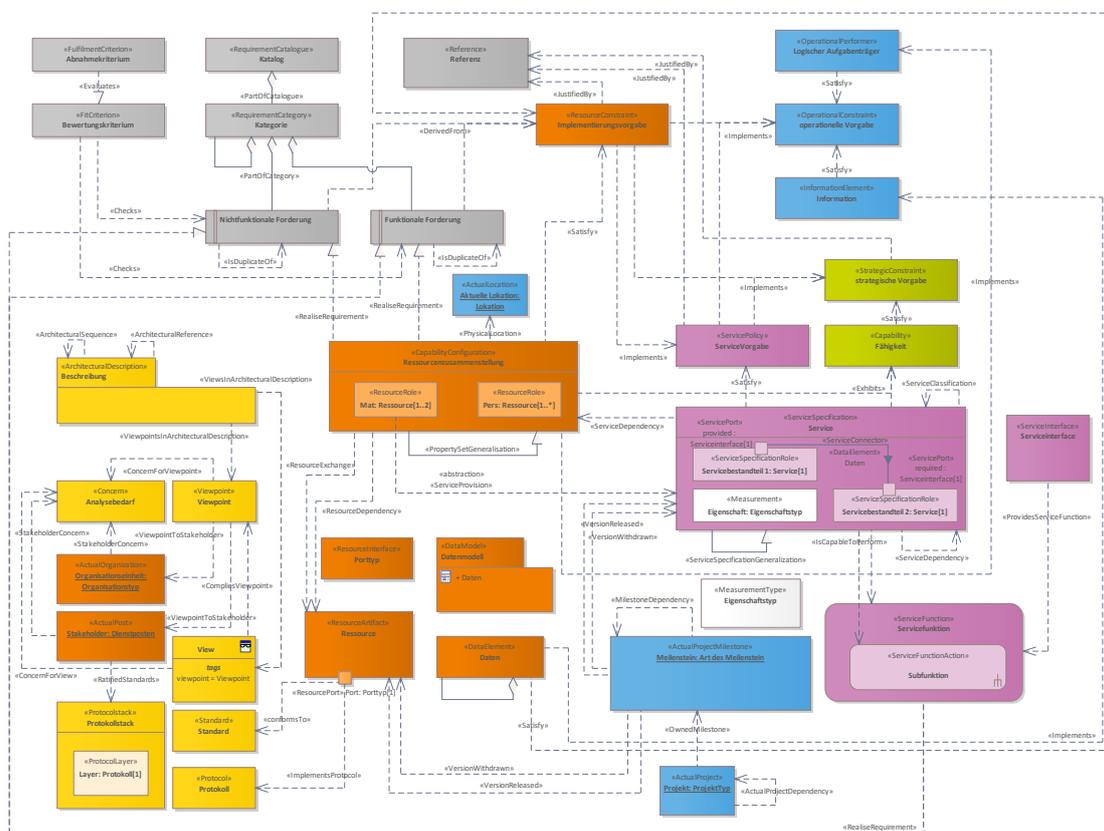


Abbildung 80: Überblick der Elemente in System- und Servicearchitekturen in der NP

Hinweis: Bei der Architekturerstellung ist es wichtig zu verstehen, dass es nicht darum geht Diagramm und Bilder zu erstellen, sondern ein Modell zu generieren. Die Viewpoints stehen nie alleine, da diese immer nur einen Ausschnitt aus einem Gesamtmodell repräsentieren. Die Zusammenhänge der Sichten ergeben sich über die Inhalte, die dargestellt werden. In einer rein theoretischen Betrachtung ist es möglich das gesamte Modell auf einer Sicht darzustellen und anschließend in handhabbare und überschaubare Diagramme aufzuteilen.

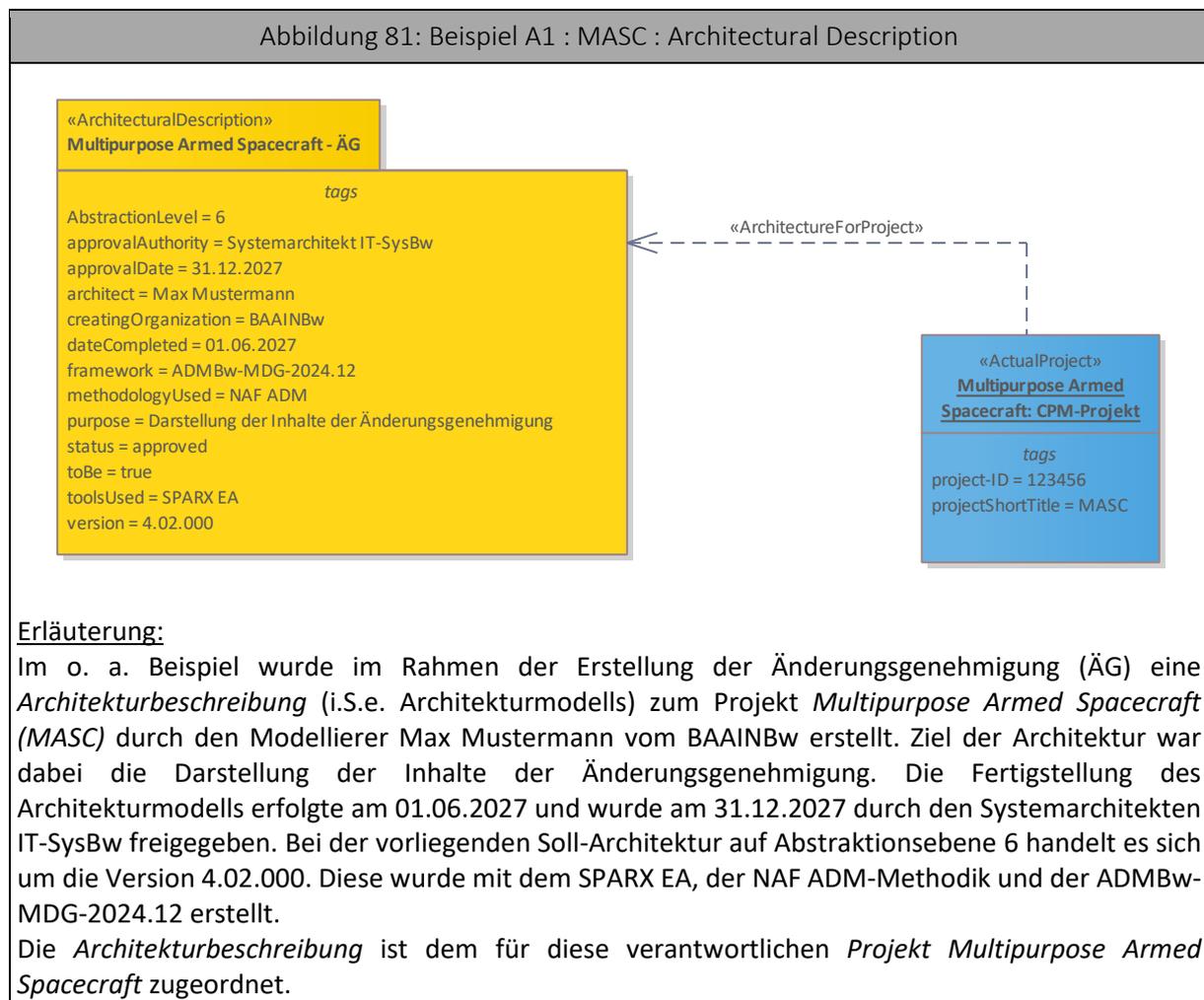
A1 – Meta–Data Definitions

Der *A1 – Meta Data Definitions* befasst sich mit den Metadaten einer Architektur, welche über die gesamte Architektur hindurch verwendet werden.

Der A1 dient in der Architektur für die Nutzungsphase dazu, weitere, noch nicht in den vorherigen Phasen des CPM erfasste Informationen zur Architektur zusammenzufassen. Durch die Festschreibung relevanter Informationen, wie z. B. des Analyse- und Informationsbedarfs, der Stakeholder der Architektur oder wichtigen Ansprechpersonen wird eine zielgerichtete Weiter- und Wiederverwendung der Architektur erst ermöglicht.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.3.

Abbildung 81: Beispiel A1 : MASC : Architectural Description



A2 – Architecture Products

Der *A2 – Architecture Products* beschreibt die Architekturprodukte, die in einer Architektur vorhanden sind.

Der A2-Viewpoint dient in der Architektur für die Nutzungsphase dazu, die Stakeholder der Architektur und deren Anliegen bzw. deren Analyse- bzw. Informationsbedarf zu definieren. Daraus werden alle weiteren notwendigen Viewpoints für die Architektur abgeleitet.

Der A2 ermöglicht eine Bewertung, inwieweit der Analyse-/Informationsbedarf durch entsprechende Diagramme abgedeckt wird.

Unter Architekturprodukten werden neben den erstellten Diagrammen (Views) auch der Analysebedarf selbst und die an der Modellierung beteiligten Personen verstanden.

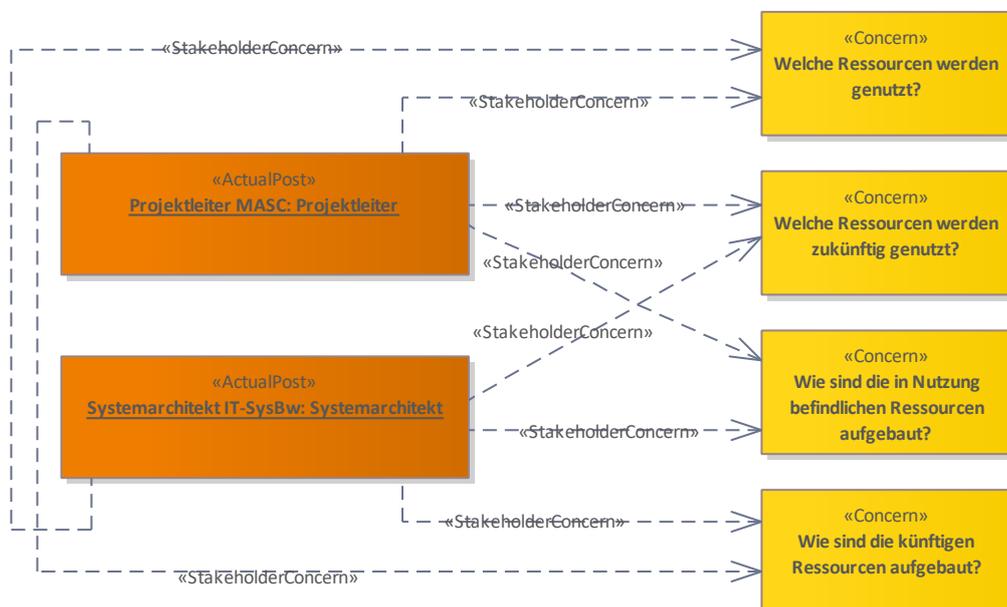
Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.4, 2.3.5, 2.3.6, 2.3.7.

Abbildung 82: Beispiel A2 : MASC : Concerns

**Erläuterung:**

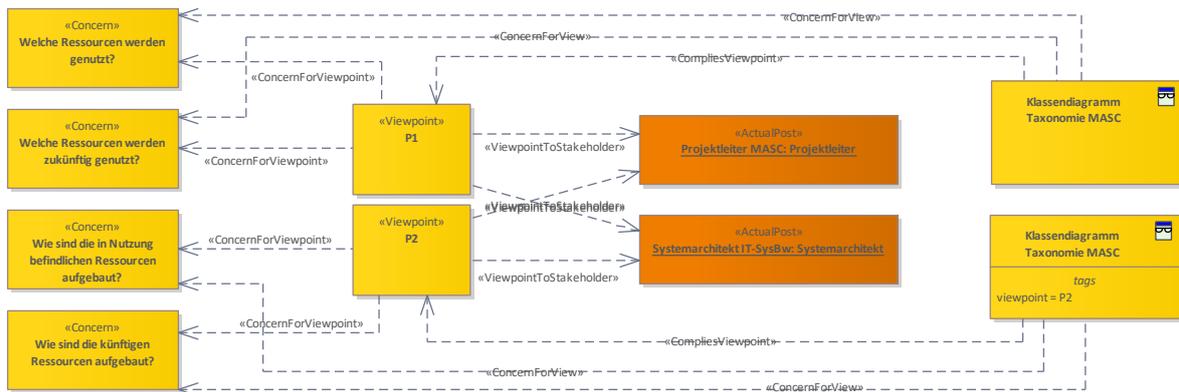
Die im Beispiel betrachteten Stakeholder des MASC möchten zum einen wissen, *welche Ressourcen aktuell genutzt werden und wie diese aufgebaut sind*. Zum anderen möchten sie wissen, *welche Ressourcen zukünftig genutzt werden sollen und wie die neuen Ressourcen aufgebaut sind*.

Abbildung 83: Beispiel A2 : MASC : Stakeholder

**Erläuterung:**

Die im Beispiel betrachteten Stakeholder des MASC sind der *Systemarchitekt IT-SysBw* sowie der *Projektleiter MASC*. Sowohl der *Projektleiter MASC* als auch der *Systemarchitekt IT-SysBw* möchten wissen, *welche Ressourcen (zukünftig) genutzt werden und wie diese (zukünftig) aufgebaut sind*.

