

DIN EN 4660-005

ICS 49.090

Ersatz für
DIN EN 4660-005:2011-09

**Luft- und Raumfahrt –
Modulare und offene Avionikarchitekturen –
Teil 005: Software;
Deutsche und Englische Fassung EN 4660-005:2019, nur auf CD-ROM**

Aerospace series –
Modular and Open Avionics Architectures –
Part 005: Software;
German and English version EN 4660-005:2019, only on CD-ROM

Série aérospatiale –
Architectures Avioniques Modulaires et Ouvertes –
Partie 005: Software;
Version allemande et anglaise EN 4660-005:2019, seulement en CD-ROM

Gesamtumfang 645 Seiten

DIN-Normenausschuss Luft- und Raumfahrt (NL)

Nationales Vorwort

Der Verband der Europäischen Luft-, Raumfahrt- und Verteidigungsindustrie — Normung (ASD-STAN) ist vom Europäischen Komitee für Normung (CEN) für zuständig erklärt worden, Europäische Normen (EN) für das Gebiet der Luft- und Raumfahrt auszuarbeiten. Durch die Vereinbarung vom 3. Oktober 1986 wurde ASD Assoziierte Organisation (ASB) des CEN.

Das vorliegende Dokument (EN 4660-005:2019) wurde von ASD-STAN, Fachbereich Elektrotechnik, unter Mitwirkung deutscher Experten des DIN-Normenausschusses Luft- und Raumfahrt (NL) erarbeitet.

Das zuständige deutsche Normungsgremium ist der Arbeitsausschuss NA 131-04-09 AA „Avionik“ im DIN-Normenausschuss Luft- und Raumfahrt (NL).

Entsprechend Beschluss 57/9 des Technischen Ausschusses des Beirats des DIN-Normenausschusses Luft- und Raumfahrt (NL) sind die europäischen Luft- und Raumfahrt-Normungsergebnisse zweisprachig, in Deutsch und Englisch, in das Deutsche Normenwerk zu überführen. Aus diesem Grund wurde der Deutschen Fassung dieses Dokumentes die Englische Fassung hinzugefügt.

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Änderungen

Gegenüber DIN EN 4660-005:2011-09 wurden folgende Änderungen vorgenommen:

- a) die Abschnitte 0 und 1 wurden überarbeitet;
- b) die normativen Verweisungen wurden an die Bearbeitung angepasst;
- c) in Abschnitt 3 wurden Abkürzungen ergänzt;
- d) im normativen Text wurden folgende Änderungen vorgenommen: Aufnahme der in Abhängigkeit von der Datenlänge dynamischen Header-Länge in alle Nachrichten, der NII-Implementierung wurde der Dienst „Select“ („Auswählen“) hinzugefügt, die MLI-Zeitsynchronisation wurde ersetzt, die Konfektionierungstabelle aktualisiert, Anwendung von robuster Partitionierung APEX als API für den Softwarestandard, Integration eines genormten Dateiübertragungsprotokolls, Streichung der Globalzeitdienste und -begriffe sowie der GLI-Definition, Definition von OpenGL ES als Graphik-API, Überarbeitung der Beschreibung von *configureInterface*, Streichung des CFM-Ressourcendienst *getPeInfo* und Ersetzen der RTBP-Definition durch XML-Schema;
- e) die Norm wurde redaktionell überarbeitet.

Frühere Ausgaben

DIN EN 4660-005: 2011-09

Deutsche Fassung

Luft- und Raumfahrt —
Modulare und offene Avionikarchitekturen —
Teil 005: Software

Aerospace series —
Modular and Open Avionics Architectures —
Part 005: Software

Série aérospatiale —
Architectures Avioniques Modulaires et Ouvertes —
Partie 005: Software

Diese Europäische Norm wurde vom CEN am 2. Dezember 2018 angenommen.

Die CEN-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist. Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim CEN-CENELEC-Management-Zentrum oder bei jedem CEN-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CEN-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Management-Zentrum mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CEN-Mitglieder sind die nationalen Normungsinstitute von Belgien, Bulgarien, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, den Niederlanden, Norwegen, Österreich, Polen, Portugal, der Republik Nordmazedonien, Rumänien, Schweden, der Schweiz, Serbien, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, der Türkei, Ungarn, dem Vereinigten Königreich und Zypern.



EUROPÄISCHES KOMITEE FÜR NORMUNG
EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION

CEN-CENELEC Management-Zentrum: Rue de la Science 23, B-1040 Brüssel

Inhalt

	Seite
Europäisches Vorwort	5
Einleitung	6
1 Anwendungsbereich	7
1.1 Allgemeiner Anwendungsbereich	7
1.2 Überblick über die Softwarearchitektur	7
1.3 Komponenten der Softwarearchitektur	7
1.3.1 Allgemeines	7
1.3.2 Funktionsanwendungen	8
1.3.3 Anwendungsmanagement (AM)	8
1.3.4 Betriebssystem (OS)	8
1.3.5 Generische Systemverwaltung (GSM)	8
1.3.6 Laufzeit-Blueprints (RTBP)	9
1.3.7 Modulunterstützungsschicht (MSL)	9
1.3.8 Schnittstelle Anwendung/Betriebssystem (APOS)	9
1.3.9 Schnittstelle Modulunterstützung/Betriebssystem (MOS)	9
1.3.10 Schnittstelle Systemverwaltung/Blueprints (SMBP)	9
1.3.11 Schnittstelle Systemverwaltung/Betriebssystem (SMOS)	9
1.3.12 Logische OS-Schnittstelle (OLI)	9
1.3.13 Logische GSM-Schnittstelle (GLI)	9
1.3.14 Logische Systemverwaltungsschnittstelle (SMLI)	9
1.3.15 Logische Modulschnittstelle (MLI)	10
2 Normative Verweisungen	10
3 Begriffe und Abkürzungen	11
3.1 Begriffe	11
3.2 Abkürzungen	11
4 Systemfunktionen	14
4.1 Systemverwaltungsfunktion	14
4.1.1 Allgemeines	14
4.1.2 GSM-Funktion	15
4.1.3 AM-Funktion	18
4.1.4 Fehlerbehandlung	19
4.1.5 Selbsttest	19
4.2 Kommunikation	21
4.2.1 MOAA-Kommunikationsmodell	21
4.2.2 Arten des Datentransfers	24
4.2.3 Kommunikationskonfiguration	24
4.2.4 Kommunikationsprotokolle	25
4.2.5 Multicast	28
4.2.6 Verteiltes Multicast	30
4.2.7 Streaming	34
4.2.8 Datendarstellung	34
4.3 Sicherheitsmanagement	40
4.3.1 Anwendungssicherheitsmanagement	40
4.3.2 Generisches Sicherheitsmanagement	41
4.3.3 Verschlüsselung/Entschlüsselung und Authentifizierung	42
4.3.4 Sicherheitsaudit	43

4.3.5	Sicherheitsreferenzüberwachung.....	43
4.4	Modulverwaltung.....	43
4.5	Massenspeicherverwaltung.....	44
4.5.1	Überblick.....	44
4.5.2	Lokale Dateiverwaltung der MMM.....	44
4.5.3	Zugriff auf Anwendungsdateien.....	45
4.5.4	CFM-Download.....	45
4.5.5	Anwendungsdownload.....	46
4.6	Graphikverwaltung.....	47
4.7	Energieverwaltung.....	47
4.7.1	GSM-gesteuerte Lösung.....	48
4.7.2	MLI-gesteuerte Lösung.....	49
4.8	Netzwerkverwaltung.....	49
4.8.1	Netzwerkdefinition.....	49
4.8.2	Netzwerkkonfiguration.....	50
4.8.3	Überwachung des Netzwerkzustands.....	50
4.8.4	Transparenz der Netzwerktechnologie.....	50
4.9	Zeitmanagement.....	51
4.9.1	Zeitreferenz.....	51
4.9.2	Uhrenhierarchie.....	52
4.9.3	Uhrenkonfiguration.....	54
4.9.4	Uhrenverwaltung.....	54
5	Definition der Softwarearchitektur.....	55
5.1	MSL.....	56
5.1.1	MSL-Modulverwaltung.....	56
5.1.2	MSL-Kommunikationsfähigkeit.....	57
5.1.3	Residente Software.....	60
5.2	OSL.....	61
5.2.1	GSM.....	61
5.2.2	OS-Funktionen.....	68
5.3	RTBP.....	85
5.3.1	Überblick.....	85
5.3.2	RTBP-Baum.....	85
5.3.3	SMBP-Dienste für den Zugriff auf die RTBP-Tabellen.....	86
5.4	Anwendungsschicht.....	87
5.4.1	Überlegungen zur Sprache.....	88
5.4.2	Behandlung von Anwendungsfehlern.....	88
6	Definitionen direkter Schnittstellen.....	89
6.1	APOS.....	89
6.2	MOS.....	92
6.2.1	Generische MOS.....	93
6.2.2	Spezifische Dienste.....	136
6.2.3	Kundenspezifische MOS-Erweiterungsdienste.....	152
6.3	SMBP.....	169
6.3.1	RTBP-Baumgrammatik.....	170
6.3.2	Dienste für Abruftabellen.....	176
6.4	SMOS.....	186
6.4.1	Prozess- und Thread-Verwaltungsdienste.....	188
6.4.2	Fehlermanagementdienste.....	189
6.4.3	VC-Konfigurationsdienste.....	192
6.4.4	Netzwerkkonfigurationsdienste.....	198
6.4.5	Sicherheitsverwaltungsdienste.....	202
6.4.6	Dienste für das Management der internen Prüfungen.....	207
6.4.7	CFM-Informationsdienste.....	211

6.4.8	Dienste der CFM-Ressourcenverwaltung	214
6.4.9	Zeitkonfigurationsdienste.....	218
6.4.10	Protokollierungsverwaltungsdienste	218
7	Definitionen logischer Schnittstellen.....	222
7.1	OLI	222
7.1.1	VC-Header.....	222
7.1.2	OLI-Dienste	222
7.2	GLI	222
7.2.1	GLI-Darstellung	222
7.2.2	GLI-Dienste.....	223
7.3	SMLI	230
7.3.1	SMLI-Darstellung	230
7.3.2	SMLI-Dienste	230
7.4	MLI.....	238
7.4.1	TC-Header.....	238
7.4.2	MLI-Dienste.....	238
7.4.3	Protokoll	259
8	Datentypdefinitionen.....	264
8.1	IDL	264
8.1.1	Allgemeines	264
8.1.2	Grundtypen	264
8.1.3	Namensräume	265
8.1.4	Einschränkungen.....	265
8.2	Datentypen.....	266
9	Konfektionierung	292
Anhang A (normativ)	RTBP-XML-Schema	298
A.1	MOAA-Typen.....	298
A.2	Erweiterungen des Typs MOAA	304
A.3	MOAA-Laufzeit-Blueprints	307
Anhang B (informativ)	Übersicht der Änderungen zur Vorgängerversion.....	321
Literaturhinweise.....		322

Europäisches Vorwort

Dieses Dokument (EN 4660-005:2019) wurde vom Verband der Europäischen Luft-, Raumfahrt- und Verteidigungsindustrie – Normung (ASD-STAN) erstellt.

Nachdem Umfragen und Abstimmungen entsprechend den Regeln dieses Verbandes durchgeführt wurden, hat diese Norm die Zustimmung der nationalen Verbände und offiziellen Behörden der Mitgliedsländer der ASD erhalten, bevor sie CEN vorgelegt wurde.

Diese Europäische Norm muss den Status einer nationalen Norm erhalten, entweder durch Veröffentlichung eines identischen Textes oder durch Anerkennung bis Februar 2020, und etwaige entgegenstehende nationale Normen müssen bis Februar 2020 zurückgezogen werden.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. CEN ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Dieses Dokument ersetzt EN 4660-005:2011.

Entsprechend der CEN-CENELEC-Geschäftsordnung sind die nationalen Normungsinstitute der folgenden Länder gehalten, diese Europäische Norm zu übernehmen: Belgien, Bulgarien, Dänemark, Deutschland, die Republik Nordmazedonien, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, Niederlande, Norwegen, Österreich, Polen, Portugal, Rumänien, Schweden, Schweiz, Serbien, Slowakei, Slowenien, Spanien, Tschechische Republik, Türkei, Ungarn, Vereinigtes Königreich und Zypern.

Einleitung

Der Zweck dieser MOAA-Norm ist, eine Reihe von Standards, Konzepten und Leitlinien für offene A3-Architekturen (en: Advanced Avionic Architectures, A3) festzulegen.

Die drei Hauptzielsetzungen der MOAA-Normen sind

- die Senkung der Kosten während des Lebenszyklus,
- die Verbesserung der Leistung bei der Erfüllung der Mission und
- die Verbesserung der Leistung während des Betriebs.

Die MOAA-Normen sind als eine Reihe von Dokumenten organisiert, die Folgendes umfassen:

- eine Reihe von vereinbarten Standards, die mit Hilfe eines Top-down-Ansatzes den Überblick über die Architektur aller Schnittstellen beschreiben, die zur Implementierung des Kerns von Avioniksystemen erforderlich sind;
- die Leitlinien für die Systemimplementierung durch Anwendung der Standards.

Die Dokumentenhierarchie ist im Folgenden dargestellt (das vorliegende Dokument ist in Bild 1 hervorgehoben):

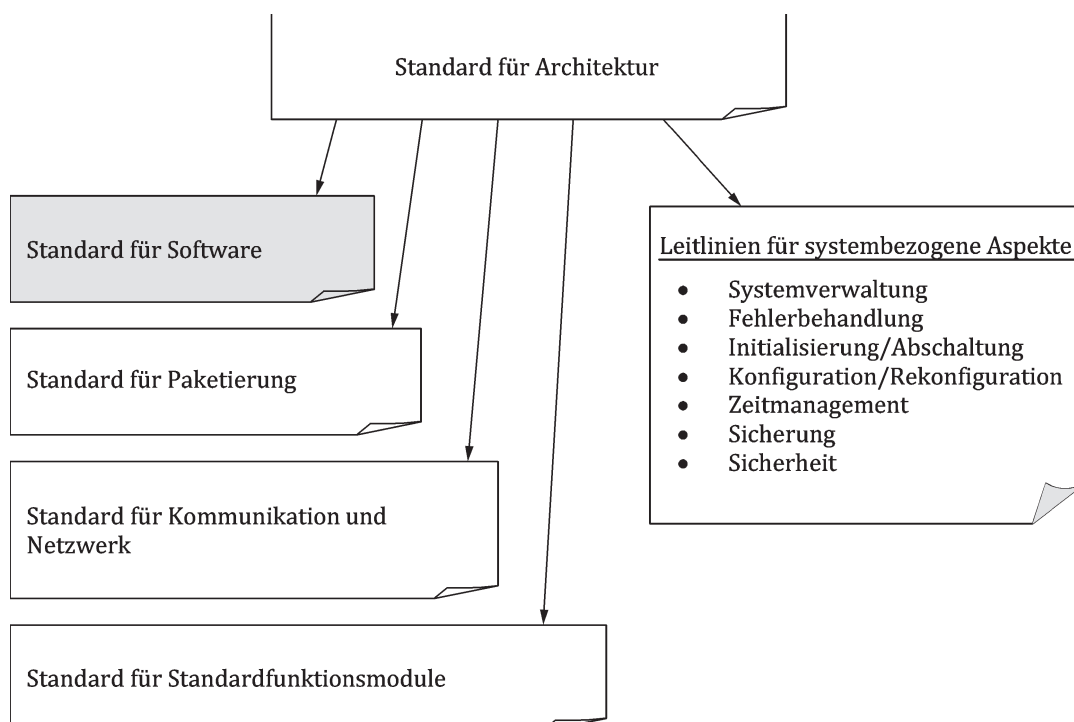


Bild 1 — Hierarchie der Dokumentation der MOAA-Normen

1 Anwendungsbereich

1.1 Allgemeiner Anwendungsbereich

Diese Europäische Norm legt einheitliche Anforderungen an den Entwurf und die Entwicklung von Softwarearchitekturen für modulare Avioniksysteme fest.

1.2 Überblick über die Softwarearchitektur

Die MOAA-Softwarearchitektur basiert auf einem Dreischichtenstapel, wie er in Bild 2 vereinfacht dargestellt ist.

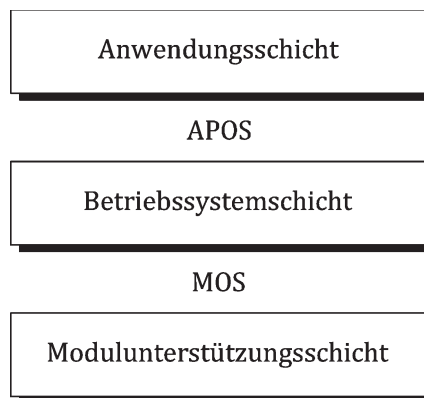


Bild 2 — MOAA-Softwarearchitektur in drei Schichten

Jede Schicht wird in Bezug auf ihre Abhängigkeit/Unabhängigkeit von sowohl dem Luftfahrzeugsystem als auch von der zugrundeliegenden Hardware beschrieben.

Tabelle 1 — Unabhängigkeit der Softwareschichten

Softwareschicht	Abhängigkeit vom Luftfahrzeug	Abhängigkeit von der Hardware
Anwendungsschicht (AL)	Abhängig	Unabhängig
Betriebssystemschicht (OSL)	Unabhängig	Unabhängig
Modulunterstützungsschicht (MSL)	Unabhängig	Abhängig

1.3 Komponenten der Softwarearchitektur

1.3.1 Allgemeines

Bild 3 zeigt eine Übersicht über die Komponenten der Softwarearchitektur und die Softwareschnittstellen.

