



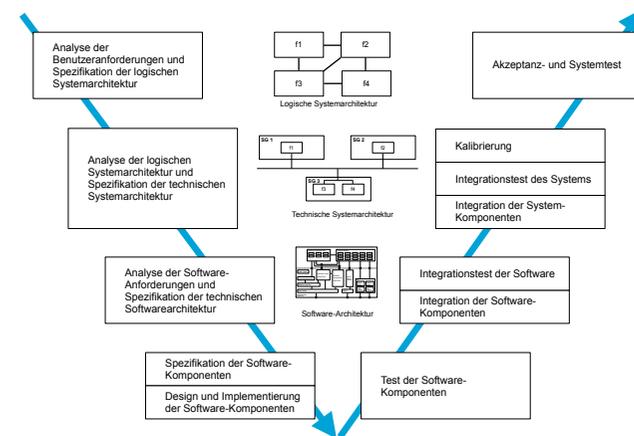
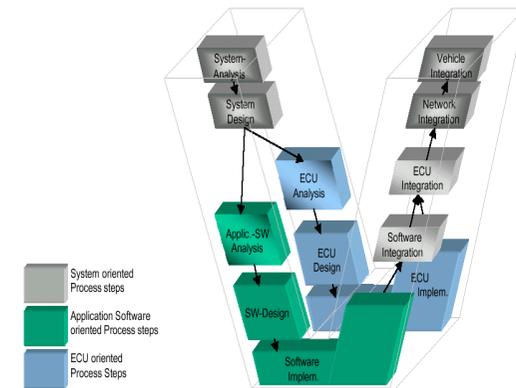
TECHNISCHE
UNIVERSITÄT
DRESDEN

Vorlesung
Automotive Software Engineering
Teil 6 SW-Entwicklung (2-1)
Sommersemester 2015

Prof. Dr. rer. nat. Bernhard Hohlfeld
Bernhard.Hohlfeld@mailbox.tu-dresden.de
Technische Universität Dresden, Fakultät Informatik
Honorarprofessur Automotive Software Engineering

Vorlesung Automotive Software Engineering

Motivation und Überblick		
Beispiele aus der Praxis	SW-Entwicklung	Normen und Standards
	E/E-Entwicklung	
	Das Automobil	
	Die Automobilherstellung	
	Die Automobilbranche	



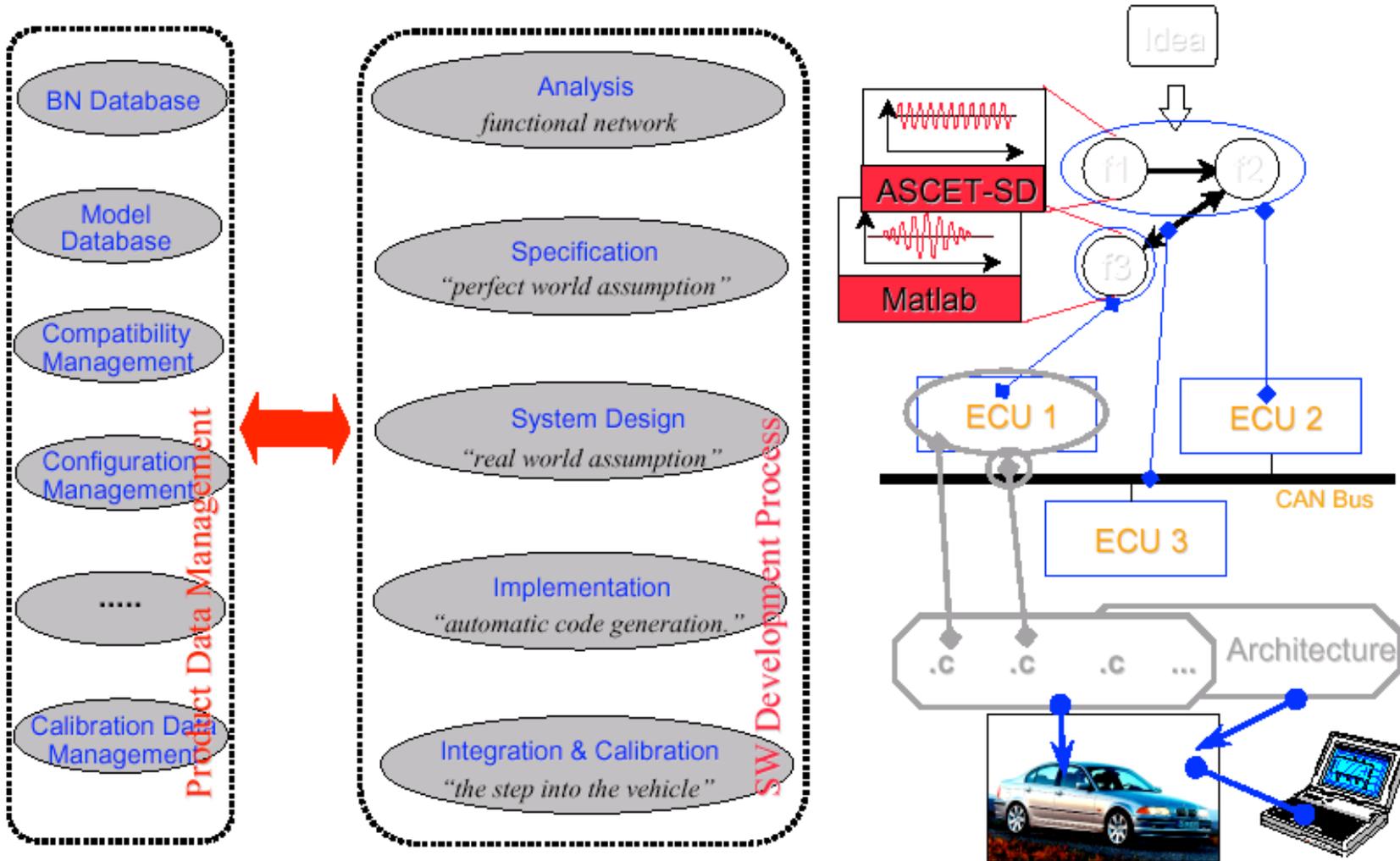
Lernziele SW-Entwicklung

- Vorgehensmodelle kennen
- Grundbegriffe des Systems Engineering kennen
- Den Kernprozess zur Entwicklung von elektronischen Systemen und Software im Fahrzeug verstehen
- Das V-Modell zur Software-Entwicklung in einer speziellen Ausprägung mit zahlreichen Beispielen kennen
- Die Unterstützungsprozesse zur Entwicklung von elektronischen Systemen und Software im Fahrzeug kennen