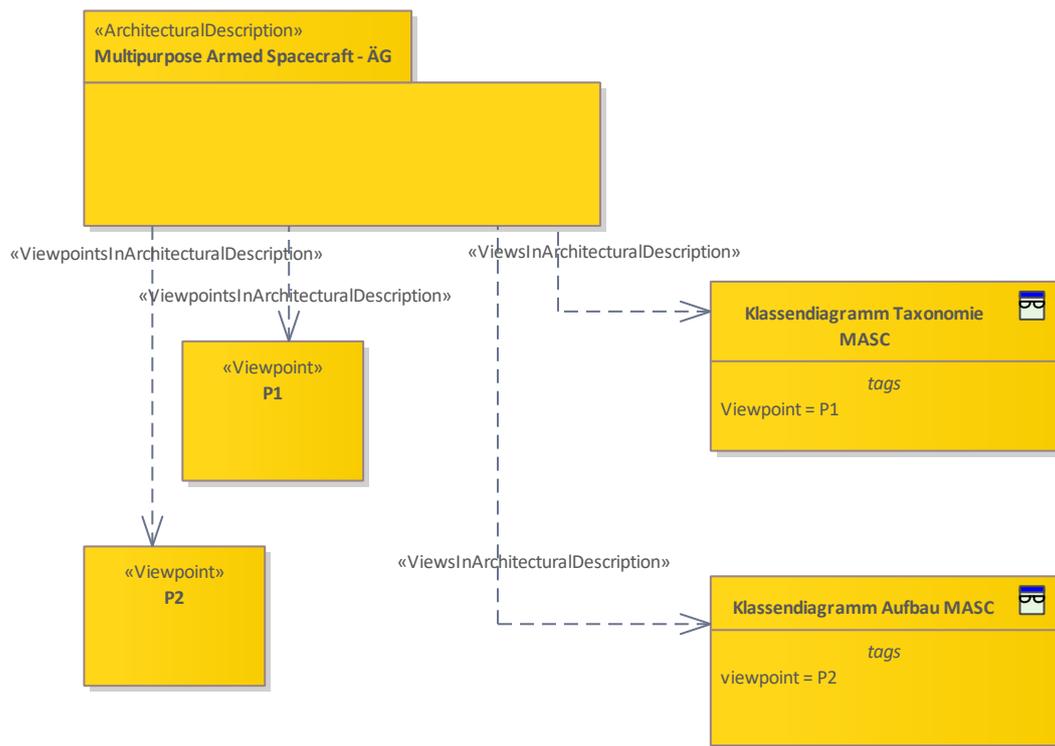
Abbildung 84: Beispiel A2 : MASC : Viewpoints



Erläuterung:

Die Viewpoints und Views, die in diesem Beispiel im Rahmen der Modellierung des MASC zu berücksichtigen sind, sind die Viewpoints P1 und P2 sowie die beiden angegebenen Klassendiagramme. Sie ergeben sich aus den Informationsbedarfen (Concerns) der Stakeholder und werden diesen ebenfalls zugewiesen.

Abbildung 85: Beispiel A2 : MASC : Architecture Products



Erläuterung:

Zum Abschluss der Modellierung des Viewpoints A2 zur ÄG werden die zu modellierenden Viewpoints (P1 und P2) und Views (Klassendiagramm Aufbau MASC und Klassendiagramm Taxonomie MASC) zur Architekturbeschreibung des MASC hinzugefügt.

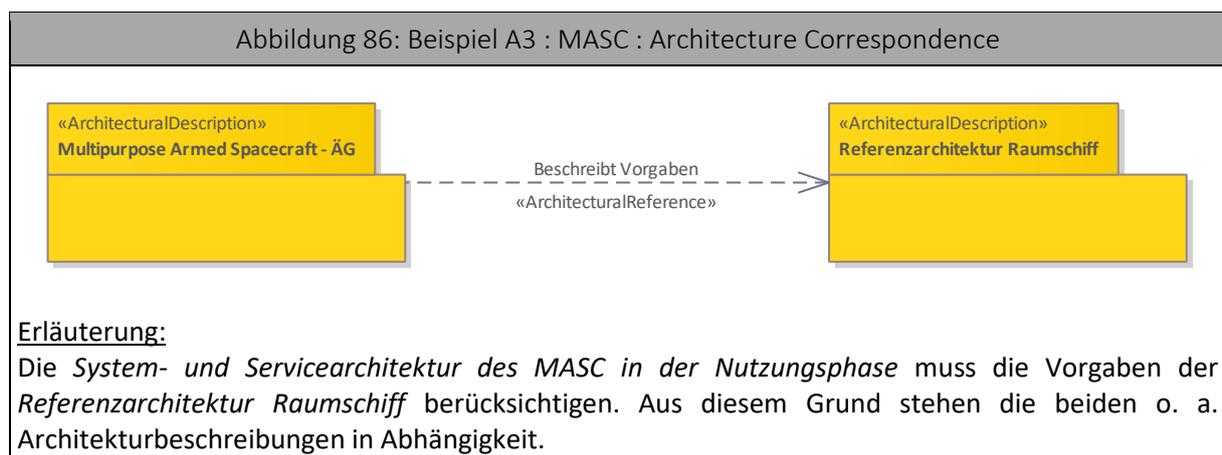
A3 – Architecture Correspondence

Der *A3 – Architecture Correspondence* befasst sich mit Abhängigkeiten zwischen Architekturen.

Der A3 dient in der Architektur für die Nutzungsphase dazu, die in den vorherigen Phasen des CPM beschriebenen Abhängigkeiten zwischen Architekturen bei Bedarf zu ergänzen und auf das Architekturmodell der Nutzungsphase umzuhängen. Die Abhängigkeiten reichen wie schon in der Analysephase Teil 1, 2 sowie der Realisierungsphase von der Nennung der entsprechenden Architekturen bis hin zur Wieder- und Weiterverwendung von Architekturartefakten.

Die Abhängigkeit zwischen Projekten ist kein Bestandteil des A3, sondern wird im Lr - Lines of Development dargestellt.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.8.



A6 – Architecture Versions

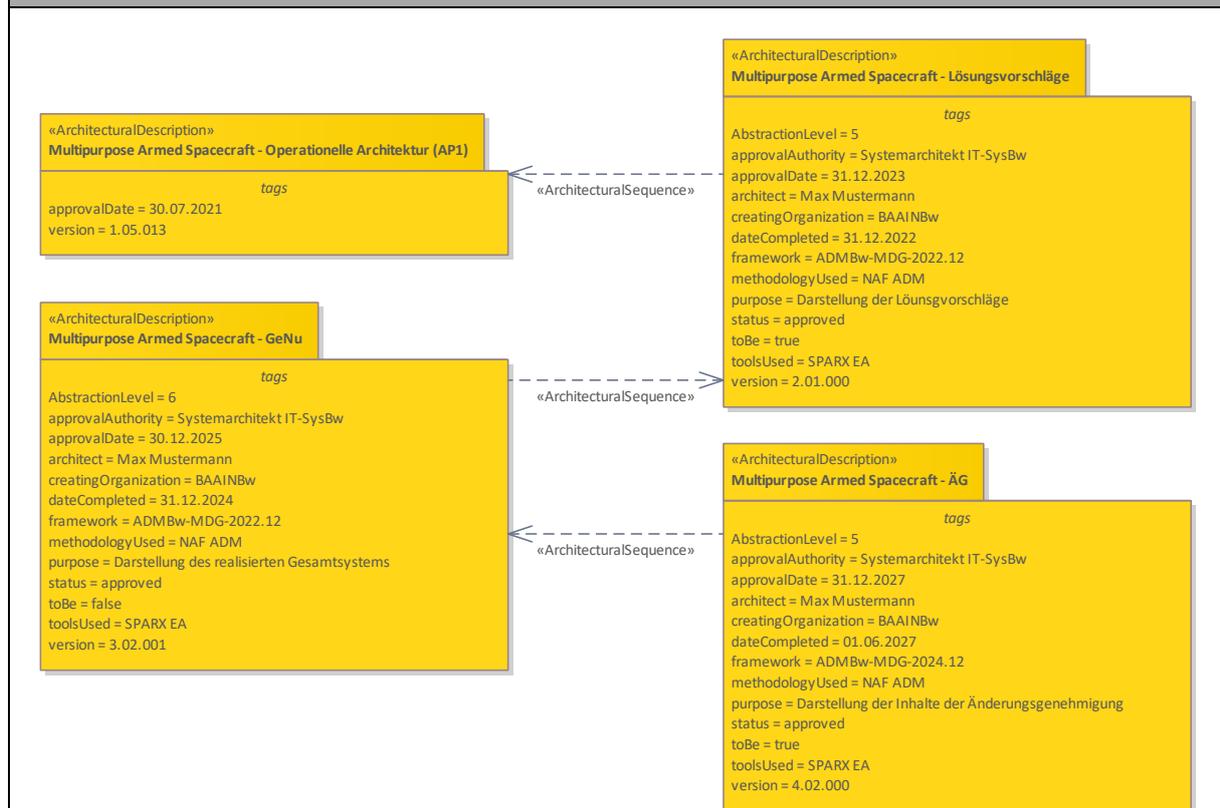
Der *A6 – Architecture Versions* befasst sich mit der Versionsgeschichte der Architektur, wobei relevante Metadaten genutzt werden.

Der A6 dient in der Architektur für die Nutzungsphase dazu, die Versionsgeschichte der betrachteten Architektur nachzuvollziehen. Ist die vorliegende Version die erste Architektur zum betrachteten Projekt, ist der A6 zu initial zu erstellen und bei jeder neuen Version der Architektur fortzuschreiben.

Wenn in der Nutzungsphase auf einer in der Realisierungsphase erstellten Architektur aufgesetzt wird, ist eine neue Architekturbeschreibung für die Nutzungsphase im A1 anzulegen und die beiden Beschreibungen im A6 zu verknüpfen.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.9.

Abbildung 87: Beispiel A6 : MASC : Architecture Versions

**Erläuterung:**

Die gerade in der Erarbeitung befindliche *System- und Servicearchitektur des Multipurpose Armed Spacecraft für die Nutzungsphase (Version 4.02.000)* ist der Nachfolger der finalen *System- und Servicearchitektur des Multipurpose Armed Spacecraft aus der Realisierungsphase (Version 3.02.001)*. Die finale *System- und Servicearchitektur des Multipurpose Armed Spacecraft aus der Realisierungsphase (Version 3.02.001)* basiert, wie schon in der Realisierungsphase beschrieben, auf der finalen *System- und Servicearchitektur des Multipurpose Armed Spacecraft aus der Analysephase Teil 2 (Version 2.01.000)*. Die finale *System- und Servicearchitektur der Analysephase Teil 2 (Version 2.01.000)* beruht selbst auf der finalen *Operationellen Architektur des Multipurpose Armed Spacecraft aus der Analysephase Teil 1 (Version 1.05.013)*.

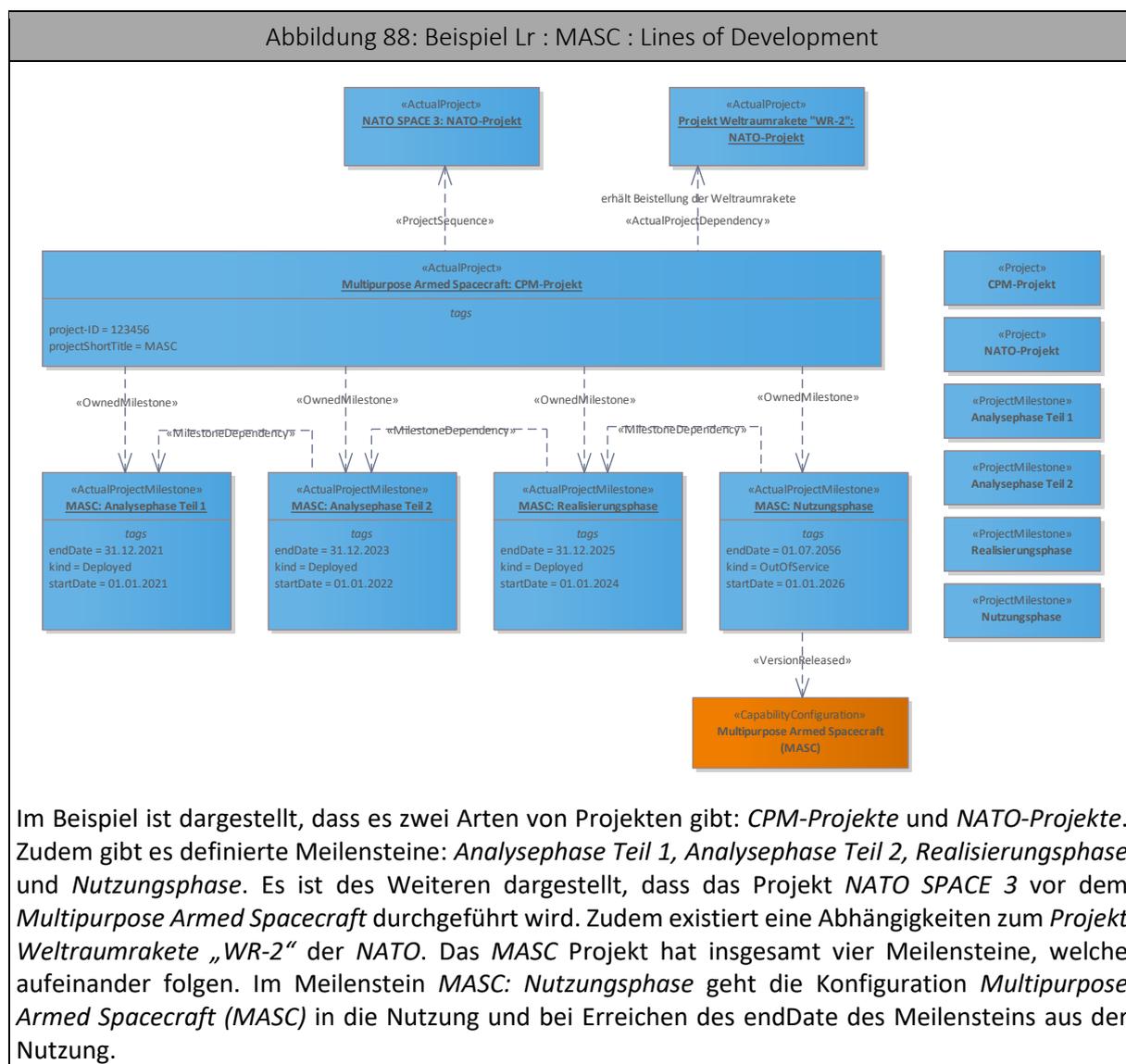
Lr – Lines of Development

Der *Lr – Lines of Development* behandelt die Identifizierung der Projektabhängigkeiten. Dazu werden die relevanten Programme und Projekte und ihre Abhängigkeiten dargestellt.