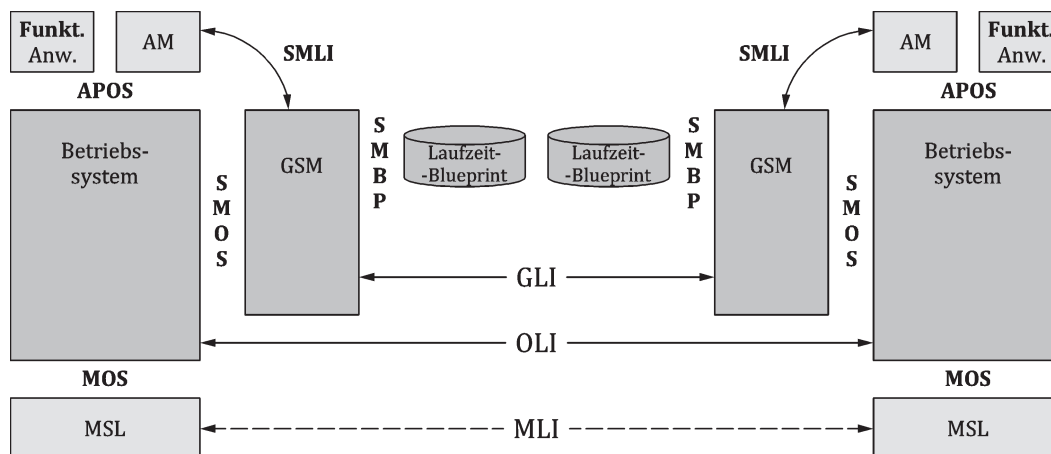


Bild 3 — Das Softwarearchitekturmodell

1.3.2 Funktionsanwendungen

Die Benennung „Funktionsanwendungen“ bezieht sich auf alle Funktionen, die die Verarbeitung von Betriebsdaten handhaben, z. B.:

- Radaranwendungen;
- Missionsmanagement;
- Lagerverwaltung;
- Fahrzeugverwaltungssystem;
- Kommunikation, Navigation und Identifizierung.

1.3.3 Anwendungsmanagement (AM)

Das Anwendungsmanagement ist für die nicht genormte Systemverwaltung zuständig, d. h. es führt die nicht generische Systemverwaltung aus. So kann das AM z. B. die Missions-/Betriebswahlverwaltung ausführen. Die Schnittstelle zwischen AM und GSM ist die logische Systemverwaltungsschnittstelle (SMLI) (siehe 4.1.3).

1.3.4 Betriebssystem (OS)

Ein Echtzeit-Betriebssystem stellt den Teil der OSL-Funktionalität bereit, der das Echtzeitverhalten des Verarbeitungselements und dessen zugeordnete Ressourcen steuert (siehe 5.2.2).

1.3.5 Generische Systemverwaltung (GSM)

Die GSM ist für die Verwaltung des Core-Processing verantwortlich (siehe 4.1.2 und 5.2.1). Diese Funktionalität ist in vier Bereiche unterteilt:

- Zustandsüberwachung;
- Fehlermanagement;
- Konfigurationsverwaltung;
- Sicherheitsmanagement.

1.3.6 Laufzeit-Blueprints (RTBP)

Die RTBP enthalten die Informationen (z. B. Prozessbeschreibung, Routing-Informationen, Fehlermanagementdaten), die für die Konfiguration und Verwaltung des Core Processing, auf dem sie gehostet werden, erforderlich sind (siehe 5.3).

1.3.7 Modulunterstützungsschicht (MSL)

Die MSL fasst die Einzelheiten der zugrundeliegenden Hardware zusammen und ermöglicht einen generischen, technologieunabhängigen Zugriff auf Low-Level-Ressourcen (siehe 5.1).

1.3.8 Schnittstelle Anwendung/Betriebssystem (APOS)

Die APOS ist eine direkte Schnittstelle, die die luftfahrzeugabhängige Software (AL) von der luftfahrzeugunabhängigen Software (OSL) trennt. Ihr Zweck ist es, die Prozesse in der AL mit einer genormten OS-unabhängigen Schnittstelle zu den vom OS bereitgestellten Diensten bereitzustellen und so die Portabilität und Wiederverwendung von Anwendungssoftware zu fördern (siehe 6.1).

1.3.9 Schnittstelle Modulunterstützung/Betriebssystem (MOS)

Die MOS ist eine direkte Schnittstelle, die die OSL von der hardwareabhängigen Software (MSL) trennt. Ihr Zweck ist, das mit OS einer hardwareunabhängigen/technologisch transparenten Schnittstelle zu der in der MSL enthaltenen Funktionalität zu versehen. Daher lässt die MOS zu, dass dieselbe OSL-Software auf unterschiedlichen Implementierungen eines bestimmten CFM vorhanden ist, und zwar unabhängig von der zugrundeliegenden Hardware (siehe 6.2).

1.3.10 Schnittstelle Systemverwaltung/Blueprints (SMBP)

Diese direkte Schnittstelle, die in der OSL zwischen GSM und den Blueprints gekapselt ist, ermöglicht, dass Struktur und Implementierung der Blueprints ungenormt bleiben, während sie eine genormte Schnittstelle zu ihnen definiert (siehe 6.2.3).

1.3.11 Schnittstelle Systemverwaltung/Betriebssystem (SMOS)

Diese in der OSL gekapselte direkte Schnittstelle beschreibt die Dienste, die der GSM vom OS zur Verfügung gestellt werden (siehe 6.3).

1.3.12 Logische OS-Schnittstelle (OLI)

Die OLI beschreibt die wechselseitige Kommunikation zwischen zwei OS-Instanzierungen im Hinblick auf die Kommunikation des virtuellen Kanals (VC) und die Datendarstellung (siehe 7.1).

1.3.13 Logische GSM-Schnittstelle (GLI)

Die GLI beschreibt die wechselseitige Kommunikation zwischen GSM-Instanzen auf getrennten RE (siehe 7.2).

1.3.14 Logische Systemverwaltungsschnittstelle (SMLI)

Die SMLI normt ein VC-basiertes Kommunikationsprotokoll zwischen dem AM und der GSM. AM und GSM müssen zusammenarbeiten; dazu kommunizieren sie und synchronisieren sich über das SMLI (siehe 7.3).

1.3.15 Logische Modulschnittstelle (MLI)

Diese logische Schnittstelle (Kommunikationsprotokoll) definiert die logischen Interaktionen zwischen Modulen, sodass die Anforderungen an die Modulinteroperabilität und die Systembaubarkeit erfüllt werden (siehe 7.4).

2 Normative Verweisungen

Die folgenden Dokumente werden im Text in solcher Weise in Bezug genommen, dass einige Teile davon oder ihr gesamter Inhalt Anforderungen des vorliegenden Dokuments darstellen. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

IEEE 1588:2008, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*¹

IEEE 754:1985, *Binary Floating-Point Arithmetic*

IEEE 802.3, *IEEE Standard for Ethernet*

ISO/IEC 14977:1996, *Information technology — Syntactic metalanguage — Extended BNF*²

ASAAC2-GUI-32450-001-CPG Issue 01, *Final Draft of Guidelines for System Issues*³

— *Volume 1 — System Management*

— *Volume 2 — Fault Management*

— *Volume 3 — Initialisation and Shutdown*

— *Volume 4 — Configuration/Reconfiguration*

— *Volume 5 — Time Management*

— *Volume 6 — Security*

— *Volume 7 — Safety*

ARINC 653P1, *Avionics Application Software Standard Interface, Part 1, Required Services*, (Version 3, 11-2010)⁴

ARINC 653P2, *Avionics Application Software Standard Interface, Part 2, Extended Services*, (Version 2, 06-2012)⁴

OpenGL® ES, The Khronos Group Inc.⁵

1 Veröffentlicht von: IEEE (Institute of Electrical and Electronics Engineers), <http://standards.ieee.org>

2 Veröffentlicht von: International Organization for Standardization (ISO), www.iso.org

3 Zum Zeitpunkt der Veröffentlichung dieser Norm in Vorbereitung.

4 Veröffentlicht von: ARINC, www.aviation-ia.com/product-categories

5 Veröffentlicht von: The Khronos group, www.khronos.org

RFC 1350:1992, *The TFTP Protocol (Revision 2)*⁶

RFC 2347:1998, *TFTP Option Extension*⁶

RFC 2348:1998, *TFTP Block Size Option*⁶

RFC 2349:1998, *TFTP Timeout Interval and Transfer Size Options*⁶

RFC 951:1985, *Bootstrap Protocol (BOOTP)*⁶

RFC 1542:1993, *Clarification and Extensions for the Bootstrap Protocol*⁶

RFC 2132:1997, *DHCP options and BOOTP Vendor Extensions*⁶

3 Begriffe und Abkürzungen

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe und Abkürzungen.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- ISO Online Browsing Platform: verfügbar unter <http://www.iso.org/obp>
- IEC Electropedia: verfügbar unter <http://www.electropedia.org/>

3.1 Begriffe

In der vorliegenden Norm werden die Verben „muss/müssen“, „sollte/sollten“ und „darf/dürfen“ wie folgt verwendet:

- MUSS/MÜSSEN drückt eine verbindliche Anforderung dieser Norm aus;
- SOLLTE/SOLLTEN drückt eine Empfehlung oder einen Ratschlag zur Implementierung einer verbindlichen Anforderung der Norm aus. Es wird erwartet, dass diese Empfehlung bzw. dieser Ratschlag befolgt wird, sofern nicht gute Gründe dafür angegeben sind, dies nicht zu tun;
- DARF/DÜRFEN drückt eine zulässige Praktik oder Handlung aus. Dabei handelt es sich nicht um eine verbindliche Anforderung der Norm.

3.2 Abkürzungen

AC	Aircraft	Luftfahrzeug
AGT	Absolute Global Time	absolute globale Zeit
AL	Application Layer	Anwendungsschicht
ALT	Absolute Local Time	absolute lokale Zeit
AM	Application manager	Anwendungsmanager
APOS	Application to OS [interface]	[Schnittstelle] Anwendung/Betriebssystem
ASAAC	Allied Standard Avionics Architecture Council	—

⁶ Veröffentlicht von: RFC Editor, www.rfc-editor.org

ATM	Asynchronous Transfer Mode	asynchrones Übermittlungsverfahren
BIT	Built-In Test	Selbsttest
BMC	Best Master Clock	—
BMC	Between Module Communication	Kommunikation zwischen Modulen
CBIT	Continuous BIT	kontinuierlicher BIT
CDR	Common Data Representation	Standarddatendarstellung
CFM	Common Functional Module	Standardfunktionsmodul
CM	Configuration Management	Konfigurationsverwaltung
COTS	Commercial-Off-The-Shelf	seriengefertigte kommerzielle Produkte
CPU	Central Processing Unit	Hauptverarbeitungseinheit
DMC	Distributed Multicast Communication	verteilte Multicast-Kommunikation
DPM	Data Processing Module	Datenverarbeitungsmodul
EBNF	Extended Backus-Naur Form	erweiterte Backus-Naur-Form
EloKa (de)	Electronic Warfare (en: EW)	Elektronische Kampfführung
FC	Fibre Channel	Fibre-Channel
FIFO	First In First Out	—
FM	Fault Management	Fehlermanagement
GLI	Generic System Management Logical Interface	logische Schnittstelle der generischen Systemverwaltung
GSM	Generic System Management	generische Systemverwaltung
HM	Health Monitoring	Zustandsüberwachung
HW	Hardware	—
IA	Integration Area	Integrationsbereich
IBIT	Initiated BIT	eingeleiteter BIT
ID	Identification	Identifizierung
IDL	Interface Definition Language	Schnittstellendefinitionssprache
IEEE	Institute of Electrical and Electronics Engineers	—
IF	Interface	Schnittstelle
IMA	Integrated Modular Avionics	integrierte modulare Avionik
IMC	Intra Module Communication	Intramodulkommunikation
IPC	Intra Processor Communication	Intraprozessorkommunikation
IPEC	Intra PE Communication	Intra-PE-Kommunikation
LC	Logical Configuration	logische Konfiguration
LSB	Least Significant Byte	niedrigstwertiges Byte
MC	Master Clock	Master-Uhr
MLI	Module Logical Interface	logische Modulschnittstelle
MMM	Mass Memory Module	Massenspeichermodul

MOS	MSL to OS [interface]	[Schnittstelle] Modulunterstützungsschicht/ Betriebssystem
MRC	Master Reference Clock	Master-Bezugsuhr
MSB	Most Significant Byte	höchstwertiges Byte
MSL	Module Support Layer	Modulunterstützungsschicht
MSU	Module Support Unit	Modulunterstützungseinheit
N/A	Not Applicable	nicht anwendbar
NC	Network Channel	Netzwerkanal
NII	Network Independent Interface	netzwerkunabhängige Schnittstelle
NIU	Network Interface Unit	Netzwerkschnittstelleneinheit
NSM	Network Support Module	Netzwerkunterstützungsmodul
NW	Network	Netzwerk
OC	Ordinary Clock	gewöhnliche Uhr
OLI	Operating System Logical Interface	logische Schnittstelle des Betriebssystems
OMG	Object Management Group	—
OS	Operating System	Betriebssystem
OSL	Operating System Layer	Betriebssystemschicht
PBIT	Power-Up BIT	BIT beim Einschalten
PCM	Power Conversion Module	Stromrichtermodul
PE	Processing Element	Verarbeitungselement
PSE	Power Supply Element	Stromversorgungselement
PTP	Precision Time Protocol	—
PU	Processing Unit	Verarbeitungseinheit
QoS	Quality of Service	Dienstgüte
RE	Resource Element	Ressourcenelement
RC	Remote Clock	externe Uhr
HF	Radio Frequency (en: RF)	Hochfrequenz
RFC	Request for Comments	—
RLT	Relative Local Time	relative lokale Zeit
RTBP	Runtime Blueprints	Laufzeit-Blueprints
RU	Routing Unit	Routing-Einheit
SCU	Switch Control Unit	Schaltsteuereinheit
SM	Security Management	Sicherheitsmanagement
SMBP	System Management to Blueprints [interface]	[Schnittstelle] Systemverwaltung/Blueprints
SMLI	System Management Logical Interface	logische Schnittstelle der Systemverwaltung
SMOS	System Management to OS [interface]	[Schnittstelle] Systemverwaltung/Betriebs- system
SPM	Signal Processing Module	Signalverarbeitungsmodul

SW	Software	—
TC	Transfer Connection	Transferverbindung
TC	Transparent Clock	transparente Uhr
tftp	trivial file transfer protocol	einfaches Dateiübertragungsprotokoll
TLS	Three Layer Stack	Dreischichtenstapel
VC	Virtual Channel	virtueller Kanal
VL	Video Library	Videothek

4 Systemfunktionen

In diesem Abschnitt werden der Systemkontext und die Abbildung von Systemfunktionen auf Komponenten der Softwarearchitektur beschrieben.

4.1 Systemverwaltungsfunktion

4.1.1 Allgemeines

ANMERKUNG Zu Anforderungen an die Systemverwaltung, einschließlich ausführlicher Beispiele, siehe die entsprechenden Teile von ASAAC2-GUI-32450-001-CPG Issue 01.

Die Systemverwaltung ist für die Verwaltung des IMA-Systems von der Initialisierung über sämtliche Betriebsphasen während des Flugs und am Boden und das Herunterfahren des Systems bis zur AUS-Schaltung verantwortlich. Zu ihren Aufgaben gehören somit

- Steuerung der Systeminitialisierungs-, Rekonfigurations- und Abschaltprozesse,
- Identifizierung, Maskierung, Filterung und Lokalisierung von Fehlern und
- Bereitstellung von sicherheitsbezogenen Diensten.

Die Systemverwaltung setzt sich aus zwei Funktionen auf der Anwendungsschicht und der OSL des IMA-Dreischichtenstapelmodells zusammen (Bild 3):

- der Anwendungsmanagementfunktion (AM-Funktion): luftfahrzeugabhängig, hardwareunabhängig;
- der GSM-Funktion: luftfahrzeugunabhängig, hardwareunabhängig.

Die dieser Architektur zugrundeliegenden Prinzipien sind Trennung zwischen den hardware- und luftfahrzeugabhängigen Schichten und Trennung zwischen Avionikfunktionen, die durch Funktionsanwendungen und Systemverwaltungsfunktionen dargestellt werden.

Die Systemverwaltungsfunktion ist auf drei Arten von Ebenen (Bild 4), die die folgenden Funktionalitäten abdecken, hierarchisch organisiert:

- Luftfahrzeugebene (AC-Ebene);
- Integrationsbereichsebene (IA-Ebene);
- Ressourcenelement-Ebene (RE-Ebene).

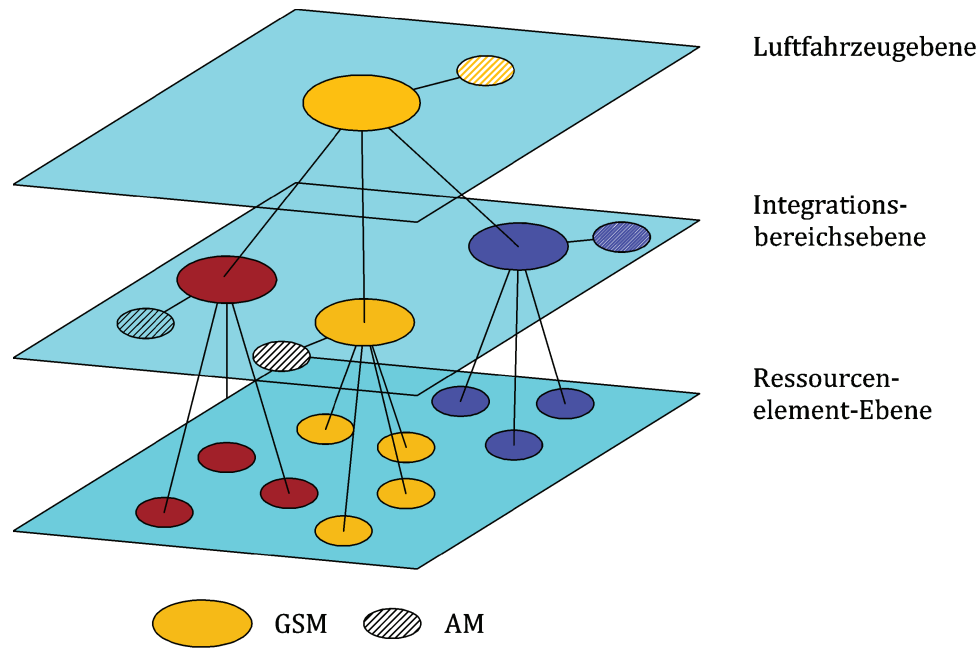


Bild 4 — Hierarchische Organisation der Systemverwaltung

Jede Ebene ist durch spezielle Merkmale gekennzeichnet:

AC-Ebene:

- Die AC-Ebene ist eine einzelne Systemverwaltungsentität, die für die Steuerung/Überwachung des gesamten IMA-Kerns verantwortlich ist.