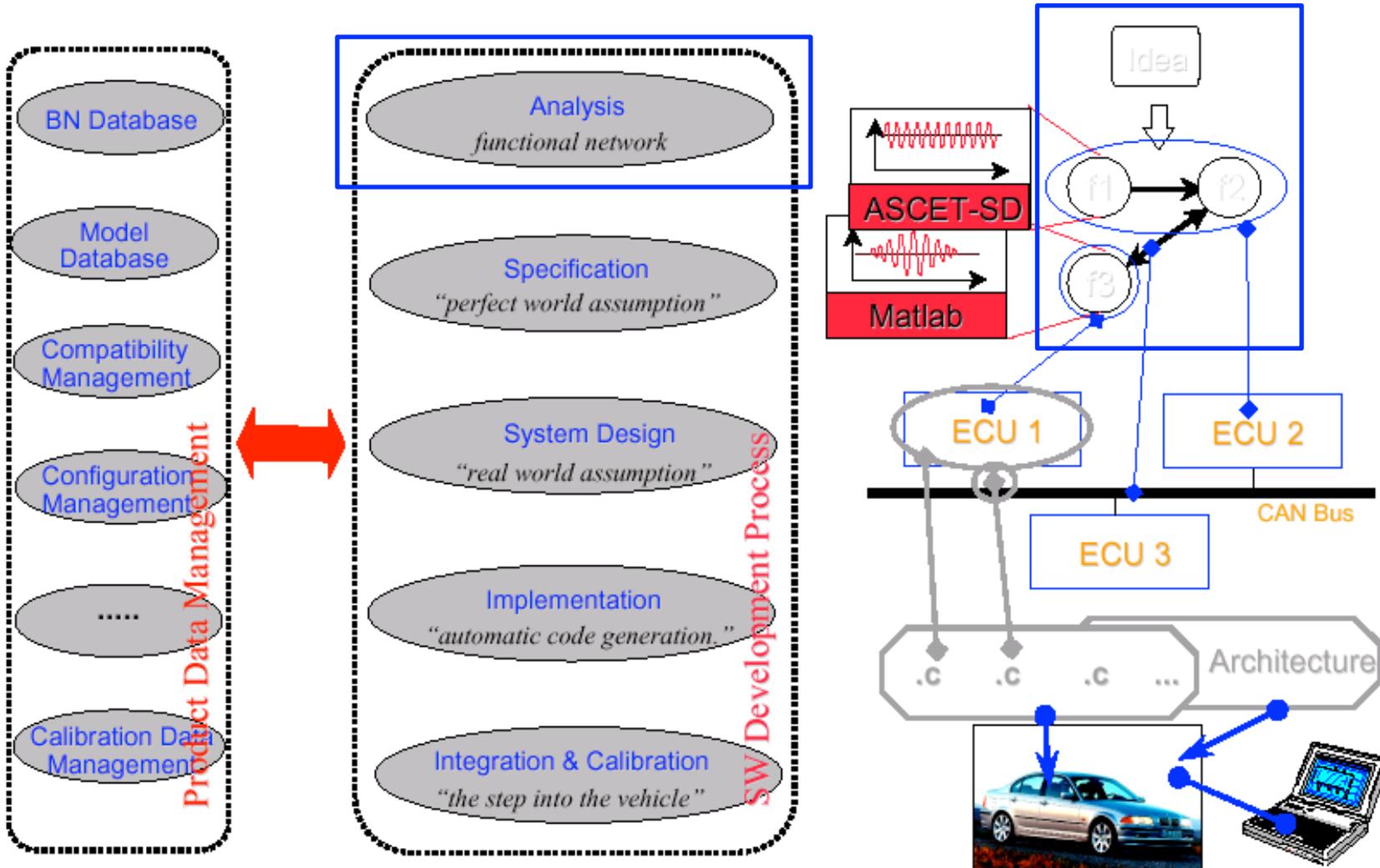
6. SW-Entwicklung

1. Vorgehensmodelle
2. **Kernprozess**
3. Unterstützungsprozesse

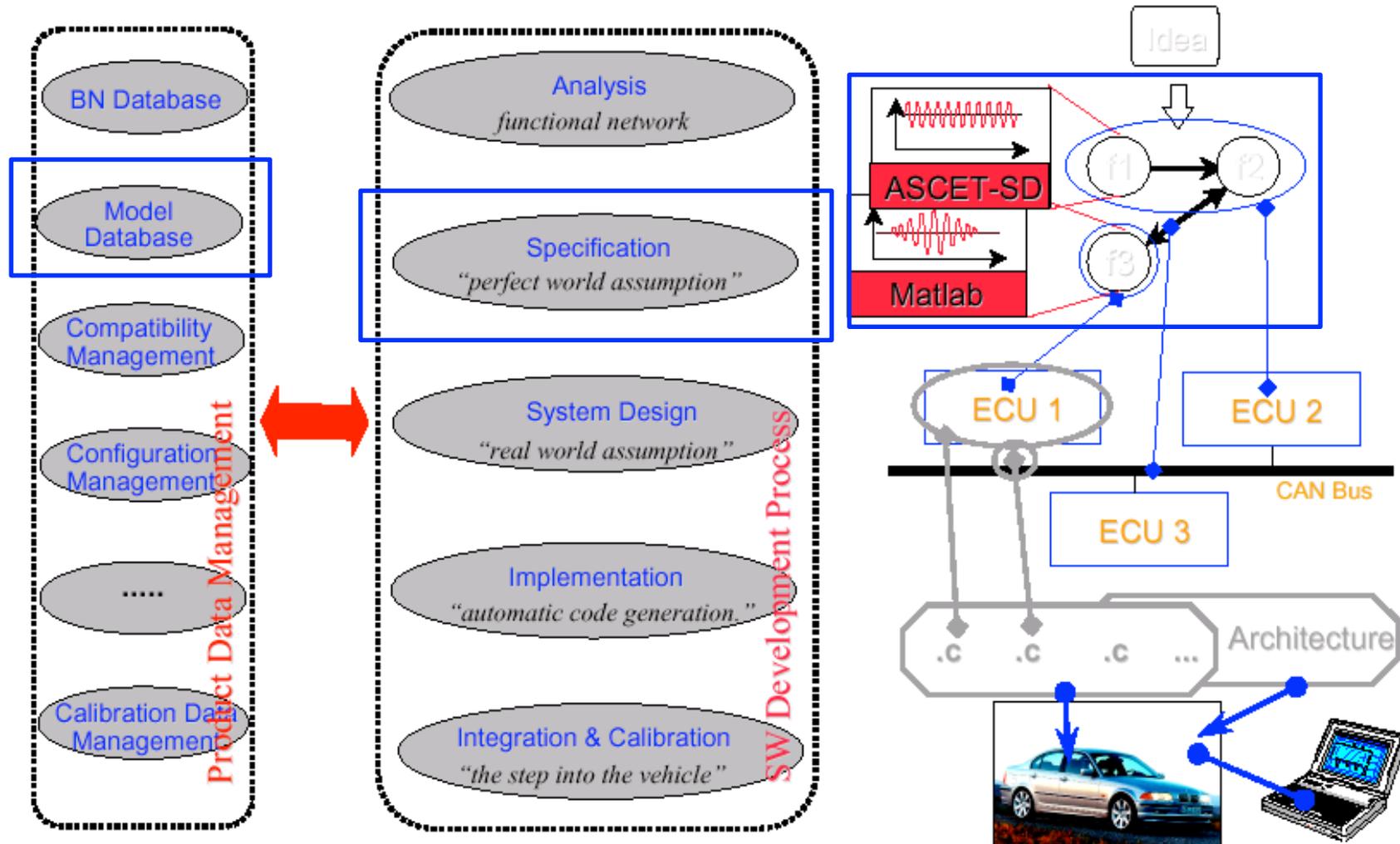
Software Entwicklungsprozess



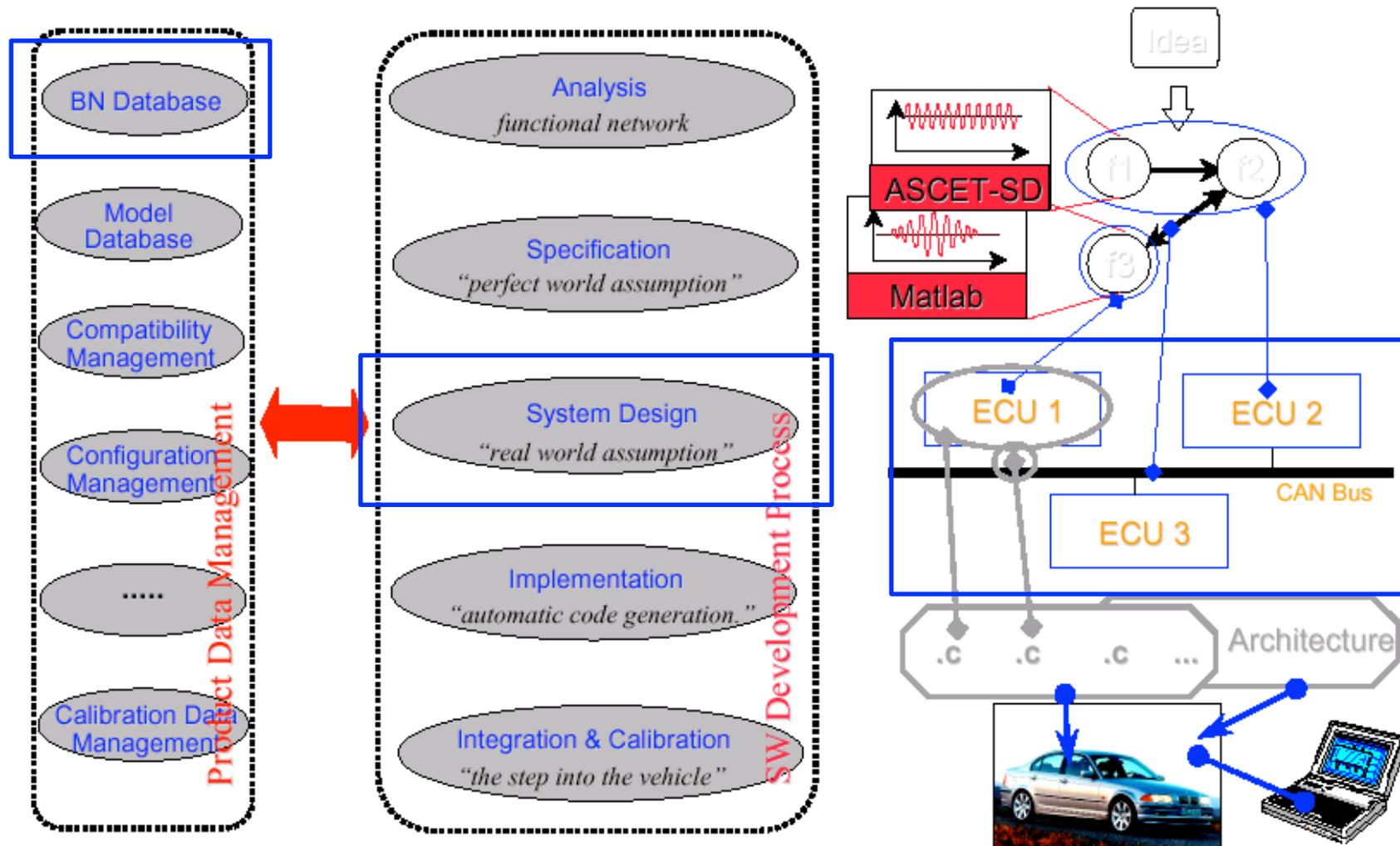
Software Entwicklungsprozess



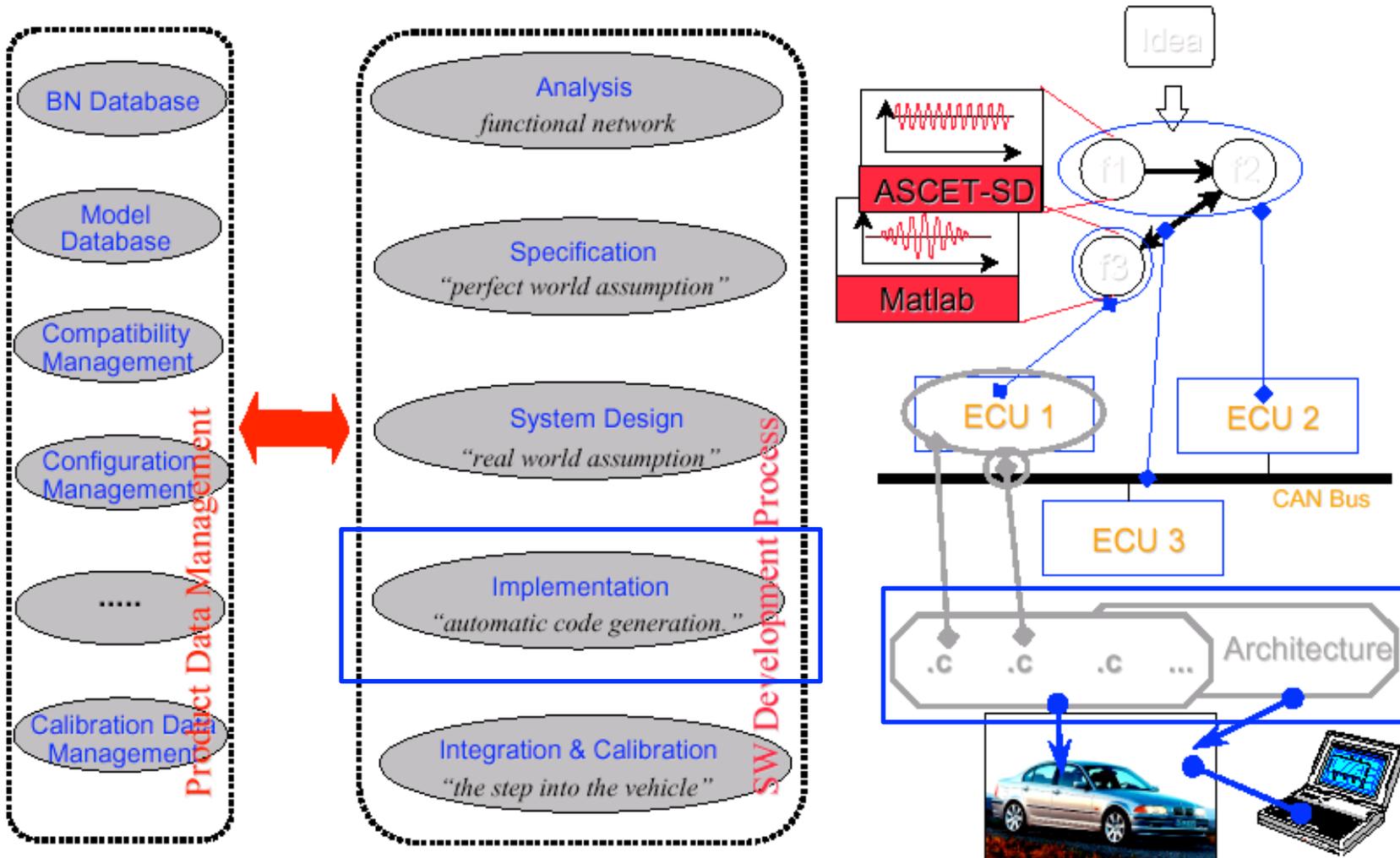
Software Entwicklungsprozess



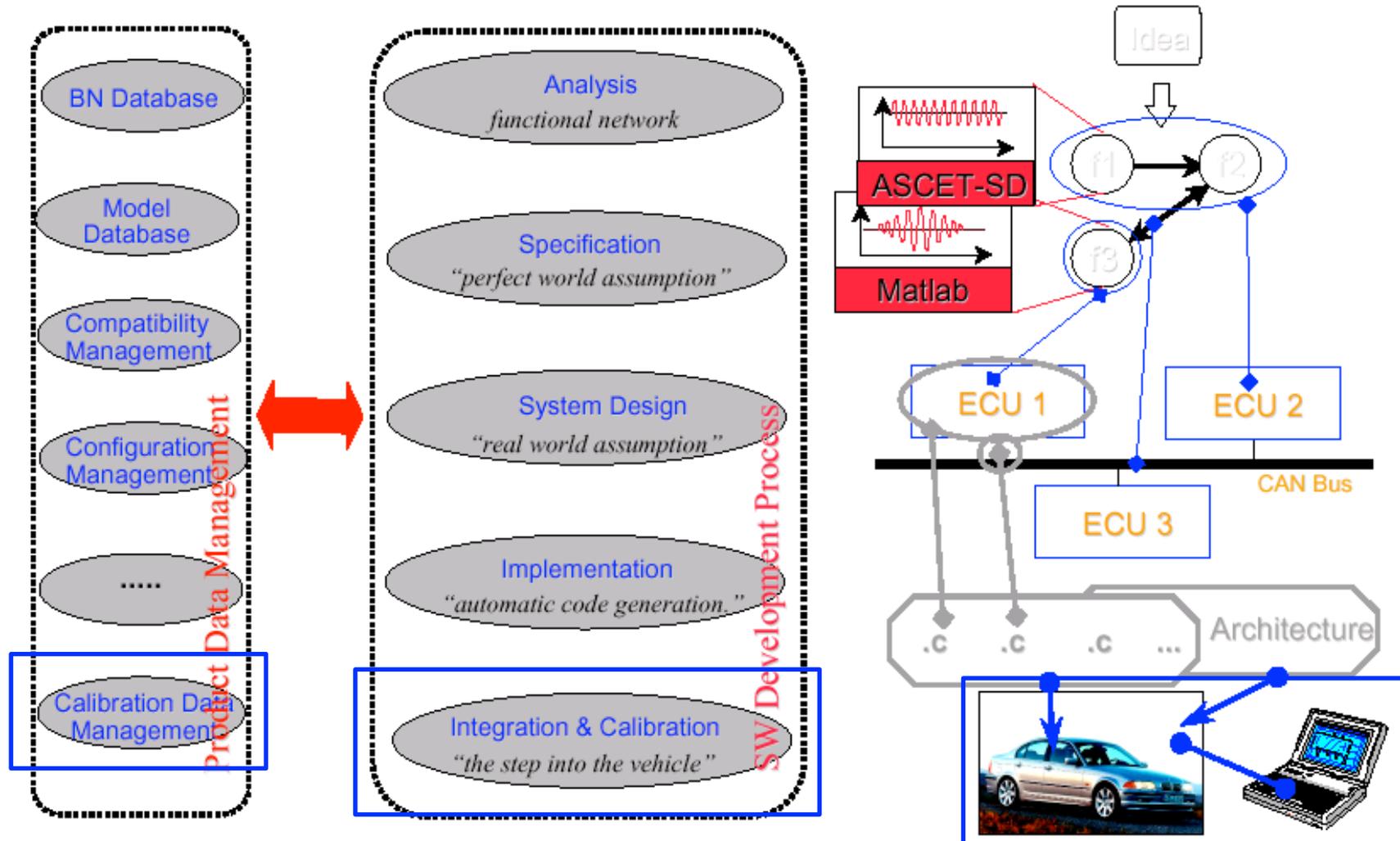
Software Entwicklungsprozess



Software Entwicklungsprozess



Software Entwicklungsprozess

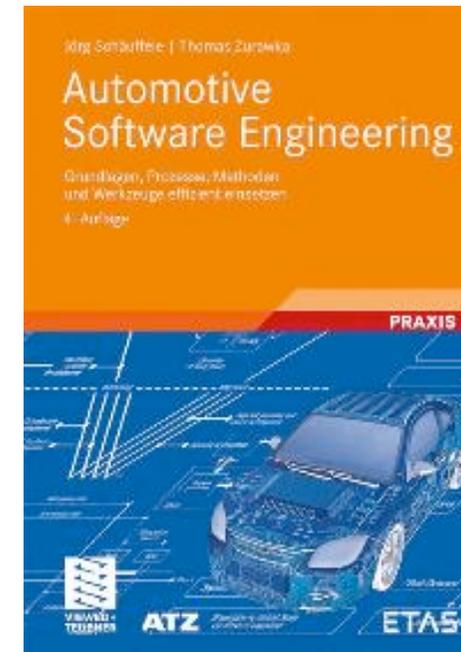
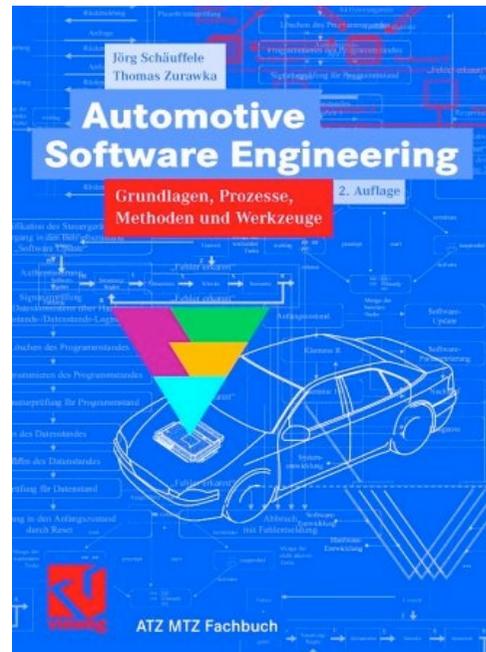


6. SW-Entwicklung / 2. Kernprozess

1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur
4. Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur
5. Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest

Quelle für Inhalte und Abbildungen

- J. Schäuuffele, Th. Zurawka: Automotive SW Engineering, Springer Vieweg, 5. Auflage, 2013.
- Kapitel 4

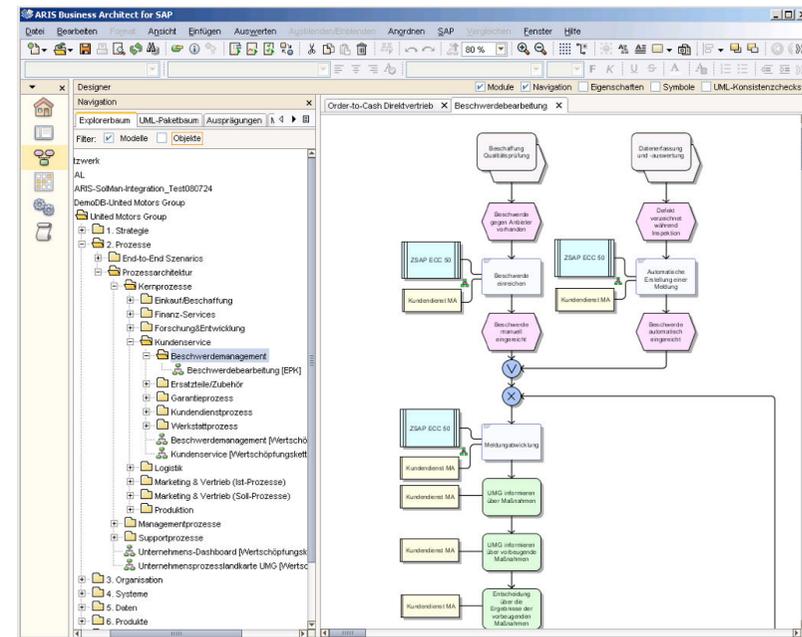


6. SW-Entwicklung / 2. Kernprozess

1. **Grundbegriffe**
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur
4. Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur
5. Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest

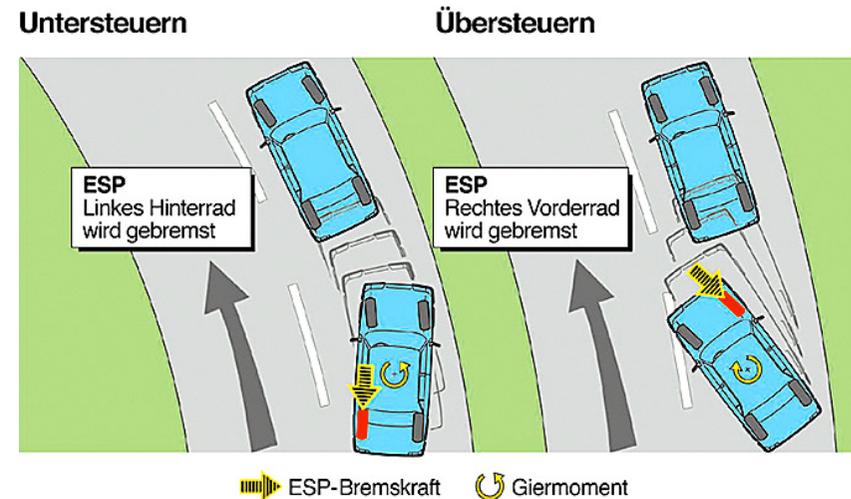
Komponentenentwicklung und Systementwicklung

- Komponentenentwicklung
 - Analyse und Entwurf von Komponenten
 - Computerspiele
 - SAP
 - Keine oder wenig Bezug zu realer Umwelt
 - Benutzer und betriebliche Abläufe müssen sich der EDV anpassen, nicht umgekehrt



Komponentenentwicklung und Systementwicklung

- Systementwicklung
 - Analyse und Entwurf des Systems als Ganzes
 - Liefert Vorgaben für Komponentenentwicklung
 - Embedded Systems / Eingebettete Systeme
 - Automotive
 - Luftfahrt
 - Bahnen
 - Medizin
 - Hoher Bezug zu realer Umwelt
 - Systeme wie ESP haben sich z.B. der Physik anzupassen



Embedded Systems

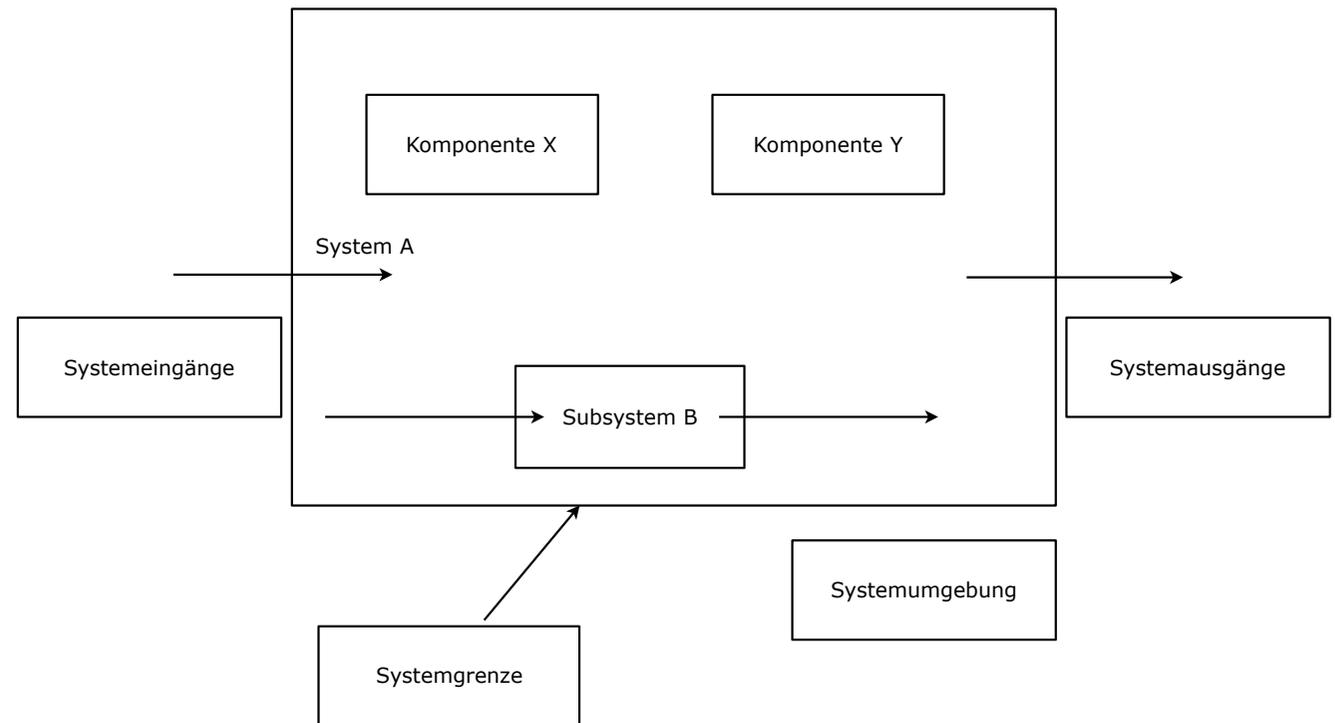
- „Eingebettete mikroelektronische Systeme sind die Treiber für innovative Technologien und Märkte des 21. Jahrhunderts. Fast alle volkswirtschaftlich relevanten Industriezweige, vom Automobilbau über die Automatisierungsbranche bis hin zur Medizintechnik erreichen ihre Spitzenstellung nur durch den Einsatz von Embedded Systems. Ohne sie fliegt kein Flugzeug, funktioniert kein Mobiltelefon und kein Computertomograph. Unentbehrlicher Bestandteil von Embedded Systems ist die Software. Meist werden die Programme in der verarbeitenden Industrie selbst geschrieben. Daher zählen viele deutsche Unternehmen zu den weltweit führenden Softwareproduzenten.“
(Presseinformation Pr 173-2009 des ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e.V.anlässlich des 4. Nationalen IT-Gipfels in Stuttgart am 8. Dezember 2009)
- Am 8. Dezember 2009 wurde auf dem 4. IT-Gipfel in Stuttgart im Beisein von Frau Bundeskanzlerin Dr. Angela Merkel die Nationale Roadmap Embedded Systems vorgestellt.
- Quelle:<http://www.zvei.org/Verband/Publikationen/Seiten/Nationale-Roadmap-Embedded-Systems.aspx>

Grundbegriffe der Systemtheorie

- Systemtheorie = Vorgehensweisen zum Umgang mit Komplexität
- Ansatz: Divide et impera / Teile und Beherrsche
- Annahmen
 - Die Aufteilung des Systems in Komponenten verfälscht nicht das betrachtete Problem.
 - Die Komponenten für sich betrachtet sind in wesentlichen Teilen identisch mit den Komponenten des Systems.
 - Die Prinzipien für den Zusammenbau der Komponenten zum System sind einfach, stabil und bekannt.
- Diese Annahmen sind bei der Entwicklung von elektronischen Systemen und Software für Fahrzeuge erfüllt bzw. erfüllbar.
- Die Eigenschaften des Systems ergeben sich aus dem Zusammenspiel der Komponenten
- Komplexität des Systems führt zu komplexen und aufwändigen Analysen der Komponenten und ihres Zusammenspiels = Kern der Systemtheorie
- Komponenten
 - Technische Bauteile, Menschen / Benutzer, Umwelt

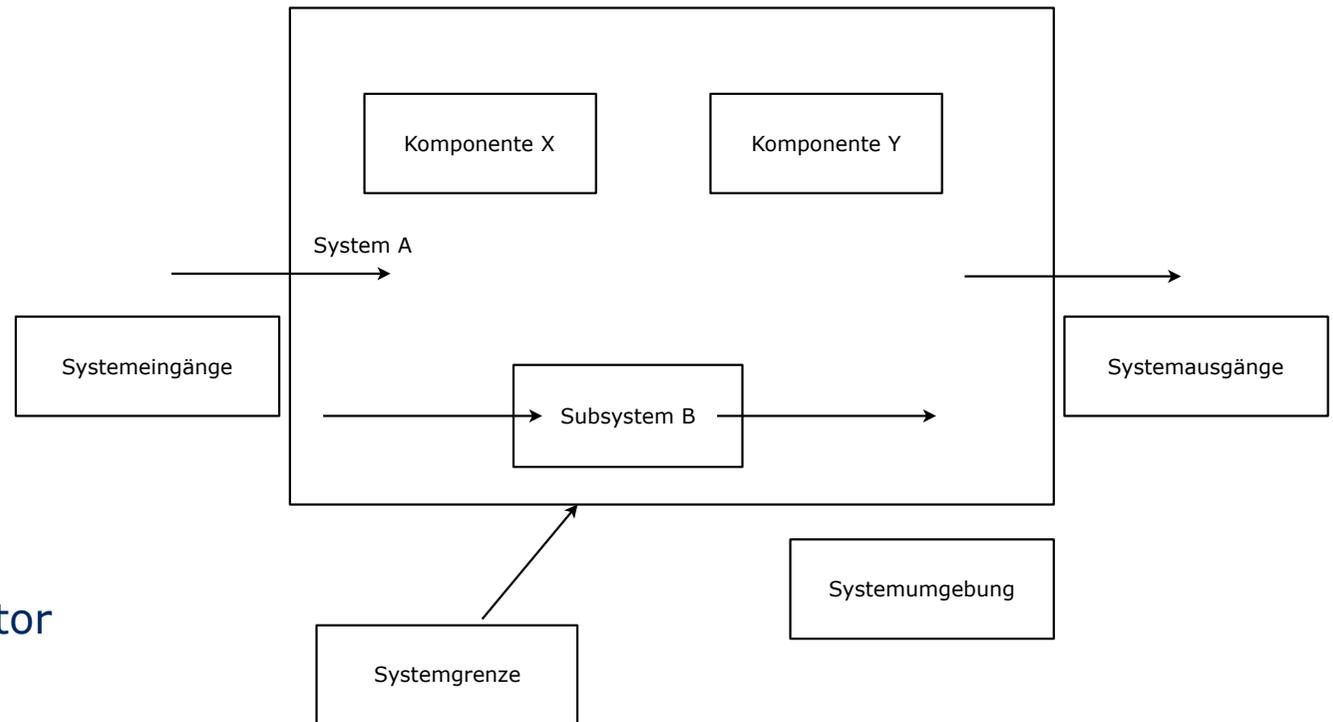
Systemdefinition (Schäuffele, Zurawka)

- Ein System ist eine von ihrer Umgebung abgegrenzte Anordnung aufeinander einwirkender Komponenten.
- Ein System interagiert mit seiner Umgebung über Systemeingänge und Systemausgänge



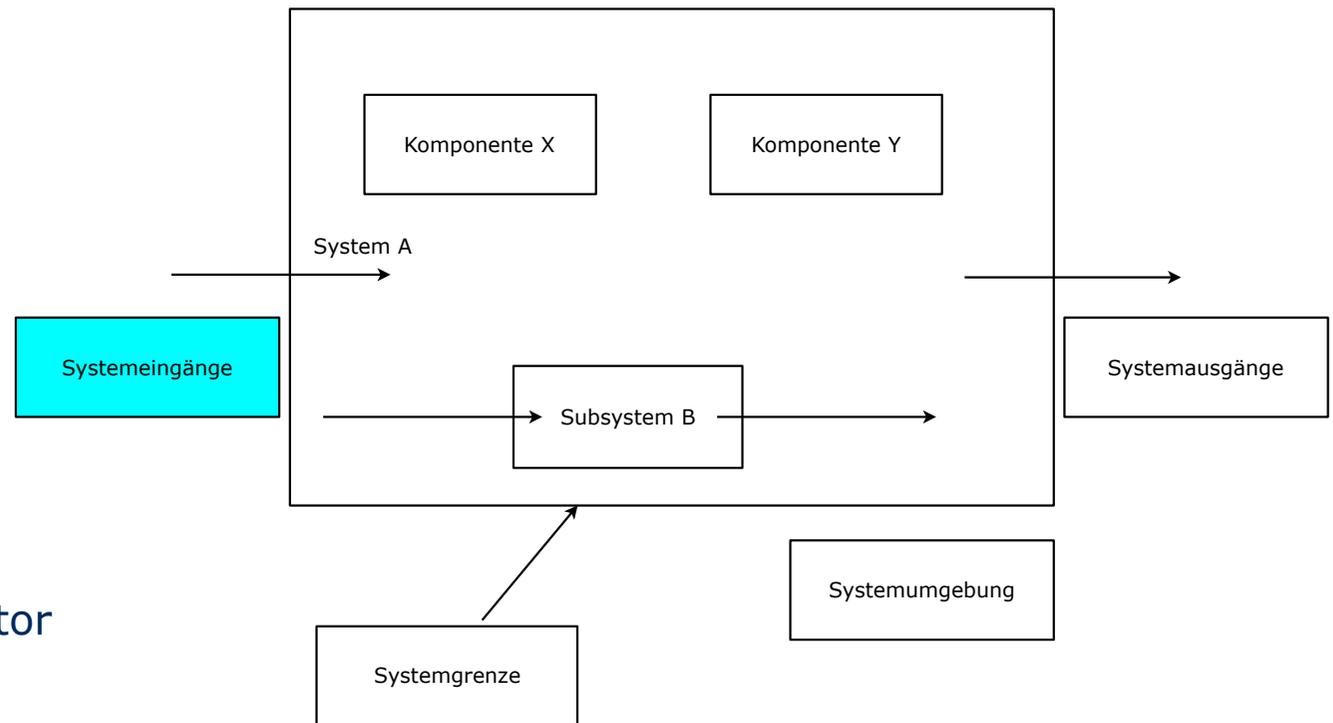
Systemdefinition (Schäuffele, Zurawka)

- Beispiel
 - Systemeingänge
 - Aktionen des Fahrers
 - Beschleunigen
 - Bremsen
 - Lenken
 - Blinken
 - Sensorsignale
 - Reibwert-erkennung
 - System
 - Fahrzeug
 - Systemausgänge
 - Geschwindigkeitsvektor
 - Aktionen
 - Blinker leuchtet