Der Lr dient in der Nutzungsphase wie schon in der Realisierungsphase sowie Analysephase Teil 2 dazu, die wesentlichen Abhängigkeiten des betrachteten Projekts zu anderen Projekten abzubilden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich bei Bedarf fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.10.

Abbildung 88: Beispiel Lr : MASC : Lines of Development



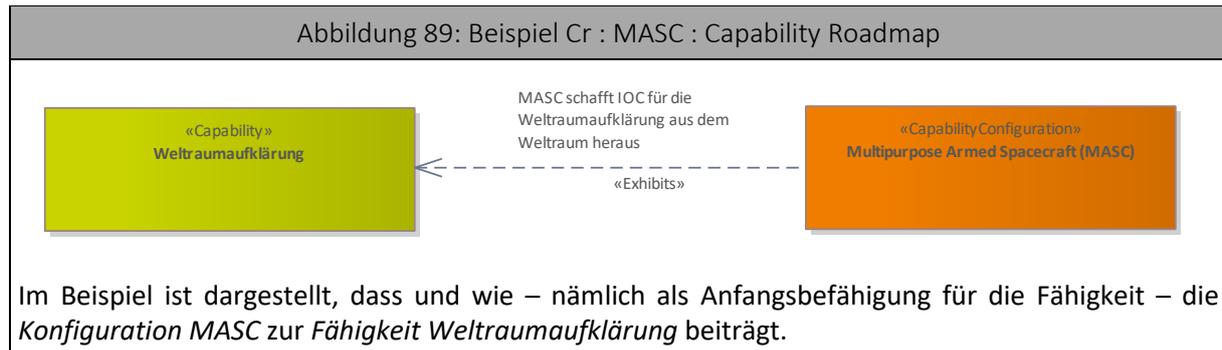
Im Beispiel ist dargestellt, dass es zwei Arten von Projekten gibt: *CPM-Projekte* und *NATO-Projekte*. Zudem gibt es definierte Meilensteine: *Analysephase Teil 1*, *Analysephase Teil 2*, *Realisierungsphase* und *Nutzungsphase*. Es ist des Weiteren dargestellt, dass das Projekt *NATO SPACE 3* vor dem *Multipurpose Armed Spacecraft* durchgeführt wird. Zudem existiert eine Abhängigkeiten zum *Projekt Weltraumrakete „WR-2“* der *NATO*. Das *MASC* Projekt hat insgesamt vier Meilensteine, welche aufeinander folgen. Im Meilenstein *MASC: Nutzungsphase* geht die *Configuration Multipurpose Armed Spacecraft (MASC)* in die Nutzung und bei Erreichen des *endDate* des Meilensteins aus der Nutzung.

Cr – Capability Roadmap (optionaler Viewpoint)

Der *Cr – Capability Roadmap* behandelt die Darstellung der Ressourcen, die zur Bereitstellung einer Fähigkeit herangezogen werden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern nur bei Bedarf fortgeschrieben. Sollte es bis zur Realisierungsphase noch keinen Cr geben, so ist dieser neu zu modellieren. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.11.

Abbildung 89: Beispiel Cr : MASC : Capability Roadmap



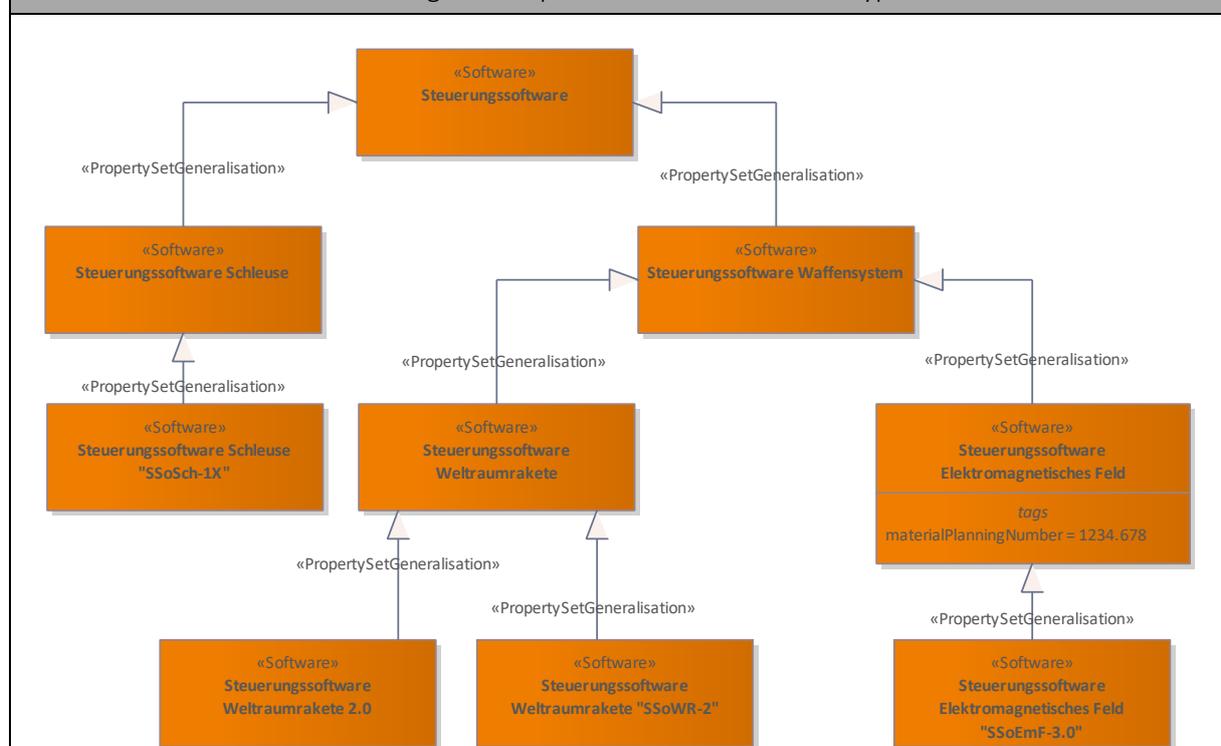
P1 – Resource Types

Der *P1 – Resource Types* behandelt die Spezifikation der Ressourcen, erforderlichen Technologien und Kompetenzen.

Im P1 sollen, sofern möglich und vorhanden, für die Abbildung der realisierten und neu zu realisierenden Ressourcen Standardelemente verwendet werden. Die Aufnahme und Zuordnung eines neuen Elementes in den Standardelementekatalog sind mit dem begleitenden Referat abzustimmen.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.12.

Abbildung 90: Beispiel P1 : MASC : Resource Types



Erläuterung:

Das o. a. Beispiel stellt einen Auszug der für das MASC relevanten Ressourcen dar. Der Fokus liegt dabei auf den verschiedenen Arten von *Steuerungssoftware*. Da bei einer ÄG nicht nur die bereits realisierten Produkte, sondern auch die zu ändernden Ressourcen benannt werden müssen, wurde die Taxonomie im Vergleich zu den Inhalten der GeNu entsprechend angepasst. Während sich die Taxonomie der *Steuerungssoftware für die Schleuse* und der *Steuerungssoftware für das Elektromagnetische Feld* sich nicht verändert haben, gibt es zukünftig aufgrund des Updates einer der beiden Weltraumrakete auf die neueste Version eine neue, dritte Ausprägung der *Steuerungssoftware für Waffensysteme*, die *Steuerungssoftware Weltraumrakete 2.0*.

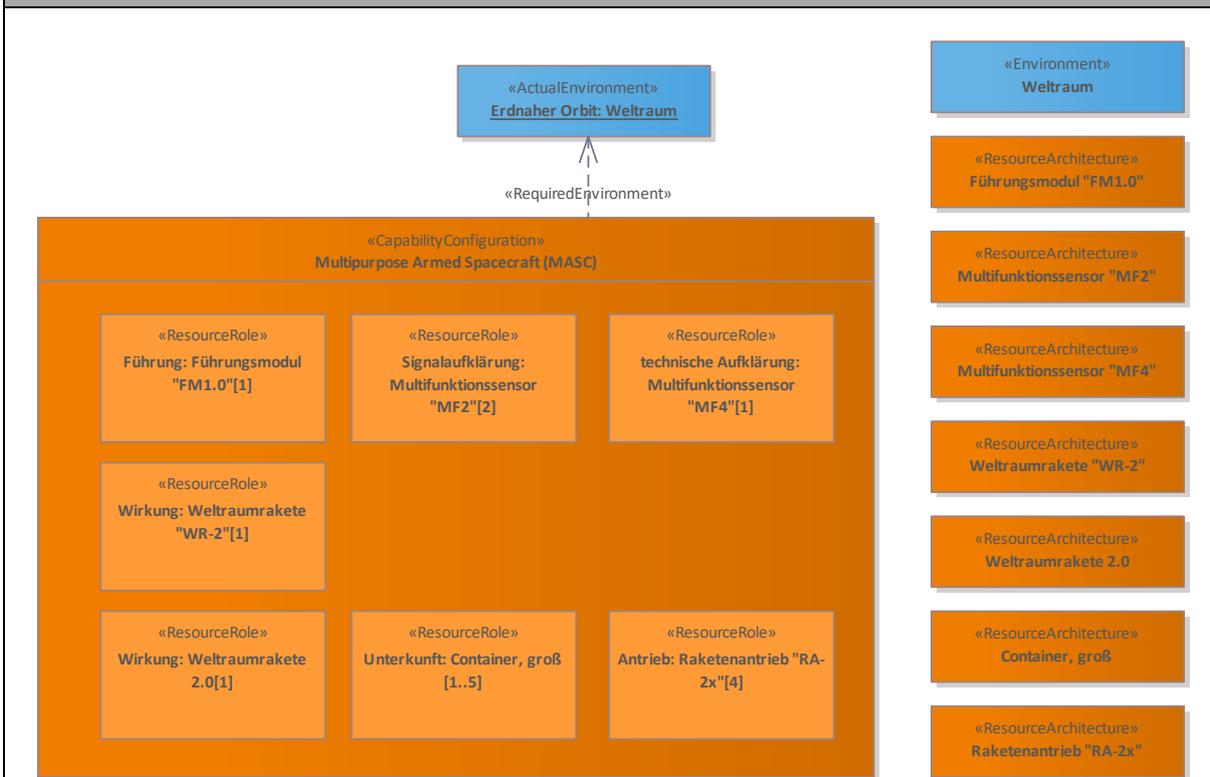
P2 – Resource Structure

Der *P2 – Resource Structure* behandelt die Zusammensetzung und Interaktion von Ressourcen.

Der P2 dient in der Architektur für die Nutzungsphase dazu, den internen Aufbau der realisierten und neu zu realisierenden Ressourcen darzustellen. Die Darstellung des Aufbaus ist notwendig, um realisierte und neu zu realisierende Anforderungen einzelnen Systembestandteilen zuordnen und somit eine zielgerichtete Wiederverwendung von Ressourcen ermöglichen zu können. Zusätzlich zum Genannten wird im P2 nicht nur dargestellt, welche operationellen Anteile durch die Ressourcen umgesetzt werden bzw. werden sollen, sondern auch, welche externen Abhängigkeiten und Datenaustauschbeziehungen existieren bzw. zukünftig existieren sollen.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.13, 2.3.14, 2.3.15, 2.3.16.