IA-Ebene:

- Die IA-Ebene ist eine logische Gruppierung eng verzahnter Funktionsanwendungen und ihrer Ressourcen. Diese Gruppierung braucht nicht statisch zu sein, sondern darf im Verlauf einer Mission dynamisch erstellt und auch wieder gelöscht werden.
- Jeder IA steuert ein oder mehrere RE.
- IA dürfen intern hierarchisch organisiert sein, d. h. dass ein System eine oder mehrere IA-Ebenen umfassen darf. In diesem Falle muss der IA der niedrigsten Ebene ein oder mehrere PE steuern.
- Ein System darf so ausgelegt sein, dass es keine IA-Ebene umfasst. In diesem Falle liegen eine AC-Ebene und eine oder mehrere RE-Ebenen vor.

RE-Ebene:

- Das Ressourcenelement ist die niedrigste adressierbare Ebene in der Systemverwaltungshierarchie; sie ist für die Verwaltung der Funktionalität eines einzelnen Verarbeitungselements (PE) verantwortlich.

4.1.2 GSM-Funktion

Die GSM-Funktion GSM ist für die Verwaltung und Lenkung aller Ressourcen und das Management des Verhaltens des IMA-Systems mit Hilfe von RTBP verantwortlich (siehe 5.3).

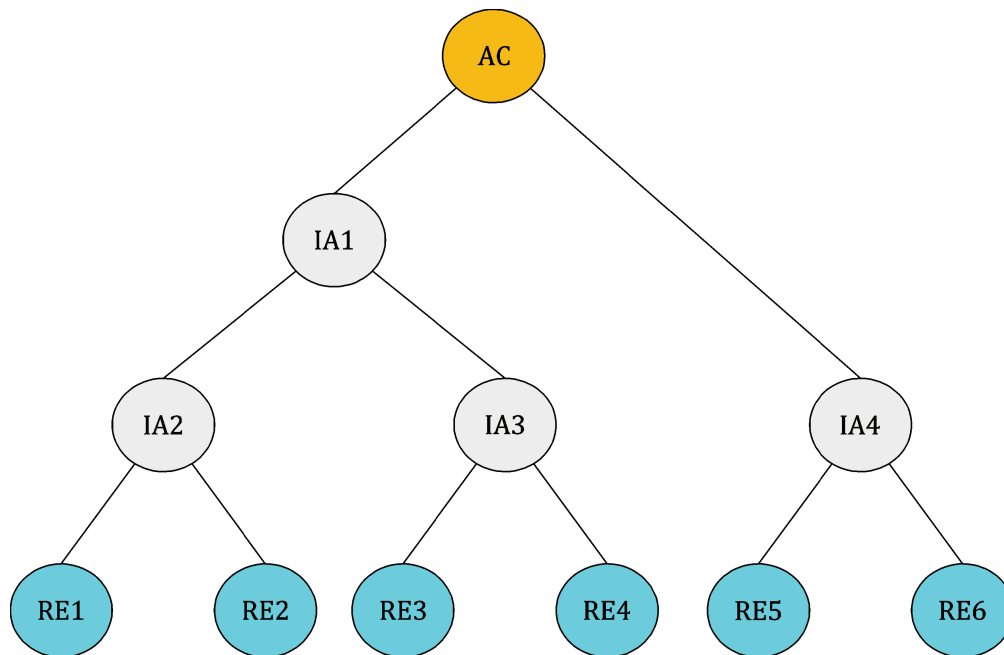


Bild 5 — GSM-Aufgliederung für die RE-Verwaltung (Beispiel)

Funktionen:

Die GSM umfasst vier Funktionen:

- Konfigurationsverwaltung:
Einrichtung der Ausgangskonfiguration, Rekonfigurationsverwaltung;
- Zustandsüberwachung:
Überwachung des Zustands und Erfassung, Filterung und Übertragung von Fehlern an das FM;
- Fehlermanagement:
Maskierung, Filterung und Lokalisierung von Fehlern einschließlich der Verarbeitung von Maßnahmen zu deren Behebung;
- Sicherheitsmanagement:
Implementierung der Sicherheitsvorgaben für das System: Authentifizierung, Entschlüsselung und Verschlüsselung.

Hierarchische Organisation:

Die nachstehenden Bilder veranschaulichen mögliche Abbildungen von Ressourcen auf die Systemverwaltungshierarchie:

- RE-Verwaltung (Bild 5): Ein AC-Manager steuert 2 IA-Manager, IA1 und IA4; IA1 steuert die IA-Manager IA2 und IA3; IA2, IA3 und IA4 steuern jeweils 2 RE.

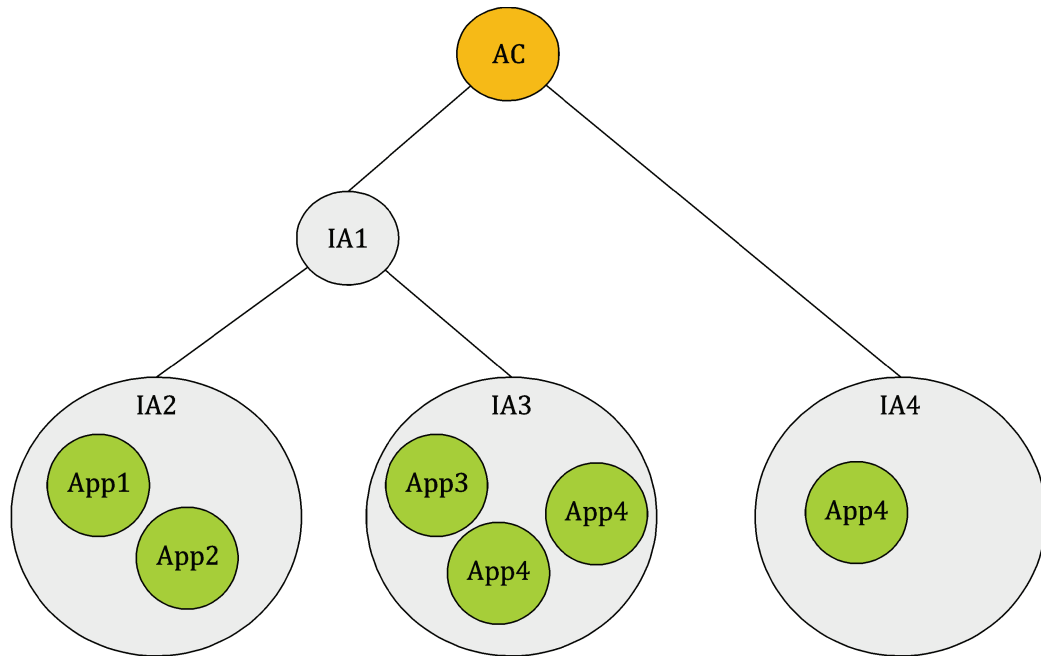


Bild 6 — IA-Anwendungssteuerung (Beispiel)

- Anwendungskonfigurationssteuerung (Bild 6): Ein AC-Manager steuert IA1 und IA4; IA1 steuert IA2 und IA3; IA2 steuert die Anwendungen App1 und App2; IA3 steuert die Anwendungen App3 und App4 (redundant); IA4 steuert die redundante Anwendung App4.

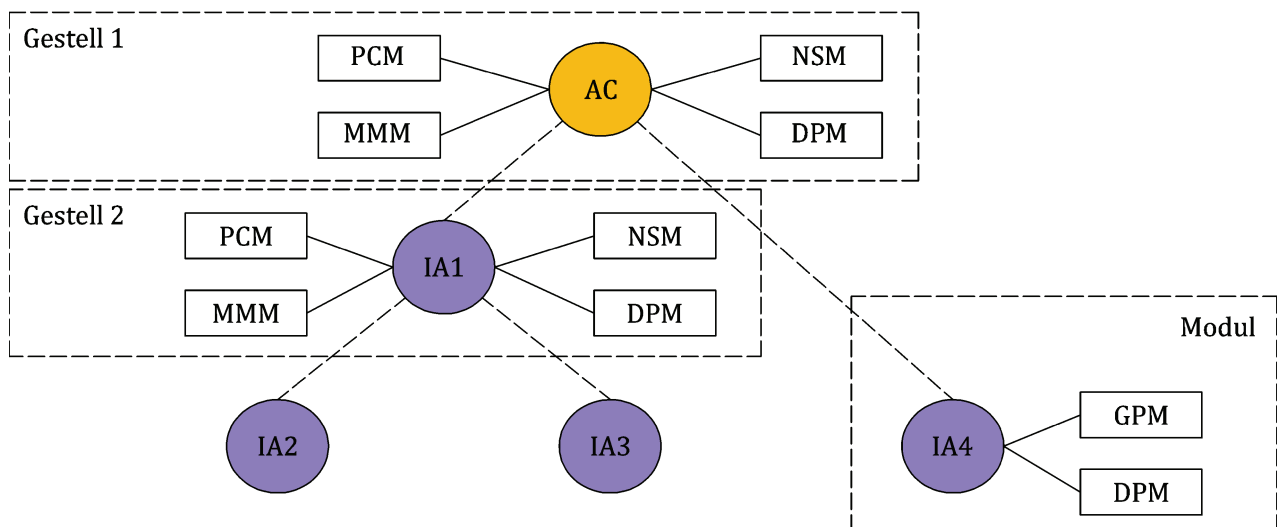


Bild 7 — GSM-Aufgliederung für die Modulverwaltung (Beispiel)

- Anwendungskonfigurationssteuerung (Bild 7): Ein AC-Manager steuert IA1 und IA4; IA1 steuert IA2 und IA3; IA2 steuert die Anwendungen App1 und App2; IA3 steuert die Anwendungen App3 und App4 (redundant); IA4 steuert die redundante Anwendung App4.

Konfigurationsdaten:

Die Konfigurationsdaten werden über die SMBP aus den RTBP bezogen. Die Rekonfiguration wird durch über die SMBP bezogene dedizierte Sequenzen definiert.

Initialisierung und AUS-Schaltung:

Initialisierung und AUS-Schaltung erfolgen auf drei verschiedenen Ebenen:

- Anwendung;
- System;
- Modul.

4.1.3 AM-Funktion

Die AM-Funktion ist für die Verwaltung und Steuerung aller vom AC abhängigen Funktionen (Funktionsanwendungen) auf der Anwendungsschicht (AL) verantwortlich. Sie fungiert als Schnittstelle zwischen den Funktionsanwendungen und einer dedizierten Instanziierung der GSM.

Hierarchische Organisation:

Der AM sollte sich nur auf der AC- und der IA-Ebene befinden, weil die RE-Ebene ressourcenorientiert ist, während die AC- und IA-Ebenen funktionsorientiert sind.

Ein Beispiel für die hierarchische Organisation des AM, das die Zuweisung von Funktionsanwendungen zu IA zeigt, ist in Bild 8 dargestellt:

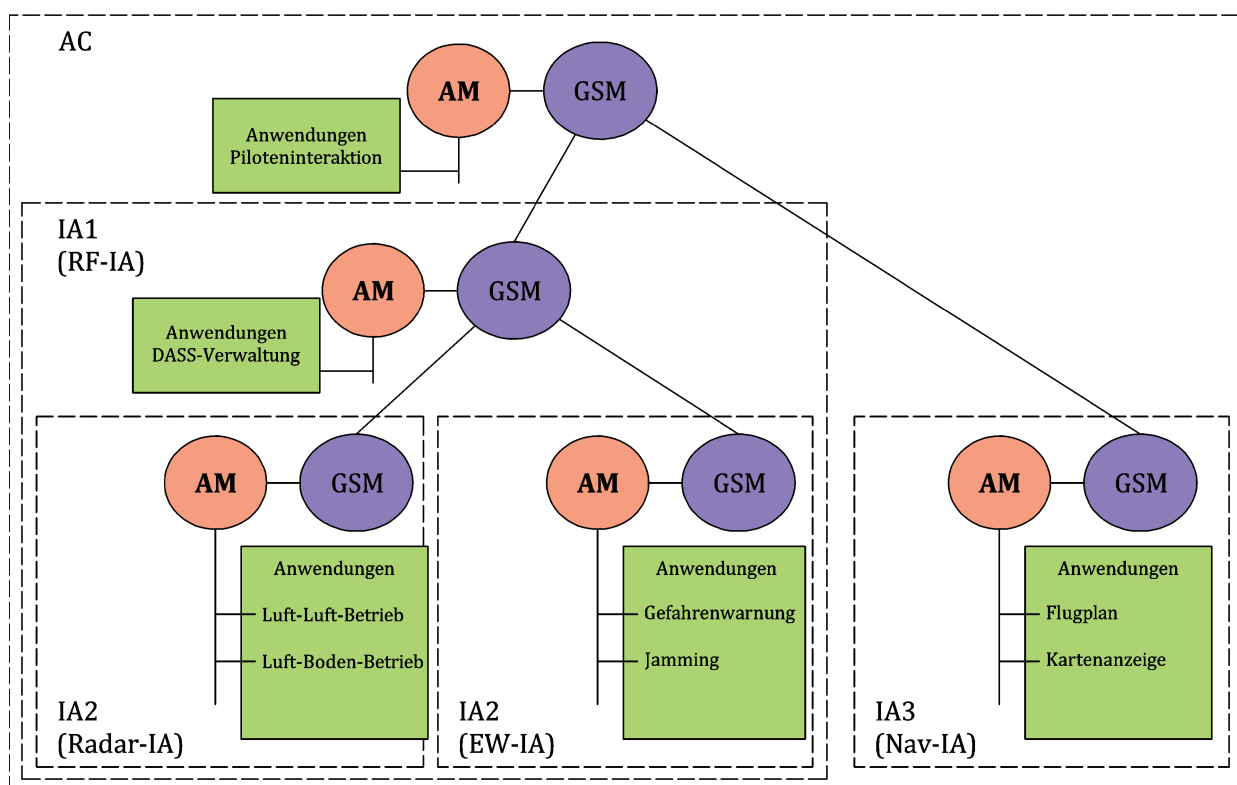


Bild 8 — Hierarchische Organisation des AM (Beispiel)

Interne Schnittstellen:

Die genormte interne Schnittstelle des AM ist die logische Schnittstelle der Systemverwaltung (SMLI). Die SMLI umfasst ein Anfrage-Antwort-Protokoll für die Änderung der logischen Konfiguration.

Externe Schnittstellen:

Es gibt keine genormten externen Schnittstellen des AM. Alle externen Schnittstellen sind anwendungsabhängig.

4.1.4 Fehlerbehandlung

Bei Systemen, die dem MOAA-Standard entsprechen, ist es erforderlich, dass die Softwareentwickler ihren Funktionsanwendungscode so schreiben, dass dieser eine Schnittstelle mit dem zugrundeliegenden OS bildet, wobei die genormten Dienstaufrufe verwendet werden, die die APOS-Schnittstelle umfassen (6.1). Während der Laufzeit ist es jedoch möglich, dass ein APOS-Dienst nicht vorschriftsmäßig funktioniert und einen Fehlerzustand an den aufrufenden Anwendungsprozess ausgibt. Dies kann auf einen tatsächlichen Fehler im zugrundeliegenden System zurückzuführen sein oder auf eine Fehlnutzung der APOS-Schnittstellen selbst (z. B. Senden eines Semaphors, bevor dieser erzeugt wurde). In jedem Fall wird der Fehler durch einen standardisierten Prozess (siehe ASAAC2-GUI-32450-001-CPG Issue 01) behandelt, bei dem die genaue Fehleridentifizierung an die Zustandsüberwachungsfunktion innerhalb der GSM übergeben wird. Jede Fehlerbehandlung muss den von der Fehlermanagementfunktion getroffenen Entscheidungen unterliegen. Im Rahmen der Fehlerbehandlung darf die Fehlermanagementfunktion die Fehlerbehandlung an einen Funktionsanwendungsprozess zurückdelegieren, indem sie den Fehlerbehandler-Thread des Anwendungsprozesses aufruft. In diesem Fall muss dieser Fehlerbehandler-Thread auf sämtliche Fehlerinformationen zugreifen können.

Es muss der Anwendung möglich sein, auf die Fehlerinformationen zuzugreifen, diese dürfen aber nur für das Debugging genutzt werden.

Ausnahmen von dieser Regel sind Zeitüberschreitungen und Ressourcen, die von der Anwendung verwaltet werden. Es ist jedoch zu beachten, dass Funktionsanwendungsprozesse mit Situationen umgehen müssen, in denen ein aufgerufener APOS-Dienst die Zeit überschritten hat. In diesem Fall muss die Anwendung, die einen Dienst aufruft, mit Hilfe eines Ausgabewertes darüber informiert werden.

4.1.5 Selbsttest

4.1.5.1 Allgemeines

Die BIT-Dienste bieten die Möglichkeit, Modul-Selbsttests durchzuführen und deren Ergebnisse zu lesen. Die Selbsttestkomponente ermöglicht den Zugriff auf alle auf dem Modul verfügbaren Selbsttestroutinen. Es gibt drei verschiedene Arten von Selbsttests:

- Selbsttest beim Einschalten (PBIT);
- kontinuierlicher Selbsttest (CBIT);
- eingeleiteter Selbsttest (IBIT).

Das OS stellt der GSM Dienste für das BIT-Management an der SMOS-Schnittstelle zur Verfügung, die mit Diensten an der MOS-Schnittstelle gepaart sind:

- get PBIT Result: ruft das gespeicherte PBIT-Ergebnis ab;
- start CBIT: führt die CBIT-Verarbeitung aus und springt anschließend zurück. Dieser Dienst ermöglicht die Ausführung entweder einer bestimmten Testart oder aller Tests;
- get CBIT Result: ruft das CBIT-Ergebnis ab;
- start IBIT: startet die IBIT-Verarbeitung;
- get IBIT Result: ruft das gespeicherte IBIT-Ergebnis ab.