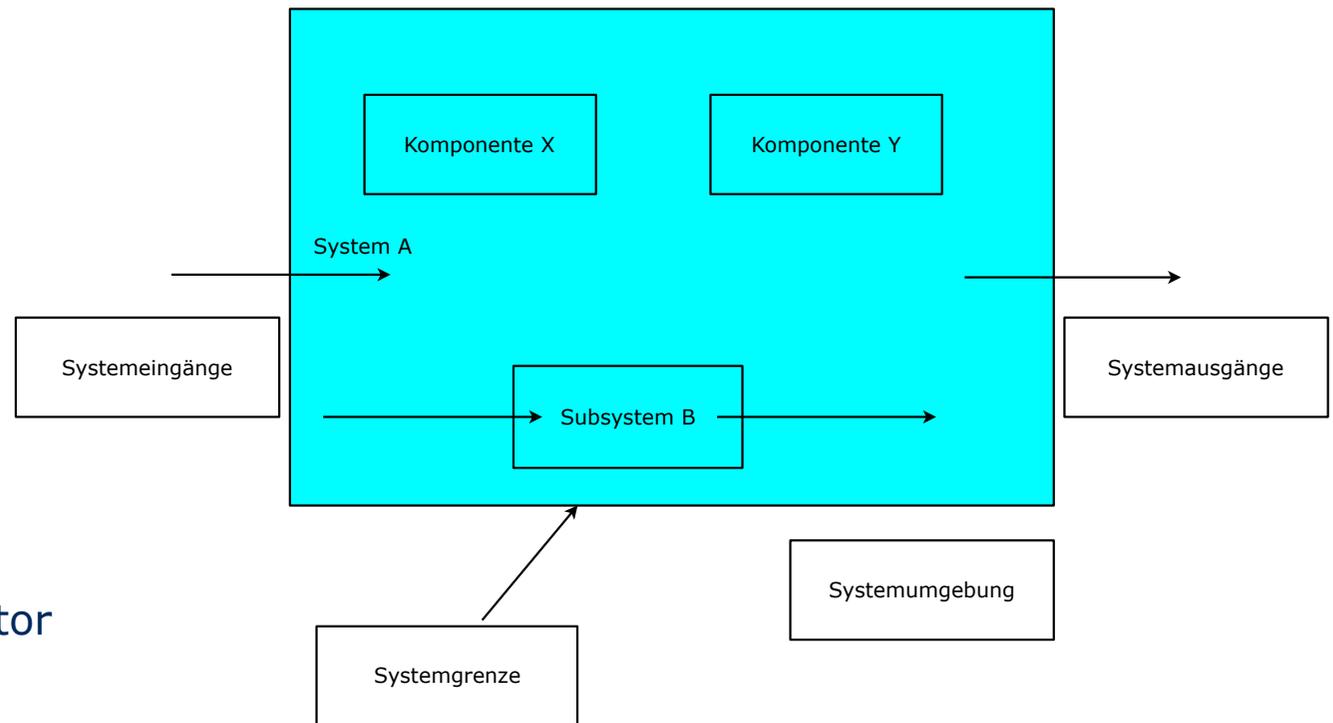
Systemdefinition (Schäuffele, Zurawka)

- Beispiel
 - **Systemeingänge**
 - Aktionen des Fahrers
 - Beschleunigen
 - Bremsen
 - Lenken
 - Blinken
 - Sensorsignale
 - Reibwert-erkennung
 - System
 - Fahrzeug
 - Systemausgänge
 - Geschwindigkeitsvektor
 - Aktionen
 - Blinker leuchtet



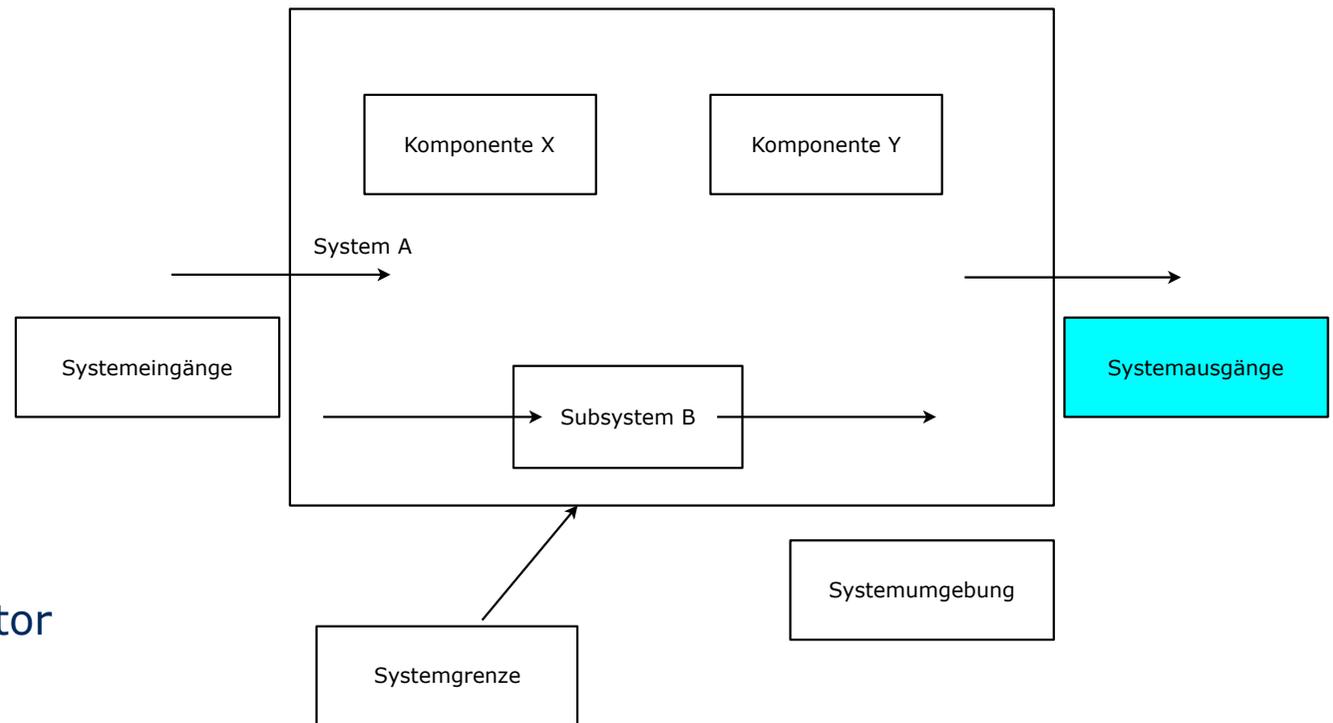
Systemdefinition (Schäuffele, Zurawka)

- Beispiel
 - Systemeingänge
 - Aktionen des Fahrers
 - Beschleunigen
 - Bremsen
 - Lenken
 - Blinken
 - Sensorsignale
 - Reibwert-erkennung
 - **System**
 - Fahrzeug
 - Systemausgänge
 - Geschwindigkeitsvektor
 - Aktionen
 - Blinker leuchtet



Systemdefinition (Schäuffele, Zurawka)

- Beispiel
 - Systemeingänge
 - Aktionen des Fahrers
 - Beschleunigen
 - Bremsen
 - Lenken
 - Blinken
 - Sensorsignale
 - Reibwert-erkennung
 - System
 - Fahrzeug
 - **Systemausgänge**
 - Geschwindigkeitsvektor
 - Aktionen
 - Blinker leuchtet



Systemzustand

- Systemzustand
 - Menge von Eigenschaften, die das System zu einem bestimmten Zeitpunkt beschreiben
 - Beispiele
 - Motor läuft mit 4862 Umdrehungen / min
 - Getriebe läuft im 5. Gang
 - Im Radio läuft MDR Jump

Systemumgebung

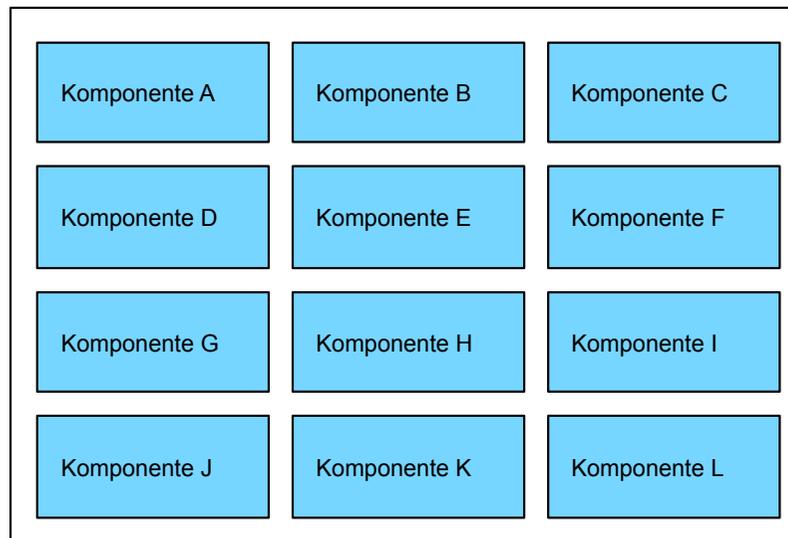
- Systemumgebung (Umgebung)
 - Menge von Komponenten, die nicht zum System gehören, aber das System beeinflussen (können)
 - Beispiel
 - System: Fahrzeug
 - Umgebung: Fahrer, Strasse
- Systemschnittstellen
 - Signale aus der Umgebung (Systemeingänge)
 - Signale an die Umgebung (Systemausgänge)
- Komponenten eines Systems können wiederum Systeme sein (Subsysteme)
 - Mehrere Systemebenen (Betrachtungsebenen, Abstraktionsebenen)
- Aussensicht
 - Abstraktion der Systemsicht auf Systemgrenze und Systemschnittstellen
 - Keine Unterscheidung zwischen Komponente und System

Modellierung von Systemen

- Methoden zur Modellierung von Systemen
 - Abstraktion durch
 - Hierarchiebildung
 - Zusammenführung von Komponenten zum System
 - Bezeichnung: Integration oder Komposition
 - Modularisierung
 - Zerlegung eines Systems in Komponenten
 - Bezeichnung: Partitionierung oder Dekomposition
- „7 +/- 2 - Regel“
 - Systeme mit mehr als $9 = 7 + 2$ Komponenten erscheinen komplex
 - Systeme mit weniger als $5 = 7 - 2$ Komponenten erscheinen trivial

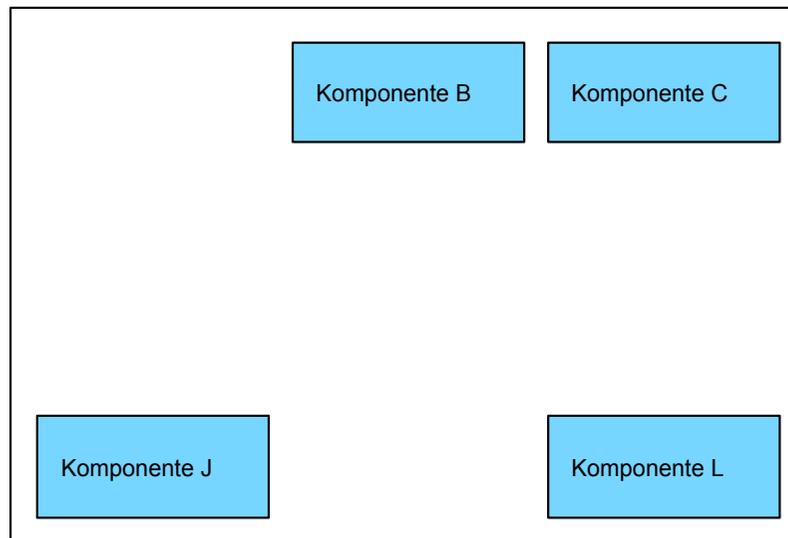
Modellierung von Systemen

- „7 +/- 2 - Regel“
 - **Systeme mit mehr als $9 = 7 + 2$ Komponenten erscheinen komplex**
 - Systeme mit weniger als $5 = 7 - 2$ Komponenten erscheinen trivial

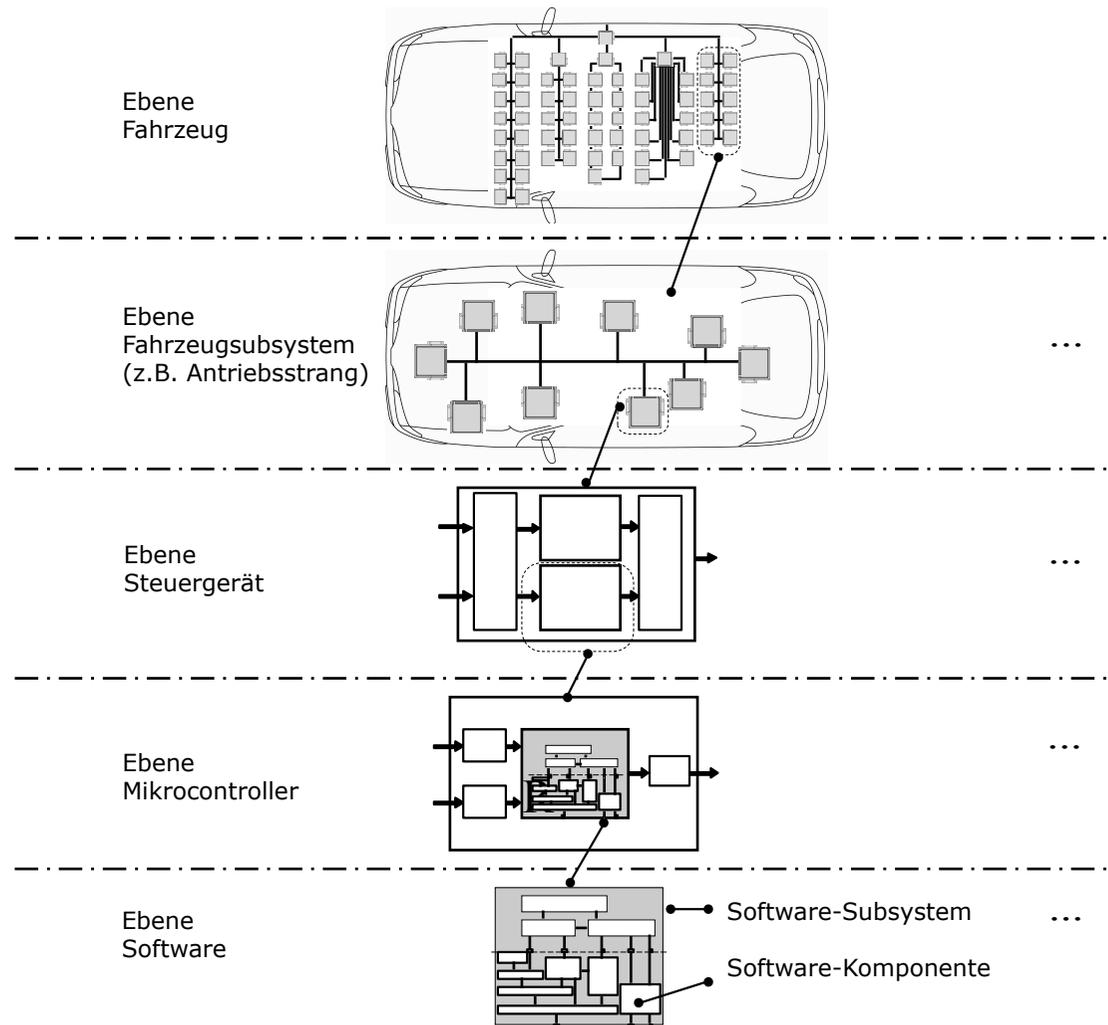


Modellierung von Systemen

- „7 +/- 2 - Regel“
 - Systeme mit mehr als $9 = 7 + 2$ Komponenten erscheinen komplex
 - **Systeme mit weniger als $5 = 7 - 2$ Komponenten erscheinen trivial**



Systemebenen in der Fahrzeugelektronik (Nach Schäuffele, Zurawka)



Systems Engineering (1)

- Im Gegensatz zur Komponentenentwicklung zielt die Systementwicklung (engl. Systems Engineering) auf die Analyse und den Entwurf des Systems als Ganzes und nicht auf die Analyse und den Entwurf seiner Komponenten. ...
- Systems Engineering ist die gezielte Anwendung von wissenschaftlichen und technischen Ressourcen
 - Zur Transformation eines operationellen Bedürfnisses in die Beschreibung einer Systemkonfiguration unter bestmöglicher Berücksichtigung aller operativen Anforderungen und nach den Maßstäben der gebotenen Effektivität.
 - Zur Integration aller technischen Parameter und zur Sicherstellung der Kompatibilität aller physikalischen, funktionalen und technischen Schnittstellen in einer Art und Weise, so dass die gesamte Systemdefinition und der Systementwurf möglichst optimal werden.
 - Zur Integration der Beiträge aller Fachdisziplinen in einen ganzheitlichen Entwicklungsansatz.
- (Nach Schäuffele/Zurawka, Einleitung zu Kapitel 4)

Systems Engineering (2)

- Die Definition ist angelehnt an Definitionen von
- SEI/CMU CMMI - Capability Maturity Model Integration
<http://www.sei.cmu.edu/cmmi>
 - Software Engineering Institute (SEI) an der Carnegie Mellon University (CMU)
- INCOSE - International Council on Systems Engineering
<http://www.incose.org>
- GfSE - Gesellschaft für Systems Engineering e.V.
German Chapter of INCOSE
<http://www.gfse.de>

Systems Engineering

- Alles hängt mit allem irgendwo irgendwie irgendwann zusammen

- „2 und 2 ist irgendwas, und 4 und 4 ist auch irgendwas, aber was anderes. Das macht alles so kompliziert.“
- „2 und 2 ist irgendwas, und 2 mal 2 ist auch irgendwas, aber diesmal das gleiche. Das macht alles noch komplizierter.“

Systems Engineering

- Alles hängt mit allem irgendwo irgendwie irgendwann zusammen
- „2 und 2 ist irgendwas, und 4 und 4 ist auch irgendwas, aber was anderes. Das macht alles so kompliziert.“
- „2 und 2 ist irgendwas, und 2 mal 2 ist auch irgendwas, aber diesmal das gleiche. Das macht alles noch komplizierter.“



Systems Engineering

Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.

Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.

Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.
- Problem

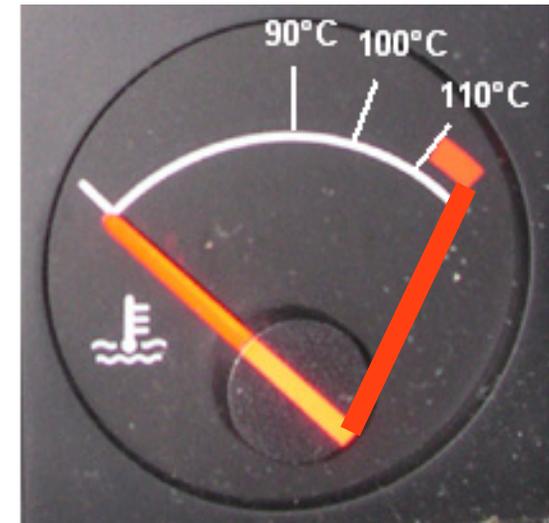
Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.
- Problem



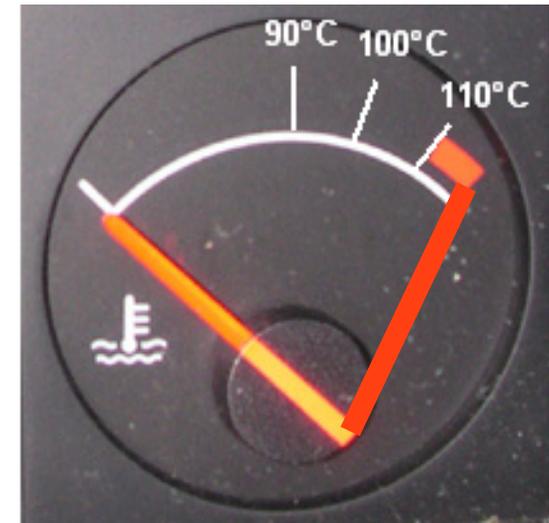
Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.
- Problem



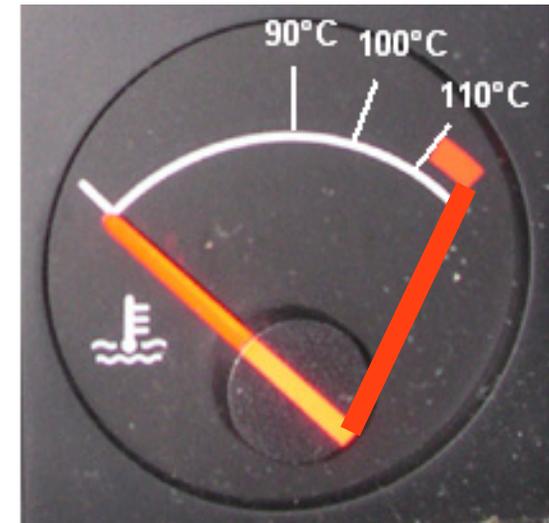
Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.
- Problem
 - Der Motor droht im Sommer zu überhitzen



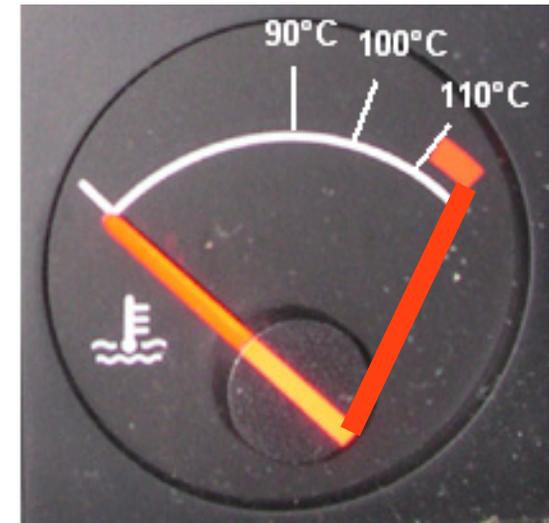
Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.
- Problem
 - Der Motor droht im Sommer zu überhitzen
 - Einfach zu realisierende Lösung (dieses Problems!)



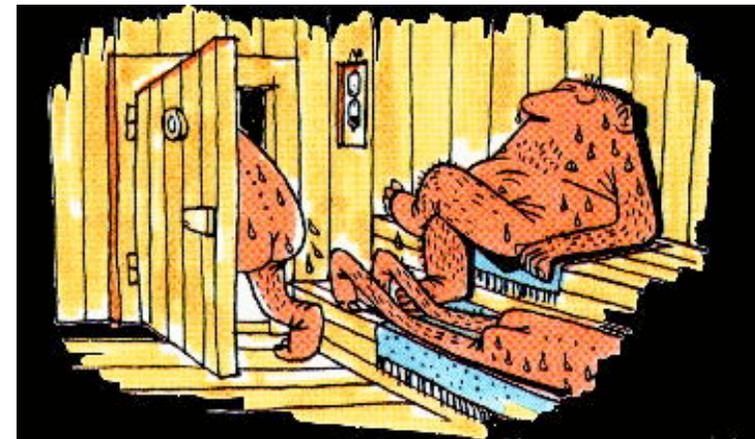
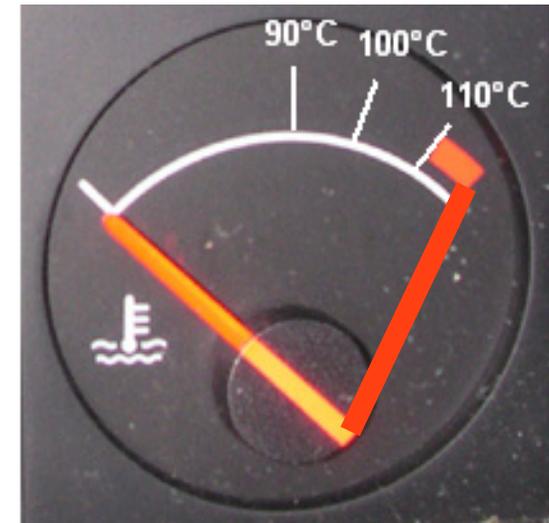
Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.
- Problem
 - Der Motor droht im Sommer zu überhitzen
 - Einfach zu realisierende Lösung (dieses Problems!)
 - Heizung auf höchste Stufe



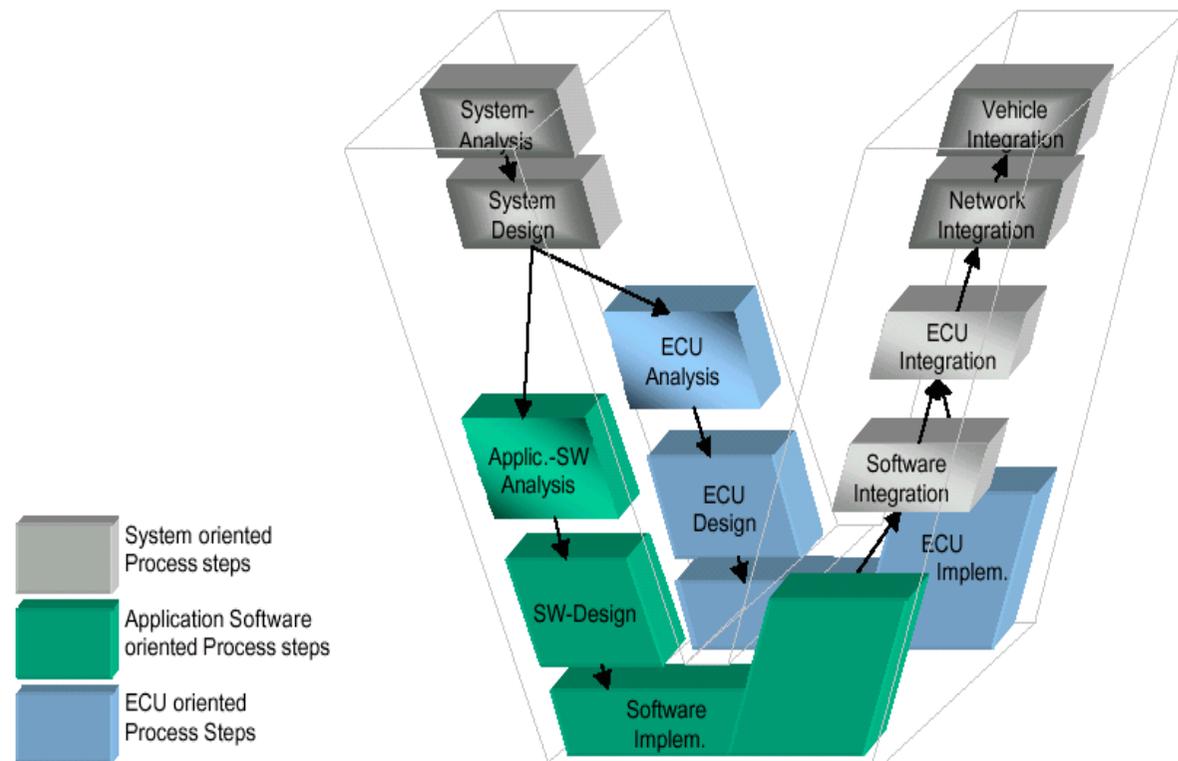
Systems Engineering

- Die Lösung eines Problems sollte nicht neue Probleme schaffen.
- Problem
 - Der Motor droht im Sommer zu überhitzen
 - Einfach zu realisierende Lösung (dieses Problems!)
 - Heizung auf höchste Stufe



Systems Engineering

- Fachdisziplinen innerhalb Systems Engineering
 - Software-Entwicklung
 - Hardware-Entwicklung
 - Sensorik
 - Aktuatorik
 - ...
- Siehe Mechatronik



Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur

Akzeptanz- und Systemtest

Anwendungsfälle

Testergebnisse

Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur

Kalibrierung
 Integrationstest des Systems
 Integration der System-Komponenten

Testfälle

Testergebnisse

Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur

Integrationstest der Software
 Integration der Software-Komponenten

Testfälle

Testergebnisse

Spezifikation der Software-Komponenten
 Design und Implementierung der Software-Komponenten

Test der Software-Komponenten

6. SW-Entwicklung / 2. Kernprozess

1. Grundbegriffe
2. **Entwicklungsobjekt: Kombiinstrument**
3. Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur
4. Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur
5. Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest

Kombiinstrument

- Tachometer
- Odometer
- Drehzahlmesser
- Tankanzeige
- Kühlmitteltemperaturanzeige
- Kontrollleuchten
- ...

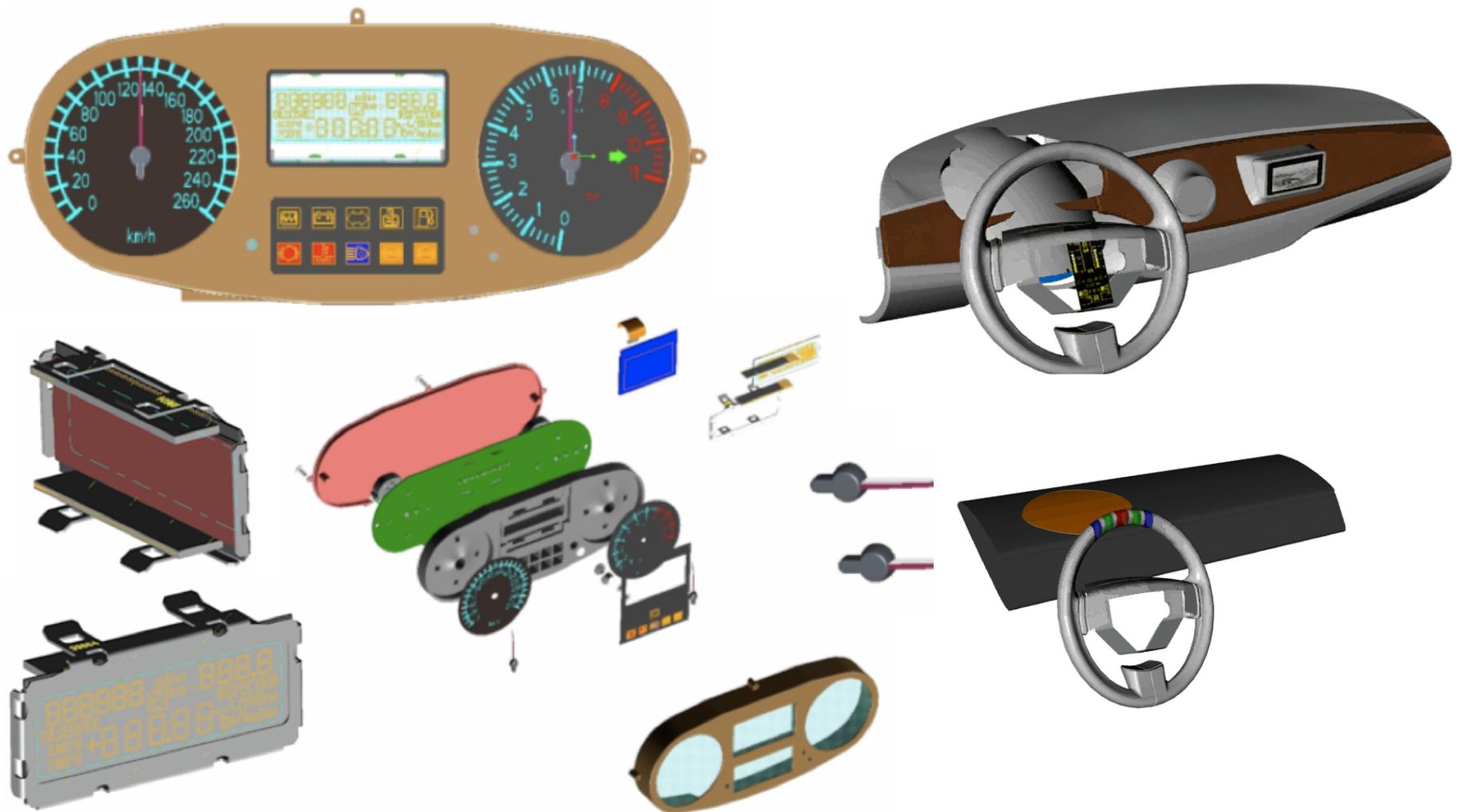


Kombiinstrument: Integration im Cockpit



Quelle: <http://www.draexlmaier.com/produkte/tuer-und-cockpitmodule.html>

Entwicklungsobjekt: Kombiinstrument



6. SW-Entwicklung / 2. Kernprozess

1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. **Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur**
4. Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur
5. Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest

Analyse der
Benutzeranforderungen und
Spezifikation der logischen
Systemarchitektur

Akzeptanz- und Systemtest

Anwendungsfälle

Testergebnisse

Analyse der logischen
Systemarchitektur und
Spezifikation der technischen
Systemarchitektur

Kalibrierung
 Integrationstest des Systems
 Integration der System-
Komponenten

Testfälle

Testergebnisse

Analyse der Software-
Anforderungen und
Spezifikation der technischen
Softwarearchitektur

Integrationstest der Software
 Integration der Software-
Komponenten

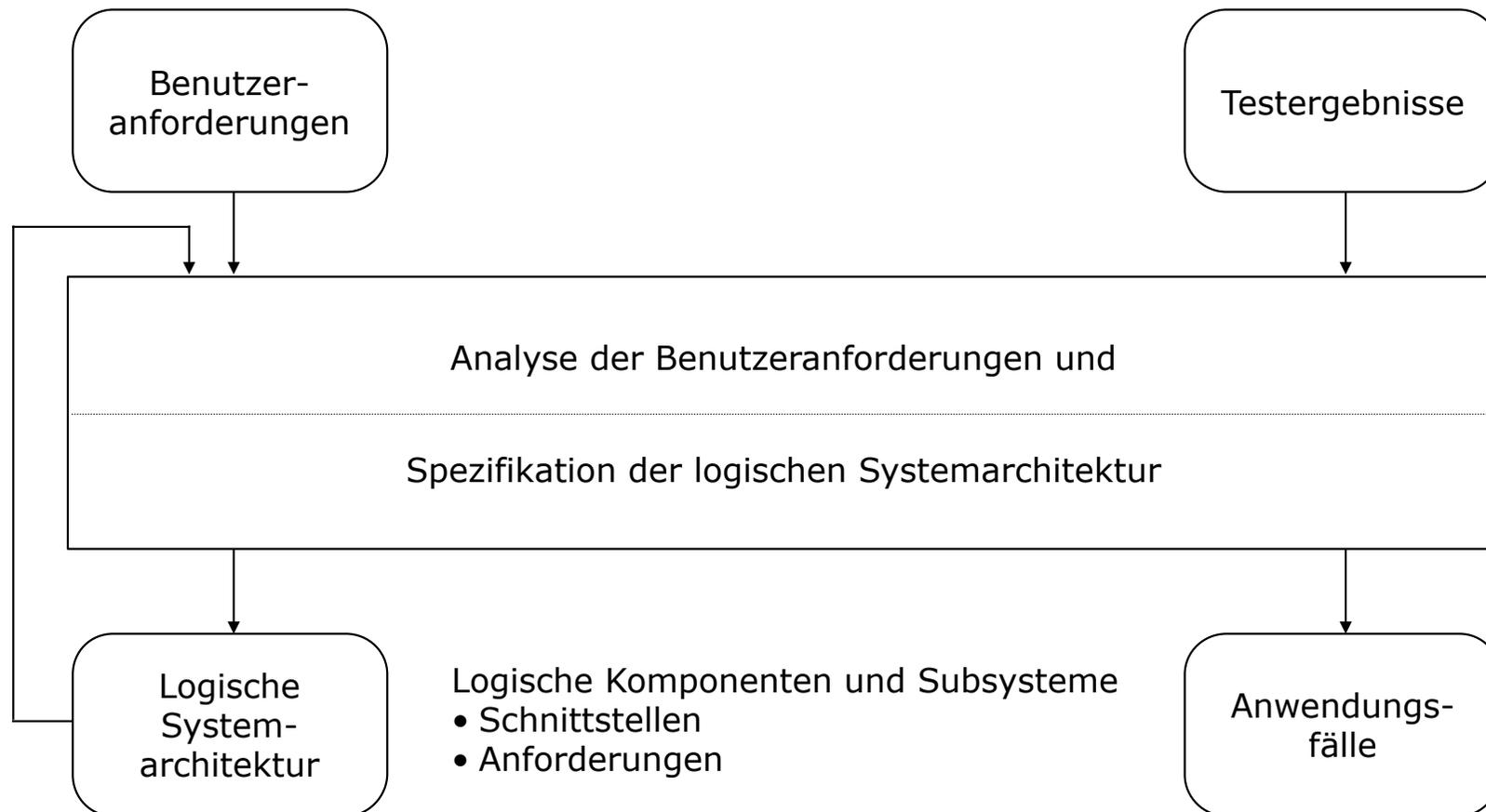
Testfälle

Testergebnisse

Spezifikation der Software-
Komponenten
 Design und Implementierung
der Software-Komponenten

Test der Software-
Komponenten

Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur



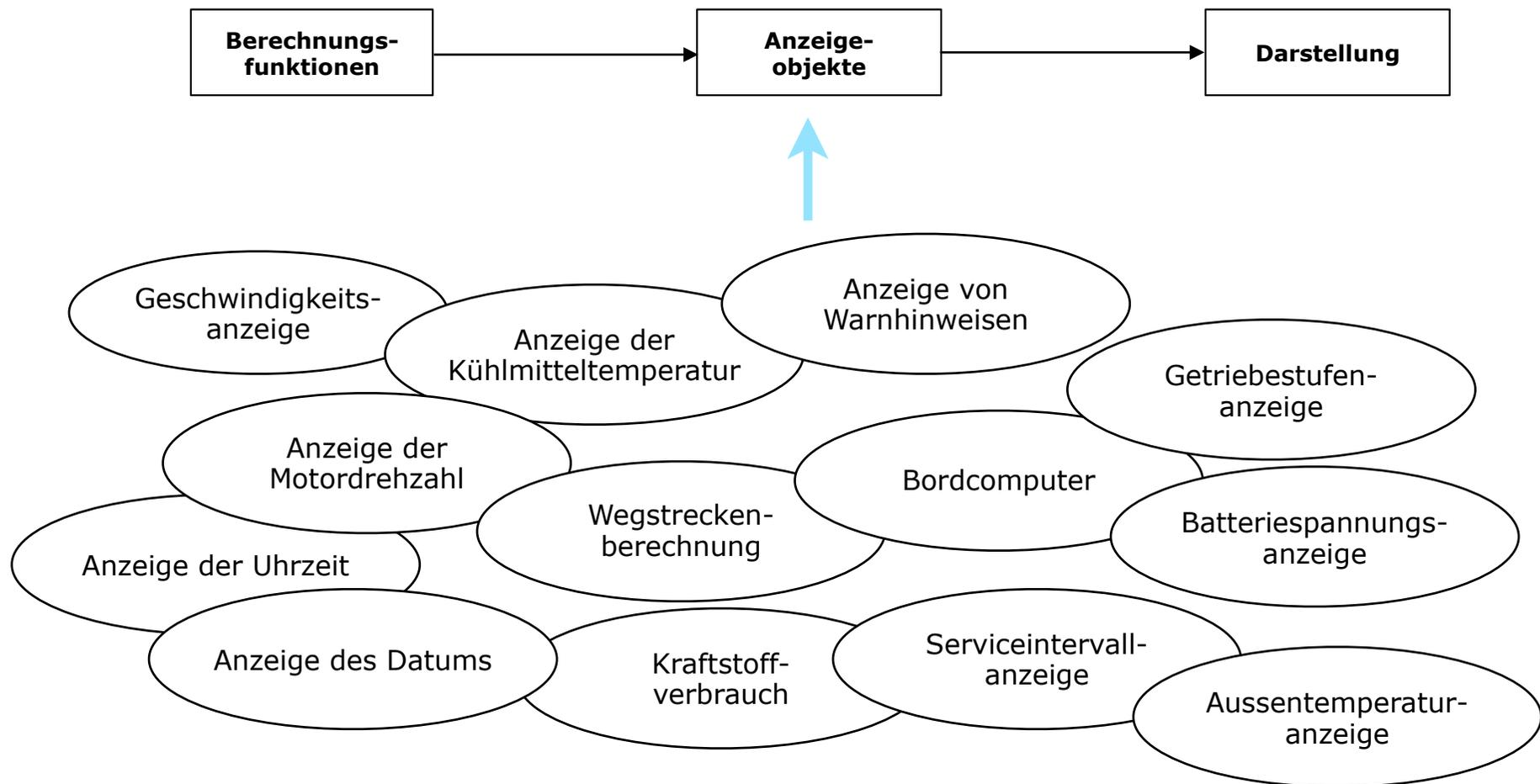
Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur

- Analyse der Benutzeranforderungen
 - Strukturierungsprozess für die Anforderungen und Randbedingungen aus Sicht der Benutzer in der frühen Phase der Systementwicklung
 - Logische Komponenten und Subsysteme definieren
 - Funktionen, Anforderungen und Schnittstellen festlegen
 - Anwendungsfälle für die Funktionen als Basis für den Systemtest festlegen
- Spezifikation der logischen Systemarchitektur
 - Abstrakte Lösung
 - Bindeglied zwischen Benutzeranforderungen und technischer Systemarchitektur
 - Modellbasierte Darstellung (Blockdiagramme, Zustandsautomaten)
 - Schrittweise Zerlegung der Funktionen