Abbildung 91: Beispiel P2 : MASC : Resource Structure

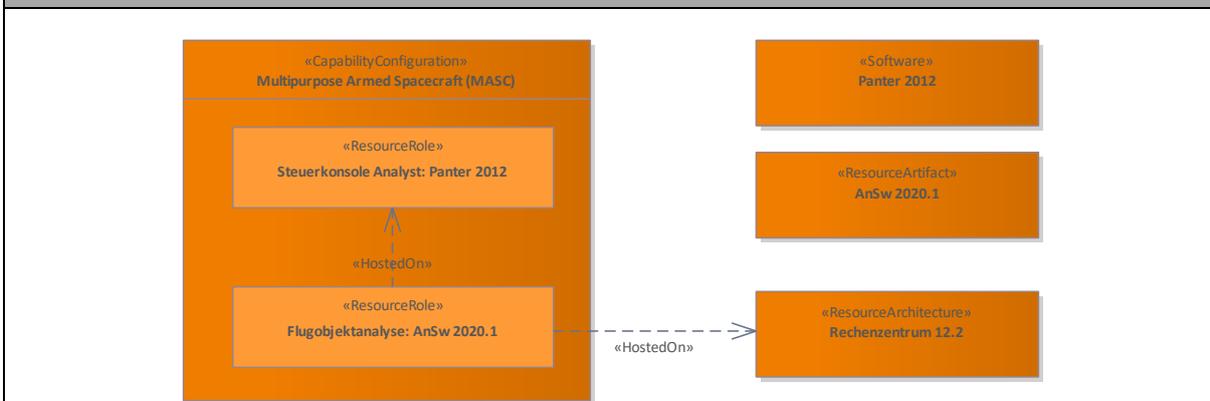


Erläuterung:

Das o. a. Beispiel zeigt den Aufbau des MASC. Da im Rahmen der ÄG die bereits realisierten und neu zu beschaffenden Ressourcen benannt werden müssen, wurde dieser View im Vergleich zu den Inhalten der GeNu entsprechend fortgeschrieben.

Wie bereits zur GeNu besteht das MASC auch weiterhin aus einem Führungsmodul „FM1.0“, zwei Multifunktionssensoren „MF2“ und einem Multifunktionssensor „MF4“, vier Raketenantrieben „RA-2x“ und Platz für ein bis fünf „Container, groß“ für die Unterbringung des Personals. Aufgrund der im Rahmen der Änderungsgenehmigung geplanten Änderungen wird eine der beiden zur GeNu implementierten Weltraumraketen „WR-2“ durch die neue Weltraumrakete 2.0 ersetzt. Am Einsatzgebiet des MASC hat sich im Vergleich zur GeNu nichts geändert.

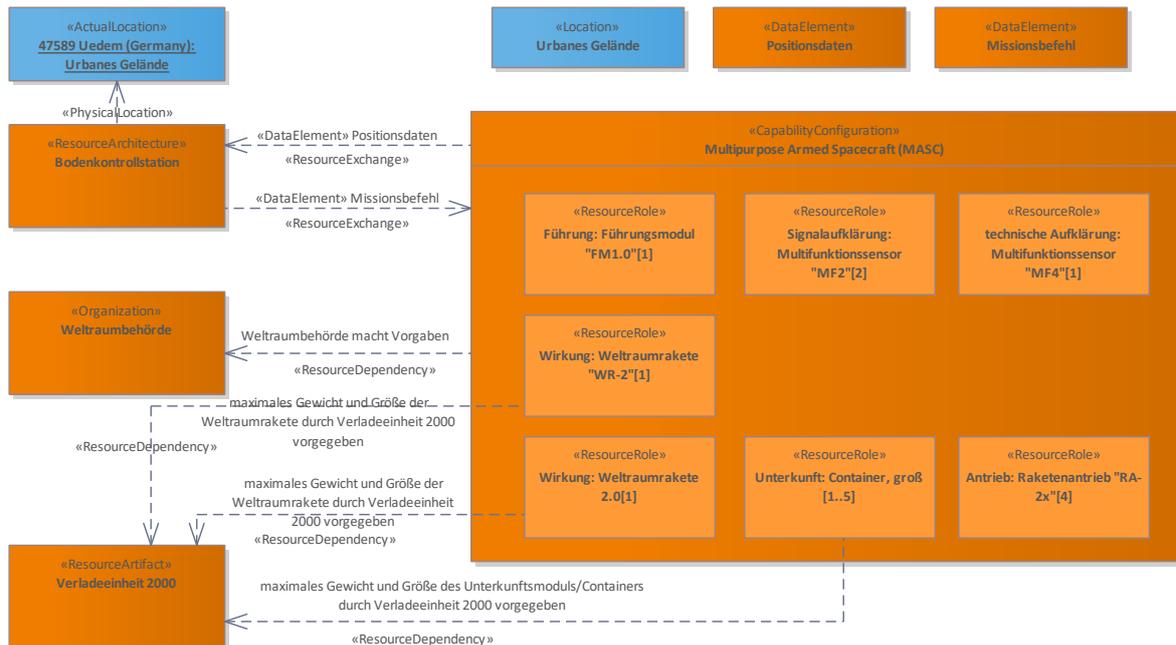
Abbildung 92: Beispiel P2 : MASC : Application Hosting



Erläuterung:

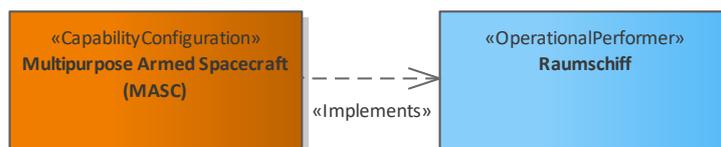
Das o. a. Beispiel zeigt die AnSw 2020.1 als Bestandteil des MASC. Diese wird auf dem Panter 2012 und im Rechenzentrum 12.2 betrieben. Es gibt keine Änderungen zur Realisierungsphase.

Abbildung 93: Beispiel P2 : MASC : External Dependencies

**Erläuterung:**

Im o. a. Beispiel wird zum einen dargestellt, dass das MASC auch weiterhin einen bidirektionalen Datenaustausch mit der *Bodenkontrollstation in 47589 Uedem* durchführt. Zum anderen hängen die Größe und das Gewicht des Containers nun nicht mehr wie bisher von der Verladeeinheit, sondern nun von der zwischenzeitlich neu entwickelten *Verladeeinheit 2000* ab, die den Container am Boden in das Raumschiff verlädt. Zusätzlich dazu kann die neue *Verladeeinheit 2000* aufgrund der neuesten Sicherheitsstandards ebenfalls *Weltraumraketen* verladen, wodurch auch die *maximale Größe und das Gewicht* der gerüsteten *Weltraumraketen „WR-2“* und *Weltraumrakete 2.0* von der *Verladeeinheit 2000* abhängig sind. An der Abhängigkeit zur *Weltraumbehörde* hat sich in der Nutzungsphase keine Änderung ergeben.

Abbildung 94: Beispiel P2 : MASC : Node Realization

**Erläuterung:**

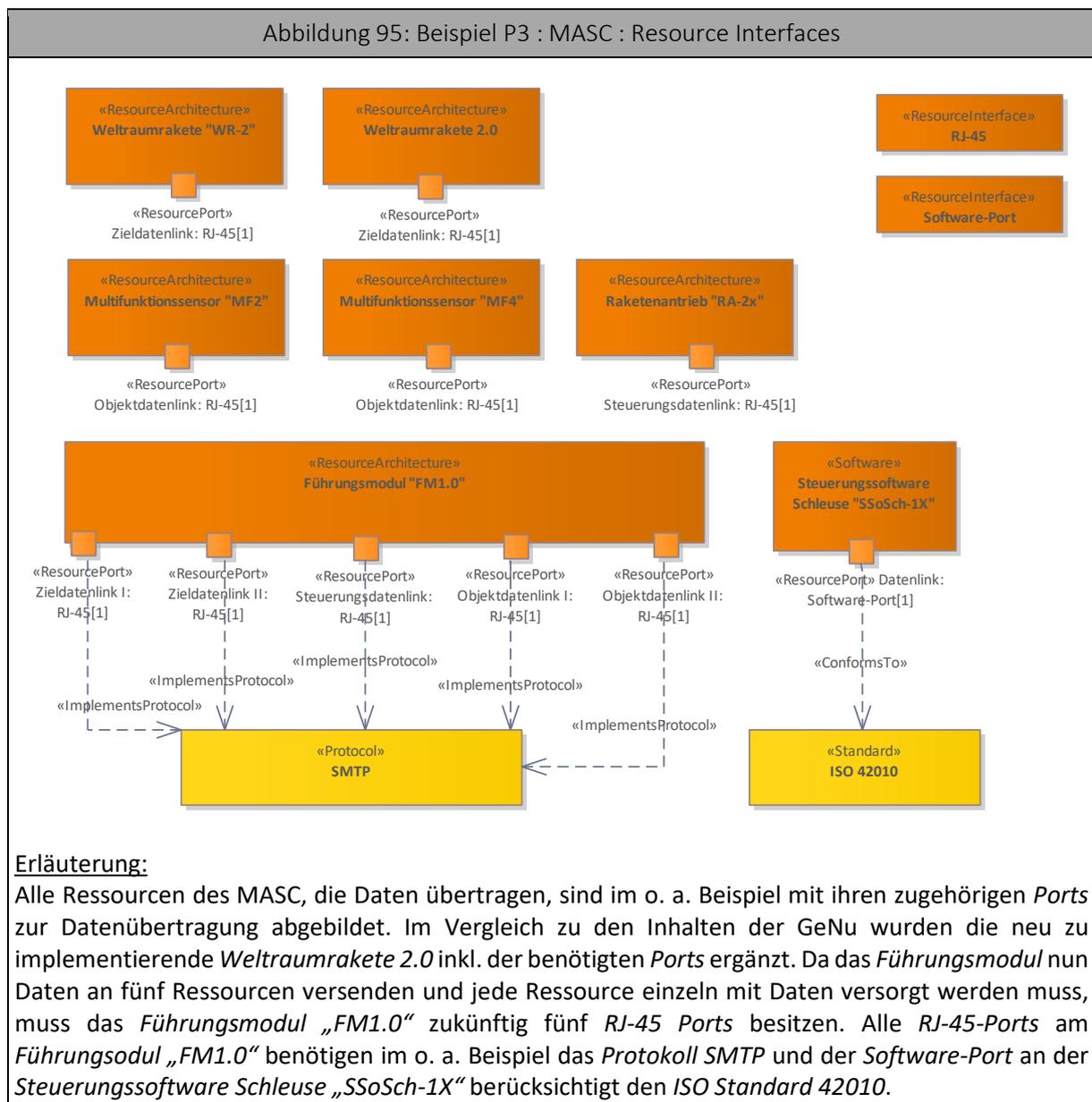
Im o. a. Beispiel wird dargestellt, dass das MASC dazu dient, den in der FFF beschriebenen, logischen Rollen- und Aufgabenträger *Raumschiff* umzusetzen. Im Vergleich zu den Inhalten der GeNu hat sich hier keine Änderung ergeben.

P3 – Resource Interfaces

Der *P3 – Resource Interfaces* beschreibt die Systemschnittstellen.

In der Nutzungsphase dient der P3 dazu, die existenten sowie die neu zu schaffenden Schnittstellen von Ressourcen inklusive der zugeordneten Standards und Protokolle zu beschreiben.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.17.



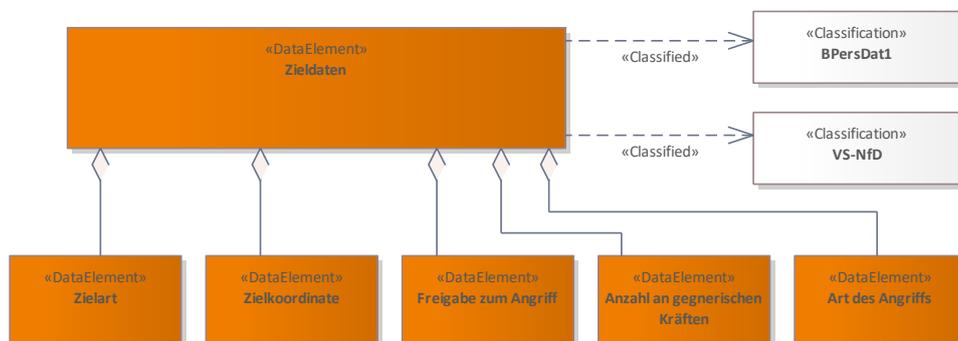
P7 – Data Model

Der *P7 – Data Model* behandelt die Struktur der Daten, die von den Ressourcentypen in der Architektur untereinander ausgetauscht werden.

Der P7 dient in der Architektur für die Nutzungsphase dazu, dass in den vorherigen Phasen des CPM entwickelte Datenmodell bei Bedarf fortzuschreiben.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.18, 2.3.19, 2.3.20.

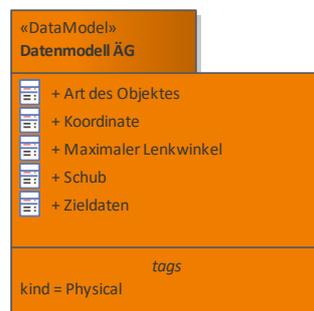
Abbildung 96: Beispiel P7 : MASC : Data Elements



Erläuterung:

Der o. a. View stellt dar, dass sich das Datenelement *Zieldaten* zukünftig aus fünf anderen Datenelementen zusammensetzt. Zu diesen zählen die *Zielart*, die *Zielkoordinate*, die *Anzahl an generischen Kräften* und die *Freigabe zum Angriff*. Im Vergleich zu den Inhalten der GeNu wurde das Datenelement *Art des Angriffs* ergänzt, da dieses vorher Bestandteil des Datenelementes *Freigabe zum Angriff* war und zur Verbesserung der Transparenz im Rahmen der Erstellung der ÄG aus der *Freigabe zum Angriff* herausgelöst und separat aufgeführt wurde.