4.1.5.2 Selbsttest beim Einschalten (PBIT)

Der PBIT wird durchgeführt, um im Rahmen des Hochfahrprozesses den Zustand der Modulhardware zu überprüfen. Die Tests werden eigenständig als Teil der MSL durchgeführt, bevor irgendeine Steuerung von außerhalb des Moduls eingreift. Das Ergebnis dieser Tests wird in der MSL aufgezeichnet, um von einer GSM auf einem Steuermodul über die MLI abgerufen werden zu können. Es steht über einen MOS/SMOS-Aufruf auch der lokalen GSM zur Verfügung.

4.1.5.3 Kontinuierlicher Selbsttest (CBIT)

CBIT werden durchgeführt, um bei Normalbetrieb fortlaufend den Zustand des Moduls zu überprüfen. CBIT sind nicht eingreifend. Die Tests können wie folgt ausgeführt werden:

- entweder autonom, wenn zur Ausführung des Tests keine Prozessorunterstützung erforderlich ist;
- oder unter dem Befehl der GSM, wenn zur Ausführung des Tests eine Prozessorunterstützung erforderlich ist.

Die Ergebnisse können ebenfalls auf zwei verschiedene Arten und Weisen bezogen werden:

- Rückruf (en: call-back);
- Umfrage (en: polled), entweder als Teil des Aufrufmechanismus oder als separater Aufruf.

Es folgt eine Beschreibung der verschiedenen Kombinationen des CBIT-Verhaltens:

Tabelle 2 — CBIT-Betriebsarten

Ausführungs- verfahren	Ergebnis- verfahren	Verhalten
Autonom	Rückruf	Der CBIT wird eigenständig ausgeführt und erfordert keine Steuerung von außerhalb der MSL. Das Nichtbestehen eines Tests wird der OSL mittels Rückruf angezeigt. Daraufhin wird der Dienst <i>getCbitResult</i> genutzt, um die ausführlichen Informationen zum Nichtbestehen abzurufen.
Autonom	Umfrage	Der CBIT wird eigenständig ausgeführt und erfordert keine Steuerung von außerhalb der MSL. Bei Nichtbestehen eines Tests wird das Ergebnis intern in der MSL gespeichert. Die OSL erhält keine Mitteilung. Anschließend wird periodisch <i>GetCbitResult</i> verwendet, um Informationen zum Nichtbestehen abzurufen. Ist es zu keinem Nichtbestehen gekommen, werden keine Maßnahmen getroffen. Bei einem Nichtbestehen werden die ausführlichen Informationen dazu ausgegeben.
Auf Befehl	Rückruf	Die CBIT-Ausführung wird von der OSL gesteuert. Das Nichtbestehen eines Tests wird der OSL mittels Rückruf angezeigt. Daraufhin wird <i>GetCbitResult</i> genutzt, um die ausführlichen Informationen zum Nichtbestehen abzurufen.
Auf Befehl	Umfrage	Die CBIT-Ausführung wird von der OSL gesteuert. Die Zeit, die zum Ausführen des CBIT bei jedem Aufruf des Dienstes <i>startCbit</i> zur Verfügung steht, ist MSL-spezifisch. Wird ein Nichtbestehen erkannt, so wird <i>GetCbitResult</i> verwendet, um die ausführlichen Informationen dazu abzurufen.

4.1.5.4 Eingeleiteter Selbsttest (IBIT)

IBIT werden durchgeführt, um im Rahmen des Fehlermanagementprozesses den Zustand der Modulhardware zu überprüfen. Es wird eine umfassende Prüfung des Moduls durchgeführt, um bei der Fehlerlokalisierung zu helfen. Die Tests können aus der Ferne ausgeführt werden, gesteuert von einer GSM auf einem Steuermodul über die MLI, oder über einen MOS/SMOS-Aufruf (*startIbit*) von der lokalen GSM aus, sofern diese verfügbar ist. Der IBIT-Betrieb kann destruktiv wirken. Das bedeutet, dass nicht garantiert werden kann, dass die aktuelle Konfiguration des Moduls nach Abschluss der Tests noch besteht. Daher muss sorgfältig darauf geachtet werden, sicherzustellen, dass das System bei Ausführung eines IBIT nicht beeinträchtigt wird. Außerdem sollte die Anwendung eines destruktiven Tests auf den Aufruf über die MLI beschränkt sein. Das Ergebnis dieser Tests wird in der MSL aufgezeichnet, um von einer GSM auf einem Steuermodul über die MLI oder über einen MOS/SMOS-Aufruf (*getIbitResult*) von der lokalen GSM aus, sofern diese den IBIT gestartet hat, abgerufen werden zu können.

4.2 Kommunikation

4.2.1 MOAA-Kommunikationsmodell

4.2.1.1 Kurzbeschreibung

Der MOAA-Kommunikationsstapel (siehe Bild 9) muss von Folgendem unterstützt werden:

- VC (die von der OSL bereitgestellt werden);
- Transferverbindungen (TC) (die von der MSL bereitgestellt werden und hardwareunabhängig sind);
- Netzwerkkanäle (NC) (die von der MSL bereitgestellt werden und hardwareabhängig sind).

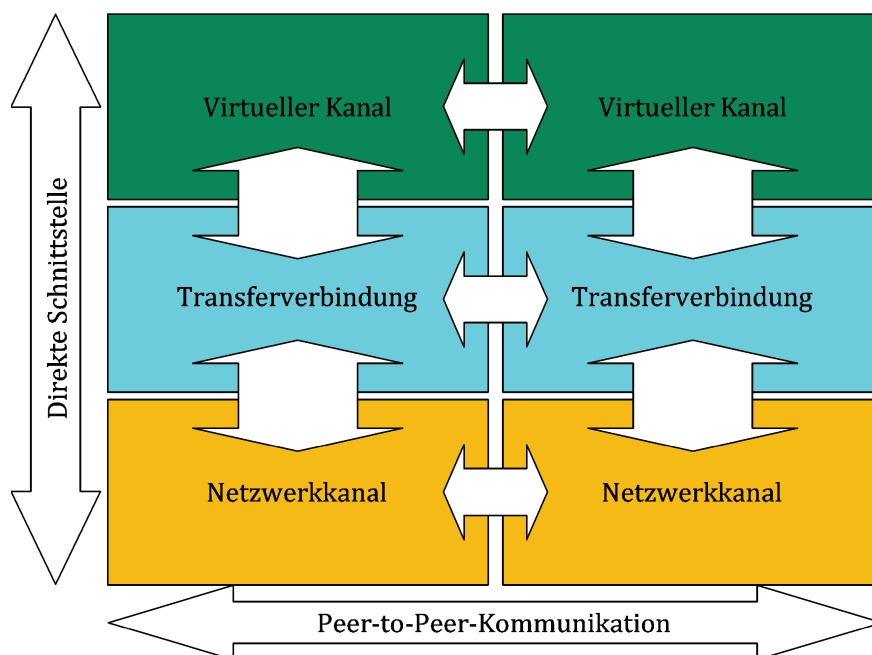


Bild 9 — Der MOAA-Kommunikationsstapel

Die MOAA-Kommunikation muss folgende Kommunikationen unterstützen:

- ein Sender an einen Empfänger (1 : 1);
- Multicast (ein Sender an N Empfänger [1 : N], der Fall „ein Sender an einen Empfänger“ ist eine Teilmenge des oben genannten Falls [1 : 1]);

- verteiltes Multicast (gilt für Signalverarbeitungsanwendungen [$M : N$]).

Die Kommunikation erfolgt auch auf unterschiedlichen Architekturebenen:

- Die VC-Kommunikation erfolgt auf den Ebenen der OSL und der AL. Die dazugehörigen Protokolle werden als OLI und logische Anwendungsschnittstelle (ALI) bezeichnet. Die ALI ist die durch die Funktion definierte Kommunikation von Anwendungen miteinander und mit nicht zum Kern gehörenden Instanzen, weswegen dieses Protokoll nicht durch den MOAA-Standard genormt wird.
- Die TC-Kommunikation erfolgt auf der MSL-Ebene.

4.2.1.2 VC

Die Interprozesskommunikation basiert auf virtuellen Kanälen. Diese weisen folgende Eigenschaften auf:

- sie sind unidirektional;
- nachrichtenorientiert (d. h. eine Nachrichtendefinition wird einem VC zugewiesen);
- werden von der OSL verwaltet (Erstellung, Löschung, Routing);
- sind in Bezug auf den Zeit- und Ressourcenverbrauch vorhersagbar.

Das Konzept ermöglicht es einem einzelnen Übertragungsprozess, Daten an einen oder mehrere Empfangsprozesse zu senden. Ein Empfangsprozess darf auf demselben Verarbeitungselement vorhanden sein, demselben CFM oder sogar einem anderen CFM als dem des Sendeprozesses. Der Sendeprozess hat keinerlei Kenntnis vom Empfangsprozess; er gibt lediglich bestimmte Daten an einen VC aus. Ebenso hat der Empfangsprozess keinerlei Kenntnis vom Sendeprozess; er empfängt lediglich bestimmte Daten von bestimmten VC.

Die Quell- und Zielprozesse, die zwischen diesen Prozessen zu übertragenden Datenelemente und die VC, über die diese übertragen werden, werden während des Systementwurfs definiert. Während der Laufzeit werden diese Informationen durch die RTBP bereitgestellt. Folglich ist der für eine gegebene Systemkonfiguration verwendete VC-Satz fest und bietet einen Bezug für die Erzeugung von Auditdaten.

Jedem VC muss eine bestimmte Nachricht zugewiesen sein bzw. werden. Daher sind den VC auch folgende Nachrichteneigenschaften zugeordnet:

- Datendarstellung: Eine Nachricht muss in den Blueprints beschrieben sein bzw. werden;
- Sicherheit (Verschlüsselung, Entschlüsselung): Der VC ist entweder gekennzeichnet oder nicht gekennzeichnet;
- Modularität: Eine Anwendung ist ein Nachrichtenserver. Während des Entwurfs der Anwendung ist nicht bekannt, wie viele Benutzer die Nachricht empfangen sollen.

4.2.1.3 TC

Die von der NII angebotene Basiskommunikationsverbindung ist die Transferverbindung (TC). Der Datentransfer über TC besitzt die folgenden Eigenschaften:

- Die TC ist unidirektional.
- Das OS verwaltet die TC im Hinblick auf Erzeugung, Löschung und Routing.

- Eine TC muss entweder von einem oder von vielen VC verwendet werden können.
- Eine TC unterstützt Streaming-Kommunikation.

4.2.1.4 NC

Für die NC legt die MOAA-Norm keine Anforderungen fest. Die NC müssen innerhalb der MSL verwaltet werden.

Die Eigenschaften eines NC müssen mittels Blueprint-Daten konfiguriert werden. Diese Daten sind implementierungsabhängig und müssen daher für die OSL-Software transparent sein.

4.2.1.5 Reihenfolge der Header

Je nachdem, ob es sich um VC-Kommunikation oder um TC-Kommunikation handelt, gibt es zwei Arten von Headern. Bild 10 zeigt die Reihenfolge der Header für die TC-Kommunikation. Die TC-Kommunikation wird bei der MLI und als Basis für die VC-Kommunikation angewendet, die auf der OSL- und der AL-Ebene angewendet wird. Bild 11 zeigt die Reihenfolge der Header für die VC-Kommunikation, bei der eine weitere Header-Komponente, der VC-Header ergänzt wird.

Medium	Transferverbindung	Nutzlast
NC Header	TC Header	TC-Daten

Bild 10 — Reihenfolge der Header für die TC-Kommunikation

Medium	Transferverbindung	Virtueller Kanal	Nutzlast
NC Header	TC Header	VC Header	VC-Daten

Bild 11 — Reihenfolge der Header für die VC-Kommunikation

Die Header-Teile werden wie folgt beschrieben:

- Der NC-Header ist von der angewendeten Netzwerktechnologie abhängig. Im Falle von AFDX oder Ethernet könnte dies z. B. ein UDP-Header sein.
- Der TC-Header ist in 7.4.1 festgelegt.
- Der VC-Header ist in 7.1.1 festgelegt.
- Spezifische TC-Daten sind als Teil des MLI-Protokolls in 7.4.2 festgelegt.
- Spezifische VC-Daten sind für die folgenden Protokolle festgelegt:
 - Das OLI-Protokoll ist in 7.1.2 festgelegt;
 - das GLI-Protokoll ist in 7.2.2 festgelegt;
 - das SMLI-Protokoll ist in 7.3.2 festgelegt; und
 - ALI wird nicht festgelegt, weil dieses Protokoll anwendungsspezifisch ist.

4.2.2 Arten des Datentransfers

Bild 12 zeigt, ausgehend vom Transferbereich, die vier Kommunikationsarten:

- Intraprozessorkommunikation (IPC):
Kommunikation innerhalb eines Prozessors, die vom VC auf OSL-Ebene gehandhabt wird;
- Intra-PE-Kommunikation (IPEC):
Kommunikation zwischen Prozessoren innerhalb eines Verarbeitungselements, die von der TC auf MSL-Ebene gehandhabt wird;
- Intramodulkommunikation (IMC):
Kommunikation zwischen PE innerhalb eines Moduls, die von der TC auf MSL-Ebene gehandhabt wird;
- Kommunikation zwischen Modulen (BMC):
Kommunikation zwischen CFMs, die von einem VC auf OSL-Ebene und von einer TC und einem NC auf MSL-Ebene gehandhabt wird.

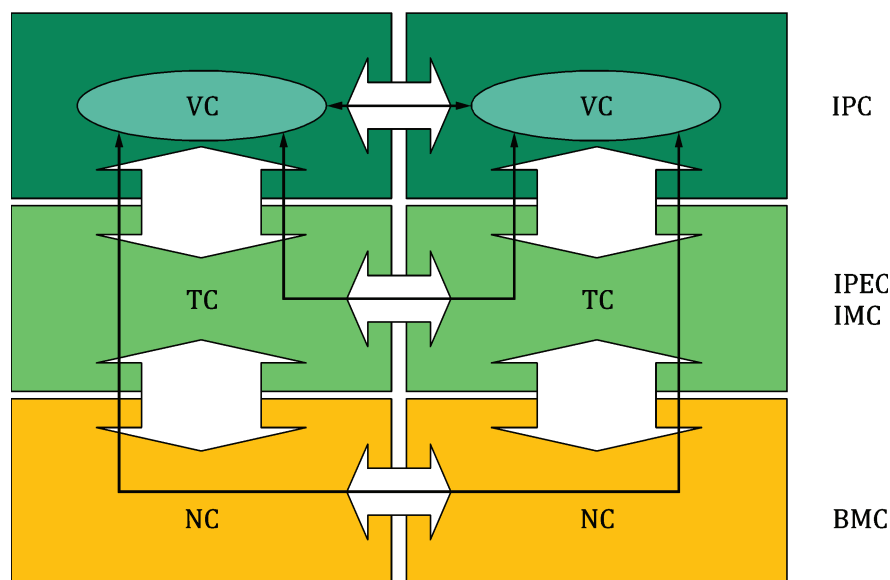


Bild 12 — Arten des Datentransfers

4.2.3 Kommunikationskonfiguration

Die MOAA-Kommunikation muss auf der Grundlage von Blueprint-Daten konfigurierbar sein (siehe 5.3).

Der Geltungsbereich einer lokalen VC-ID ist auf einen einzelnen Anwendungsprozess begrenzt. Deren Wert wird durch die Implementierung des zugehörigen Anwendungsprozesses bestimmt. Dieser Wert ist innerhalb dieses Anwendungsprozesses eindeutig. Er ist von der Auslegung der übrigen Anwendungsprozesse unabhängig.

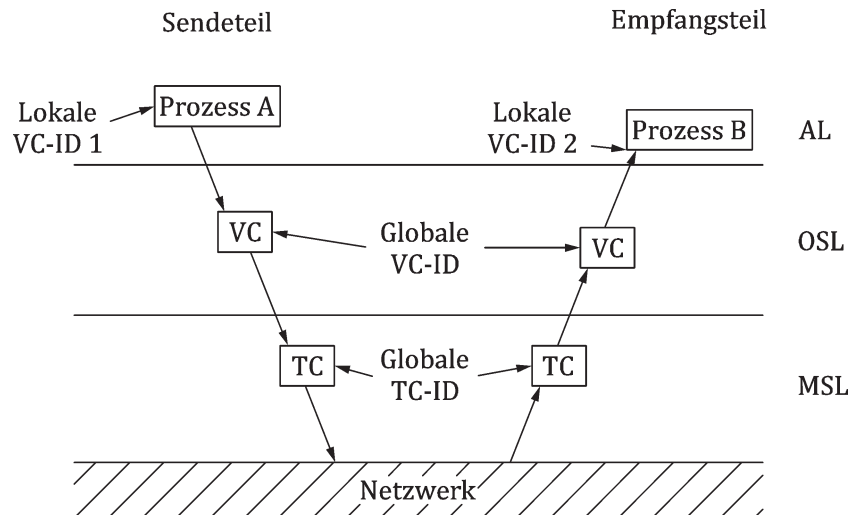


Bild 13 — Kommunikationskonzept

Die GSM verwendet Blueprint-Daten sowohl auf Sende- als auch auf Empfangsseite und muss das OS dazu veranlassen, auf der Basis von Blueprint-Informationen Kommunikationsobjekte zu erzeugen, um die Kommunikation zu konfigurieren:

- Das OS erzeugt eine Instanz eines VC. Eine innerhalb des Systems eindeutige globale VC-ID identifiziert einen VC. Sie muss auf Sende- und auf Empfangsseite identisch sein.
- Das OS bindet einen anwendungsprozesslokalen VC an einen systemglobalen VC.
- Das OS erzeugt eine Instanz einer TC. Eine innerhalb des Systems eindeutige globale TC-ID identifiziert eine TC. Sie muss auf Sende- und auf Empfangsseite identisch sein.
- Das OS bildet einen VC auf eine entsprechende TC ab.

Die Definition der NC muss Teil der TC-Parametermenge sein, da NC der Definition einer TC zugeordnet sind. Daher werden NC nicht als eine separate Entität behandelt.