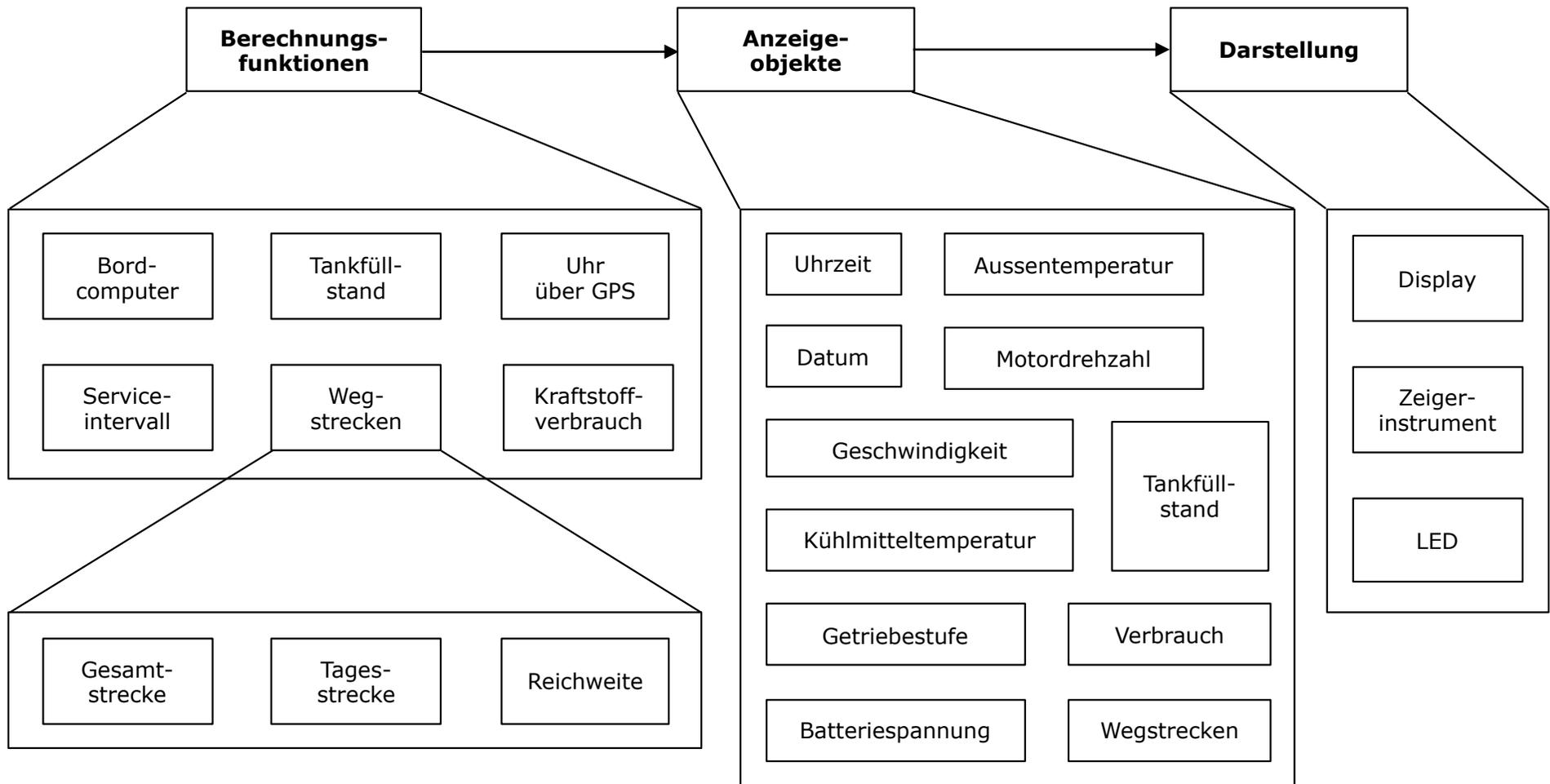
Kombiinstrument: Benutzeranforderungen



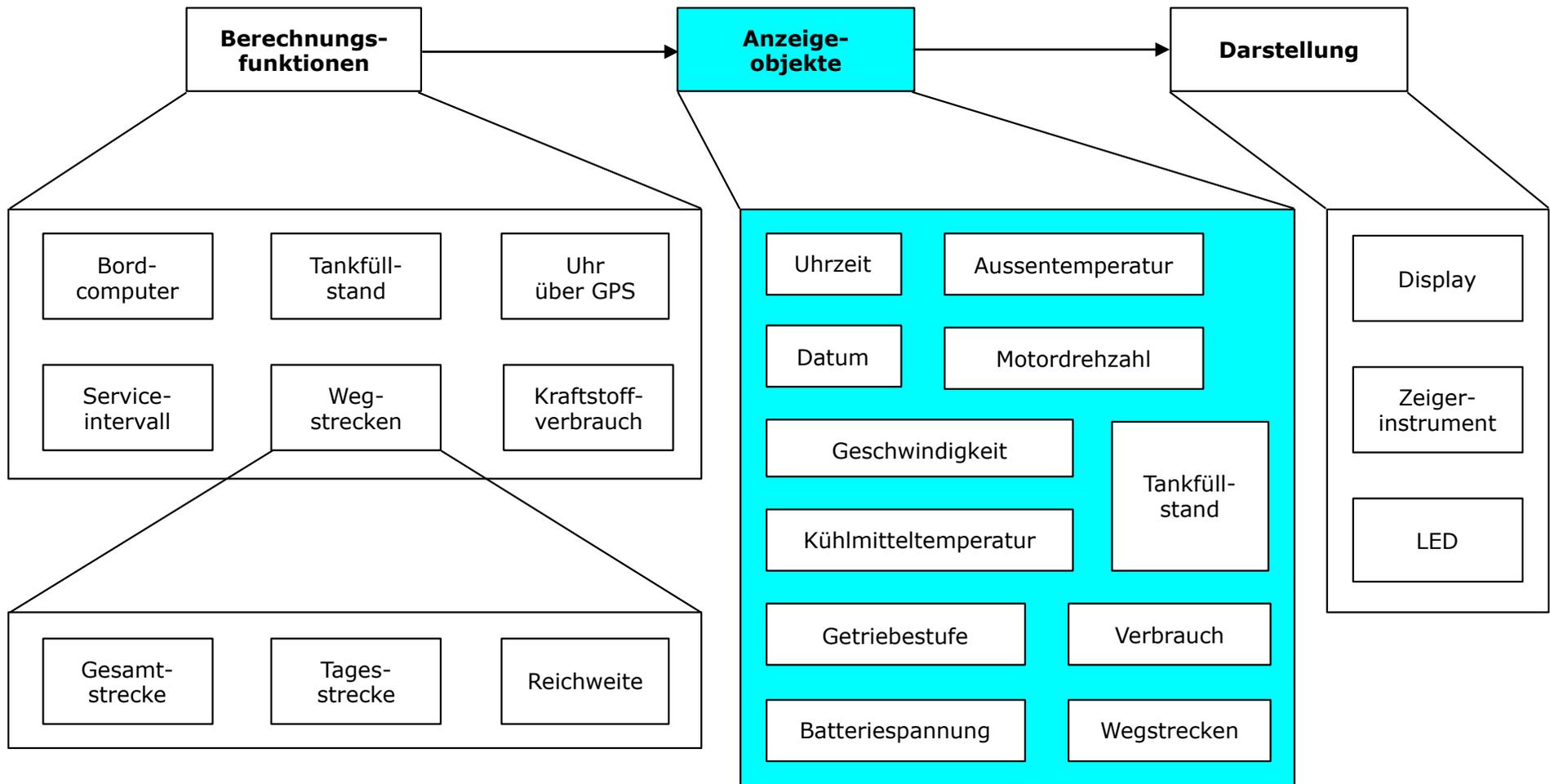
Kombiinstrument: Logische Systemarchitektur (1)



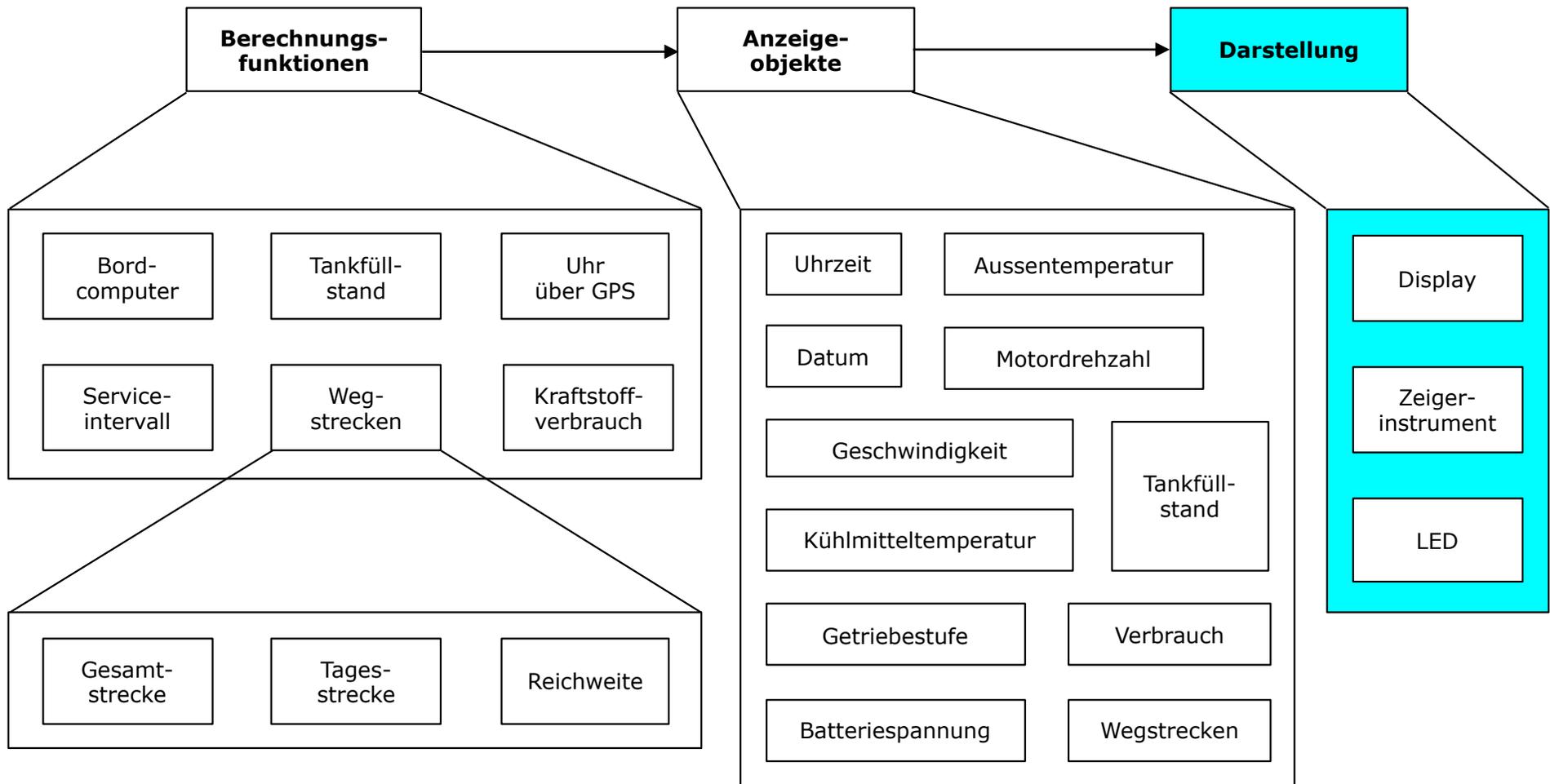
Kombiinstrument: Logische Systemarchitektur (2)



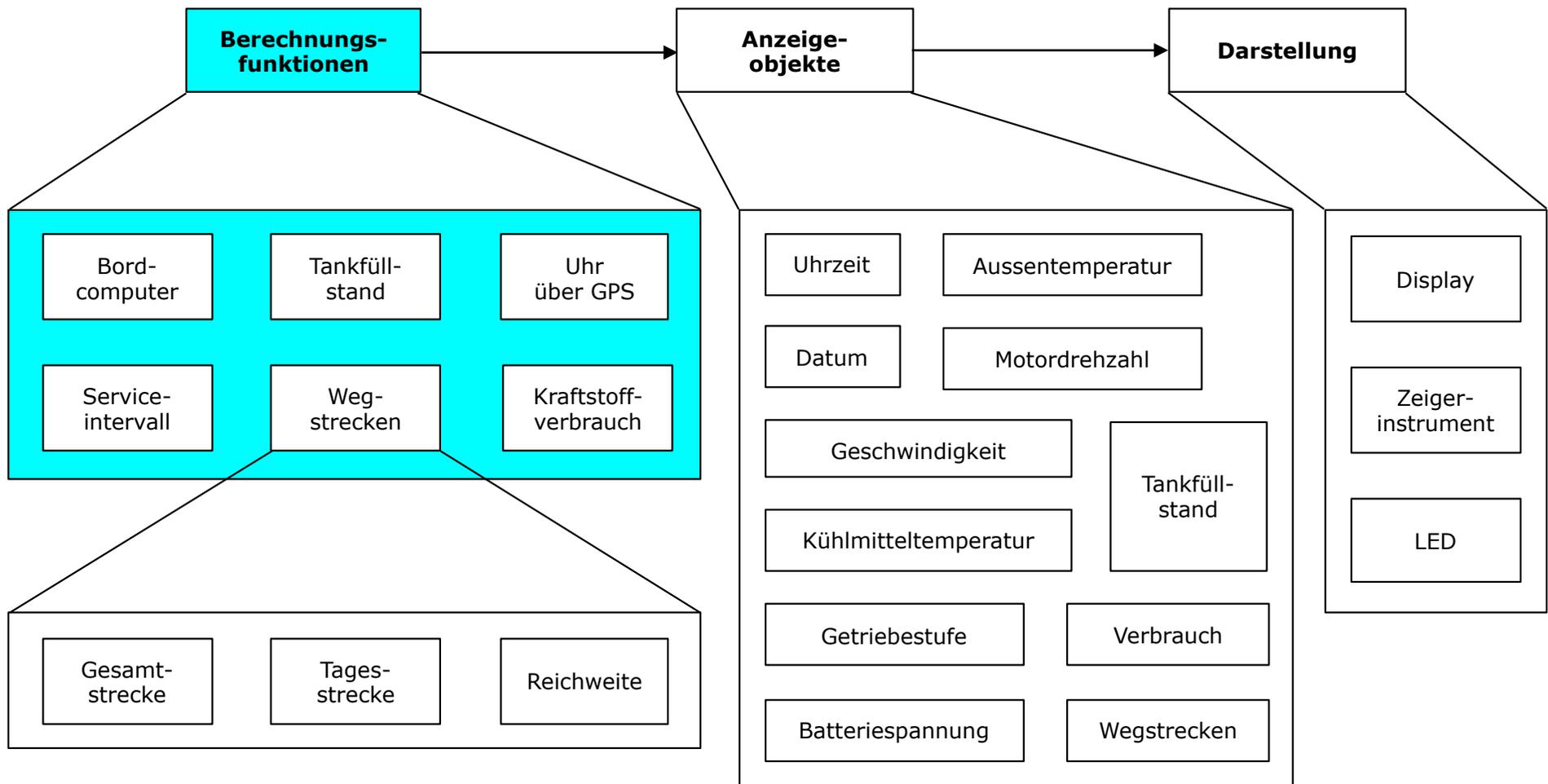
Kombiinstrument: Logische Systemarchitektur (3)



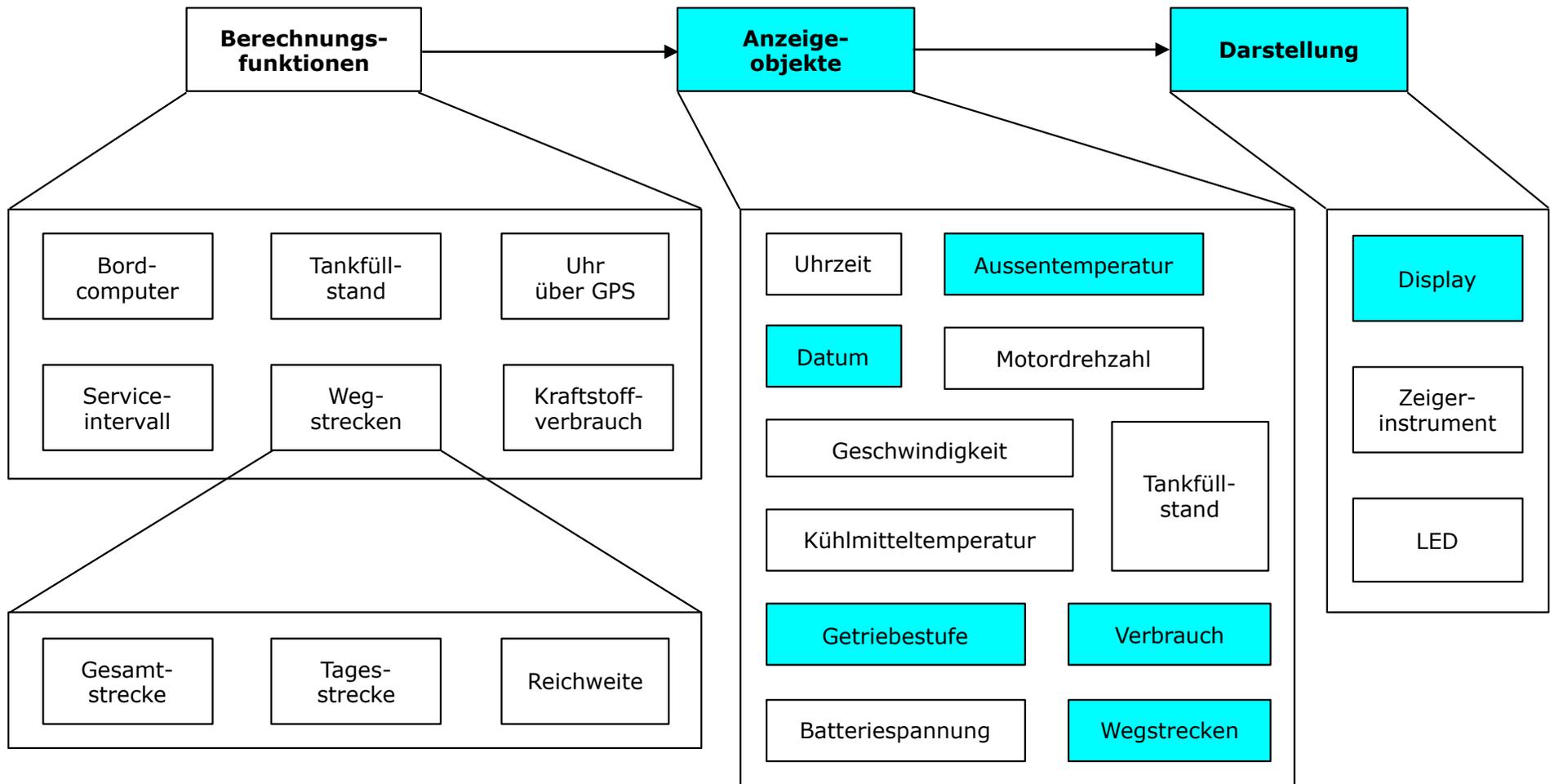
Kombiinstrument: Logische Systemarchitektur (4)