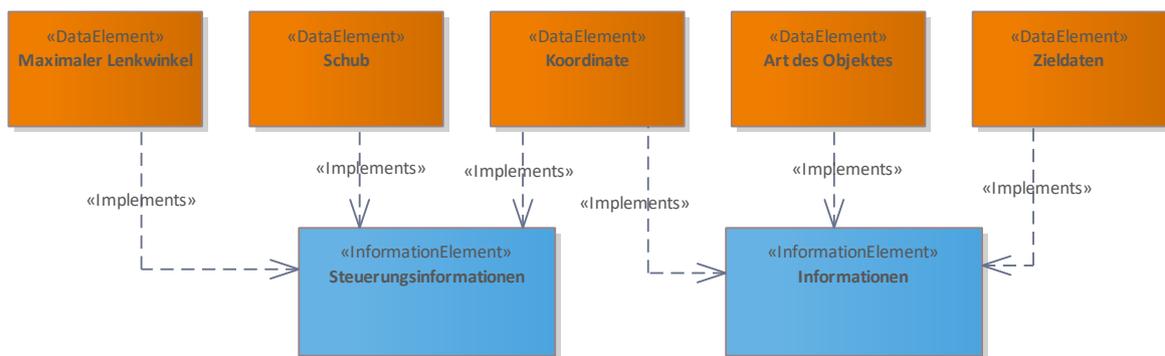
Abbildung 97: Beispiel P7 : MASC : Data Model



Erläuterung:

Alle im Rahmen der ÄG zum MASC benannten, weiterhin sowie zukünftig zu übertragenden Daten werden in diesem View zum physischen *Datenmodell der ÄG* hinzugefügt. Neben den *Zieldaten* gehören wie schon zur GeNu auch weiterhin die *Art des Objektes*, die *Koordinate* sowie die benötigte *Schubkraft* zum dargestellten Datenmodell. Neu hinzugefügt wurde aufgrund eines in der Nutzungsphase erkannten Fehlers das Datenelement *Maximaler Lenkwinkel*.

Abbildung 98: Beispiel P7 : MASC : InformationElement Realization

**Erläuterung:**

Wie schon im P2 Node Realization geht es auch in diesem View um die aktuelle sowie geplante Umsetzung von Elementen, die in der FFF beschrieben sind. Dieses Mal liegt der Fokus jedoch nicht auf Rollen- und Aufgabenträgern und Ressourcen, sondern auf Informationselementen und Datenelementen. Die in der FFF beschriebenen *Steuerungsinformationen* werden durch die in der Nutzungsphase aufgetretenen Fehler nun nicht mehr nur durch die *Schubkraft* und die *Koordinate*, sondern zukünftig auch durch den *Maximalen Lenkwinkel* umgesetzt. An der Realisierung der in der FFF benannten Informationen haben sich im Rahmen der ÄG keine Änderungen ergeben. Sowohl die *Koordinate* als auch die *Art des Objektes* sowie die *Zieldaten* dienen auch weiterhin der Realisierung der in der FFF benannten *Informationen*.

P8 – Resource Constraints

Der *P8 – Resource Constraints* stellt die Implementierungsvorgaben für die Ressourcen dar.

Der P8 dient in der Architektur für Nutzungsphase dazu, die Vorgaben an die geplanten Ressourcen aufzubereiten sowie mögliche Änderungen der Vorgaben an Ressourcen darzustellen.

Zu den Vorgaben zählen:

- die Nutzung bereits vorhandener Ressourcen,
- die Herstellung von Interoperabilität zu anderen Ressourcen oder
- Beschränkungen beim Einsatz bestimmter Ressourcen.

Implementierungsvorgaben werden entweder durch strategische Vorgaben begründet oder aus anderen Vorgaben abgeleitet.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.21, 2.3.22.

Abbildung 99: Beispiel P8 : MASC : Resource Constraints



Erläuterung:

Der o. a. View stellt dar, dass sich aus dem Dokument *Konzept Weltraum 2030* eine neue Vorgabe und Rahmenbedingung ableitet, die im Rahmen der ÄG zu berücksichtigen ist. Während im Rahmen der Lösungsvorschläge vom Wirkungsmodul gefordert wurde, dass es *Ziele im Weltraum in mittlerer Entfernung bekämpfen* können muss, so gibt das *Konzept Weltraum 2030* auf Seite 17 im 8. Absatz vor, dass zukünftig auch *Ziele im Weltraum in großer Entfernung bekämpft* werden können müssen. Da diese Vorgabe nicht durch die in Nutzung befindliche Weltraumrakete „WR-2“ erfüllt werden kann, hat man sich im Rahmen der ÄG aufgrund des Funktionsumfangs für ein Update auf die neueste Version der Weltraumrakete, die *Weltraumrakete 2.0*, entschieden.

Abbildung 100: Beispiel P8 : MASC : Resource Constraint Realization



Erläuterung:

Die Vorgabe, dass das MASC *Ziele in großer Entfernung im Weltraum bekämpfen* können muss, setzt die operationelle Vorgabe, dass das MASC *Ziele im Weltraum bekämpfen* können muss, um.

S1 – Service Description

Der *S1 – Service Description* dient der Erfassung der Servicebeschreibung und der Einordnung des Service in eine der drei Ebenen des Portfolios.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapite l 2.3.24.

Abbildung 101: Beispiel S1 : MASC : Service Description (Spacecraft Information Service Bw)



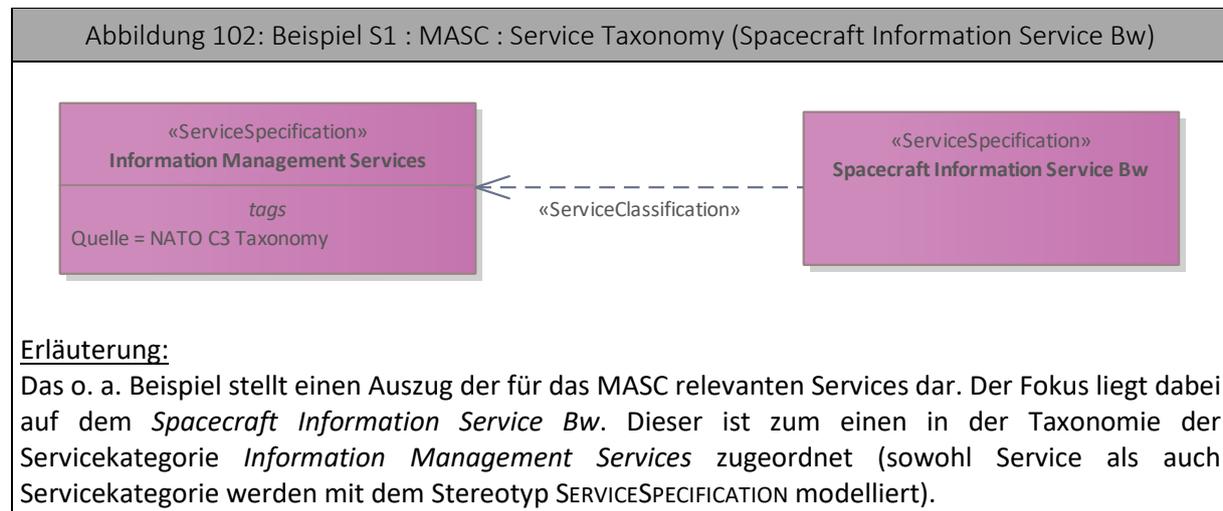
Erläuterung:

Das Beispiel stellt den *Spacecraft Information Service Bw* inkl. Beschreibung dar. Es handelt sich um einen Service der Ebene 1 (am Begriff *Service Bw* am Ende im Namen zu erkennen).

S1 – Service Taxonomy

Der *S1 – Service Taxonomy* ordnet den Service in die C3-Taxonomie ein.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.25.



S1 – Service Varianten (optionaler Viewpoint)

Der *S1 – Service Varianten* erfasst die verschiedenen Varianten eines Service.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.26.

Abbildung 103: Beispiel S1 : MASC : Service Varianten (Spacecraft Information Service Bw)



Erläuterung:

Das o. a. Beispiel stellt einen Auszug der für das MASC relevanten Services dar. Der Fokus liegt dabei auf dem *Spacecraft Information Service Bw*. Der *Spacecraft Information Service Bw* wird in den zwei Varianten *Spacecraft Information Service Bw verlegfähig* & *Spacecraft Information Service Bw stationär* ausgebracht (sowohl Service als auch Servicevariante werden mit dem Stereotyp SERVICE SPECIFICATION modelliert).

C1-S1 – Capability to Service Mapping

Der *C1-S1 – Capability to Service Mapping* dient der Einordnung des Services in die Fähigkeitslage.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.27.

Abbildung 104: Beispiel C1-S1 : MASC : Capability to Service Mapping (Spacecraft Information Service Bw)



Erläuterung:

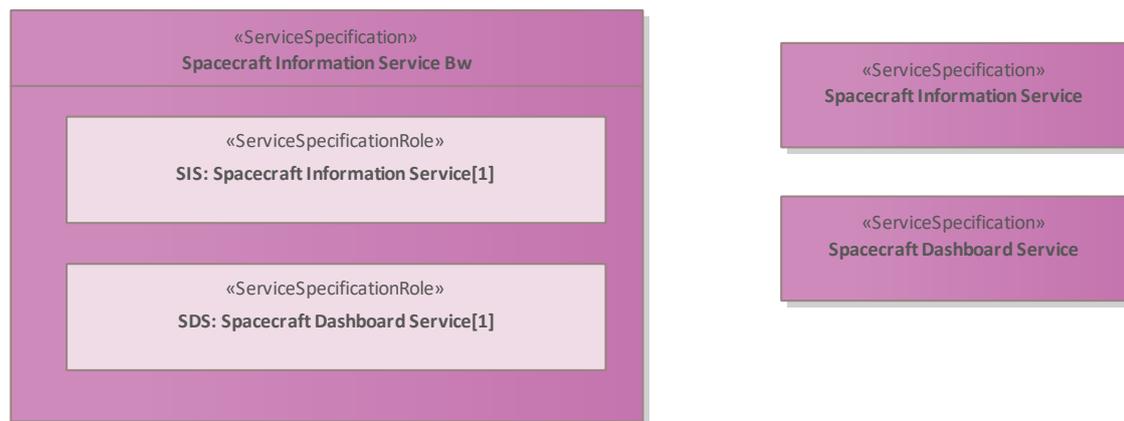
Im o. a. Beispiel leistet der Service *Spacecraft Information Service Bw* einen Beitrag zur Fähigkeit *Weltraumaufklärung*.

S2 – Service Structure

Der *S2 – Service Structure* beschreibt die Zusammensetzung des Service aus dritten Services.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.28.

Abbildung 105: Beispiel S2 : MASC : Service Structure (Spacecraft Information Service Bw)



Erläuterung:

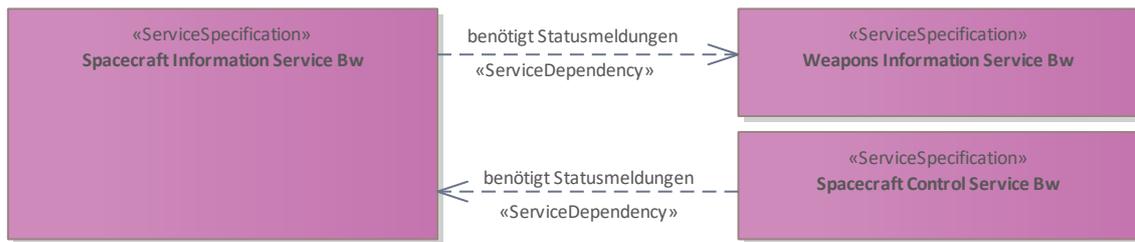
Der im o. a. Beispiel ausgewählte *Spacecraft Information Service Bw* setzt sich aus den beiden Teilservices, dem *Spacecraft Information Service* und dem *Spacecraft Dashboard Service*, zusammen.

S2 – External Service Dependencies

Der *S2 – External Service Dependencies* stellt die Abhängigkeiten des Service dar.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.29.

Abbildung 106: Beispiel S2 : MASC : External Service Dependencies (Spacecraft Information Service Bw)



Erläuterung:

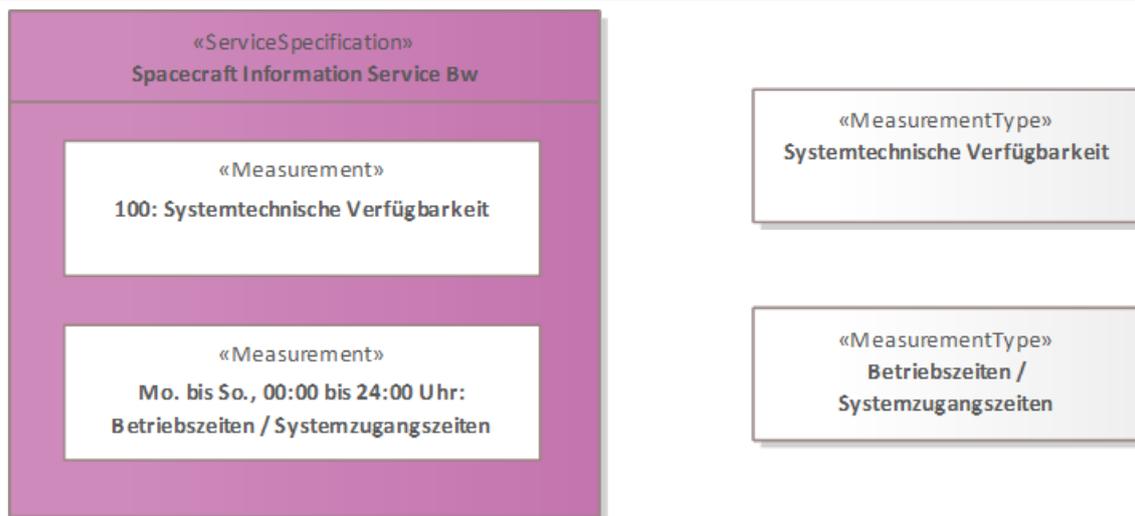
Im o. a. Beispiel sind einige Abhängigkeiten des *Spacecraft Information Service Bw* aufgezeigt. Der Service ist abhängig vom *Weapons Information Service Bw* und vom *Spacecraft Control Service Bw*.