4.2.4 Kommunikationsprotokolle

4.2.4.1 Einleitung

Der MOAA-Kommunikationsstapel definiert die Protokolle zur Herstellung einer Kommunikation zwischen MOAA-Schichten.

4.2.4.2 VC zu VC

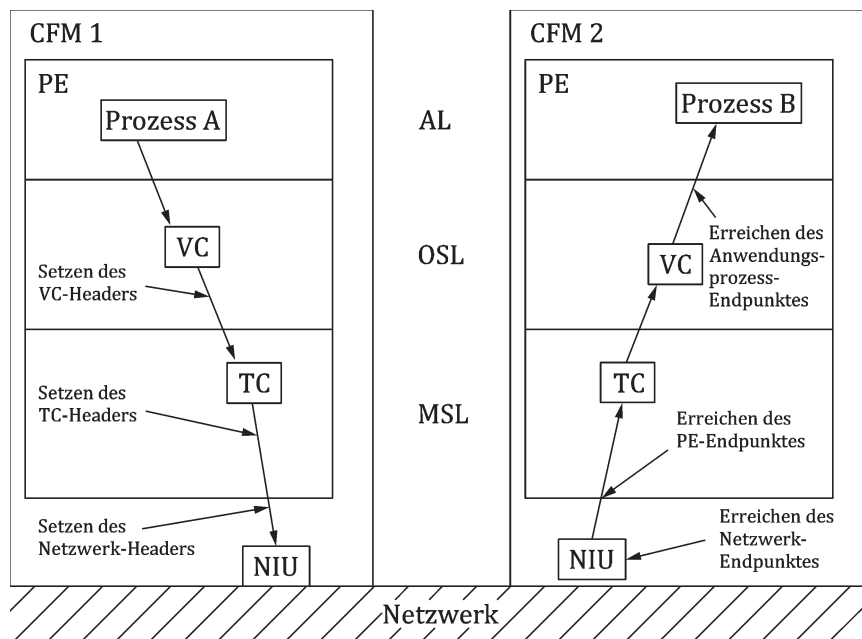


Bild 14 — Routing der Kommunikation zwischen AL

Bei einer Kommunikation zwischen Prozessen auf zwei unterschiedlichen CFM können folgende Mechanismen angewendet werden:

- Prozess-zu-Prozess: Der VC lässt das Erreichen eines Prozesses innerhalb eines PE zu.
- PE-zu-PE: Die TC lässt das Erreichen eines PE innerhalb eines CFM zu.
- CFM-zu-CFM: Der NC lässt das Erreichen eines CFM im Netzwerk zu.

VC-zu-VC unterstützt die Kommunikation zwischen

- Anwendungsprozessen,
- GSMs, die Inter-GSM-Kommunikation ist eine genormte logische Schnittstelle, nämlich die GLI, siehe 7.2 und
- Anwendungsmanager und GSM, einer genormten logischen Schnittstelle, nämlich der SMLI, siehe 7.3.

4.2.4.3 TC zu TC

Der Zugriff auf die TC und deren Steuerung müssen entweder an der Grenze zwischen OSL und MSL, d. h. der NIU, oder innerhalb der MSL erfolgen.

Wie oben beschrieben, unterstützt die TC-zu-TC-Kommunikation die BMC-, IMC- und IPE-Kommunikation. Für IPEC und IMC ist der TC-Header erforderlich. Die BMC erfordert sowohl TC- als auch Netzwerk-Header.

Zu beachten ist, dass es sich bei der Inter-MSL-Kommunikation um eine genormte logische Schnittstelle, nämlich die MLI, handelt, siehe 7.4.

4.2.4.4 VC-ohne-Header-Transfer

Ein VC-ohne-Header-Transfer ist erforderlich, wenn ein Anwendungsprozess mit einer MSL auf einem anderen PE oder mit nicht zum Kern gehörenden Geräten kommuniziert; ein Beispiel dafür ist die Graphikverwaltung, siehe 4.6.

Eine Anwendung sendet einen VC ohne Header:

- Der Sendeprozess verwendet einen VC, der auf eine TC abgebildet wird.
- Die APOS-VC-Nachricht wird so, wie sie ist, bereitgestellt, ohne dass ein VC-Header angebunden wird.
- Ein CFM oder ein nicht zum Kern gehörendes Gerät empfängt die TC. Dieser wird je nach Erfordernis entschlüsselt.

Ein VC ohne Header wird empfangen:

- Das CFM oder das nicht zum Kern gehörende Gerät verschlüsselt die TC-Nutzdaten je nach Erfordernis.
- Ein VC ohne Header wird mittels seiner Eigenschaften identifiziert, die durch die Blueprint-VC-Informationen definiert sind.
- Erkennt das OS den VC ohne Header als einen solchen, so ist er ohne Entfernen des VC-Headers zu verarbeiten.
- Die Verwendung eines VC ohne Header bedeutet, dass jede TC, die einen VC ohne Header trägt, auf einen einzelnen VC beschränkt ist.

4.2.4.5 Routing-Header

Um eine Nachricht zwischen Anfangs- und Endpunkt zu leiten, sind folgende Informationen erforderlich:

- ein Header, der Informationen zu seinen Merkmalen enthält;
- ein Bezeichner, der jede Nachrichtenschicht beschreibt. Dieser Bezeichner erlaubt die Identifizierung der Routing-Informationen in internen Tabellen.

Jeder Header muss Informationen zum Erreichen eines Endpunktes bereitstellen (siehe Bild 15):

- der Netzwerk-Header stellt Informationen zum Erreichen des vorgesehenen CFM bereit;
- da der NC von mehreren TC genutzt werden kann, muss der TC-Header die TC identifizieren. Diese Identifizierung ermöglicht, dass das vorgesehene PE erreicht wird;
- da die TC von mehreren VC genutzt werden kann, muss der VC-Header den VC identifizieren. Diese Identifizierung ermöglicht, dass der vorgesehene Prozess erreicht wird.

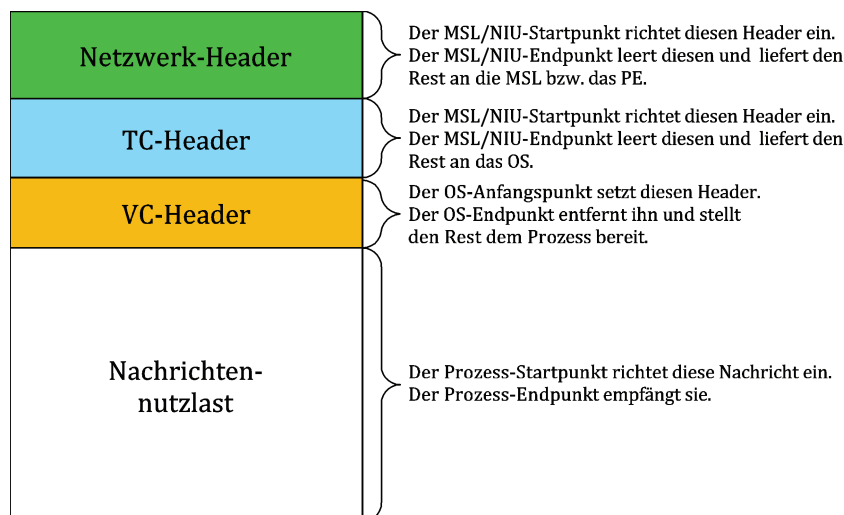


Bild 15 — MOAA-Nachricht bei BMC-Datentransfer

Die Schicht innerhalb des MOAA-Kommunikationsstapels, auf der der Transfer erfolgt, bestimmt den Satz der erforderlichen Routing-Daten. Wenn z. B. nur VC-Kommunikationsfunktionen beteiligt sind, ist für IPC lediglich ein VC-Header ausreichend.

Der TC-Header und der VC-Header sind genormt. Tabelle 3 definiert ihre Anwendbarkeit auf die Datentransferarten. Der Netzwerk-Header ist nicht genormt, weil er technologieabhängig ist.

Tabelle 3 — Routing-Informationen und Datentransfer

Datentransfer	Netzwerk	TC	VC
BMC	Ja	Ja	Ja
IMC	Nein	Ja	Ja
IPEC	Nein	Ja	Ja
IPC	Nein	Nein	Ja

4.2.5 Multicast

Die MOAA-Kommunikation muss die Multicast-Kommunikation unterstützen. Diese Fähigkeit ermöglicht die Verbindung eines Senders mit vielen Empfängern.

Die Multicast-Fähigkeit muss auf VC- und TC-Ebene des MOAA-Kommunikationsstapels unterstützt werden.

Bei mehreren Empfängern kommt es wahrscheinlich zu einer Kombination von mehreren gleichzeitig angewendeten Datentransferarten (IPC, IPEC, IMC und BMC).

Der VC-Multicast besitzt einen eindeutigen Bezeichner. Der Senderprozess hat keine Kenntnis von seinem Multicast-Verhalten. Bei IPC muss dieser VC an alle Empfängerprozesse gebunden sein.

Wenn der VC TC nutzt (IPEC, IMC und BMC), bestehen zwei Möglichkeiten:

- Verwendung einer einzelnen TC, die Multicast-Fähigkeit bietet;
- Verwendung mehrerer einfacher TC.

4.2.5.1 Einzelne TC mit Multicast-Fähigkeit

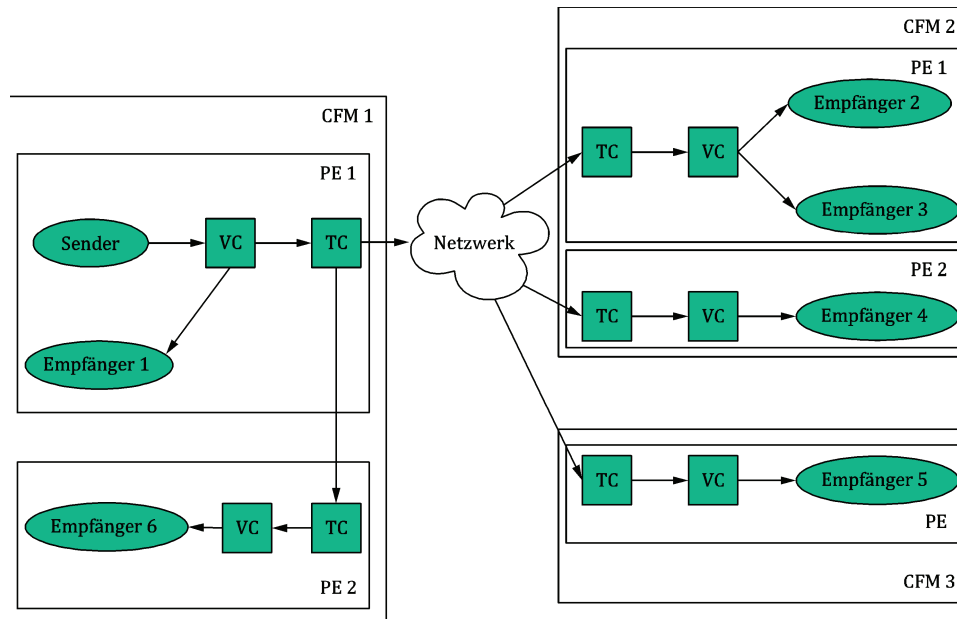


Bild 16 — Multicast-Schema mit einer einzelnen TC

In diesem Beispiel ist der TC-Bezeichner eindeutig. Bild 16 zeigt folgende Arten des Datentransfers (vom Sender an):

- Empfänger 1: IPC; der VC ist an den Sender als abgehend und an den Empfänger als eingehend gebunden;
- Empfänger 2 und Empfänger 3: BMC; die eingehende TC ist an einen VC gebunden, dieser eingehende VC ist an beide Empfänger gebunden;
- Empfänger 4: BMC; zu beachten ist, dass die TC auf CFM2/PE1 und CFM2/PE2 denselben Bezeichner hat. Das bedeutet, dass die MSL in der Lage sein muss, mehrere Endpunkte mit demselben Bezeichner zu handhaben;
- Empfänger 5: BMC;
- Empfänger 6: IMC; die abgehende TC von PE1 führt sowohl zum Netzwerk als auch zu PE2. Auch hier wird ein TC mit mehreren Endpunkten gehandhabt.

Für das Durchqueren des Netzwerkes stehen zwei Verfahren zur Verfügung:

- 1) Die MSL auf NIU-Ebene dupliziert die TC-Nachricht in mehreren Nachrichten, wobei der Netzwerk-Header dem Endpunkt im Netzwerk entspricht.
- 2) Die MSL auf NIU-Ebene sendet eine eindeutige TC-Nachricht und einen Netzwerk-Header, der das Multicast-Verhalten angibt. Anschließend handhabt die Netzwerkressource den Multicast-Vorgang und führt die Nachricht den Endpunkten zu.

4.2.5.2 Mehrere einfache TC

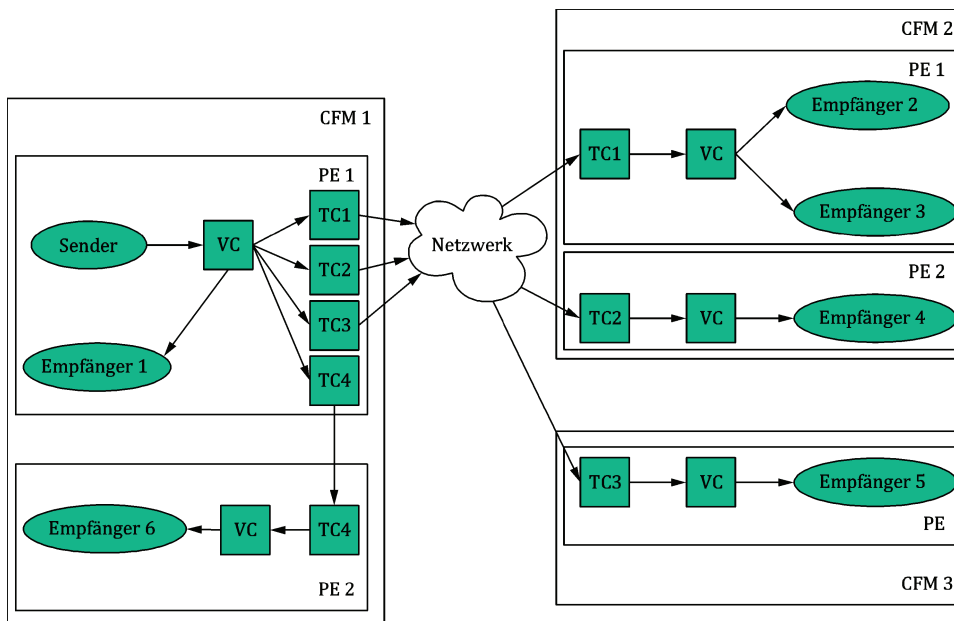


Bild 17 — Multicast-Schema mit mehreren einfachen TC

Auf Sendeseite ist der Multicast-VC an mehrere TC gebunden. Auf OS-Ebene wird die VC-Nachricht über mehrere TC ausgesendet.

4.2.6 Verteiltes Multicast

4.2.6.1 Einleitung

Die verteilte Multicast-Kommunikation gilt nur für die Signalverarbeitung. Bei der verteilten Multicast-Kommunikation handelt es sich um eine $N : M$ -Kommunikation (N Sender an M Empfänger), die sowohl Multicasting als auch Fragmentierung nutzt.

Verteiltes Multicast wird verwendet, wenn es erforderlich ist, zur Verringerung der Zeitverarbeitung einer Folge von Signalverarbeitungsoperationen Datenparallelität zu nutzen. Die Eingangsdaten werden in Unterteile aufgeteilt, die von mehreren Prozessoren in derselben Zeit verarbeitet werden. Daten werden von unterschiedlichen Prozessoren verarbeitet, der ausgeführte Algorithmus ist jedoch derselbe.

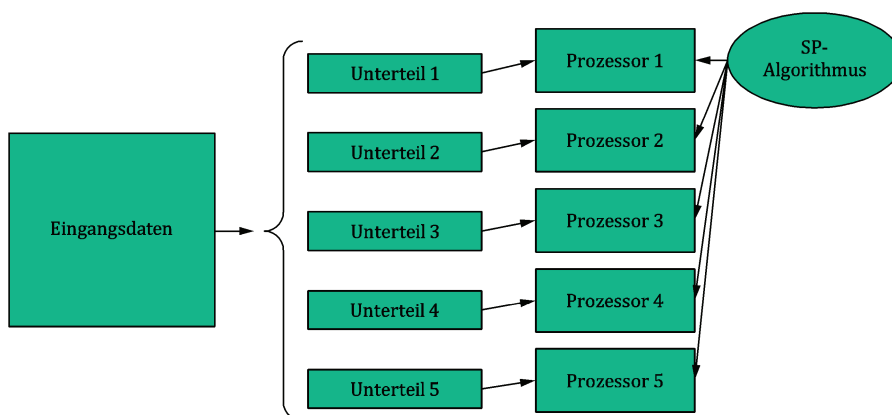


Bild 18 — Datenparallelität

Der verteilte Multicast erlaubt die Neuverteilung von Daten, die über mehrere Prozessoren verteilt sind. Ein typischer Fall für die Anwendung von verteiltem Multicast ist der „Corner Turn“. Diese Funktion kehrt die Zeilen und Spalten einer Tabelle um, wie unten dargestellt.

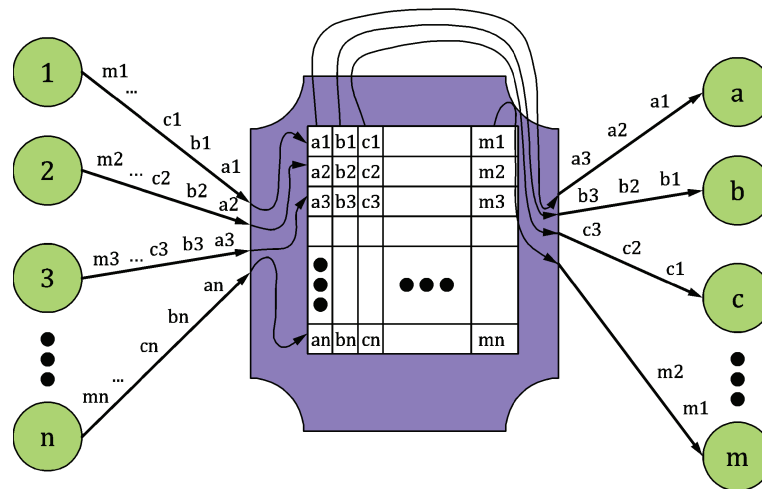


Bild 19 — Corner-Turn

In Bild 19 sind $[1 \dots n]$ Sender und $[a \dots m]$ Empfänger. Die umzukehrende Matrix wird vom Kommunikationsmechanismus virtuell konstruiert.