S2 – Service Warranties

Der *S2 – Service Warranties* wird die Warranty des Service erfasst.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.30.

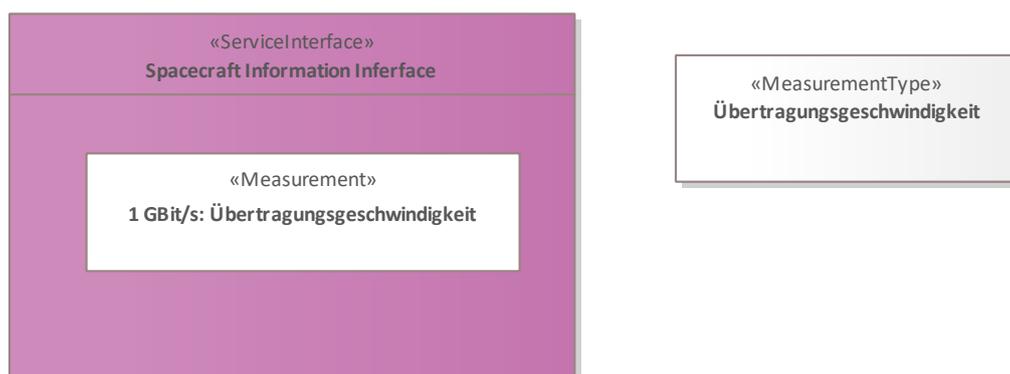
Abbildung 107: Beispiel S2 : MASC : Service Warranties (Spacecraft Information Service Bw)



Erläuterung:

Im o. a. Beispiel sind einige Eigenschaften des *Spacecraft Information Service Bw* aufgezeigt. Dem Service sind die Eigenschaften *Systemtechnische Verfügbarkeit* mit einem Wert von *100* und *Betriebszeiten / Systemzugangseiten* von *Mo. Bis So., 00:00 bis 24:00 Uhr* zugeordnet.

Abbildung 108: Beispiel S2 : MASC : Service Interface (Spacecraft Information Service Bw)



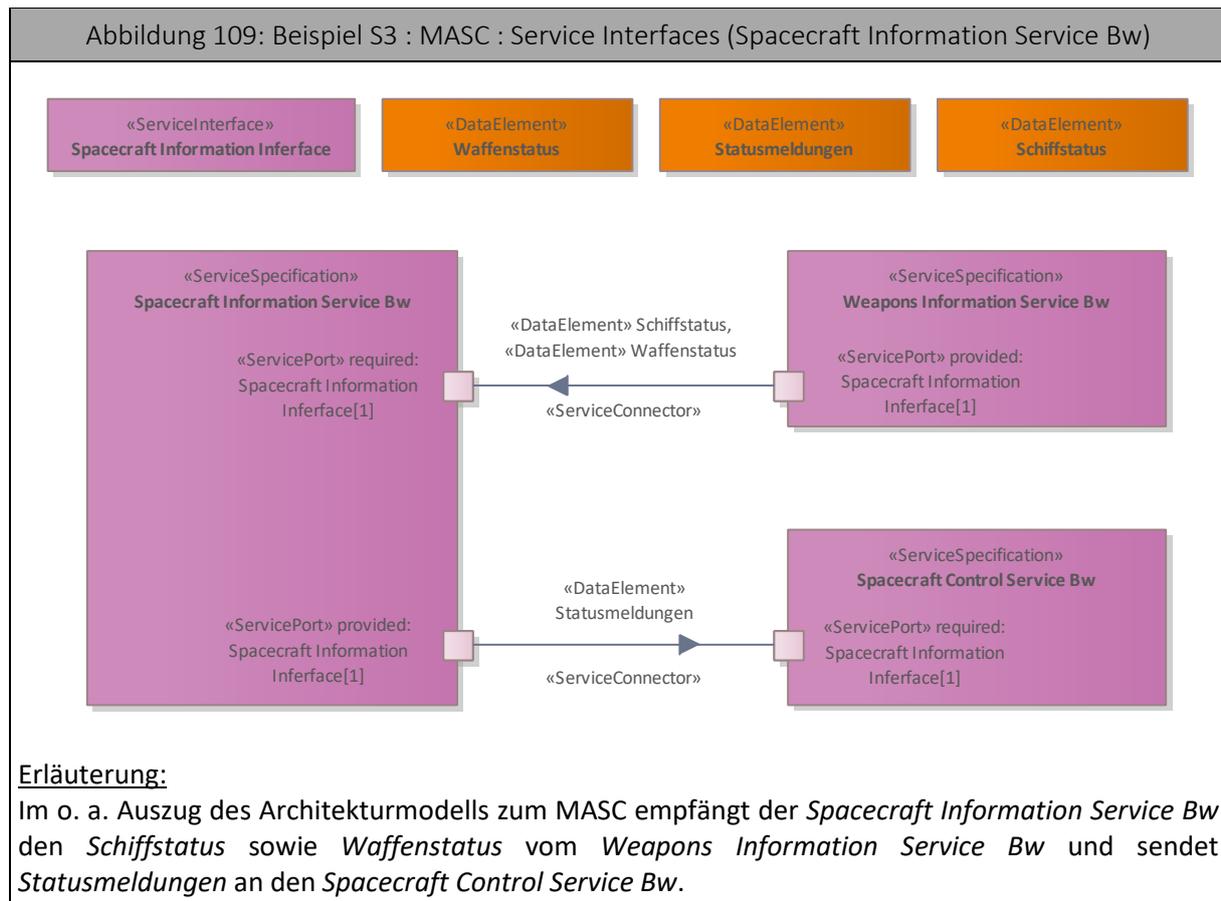
Erläuterung:

Im o. a. Beispiel sind einige Eigenschaften des *Spacecraft Information Interface* aufgezeigt. Dem Interface ist die Eigenschaft *Übertragungsgeschwindigkeit* mit einem Wert von *1 Gbit/s* zugeordnet.

S3 – Service Interfaces

Der *S3 – Service Interfaces* bildet die Schnittstellen des Service und die Austauschbeziehungen zu anderen Services ab.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.31.

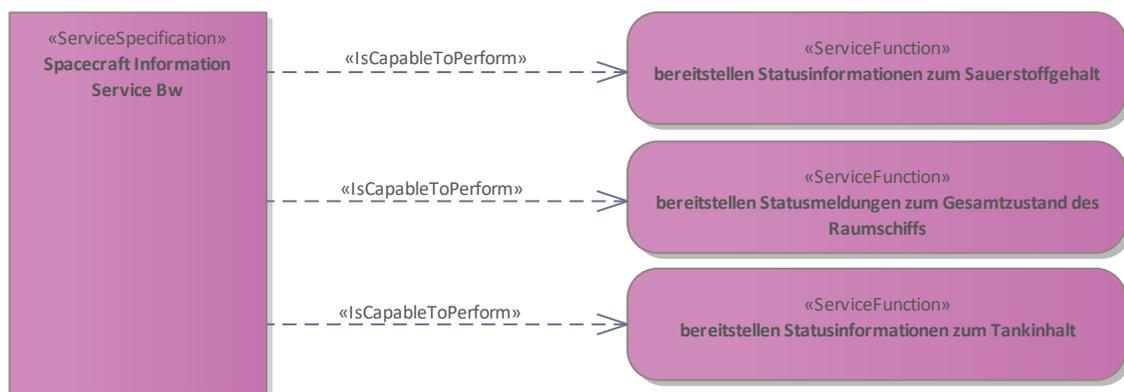


S4 – Service Functions

Der *S4 – Service Functions* dient der Darstellung der Funktionen, die vom Service zur Verfügung gestellt werden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.32.

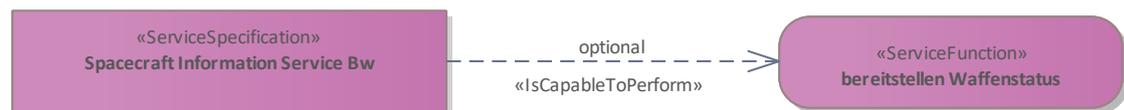
Abbildung 110: Beispiel S4 : MASC : Service Functions (Spacecraft Information Service Bw)



Erläuterung:

Der *Spacecraft Information Service Bw* stellt drei verschiedene Servicefunktionen zur Verfügung. Zu diesen zählen neben dem *Bereitstellen von Statusinformationen zum Sauerstoffgehalt*, das *Bereitstellen von Statusmeldungen zum Gesamtzustand des Raumschiffs* sowie das *Bereitstellen von Statusinformationen zum Tankinhalt*.

Abbildung 111: Beispiel S4 : MASC : Service Functions - optional (Spacecraft Information Service Bw)



Erläuterung:

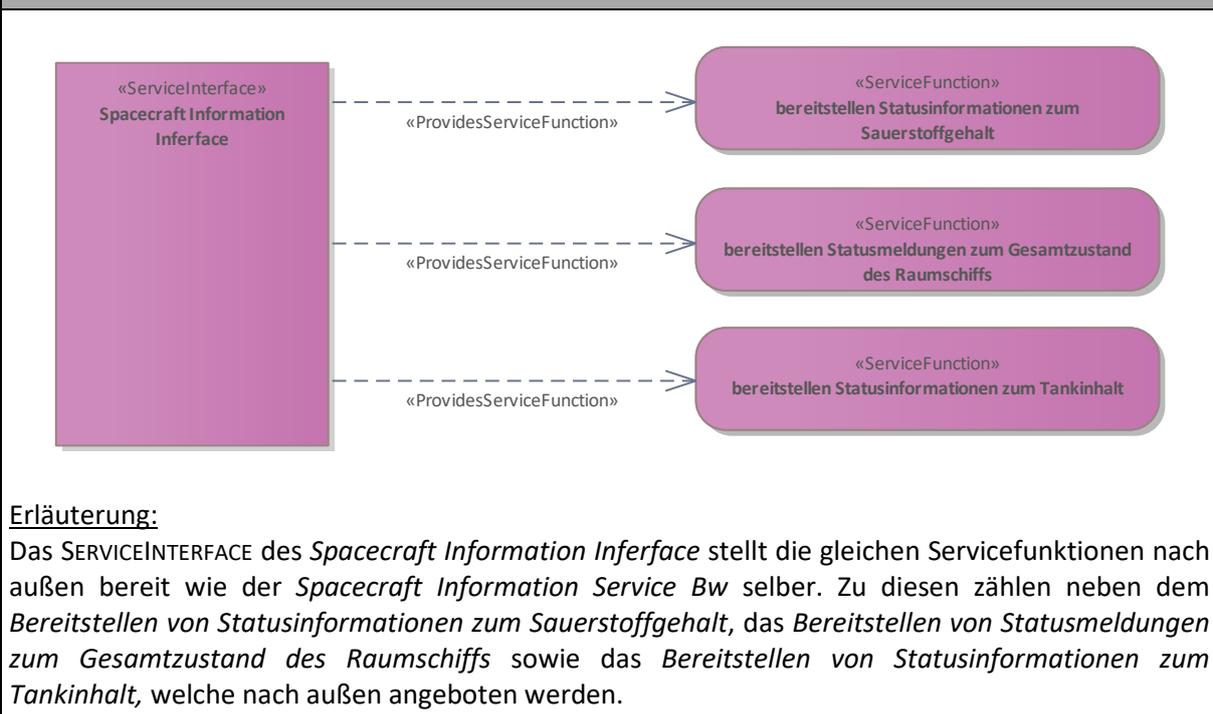
Dem *Spacecraft Information Service Bw* ist die *optionale Servicefunktion bereitstellen Waffenstatus* zugeordnet.

S7 – Service Interface Parameters

Der *S7 – Service Interface Parameters* beschreibt, welche Servicefunktionen eine Serviceschnittstelle nach außen anbietet.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.33.

Abbildung 112: Beispiel S7 : MASC : Service Interface Parameters (Spacecraft Information Service Bw)



Erläuterung:

Das SERVICEINTERFACE des *Spacecraft Information Interface* stellt die gleichen Servicefunktionen nach außen bereit wie der *Spacecraft Information Service Bw* selber. Zu diesen zählen neben dem *Bereitstellen von Statusinformationen zum Sauerstoffgehalt*, das *Bereitstellen von Statusmeldungen zum Gesamtzustand des Raumschiffs* sowie das *Bereitstellen von Statusinformationen zum Tankinhalt*, welche nach außen angeboten werden.

Sr – Service Change

Der *Sr – Service Change* beschreibt, an welchen Services im Rahmen der Servicedokumentation ein Änderungsbedarf besteht.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.34.