Jeder Sender muss für die Adressierung von Daten an jeden Empfänger eine Verteilungsregel anwenden. Eine Verteilungsregel ist ein lineares Gesetz, das definiert, wie ein gesendeter Puffer zu fragmentieren ist, um die einzelnen an jeden Empfänger zu sendenden Puffer zu generieren.

Jeder Empfänger muss für die Adressierung von Daten von jedem Sender eine Sammlungsregel anwenden. Eine Sammlungsregel ist ein lineares Gesetz, das definiert, wie aus den von mehreren Sendern empfangenen fragmentierten Puffern ein Puffer zu bilden ist.

Die $N : M$ -Kommunikation bzw. verteilte Multicast-Kommunikation darf wie folgt betrachtet werden:

- vom Sendeteil aus als eine $1 : M$ -Multicast-Kommunikation, bei der die M Empfänger jeweils ein anderes Fragment erhalten;
- vom Empfangsteil aus als eine $N : 1$ -Kommunikation, bei der N Sender ein Fragment abgeben.

Der Corner Turn darf in einem Kontext angewendet werden, der mehrere Dimensionen erfordert. Daten in einer Radaranwendung sind in mehrdimensionalen Arrays organisiert. Im folgenden Beispiel einer Radaranwendung ist ein Wert an einen Antennenstrahl, einen Impuls und ein Entfernungsfenster (en: range gate) gebunden. Dies führt zur Verwendung einer Ecke in drei Dimensionen (Impuls, Entfernungsfenster, Strahl).

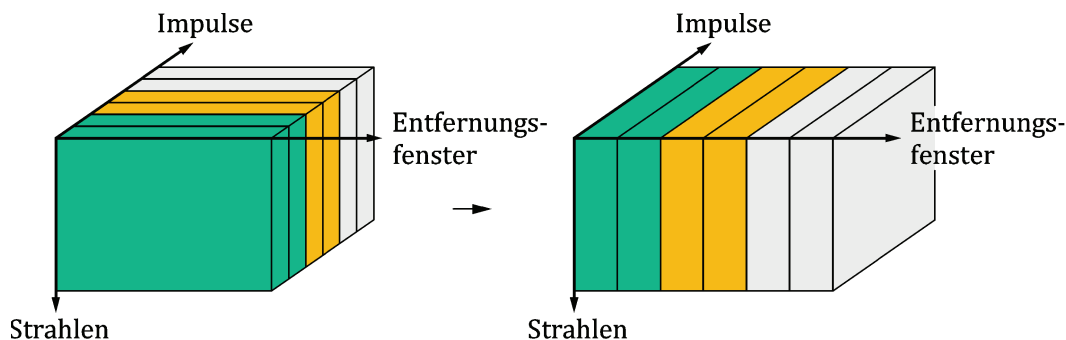


Bild 20 — Corner-Turn in drei Dimensionen

Die einzelnen Scheiben stellen einen Teil des Würfels dar, der von einem Prozessor verarbeitet wird. Sie dürfen einander überlappen.

4.2.6.2 Anforderungen

Das verteilte Multicast muss folgende Merkmale bieten:

- Ein am verteilten Multicast beteiligter Prozessor darf ein Sender und ein Empfänger für diese Transaktion sein.
- Jeder Sender und jeder Empfänger darf sich in einem oder mehreren Verarbeitungselementen befinden, die sich wiederum in einem oder mehreren SPM befinden dürfen.
- Für jeden sendenden Prozessor muss eine Verteilungsregel je Empfangsprozessor definiert werden.
- Für jeden empfangenden Prozessor muss eine Sammlungsregel je Sendeprozessor definiert werden.
- Die Signalverarbeitungsanwendung muss in der Lage sein, eine Fragmentierung in einer, zwei oder drei Dimensionen zu verarbeiten.
- Jeder Nachricht, die auf einer für die verteilte Multicast-Kommunikation bestimmten TC gesendet wird, müssen der TC-Bezeichner für das Routing an den Empfänger sowie ein Fragmentbezeichner zur Wiederherstellung vorausgehen.

4.2.6.3 Verteilte Multicast-Kommunikation in TLS

Der vorliegende Abschnitt stellt aus der Sicht eines Prozessors vor, wie Daten gesendet und dann empfangen werden.

Die Anwendung muss für das Senden der Daten einen VC verwenden. Der VC ist als ein VC ohne Header festzulegen. Dieser VC ist an eine als DMC-TC festgelegte TC gebunden.

Das OS muss den MOS-Dienst ausführen, der für DMC-Zwecke bestimmt ist: *sendFragmentedTransfer*.

Die Verteilung der DMC-TC erfolgt innerhalb der MSL. Die einzelnen Fragmente müssen separaten TC zugewiesen werden. Jedem Empfänger ist eine TC zuzuweisen. Der TC-Bezeichner identifiziert den Empfänger.

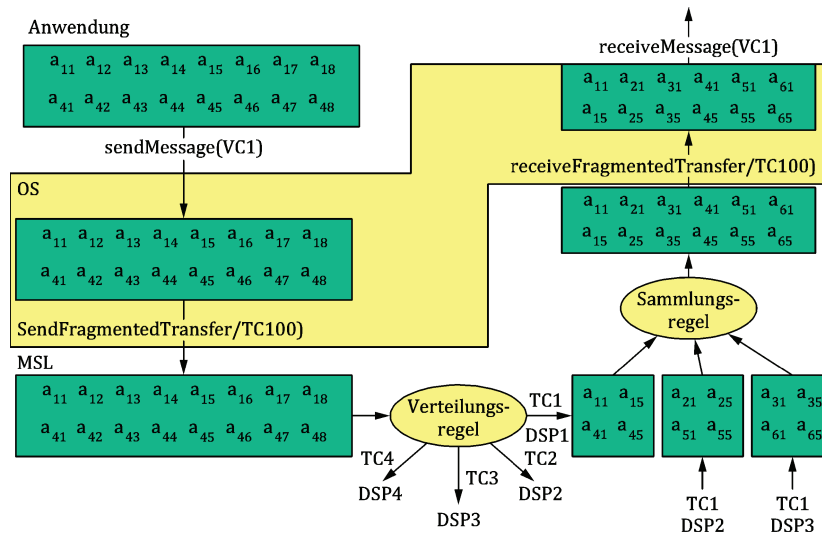


Bild 21 — Darstellung der beteiligten Dienste in DSP1

Die empfangende MSL muss die TCs mit dem als DMC-TC festgelegten TC-Bezeichner sammeln und die Nachricht auf der Grundlage des Fragmentbezeichners wiederherstellen, der den Sender identifiziert. Wenn die Wiederherstellung des Puffers abgeschlossen ist, wird er vom MOS-Dienst `receiveFragmentedTransfer` abgerufen. Der MOS-Dienst darf dieselbe TC-ID verwenden wie der Sendepunkt.

Das OS muss den MOS-Dienst `receiveFragmentedTransfer` zur Weiterleitung von Daten an die Anwendung durch einen VC ohne Header ausführen.

Bild 21 zeigt ein Beispiel für einen Corner Turn für die DMC-Verarbeitung auf einem Prozessor.

4.2.6.4 Nachrichtenfragmentierung

Die Datenauswahl der Nachrichtenfragmentierungsprozesse folgt einer linearen Regel, welche, je nachdem, ob diese auf Sende- oder Empfangsseite angewendet wird, entweder eine Verteilungs- oder eine Sammlungsregel bereitstellt.

In dem Corner-Turn-Beispiel müssen Datenadressen nach der folgenden C-ähnlichen Formel berechnet werden:

```
for( a3 = 0; a3<N3; a3++ ) {
  for( a2 = 0; a2<N2; a2++ ) {
    for( a1= 0; a1<N1; a1++ ) {
      DataAddress = Address0 + a1*Inc1 + a2*Inc2 + a3*Inc3 ;
    }
  }
}
```

Dabei ist

$Address_0$ = *initialAddress* des internen Puffers, an der mit der Fragmentierung begonnen wird;

N_x die Datenlänge der Dimension X ;

a_x der aktuelle Index der Dimension X ;

Inc_x	die Schrittweitzahl der Dimension X ;
x	die Nummer der Dimension: 1 (erste Dimension) 2 (zweite Dimension) 3 (dritte Dimension).

Wird die Fragmentierung im Kontext von zwei Dimensionen verarbeitet, so gilt N_3 und Inc_3 sind gleich null.

4.2.7 Streaming

Streaming-Daten sind kontinuierliche Daten. Zwei grundlegende Anwendungen sind Video und Audio. Streaming-Daten dürfen nur auf der TC-Ebene gehandhabt werden. Dieser Datenstrom darf identifizierte Blöcke von Unterdaten enthalten, z. B. dürfen Videodaten einzelne Bilder enthalten.

Um dem System zu ermöglichen, mit diesem Strom Schritt zu halten, müssen Daten von der Netzwerkschnittstelleneinheit (NIU) an die Verarbeitungseinheit (PU) mit derselben Geschwindigkeit übertragen werden wie Daten, die an der Netzwerkschnittstelleneinheit angekommen sind. Ist diese Anforderung nicht erfüllt, so wird der Strom unterbrochen, was als Fehler zu behandeln ist.

Bei einer Streaming-TC werden der NIU Informationen bereitgestellt, um den direkten Transfer von Daten an die PU zu erlauben. Die Implementierung der MSL, die die NIU und die PU umfasst, ist projektabhängig. Daher müssen die Konfigurationsdaten, die das Streamingverhalten festlegen, implementierungsabhängig sein.

4.2.8 Datendarstellung

Verschiedene Komponenten eines CFM können Daten an Komponenten entfernter CFM senden. Diese Komponenten befinden sich auf der MSL-, OSL- oder AL-Ebene und können Mechanismen zur Datendarstellung umfassen.

Die unterschiedlichen Nachrichtenarten, für die eine Datendarstellung erfolgen kann, sind

- auf MSL-Ebene:
 - TC, die VC-Nachrichten unterstützt,
 - TC, die MLI-Nachrichten unterstützt;
- auf OSL-Ebene:
 - VC, der Anwendungs-, GLI- und SMLI-Nachrichten unterstützt,
 - VC, der OLI-Nachrichten unterstützt;
- auf APOS-Client-Ebene:
 - Anwendungsnachrichten als typisierte Nachrichten,
 - Anwendungsnachrichten als Nachrichten ohne Header.

Die Datendarstellung beeinflusst jede MOAA-Schicht.

4.2.8.1 MSL-Ebene

Die Datendarstellung ist auf MSL-Ebene nicht konfigurierbar. Das bedeutet, dass alle Teile einer Nachricht, die in den Verantwortungsbereich der MSL fallen, festkodiert sind.

- TC, die VC-Nachrichten unterstützt: Die TC-Header-Struktur ist fest und bekannt; die Nutznachricht wird nicht modifiziert und wird an das OS übergeben.
- TC, die MLI-Nachrichten unterstützt: Die Struktur aller MLI-Nachrichten — des Headers und der Nutznachricht — ist bekannt. Die Nutznachricht darf verarbeitet und im erwarteten Format an das OS übergeben werden (d. h. wenn der CFM-Status angefordert wird, muss die Antwort in einem Format erfolgen, die für den Host-Prozessor verständlich ist). Zu weiteren Einzelheiten siehe MLI in 7.4.

4.2.8.2 OSL-Ebene

4.2.8.2.1 Kurzbeschreibung

Die Datendarstellung darf, wenn angefordert, auf den VC-Header angewendet werden.

Die Datendarstellung darf nicht auf Nachrichten ohne Header angewendet werden.

Die Datendarstellung muss, wenn gefordert, auf typisierte Nachrichten angewendet werden.

Die Datendarstellung darf nicht auf VC zwischen Prozessen auf demselben PE angewendet werden.

Die Datendarstellung muss auf einer Teilmenge der OMG-CDR beruhen. Es wird die Big-Endian-Ausrichtung ausgewählt.

Bei typisierten Nachrichten nimmt das OS gemäß der in Blueprints enthaltenen Informationen eine lokale Übersetzung einer Nachricht vor. Es übersetzt die Nachricht:

- auf Sendeseite in das OMG-CDR-Format. Siehe CORB-Architektur (Common Object Request Broker Architecture), 15.3;
- auf Empfängerseite in das für den Host-Prozessor verständliche Format.

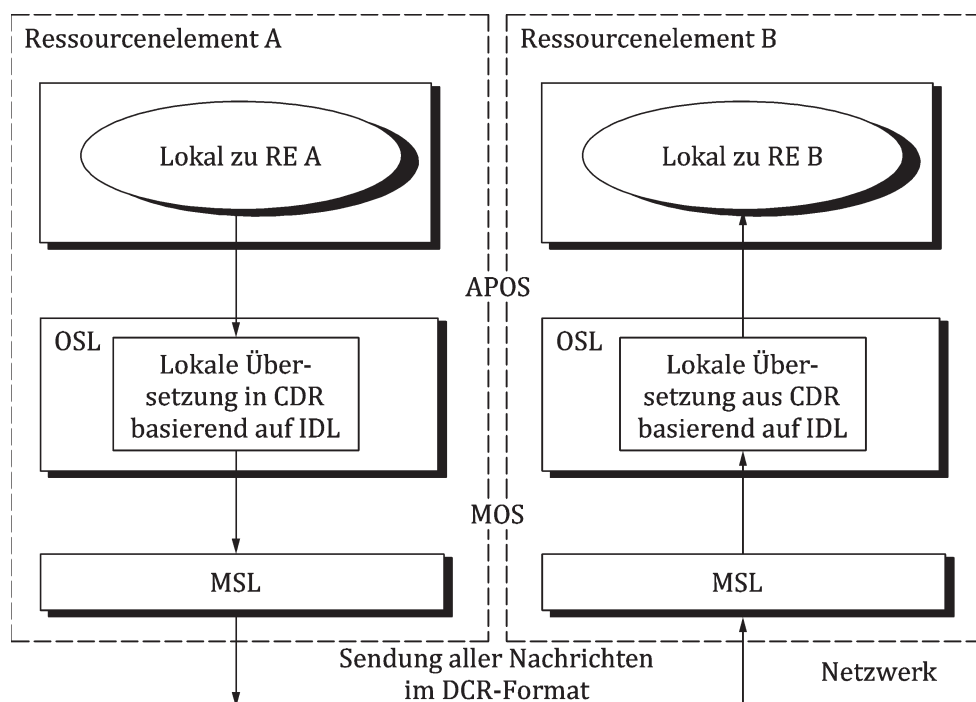


Bild 22 — Datendarstellung

Bild 22 zeigt die Folge der Übersetzungsaktionen, die sich durch eine von Ressourcenelement A gesendete und von Ressourcenelement B empfangene Nachricht ergeben.

4.2.8.2.2 Konfiguration der Datendarstellung

Blueprints enthalten Informationen zur Datendarstellung:

- ein Kennzeichen (Flag) in der SMBP-Struktur *VcToTcMappingDescription* gibt an, ob der Datendarstellungsmechanismus erforderlich ist. Ist dieses Flag gesetzt, so müssen alle VC auf dieser TC verarbeitet werden, andernfalls darf keine Verarbeitung erfolgen. Bei der Multicast-Kommunikation müssen alle Empfänger dieselben Merkmale aufweisen;
- die SMBP-Struktur *VcDescription*:
 - ein Flag gibt an, ob die Nachricht typisiert ist oder nicht;
 - wenn die Nachricht als typisiert festgelegt ist, eine anbieterabhängige Darstellung des Nachrichtendatentyps in der *VcDescription* im Anschluss an die IDL-Nachrichtendefinition.

Das GSM muss die Verarbeitung der Datendarstellung des OS mit den Blueprint-Informationen konfigurieren.

4.2.8.2.3 OLI-Nachrichten

OLI-Nachrichten sind typisierte Nachrichten. Ihre Beschreibung ist im IDL-Format festgelegt. Da sie genormt sind, sind sie in den Blueprints nicht beschrieben.

Zu weiteren Einzelheiten siehe 7.1.

4.2.8.3 APOS-Clients

Die möglichen APOS-Clients sind die Prozesse in der AL oder im GSM. APOS-Clients verwalten keine Datendarstellung und dürfen von derartigen Übersetzungsvorgängen keine Kenntnis haben. APOS-Clients senden über VC typisierte Nachrichten oder Nachrichten ohne Header.

4.2.8.3.1 Nachricht ohne Header

Für Nachrichten ohne Header (Bytestrom) müssen Anwendungsdaten ohne Interpretation gesendet und empfangen werden. In diesem Fall darf der Datendarstellungsmechanismus nicht angewendet werden.

4.2.8.3.2 Typisierte Nachricht

Um die Datendarstellung auf typisierte Nachrichten anzuwenden, müssen beim Aufbau von APOS-Clients einige Einschränkungen beachtet werden:

- Die Definitionen der Nachrichtendatentypen basieren auf einer Teilmenge der IDL-Typen.
- Die Programmiersprachen und Compiler-Optionen für den Aufbau der APOS-Client-Prozesse und der OS-Komponenten müssen konform sein, um die standardisierte Speicheranordnung erzeugen zu können.

4.2.8.3.2.1 Definition typisierter Nachrichten

Die Definition der Nachrichtendatentypen muss auf einer IDL-Teilmenge basieren und Folgendes umfassen:

- einfache Typen;
- zusammengesetzte Typen.

In Tabelle 4 ist die Teilmenge primitiver IDL-Typen festgelegt, die von einfachen Typen oder zusammengesetzten Typen verwendet werden.