Abbildung 113: Beispiel Sr : MASC : Service Change (Spacecraft Information Service Bw)



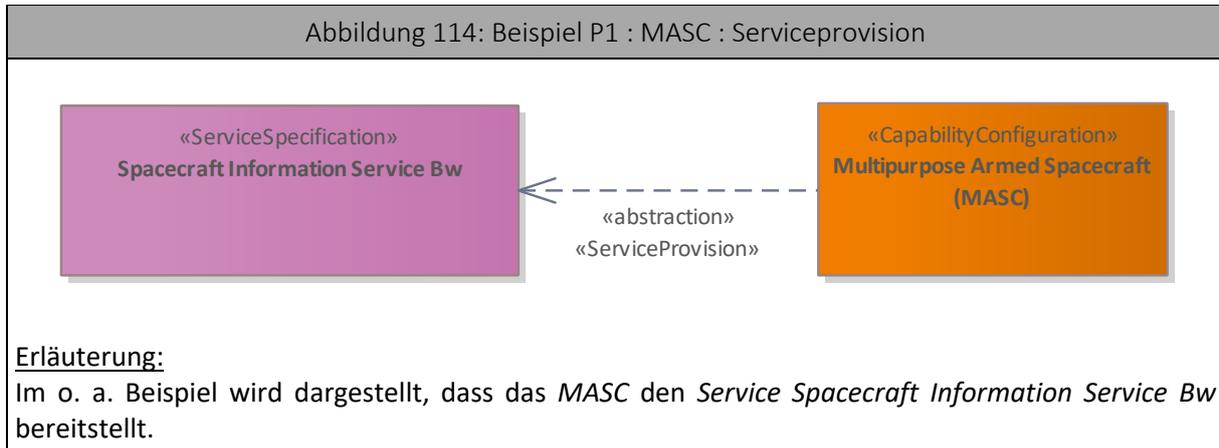
Erläuterung:

Am *Spacecraft Information Service Bw* soll eine Anpassung der Schnittstellen erfolgen.

P1 – Projektbezug

Der *P1 – Projektbezug* erfasst die einem Service zugeordneten Ressourcen.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.35.



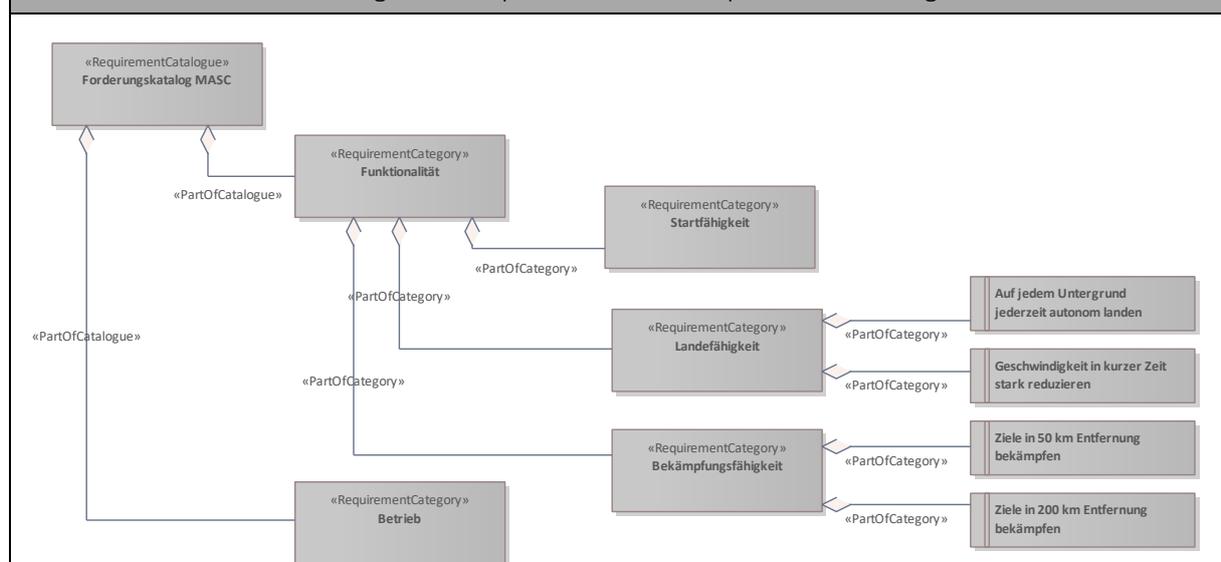
R2 – Requirement Catalogue

Der *R2 – Requirement Catalogue* behandelt die Struktur der Forderungen. Dies geschieht, indem die Forderungen in einem Katalog in Kategorien eingeteilt werden.

Der R2 dient in der Nutzungsphase dazu die die bisherigen und neuen Forderungen in Kategorien einzuordnen und so inhaltlich zusammengehörende Forderungen für unterschiedliche Systemelemente zusammenzufassen. Die Strukturierung ist teilweise durch die Regelung für den priorisierten Forderungskatalog vorgegeben, kann jedoch individuell erweitert werden. Die vorgegebene Struktur kann in Anlage 0 heruntergeladen und in der Architektur verwendet werden. Der R2 entspricht inhaltlich und in seiner Struktur dem Priorisierten Forderungskatalog gemäß A1-1500/3-7000.

Sofern der R2 bereits in der Analysephase Teil 1 oder 2 erstellt wurde, ist dieser in der Nutzungsphase nicht neu anzulegen, sondern fortzuschreiben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.37.

Abbildung 115: Beispiel R2 : MASC: Requirement Catalogue



Erläuterung:

Der o. a. Auszug des Forderungskatalog MASC besteht aus den zwei Überkategorien *Funktionalität* und *Betrieb*. Die Kategorie *Funktionalität* untergliedert sich in die Kategorien *Startfähigkeit*, *Landefähigkeit* und die zur ÄG hinzugefügte Kategorie *Bekämpfungsfähigkeit*. Der Kategorie *Landefähigkeit* sind wie bisher die beiden Forderungen *Auf jedem Untergrund jederzeit autonom landen* und *Geschwindigkeit in kurzer Zeit stark reduzieren* zugeordnet. Der Kategorie *Bekämpfungsfähigkeit* und der bisherigen Forderung, *Ziele in 50 km Entfernung bekämpfen* können, ist die zur ÄG ergänzte Forderung *Ziele in 200 km Entfernung bekämpfen* untergeordnet.

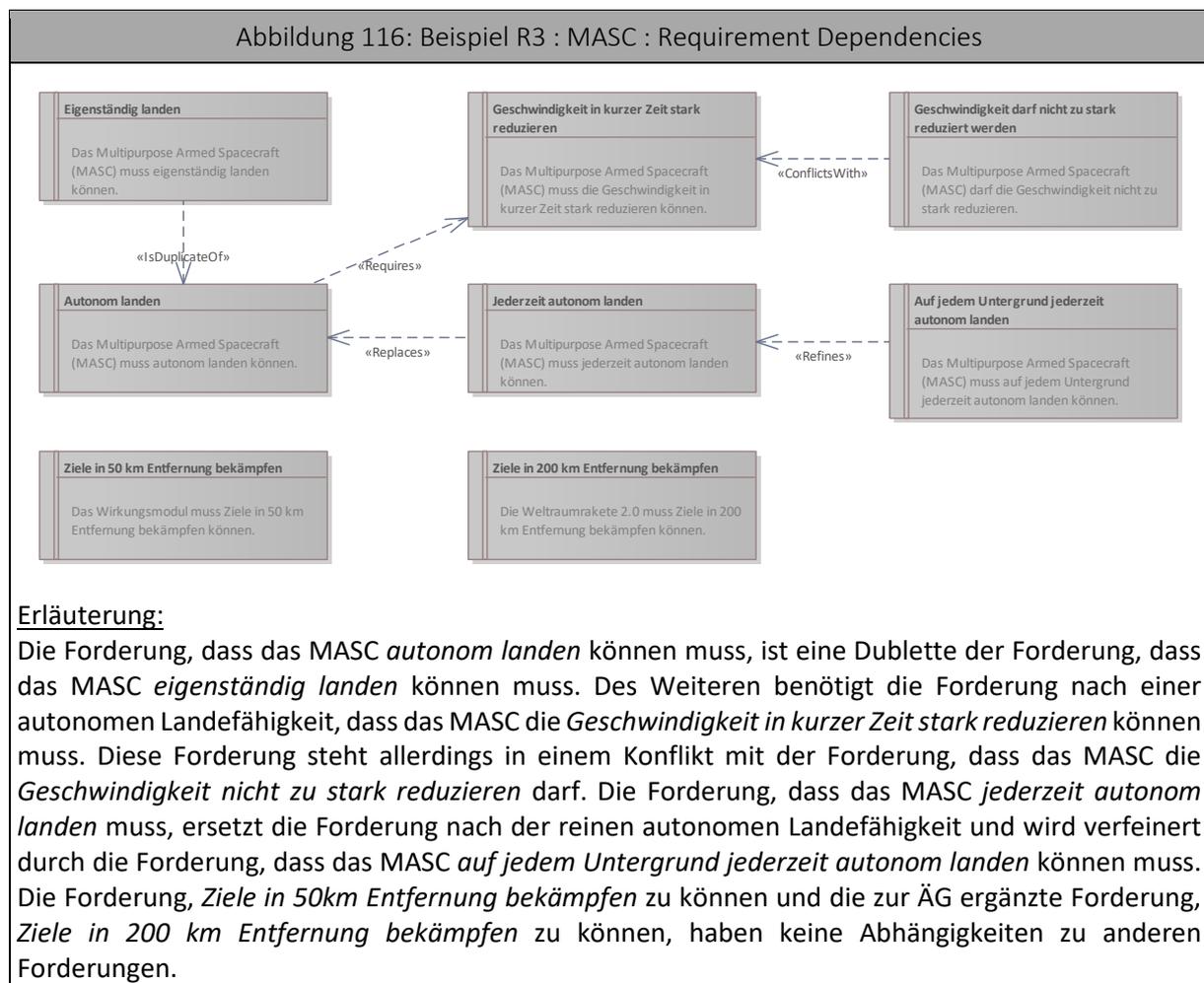
R3 – Requirement Dependencies

Der *R3 – Requirement Dependencies* behandelt die Beziehungen zwischen Forderungen.

Der R3 dient in der Nutzungsphase dazu Abhängigkeiten, Konflikte, Dubletten und Verfeinerungen zwischen alten und neuen Forderungen darzustellen. Er ist damit ein wesentliches Hilfsmittel zur Forderungsanalyse.

Abhängigkeiten zwischen den Forderungen sind vollständig beizubehalten, so dass sich aus den Abhängigkeiten ergebende Konsequenzen nachvollziehbar sind. Konflikte zwischen Forderungen sind zum Ende der Forderungserhebung aufzulösen. Dubletten können, insofern die Forderung für mehrere Systemelemente benötigt werden, erhalten bleiben.

Sofern der R3 bereits in der Analysephase Teil 1 oder 2 erstellt wurde, ist dieser in der Nutzungsphase nicht neu anzulegen, sondern fortzuschreiben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.38.



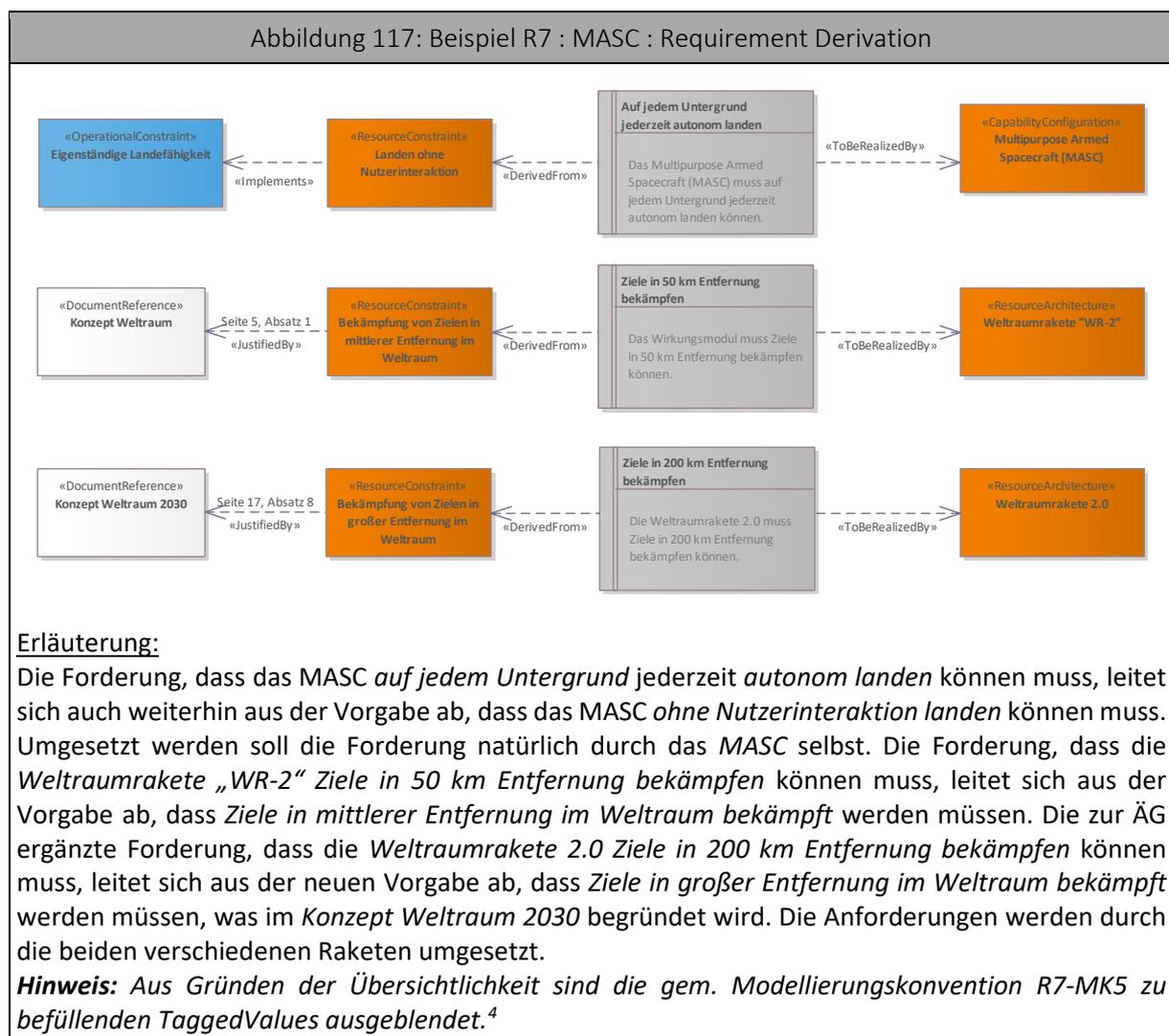
R7 – Requirement Derivation

Der *R7 – Requirement Derivation* behandelt die Ableitung der Forderungen aus operationellen Vorgaben und Rahmenbedingungen und den dazugehörigen Elementen.