Tabelle 4 — Primitive IDL-Typen

IDL-Typ	Oktettanordnung	CDR-Darstellung	
octet	1	Nicht interpretierte 8-Bit-Darstellung	
short unsigned short	2	Oktett	Big-Endian
		0	MSB
		1	LSB
long unsigned long	4	Oktett	Big-Endian
		0	MSB
		1	
		2	
		3	LSB
long long unsigned long long	8	Oktett	Big-Endian
		0	MSB
		1	
		2	
		3	
		4	
		5	
		6	
		7	LSB
float	4	IEEE 754-1985	
		Oktett	Big-Endian
		0	s e1
		1	e2 f1
		2	f2
		3	f3
double	8	IEEE 754-1984	
		Oktett	Big-Endian
		0	s e1
		1	e2 f1
		2	f2
		3	f3
		4	f4
		5	f5
		6	f6
		7	f7

IDL-Typ	Oktettanordnung	CDR-Darstellung		
long double	16	IEEE 754-1984		
		Oktett	Big-Endian	
		0	s	e1
		1	e2	
		2	f1	
		3	f2	
		4	f3	
		5	f4	
		6	f5	
		7	f6	
		8	f7	
		9	f8	
		10	f9	
		11	f10	
		12	f11	
		13	f12	
		14	f13	
		15	f14	

In Tabelle 5 sind die zusammengesetzten Typen festgelegt. Der APOS-Client muss Typen der Spalte „Zusammengesetzter IDL-Typ“ verwenden. Das OS stellt die Datendarstellung sicher und muss diese entsprechend der Spalte „CDR-Kodierung“ kodieren.

Tabelle 5 — Zusammengesetzte IDL-Typen

Zusammen- gesetzter IDL-Typ	CDR-Kodierung
<code>struct</code>	Komponenten in Definitionsreihenfolge kodiert. Eine Struktur darf aus primitiven Typen und zusammengesetzten Typen aufgebaut sein.
<code>array</code>	Kodierte Array-Elemente in Folge. Das kodierte Element darf entweder ein primitiver Typ oder ein zusammengesetzter Typ sein. Ein Array muss einem eindeutigen Elementtyp zugeordnet sein.
<code>sequence</code>	Anzahl der Elemente, gefolgt von den kodierten Sequenz-Elementen in Folge. Die Anzahl der Elemente muss ein einfacher vorzeichenloser Integertyp sein. Das kodierte Element darf entweder ein primitiver Typ oder ein zusammengesetzter Typ sein. Eine Sequenz (sequence) muss einem eindeutigen Elementtyp zugeordnet sein.
<code>string</code>	Länge des Strings, gefolgt von der Folge von Zeichen, die in Einzelbyte-Form kodiert sind, einschließlich eines Null-Zeichens. Die Länge des Strings muss ein einfacher vorzeichenloser Integertyp sein. Ein Leerstring hat die Länge eins. Die Länge des Strings muss ein einfacher Integertyp sein. Die kodierten Elemente müssen ein Oktett sein.
<code>union</code>	Diskriminanz-Tag, gefolgt von der Darstellung des gewählten Mitglieds. Das Diskriminanz-Tag muss ein einfacher vorzeichenloser Integertyp sein. Das gewählte Mitglied darf entweder ein primitiver Typ oder ein zusammengesetzter Typ sein.

Es ist möglich, mehrere zusammengesetzte IDL-Typen mit entweder fester oder variabler Größe zusammenzuführen, solange sie in einer Struktur gekapselt sind und die Zusatzinformation, die durch die CDR-Kodierung ergänzt wird, ebenfalls ein Element der Struktur ist. In der Datendarstellungsspezifikation der Nachricht muss diese Zusatzinformation ein Schlüsselwort in Abhängigkeit vom jeweiligen zusammengesetzten IDL-Typ sein.

- Sequenz (sequence): Element-Anzahl muss Teil der Struktur sein, festgelegt durch ein Schlüsselwort;
- String (string): Länge des Strings muss Teil der Struktur sein, festgelegt durch ein Schlüsselwort;
- Vereinigung (union): Diskriminanz-Tag muss Teil der Struktur sein, festgelegt durch ein Schlüsselwort.

4.3 Sicherheitsmanagement

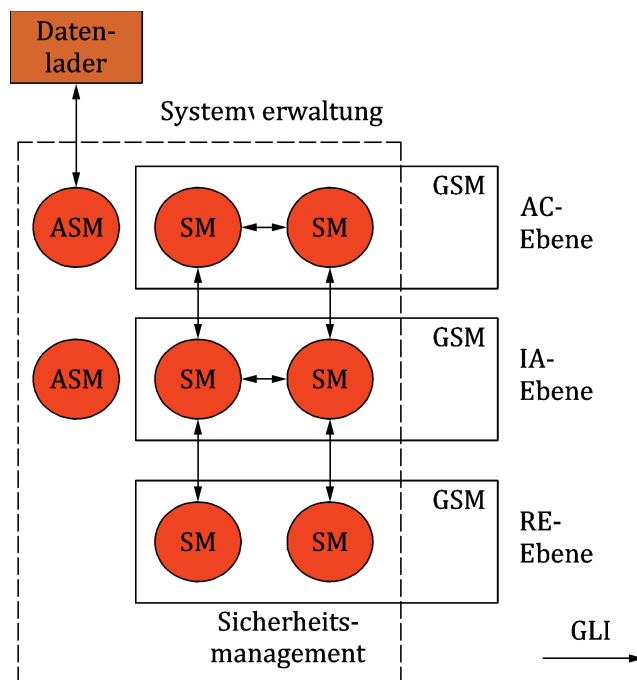


Bild 23 — GSM-Schnittstellen

Das Sicherheitsmanagement ist die GSM-Funktion, die für den sicheren Transfer von als zu schützend gekennzeichneten Daten durch das System verantwortlich ist. Die Hauptaufgaben des Sicherheitsmanagements sind

- die Verwaltung und Verteilung von Schlüsseln zur Authentifizierung und Verschlüsselung nach Bedarf,
- die Bereitstellung von Klassifizierungs- und Freigabedaten für die Bezugsüberwachungsfunktion,
- die Implementierung von Authentifizierungs- und Softwareverschlüsselungsalgorithmen und
- die Erstellung eines Sicherheitsauditprotokolls — Protokollierung und Reihung.

Die Sicherheitsmanagementfunktion ist implementiert in Form eines plattformspezifischen Teils, dem Anwendungssicherheitsmanagement (als Teil des AM), und der Hierarchie des Sicherheitsmanagements des generischen Systems.

4.3.1 Anwendungssicherheitsmanagement

Das Anwendungssicherheitsmanagement ist der Teil der Sicherheitsmanagementfunktion, der speziell für eine bestimmte Plattform bereitgestellt wird und sich daher über der APOS neben dem Anwendungsmanager befindet.

Es erfüllt zwei Zwecke:

- Es dient als Schnittstelle zwischen dem Datenlader und dem übrigen IMA-System;
- es dient als Schnittstelle zwischen der Datenbank, die die als zu schützend gekennzeichneten Daten enthält, und den Funktionsanwendungen, die diese als zu schützend gekennzeichneten Daten verarbeiten sollen.

ANMERKUNG Der Datenlader und dessen Schnittstelle befinden sich außerhalb des MOAA-Kerns.

4.3.2 Generisches Sicherheitsmanagement

Das GSM-SM ist der Teil der Sicherheitsmanagementfunktion, der für alle Plattformen gilt.

Es erfüllt zwei Zwecke:

- es ermöglicht die sichere Übertragung von als zu schützend gekennzeichneten Daten durch Anwendung von Verschlüsselung/Entschlüsselung und Authentifizierung;
- es schreibt erkannte Versuche der Sicherheitsverletzung in das Sicherheitsprotokoll.

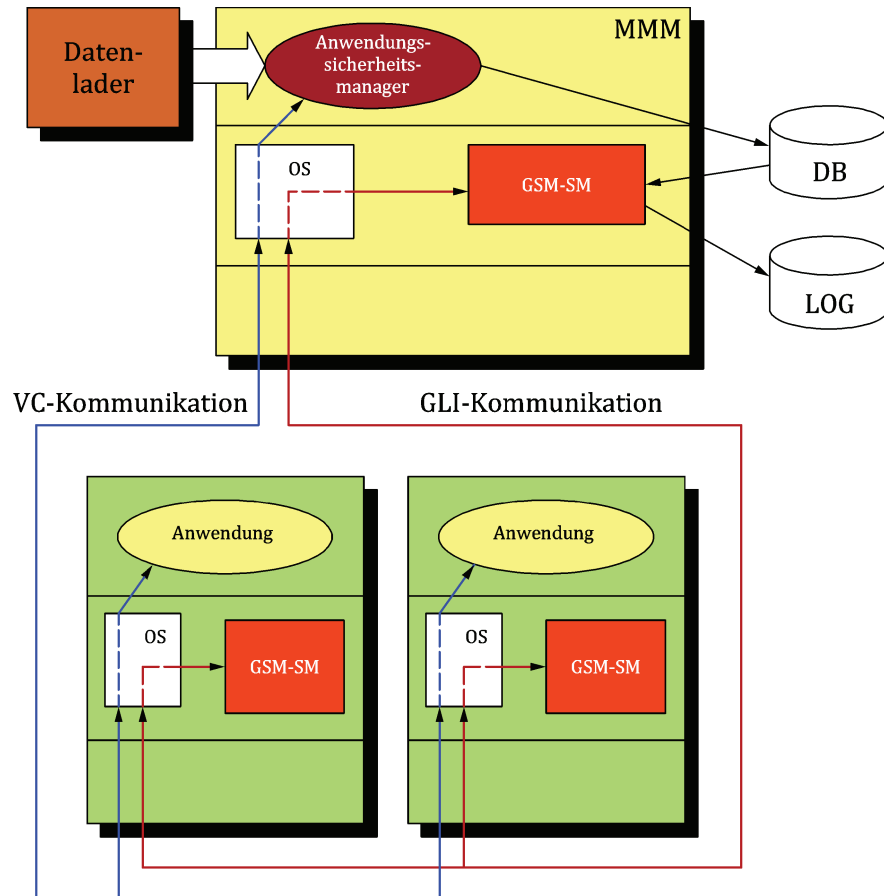


Bild 24 — Wesentliche sicherheitsbezogene Architekturkomponenten

4.3.3 Verschlüsselung/Entschlüsselung und Authentifizierung

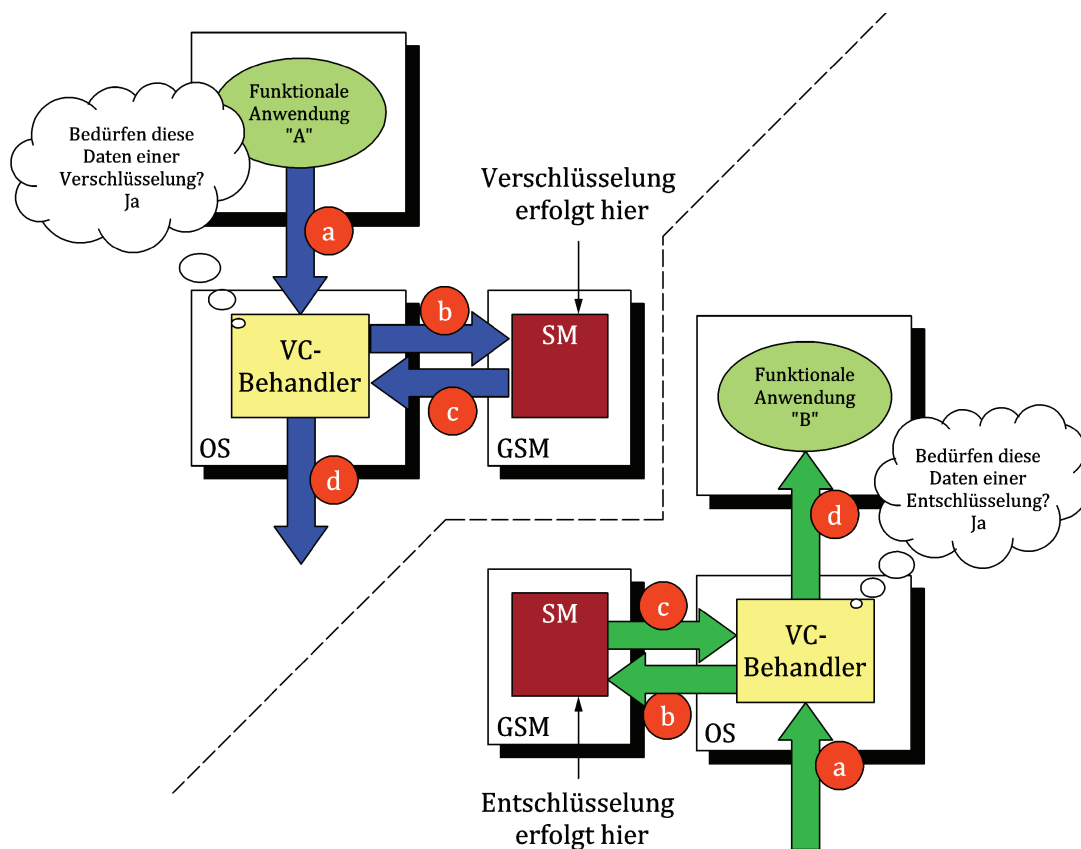


Bild 25 — VC überträgt Daten, die einer Ver- und Entschlüsselung bedürfen

Das OS fängt übertragene oder empfangene Daten ab und ermittelt mit Hilfe einer Lookup-Tabelle, ob die übertragenen Daten als zu schützend gekennzeichnet sind. Ist dies der Fall, so ist das OS dafür verantwortlich, diese Daten an das GSM-SM zu senden, wobei es die Daten und die VC-ID weitergibt, sodass auf der Grundlage von Blueprint-Informationen an den Daten entsprechende Aktionen vorgenommen werden können, z. B.: Verschlüsselung, Entschlüsselung und Authentifizierung.

Der Satz der als zu schützend gekennzeichneten Nachrichten hängt von der Betriebsart des Systems ab und ist mittels Blueprints definiert. Als Teil des Initialisierungs-/Konfigurationsprozesses durchläuft das GSM-SM, das auf RE resident ist, welche als zu schützend gekennzeichnete Daten verarbeiten, eine Menge vordefinierter Aktionen. Die abschließende Aktion des GSM-SM in diesem Betriebsartänderungsprozess muss darin bestehen, den SMOS-Dienst *getPMDData* aufzurufen.

Dabei muss es sich um einen blockierenden Aufruf handeln, der nur dann behandelt wird, wenn das OS die Ausführung einer sicherheitsbezogenen Aktion an als zu schützend gekennzeichneten Daten fordert. Dieser Aufruf gibt die ID des VC, der die betreffenden Daten überträgt, sowie die übertragenen Daten aus. Das GSM-SM verwendet dann die VC-ID als Eintrag in einer Lookup-Tabelle, die während der Konfigurationsphase erzeugt wurde, um zu bestimmen, welche Aktion an den Daten auszuführen und welcher Schlüssel zu verwenden ist.

Die verarbeiteten Daten werden daraufhin mit Hilfe des SMOS-Dienstes *returnPMDData* an die OS-VC-Dienste ausgegeben.

4.3.4 Sicherheitsaudit

Das Sicherheitsaudit erfüllt zwei Zwecke:

- Verlaufsaufzeichnung aller sicherheitsbezogenen Funktionen, um eine Akkreditierung zu unterstützen (auswählbar);
- Protokollierung von versuchten Sicherheitsverletzungen.

Hauptzweck des Sicherheitsaudits ist die Protokollierung von versuchten Sicherheitsverletzungen während des Betriebs des Luftfahrzeugs.

Diese sind dem GSM-SM mit Hilfe des SMOS-Dienstes *getAuditData* zu melden.

- Das OS muss für die Meldung von Verletzungen der Sicherheitsvorgaben verantwortlich sein.
- Die Referenzüberwachung muss für die Erfassung von Verletzungen der Mehrebenen-Sicherheitsvorgaben verantwortlich sein.
- Das GSM-SM selbst ist dafür verantwortlich, ein Fehlschlagen der Authentifizierung zu erkennen.

4.3.5 Sicherheitsreferenzüberwachung

Die Funktion der Sicherheitsreferenzüberwachung ist, die Implementierung der Sicherheitsvorgaben durchzusetzen und versuchte Verletzungen der Vorgaben zu erkennen. Die Hauptaufgaben der Referenzüberwachung sind

- die Implementierung eines Algorithmus zur Laufzeit-Freigabe/Klassifizierungsvalidierung (z. B. Bell-LaPadula), d. h. Überprüfen (en: checking),
- welcher Prozess-Thread welchen Dienst aufruft und
- der zwischen dem Prozess-Thread und dem Dienst übertragenen Daten.

4.4 Modulverwaltung

Die Modulverwaltung darf in zwei Teile aufgeteilt werden:

- aus interner Sicht: den Teil, bei dem das CFM hochgefahren wird und dann in einen Zustand übergeht, in dem es bereit ist, den gesamten TLS herunterzuladen;
- aus externer Sicht: den Teil, bei dem ein übergeordnetes CFM die Initialisierung eines untergeordneten CFM steuert.

Die logische Modulschnittstelle (MLI) unterstützt diese beiden Aspekte der Modulverwaltung. Die MLI muss den Zugriff auf ein CFM erlauben, nachdem dieses eingeschaltet und über eine definierte Netzwerkroute initialisiert wurde. Sie muss folgende Funktionen aufweisen:

- Bereitstellung des CFM-PBIT (Selbsttest beim Einschalten);
- Abrufen des aktuellen CFM-Status;
- Abrufen der CFM-Merkmale;
- Herunterladen des MOAA-Softwarestacks auf ein CFM;
- Konfigurieren eines CFM (für NSM und PCM);

- Synchronisieren der CFM-Zeit;
- Senden und Empfangen von Prüfnachrichten an ein und von einem CFM.

Das MLI-Protokoll muss für alle CFM-Typen gleich sein. Die MLI erlaubt die Initialisierung, Konfiguration und Synchronisation der CFMs in einem IMA-Kernsystem.

Die MLI-Dienste sind in 7.4 festgelegt.

4.5 Massenspeicherverwaltung

4.5.1 Überblick

Das MMM stellt eine Massenspeicherfähigkeit bereit. Es stellt den übrigen CFMs die Dateizugriffs- und Speicherfunktionen in einem MOAA-System bereit.

Das MMM darf, je nach Implementierung, mehrere Dateiarten enthalten, z. B.:

- Abbilder, die die Komponenten des TLS enthalten: MSL, OS, GSM und Blueprints;
- Softwareanwendungen;
- Datenkonfiguration für das Netzwerk;
- Datenkonfiguration für die Leistungsschalter;
- Datenkonfiguration für die CFM-Kommunikation zum Zeitpunkt des Einschaltens (Routing-Tabelle);
- Daten für die Anwendung;
- sicherheitsbezogene Daten (z. B. Verschlüsselungsschlüssel).

Das MMM stellt für CFMs, die auf diese Dateien zugreifen müssen, Server bereit. Welche Schicht den Server enthält, hängt davon ab, welche Schicht den Requester enthält.

Der direkte Dateizugriff erfolgt nur auf der MMM. Die übrigen CFMs nutzen die OLI zum Fernzugriff auf die Datei.

Zu beachten ist, dass die Massenspeicherverwaltung keine Dateizugriffsrechte für entfernte CFMs unterstützt. Diese sind Teil der Sicherheitsmanagementvorgaben. Die Grundlage bildet eine sichere Kommunikation und mögliche Verschlüsselung.

4.5.2 Lokale Dateiverwaltung der MMM

Die den Dateizugriff betreffenden Dienste sind spezifisch für die MMM. Außer auf der MMM darf auf einem CFM keine lokale Dateiverwaltung stattfinden.

In der APOS muss ein für die MMM spezifischer Satz von Dateihandhabungsdiensten vorhanden sein. In der MOS muss ein für die MMM spezifischer Satz von Datenspeicherhandhabungen vorhanden sein.

Die APOS-Dateihandhabung stellt einen Satz von Diensten bereit, die einen Mehrfachzugriff ermöglichen. Das bedeutet, dass mehrere Benutzer in der Lage sein müssen, eine Datei zu verarbeiten (unterschiedliche Prozesse bzw. unterschiedliche Threads innerhalb eines Prozesses). Die unterschiedlichen Benutzer müssen in der Lage sein, gleichzeitige Operationen auszuführen. Nur die Dateizugriffsrechte dürfen Zugriffsbeschränkungen auferlegen.

4.5.3 Zugriff auf Anwendungsdateien

Für den Zugriff auf Anwendungsdateien muss das tftp-Protokoll angewendet werden, wie in der OLI-Definition in 7.1 beschrieben. Das erfordert die Verfügbarkeit eines entsprechenden APOS-Dienstes.

4.5.4 CFM-Download

Beim Starten verlässt sich ein IMA-Kernsystem darauf, dass ein CFM anfangs eigenständig bootet. Alle anschließend initialisierten weiteren CFMs besitzen nur eine residente MSL. Die MSL-Software wird lokal gebootet, woran ein weiterer, lokaler Boot-Lader beteiligt sein kann. Ein CFM mit einem vollständigen Softwarestapel stellt die GSM-Steuerung der Module bereit. Diese Funktion sorgt für das Booten der übrigen CFMs.

Software- oder Konfigurationsinformationen können von einem CFM/PE mit einem vollständigen TLS auf ein entferntes PE oder eine entfernte MSU mit nur einer residenten MSL geladen werden.

- Abbilder, die die Komponenten des TLS enthalten: OS, GSM und Blueprints;
- Datenkonfiguration für das Netzwerk;
- Datenkonfiguration für die CFM-Kommunikation zum Zeitpunkt des Einschaltens.

Der Kommunikationsmechanismus dafür ist die logische Modulschnittstelle (MLI) auf der Basis von TC.

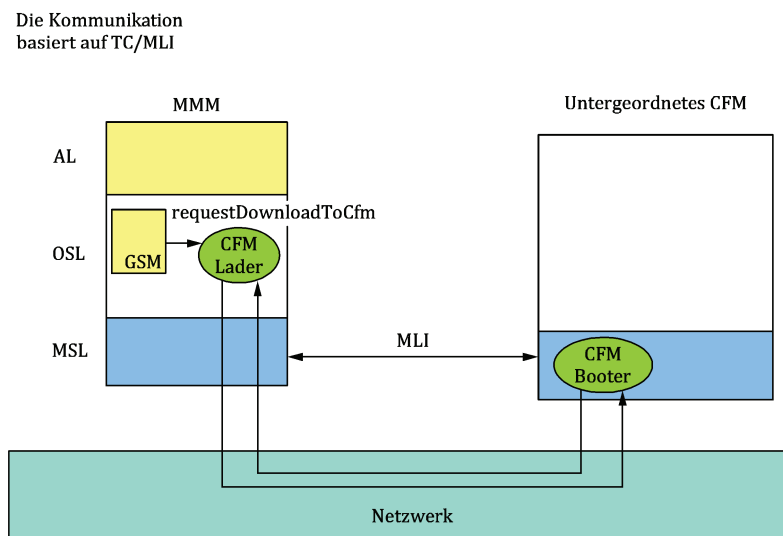


Bild 26 — MMM-Download auf ein CFM, das nur die MSL besitzt

Der MMM ist die Verantwortung für die GSM-Modulsteuerung zugewiesen. Die TC zwischen der MMM und einem Satz gesteuerter CFM sind in den Blueprints definiert und wurden bei der MMM konfiguriert. Aufseiten des zu bootenden CFM lautet die für die Download-TC geltende Download-Sequenz:

- Die TC auf der MMM sind im Blueprint definiert und wurden konfiguriert.
- Die GSM schaltet den Strom für ein untergeordnetes CFM ein.
- Das untergeordnete CFM initialisiert, bootet und führt den PBIT aus.
- Falls das PBIT-Ergebnis nicht zufriedenstellend ausfällt, schweigt das nachgeordnete CFM.

- Wurde der PBIT bestanden, so muss das nachgeordnete CFM einen tftp-Transfer von der MMM über zuvor festgelegte TCs initiieren. Die TC-Bezeichner müssen die vom tftp-Protokoll definierten Portnummern widerspiegeln.
- In Reaktion auf diese Anfrage wird das tftp-Protokoll zwischen dem nachgeordneten CFM (Client) und der zugehörigen MMM (Server) ausgeführt. Die verwendeten tftp-Nachrichten sind in einem TC-Nachrichtenframe gekapselt, der in der Lage sein muss, eine zuvor aufgebaute Verbindung zum Server zu nutzen. Das angewendete tftp-Protokoll wird durch die folgenden Standards definiert:
 - RFC 1350:1992, *Trivial File Transfer Protocol (Revision 2)*;
 - RFC 2347:1998, *TFTP Option Extension*;
 - RFC 2348:1998, *TFTP block size Option*;
 - RFC 2349:1998, *TFTP Timeout Interval and Transfer Size Option*.

Bei der Initialisierung hat das nachgeordnete CFM die TCs definiert, die zur Ausführung des Protokolls erforderlich sind.

Als erstes ist es erforderlich, die Netzwerkkonfiguration auf das NSM herunterzuladen. Dies ist eine Voraussetzung, um alle anderen CFMs erreichen zu können.

4.5.5 Anwendungsdownload

Bevor ein Anwendungsprozess heruntergeladen werden kann, muss der TLS auf einem CFM/PE laufen. Die GSM/das CFM führt einen SMOS-Dienst zur Prozesserzeugung aus. Vom OS/CFM wird auf dem VC eine OLI-Anfrage an das OS/die MMM gesendet.

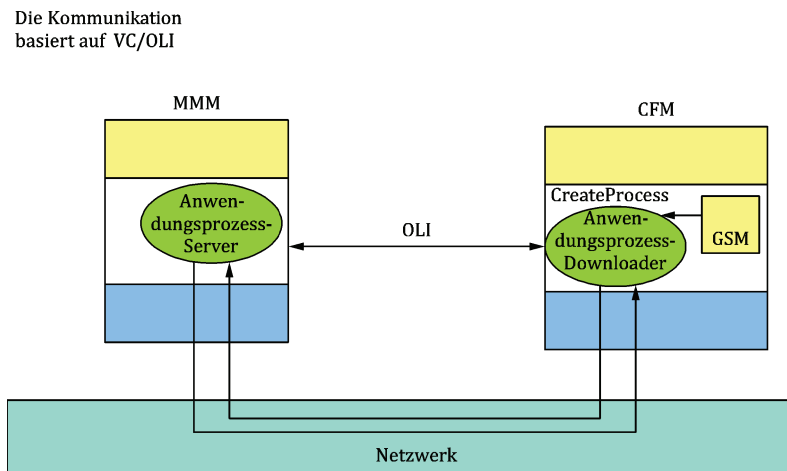


Bild 27 — Anwendungsdownload

Es wird folgende Sequenz ausgeführt:

- die VCs zwischen der MMM und dem CFM sind in den Blueprints definiert und wurden konfiguriert;
- beim Lesen der Blueprints initiiert das CFM/die GSM eine Prozesserzeugung;
- das CFM/OS sendet eine OLI-Anfrage unter Verwendung eines VC auf der Basis eines Protokolls, das mit dem OLI-Dateilesen kompatibel ist;

- die MMM/das OS empfängt die OLI-Anfrage und greift auf die zu sendende Datei zu;
- die MMM/das OS sendet auf einem VC eine OLI-Antwort mit den erforderlichen Daten an das CFM/OS. Die VC-ID der OLI-Antwort wird nach einer von der OS-Implementierung abhängigen Regel berechnet.

Die Dateianwendung wird von dem CFM/OS abgerufen, wodurch der Anwendungsprozess erzeugt wird.

4.6 Graphikverwaltung

Das MOAA-Graphikkonzept stellt eine Systemgraphikfähigkeit für das Steuern der Cockpit- und Instandhaltungspanelanzeigen bereit. Das Konzept maximiert das Potential der Portabilität und Wiederverwendung von Software und verwendet, wo möglich, die Standardkomponenten der MOAA-Architektur (z. B. APOS, VC usw.).

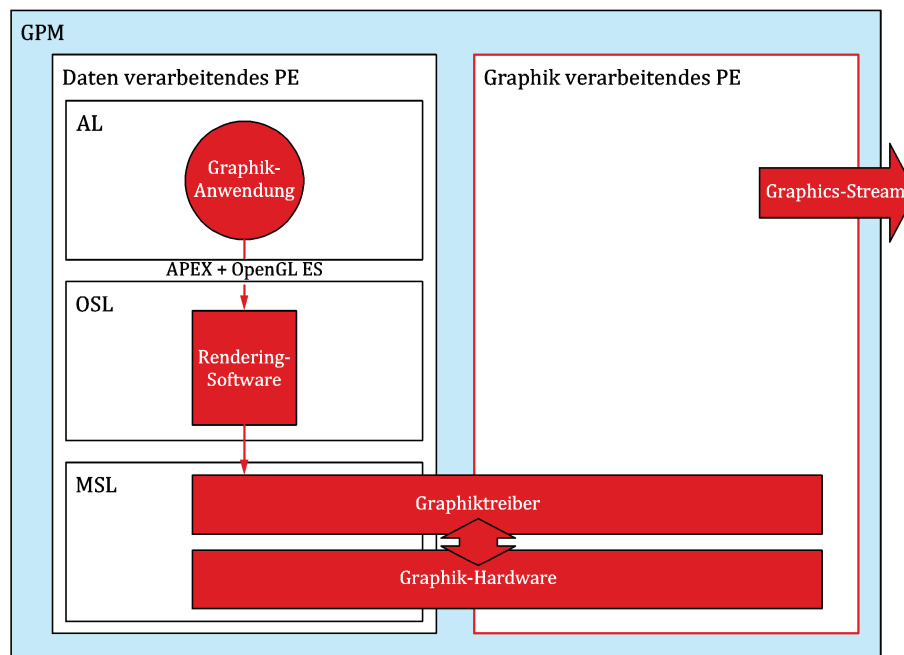


Bild 28 — MOAA-Graphikarchitektur

Die IMA-Graphikarchitektur wird in Bild 28 veranschaulicht. Üblicherweise erfordert ein GPM mindestens ein PE zur Verarbeitung von Daten und mindestens ein PE zur Verarbeitung von Graphik. Die Erzeugung und Kommunikation von Graphik müssen im Daten verarbeitenden Teil und das Graphik-Rendering und die Bilderzeugung im Graphik verarbeitenden Teil erfolgen. Die Graphikerzeugung führt zu einem verschlüsselten Graphik-Stream, der über das Kernnetzwerk übertragen wird. Das Verschlüsselungsformat ist nicht genormt, weil dieser Stream üblicherweise an Anzeigegeräte außerhalb des Kerns gesendet wird, die ebenfalls nicht genormt sind.

Als API für die Graphikerzeugung wurde OpenGL ES als der De-facto-Standard gewählt.

4.7 Energieverwaltung

Ein PCM besitzt die Fähigkeit, den Strom von anderen CFMs ein- und auszuschalten, wodurch ein kontrolliertes Ein- und Ausschalten der CFMs ermöglicht wird. Der vorliegende Abschnitt behandelt die Dienste, die zur Steuerung der Leistungsschalter notwendig sind.

Das Energieverwaltungskonzept des IMA-Kernsystems bietet drei mögliche Lösungen dafür, wie diese Schalter verwaltet werden könnten. Die von einem Programm gewählte Lösung bleibt ganz dem Systemdesigner überlassen. Es handelt sich um die Optionen:

- anwendungsgesteuert;
- GSM-gesteuert;
- MLI-gesteuert.

Die PCM-MSL enthält „Leistungsschaltdienste“ (verfügbar an den Schnittstellen APOS und MOS), die ihr ermöglichen, eine Bank von Leistungsschaltern zu steuern; diese Schalter sind für das Ein- bzw. Ausschalten der Stromversorgung der übrigen CFMs im Gestell verantwortlich.

Tabelle 6 — Leistungsschaltdienste

<i>setPowerSwitch</i>	Stromversorgung eines Moduls ein-/ausschalten
<i>resetPowerSwitch</i>	Leistungsschalter zurücksetzen (alle Module ausschalten)
<i>getPowerSwitchStatus</i>	Den Status des Leistungsschalters beziehen

Diese Dienste gelten nur für die anwendungsgesteuerten oder GSM-gesteuerten Lösungen.

4.7.1 GSM-gesteuerte Lösung

Wie bereits im vorangegangenen Abschnitt erwähnt, kann der Datenprozessor (DP) auf dem PCM einen TLS hosten; dadurch verfügt die OSL über eine umfassende GSM-Funktionalität.

Die auf einem PCM residente GSM der RE-Ebene ist eine GSM wie jede andere. Folglich nimmt sie ihren Platz in der Systemverwaltungshierarchie ein und wird, wie alle übrigen RE-GSM, von einer übergeordneten GSM gesteuert.

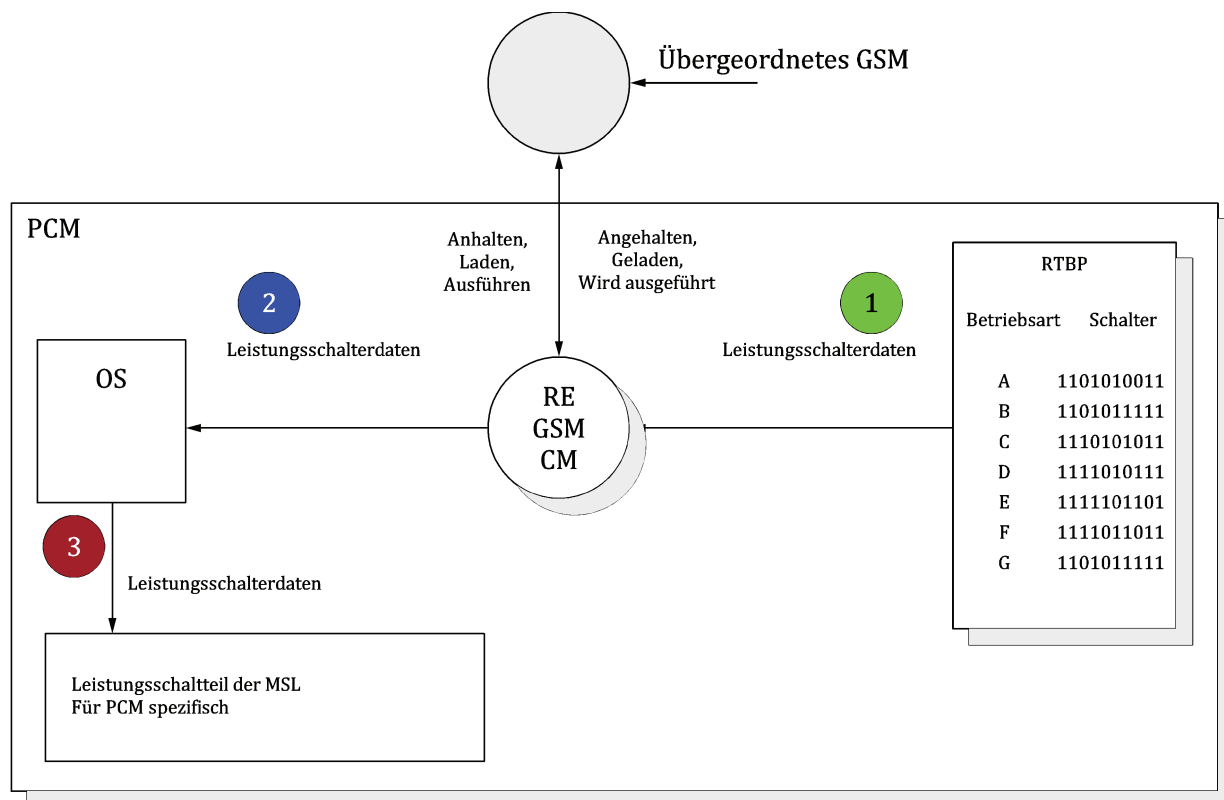


Bild 29 — PCM-Verwaltung (Beispiel)