Der R7 dient in der Nutzungsphase dazu, die in der Analysephase Teil 2 erhobene Forderungslage zu präzisieren und fortzuschreiben. Des Weiteren wird im R7 in der Nutzungsphase die Ableitung der neuen Forderungen dargestellt.

Sofern der R7 bereits in der Analysephase Teil 2 erstellt wurde, ist dieser in der Analysephase Teil 2 nicht neu anzulegen, sondern fortzuschreiben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.39.

Abbildung 117: Beispiel R7 : MASC : Requirement Derivation



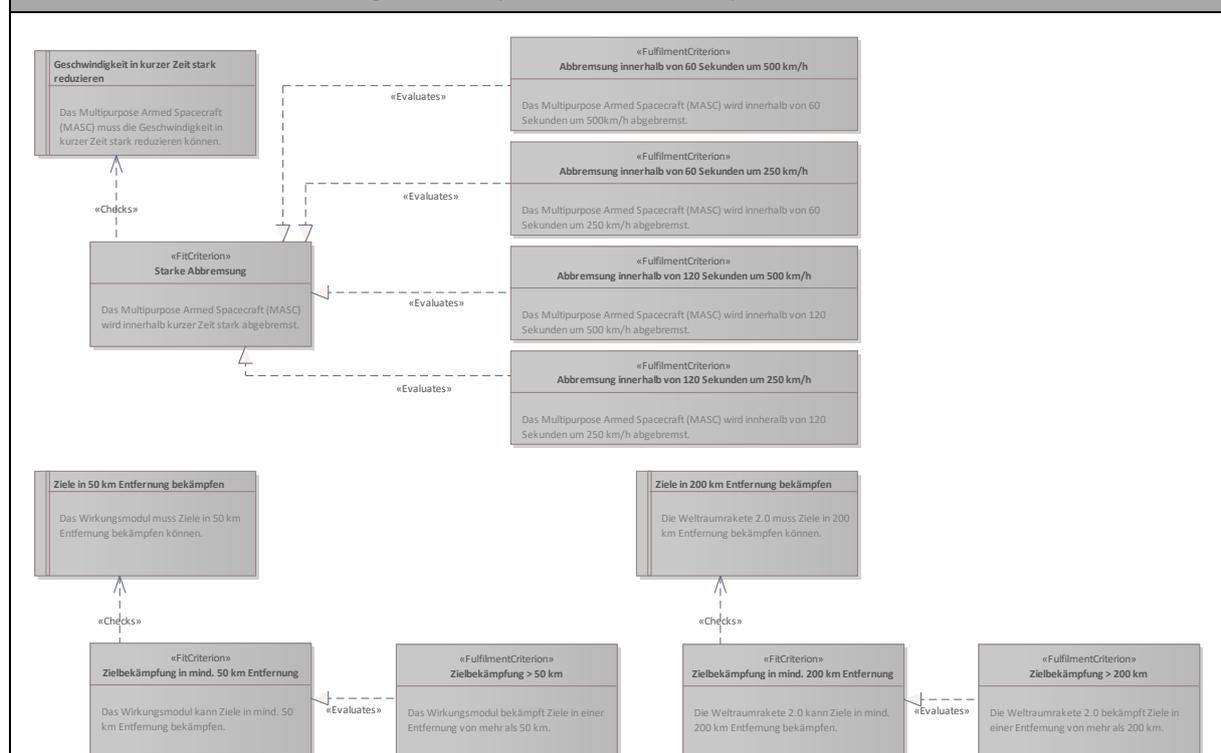
⁴ Ein Beispiel zur Befüllung der TaggedValues finden Sie in der Dokumentation „Verwendung des ADMBw im SPARX Enterprise Architect“ in Kapitel 4.5.3.

R8 – Requirement Fulfilment (optionaler Viewpoint)

Der *R8 – Requirement Fulfilment* behandelt die Abnahme- und Prüfkriterien für die Überprüfung der Forderungen. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.3.40.

Hinweis: Der Viewpoint dient der Verfeinerung der Anforderung. Dabei soll diese in für den Operateur verständliche, messbare Aussagen aufgegliedert werden. Sofern die Anforderung in einer messbaren bzw. überprüfbaren Art beschrieben ist, besteht keine Notwendigkeit für diesen Viewpoint.

Abbildung 118: Beispiel R8 : MASC : Requirement Fulfilment



Erläuterung:

Für die Forderung, dass das MASC die Geschwindigkeit in kurzer Zeit stark reduzieren können muss, gibt es im o. a. Beispiel auch weiterhin ein Bewertungskriterium. Dieses dient der Bewertung, ob das MASC in kurzer Zeit stark abgebremst werden kann. Zur Abnahme des MASC kommen wie bisher vier Abnahmekriterien zum Tragen: Eine Abbremsung innerhalb von 60 Sekunden um 500 km/h, eine Abbremsung innerhalb von 60 Sekunden um 250 km/h, eine Abbremsung in 120 Sekunden um 500 km/h und eine Abbremsung in 120 Sekunden um 250 km/h.

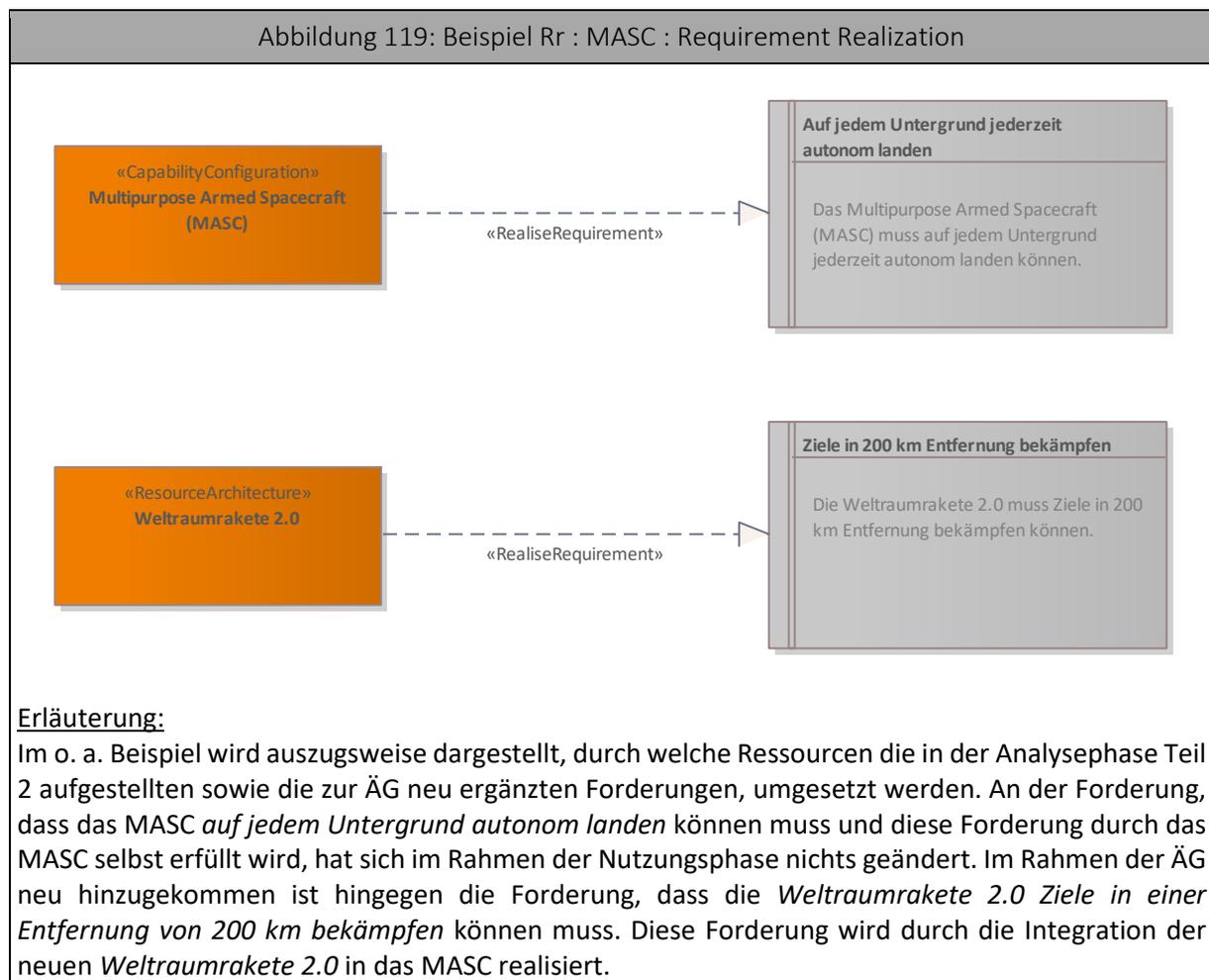
Zusätzlich zur o. a. Forderung zur Abbremsung existiert ebenfalls eine Forderung, dass das *Wirkungsmodul Ziele in 50 km Entfernung bekämpfen* können muss. Diese Forderung soll dadurch bewertet werden, dass das *Wirkungsmodul* tatsächlich *Ziele in min. 50 km Entfernung bekämpfen* kann. Zur Abnahme gibt es daher genau ein Kriterium. Das *Wirkungsmodul* muss *Ziele in einer Entfernung größer 50 km bekämpfen*. Zur ÄG neu hinzugekommen ist die Forderung, dass die *Weltraumrakete 2.0* nun *Ziele in 200 km Entfernung bekämpfen* können muss. Diese Forderung wird durch das Bewertungskriterium geprüft, dass die *Weltraumrakete 2.0* tatsächlich *Ziele in min. 200 km Entfernung bekämpfen* kann. Auch hier gibt es genau ein Abnahmekriterium. Die *Weltraumrakete 2.0* muss *Ziele in einer Entfernung größer 200 km bekämpfen*.

Rr – Requirement Realization

Der *Rr – Requirement Realization* hat das Ziel, die Realisierung von Forderungen darzustellen.

In der Nutzungsphase dient der Rr der Darstellung, welche Forderungen aus der Analysephase Teil 1 und 2 sowie welche neuen Forderungen durch welche konkreten Ressourcen oder Services realisiert werden.

Prinzipiell wird dieser Viewpoint zu diesem Zeitpunkt nicht neu modelliert, sondern lediglich fortgeschrieben. Die dabei zu berücksichtigenden Modellierungskonventionen finden Sie in Kapitel 2.4.24.



Auswertediagramme

Wie schon in den vorherigen Phasen des CPM kann es auch in der Nutzungsphase zielführend sein, Sichten zu definieren, die dazu dienen, einen ganz bestimmten und nur für das Projekt bestimmten Analysebedarf zu decken.

Alle relevanten Informationen zu den Auswertediagrammen finden Sie in Kapitel 2.3.41.

ANLAGEN

Glossar und Abkürzungen

Anlage 3 Template Initialisierung

Das aktuelle Template zur initialen Erfassung von Architekturbeschreibung, Umfang, Ziel, Stakeholder und Analysebedarf.

Anlage 4 Architekturblock Forderungskatalog

Der Forderungskatalog kann über die Import-Funktion des SPARX EA in die Architektur importiert werden. Hierbei ist zwingend die Option STRIP GUIDS zu aktivieren.

Bei Fragen wenden Sie sich an folgendes Gruppenpostfach:

ZDigBwII5MethZukE@bundeswehr.org

ÄNDERUNGSPROZESS

Sollten Sie Anregungen und Verbesserungsvorschläge zu Inhalt und Gestaltung dieses Leitfadens haben, würden wir uns freuen, wenn Sie uns Ihre Vorschläge zusenden:

BAAINBwStabDigitalisierungD1.1@bundeswehr.org