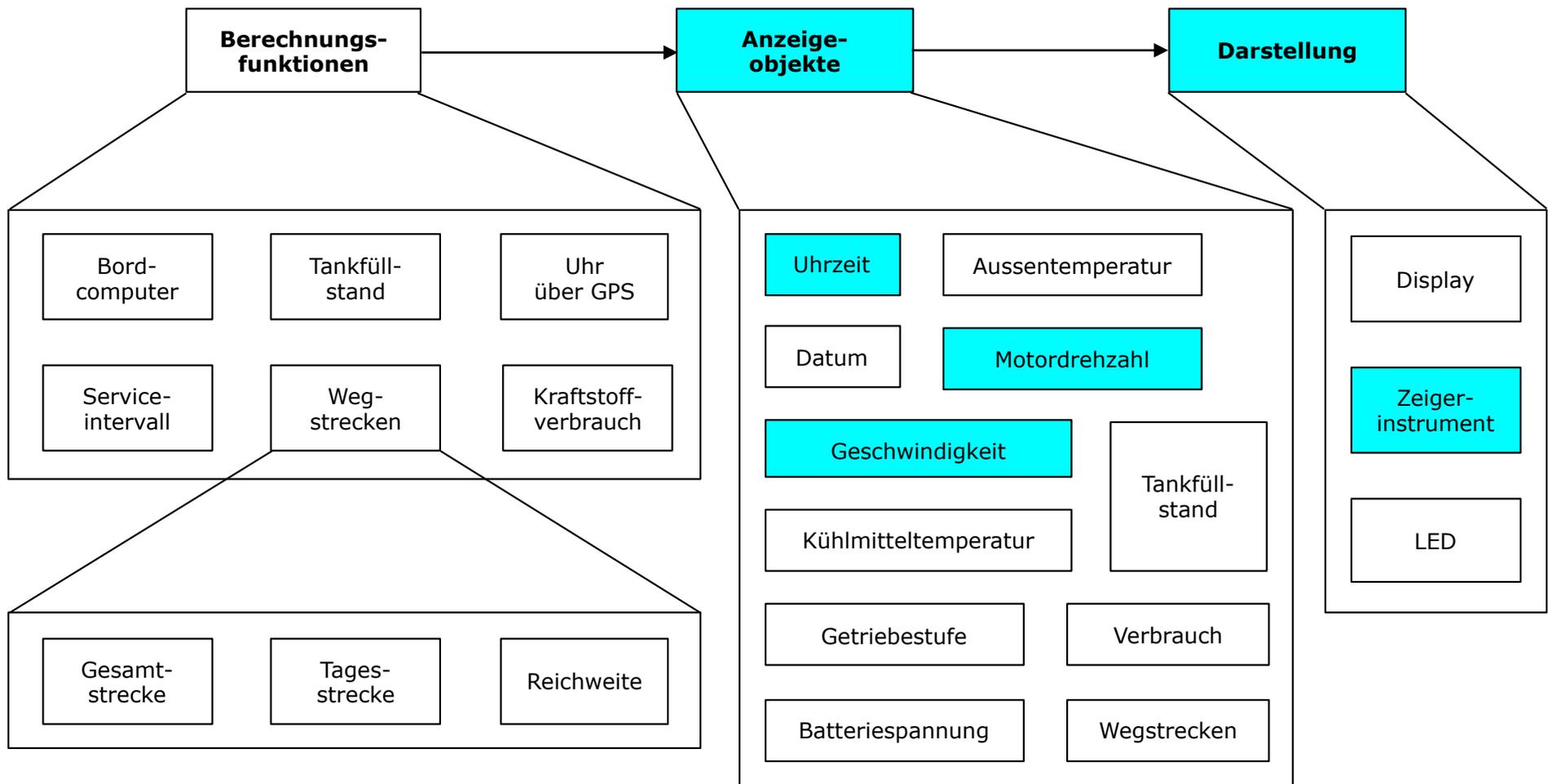
Kombiinstrument: Logische Systemarchitektur (5)



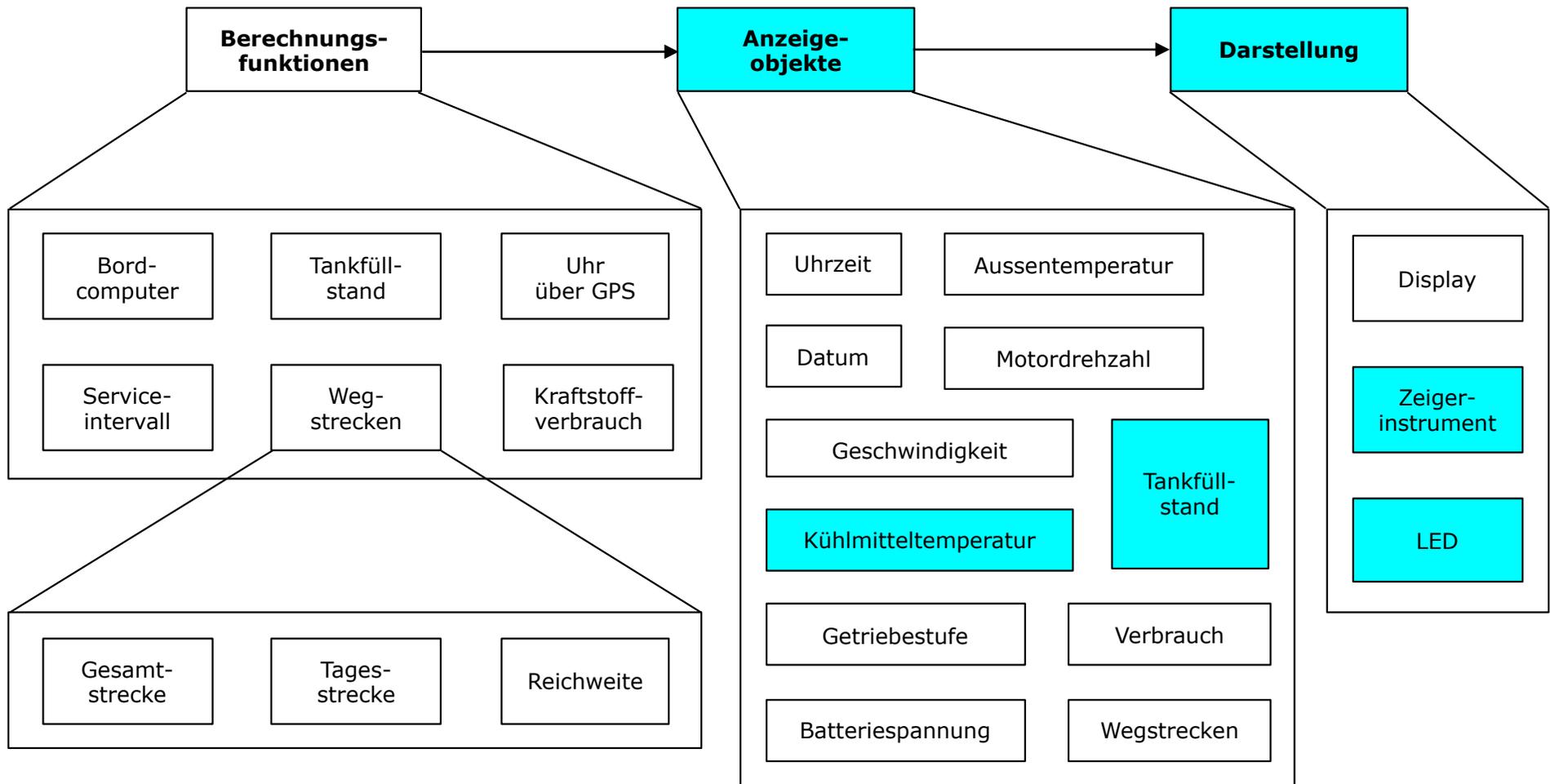
Kombiinstrument: Logische Systemarchitektur (6)



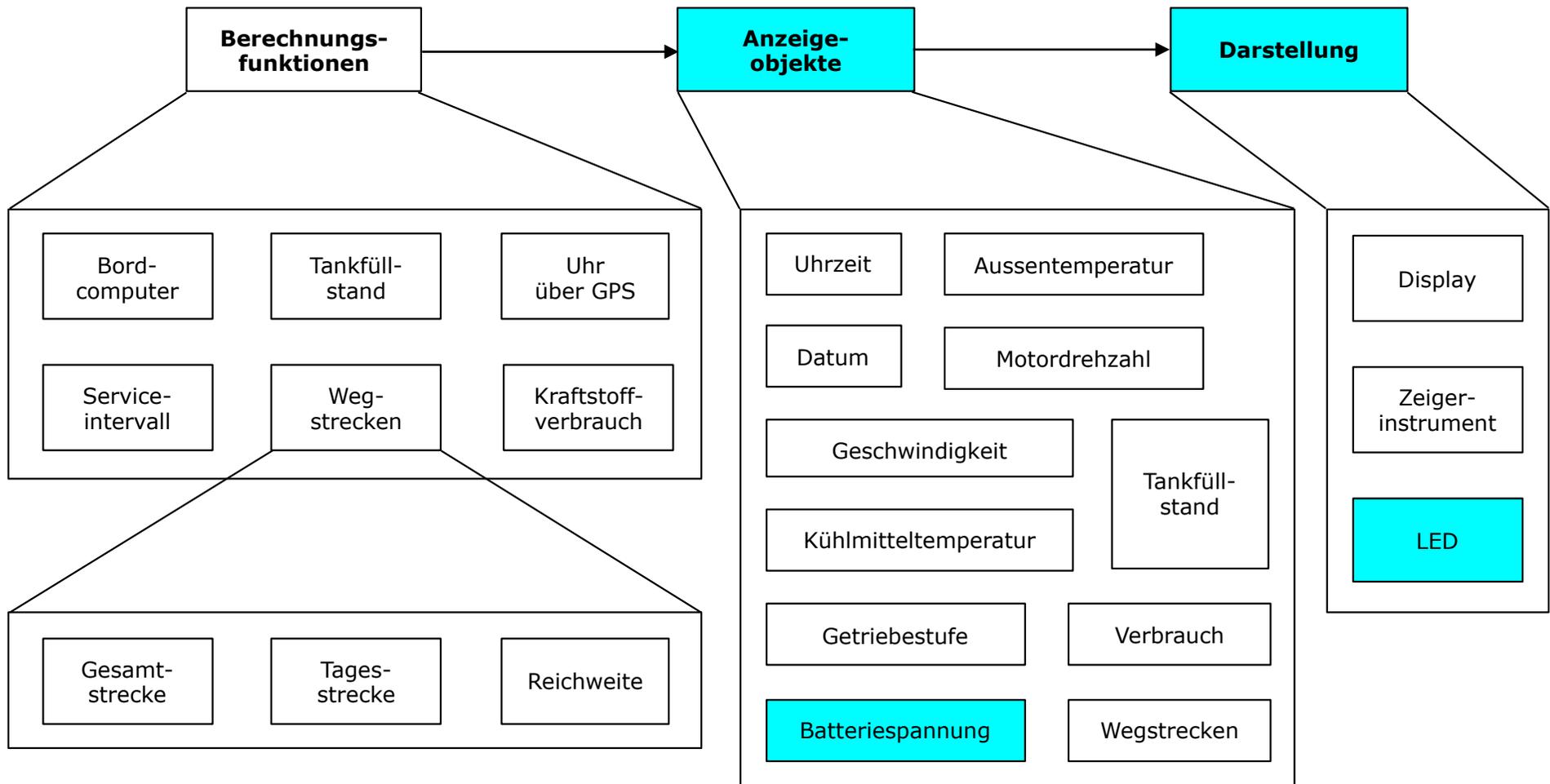
Kombiinstrument: Logische Systemarchitektur (7)



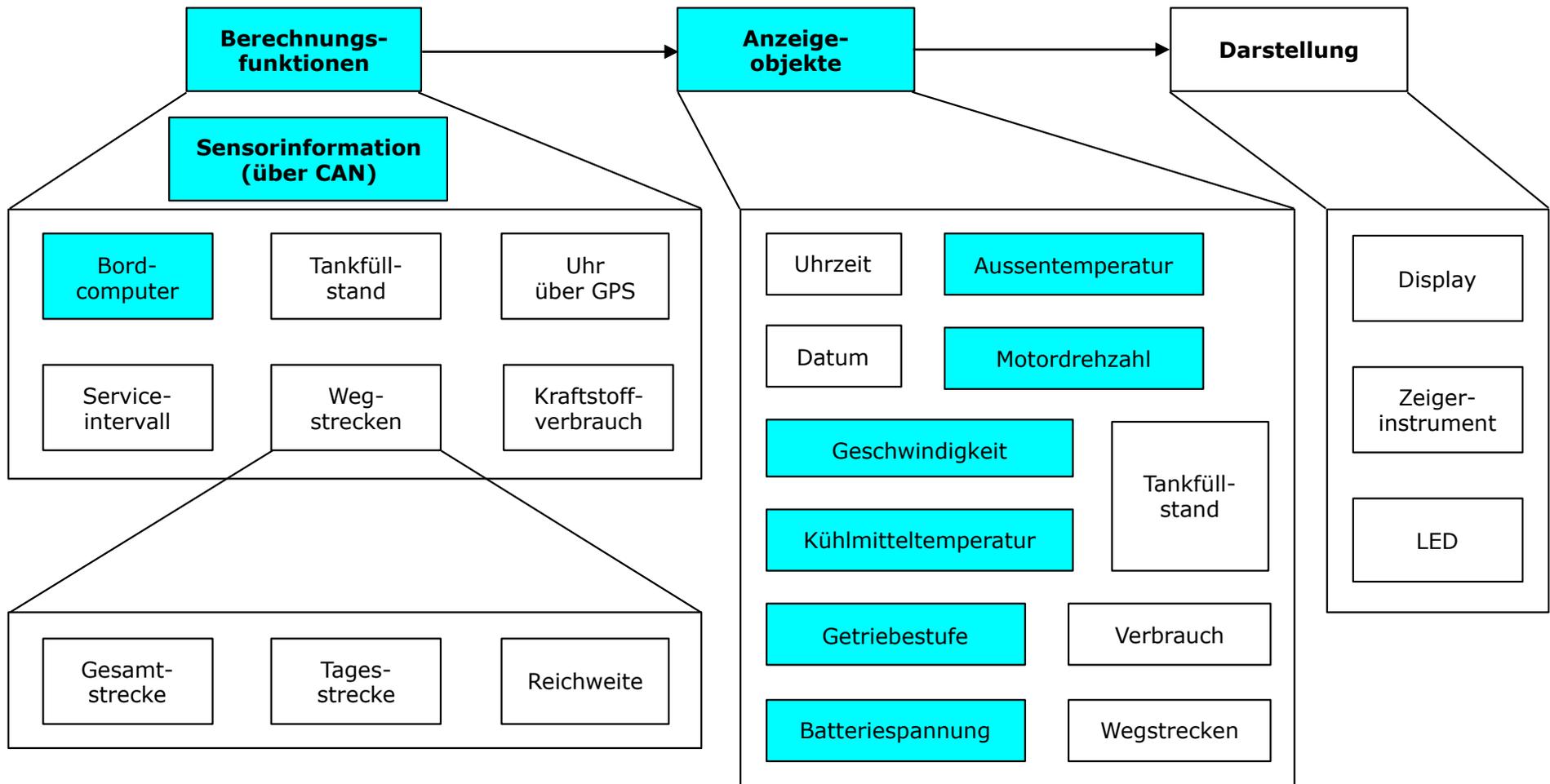
Kombiinstrument: Logische Systemarchitektur (8)



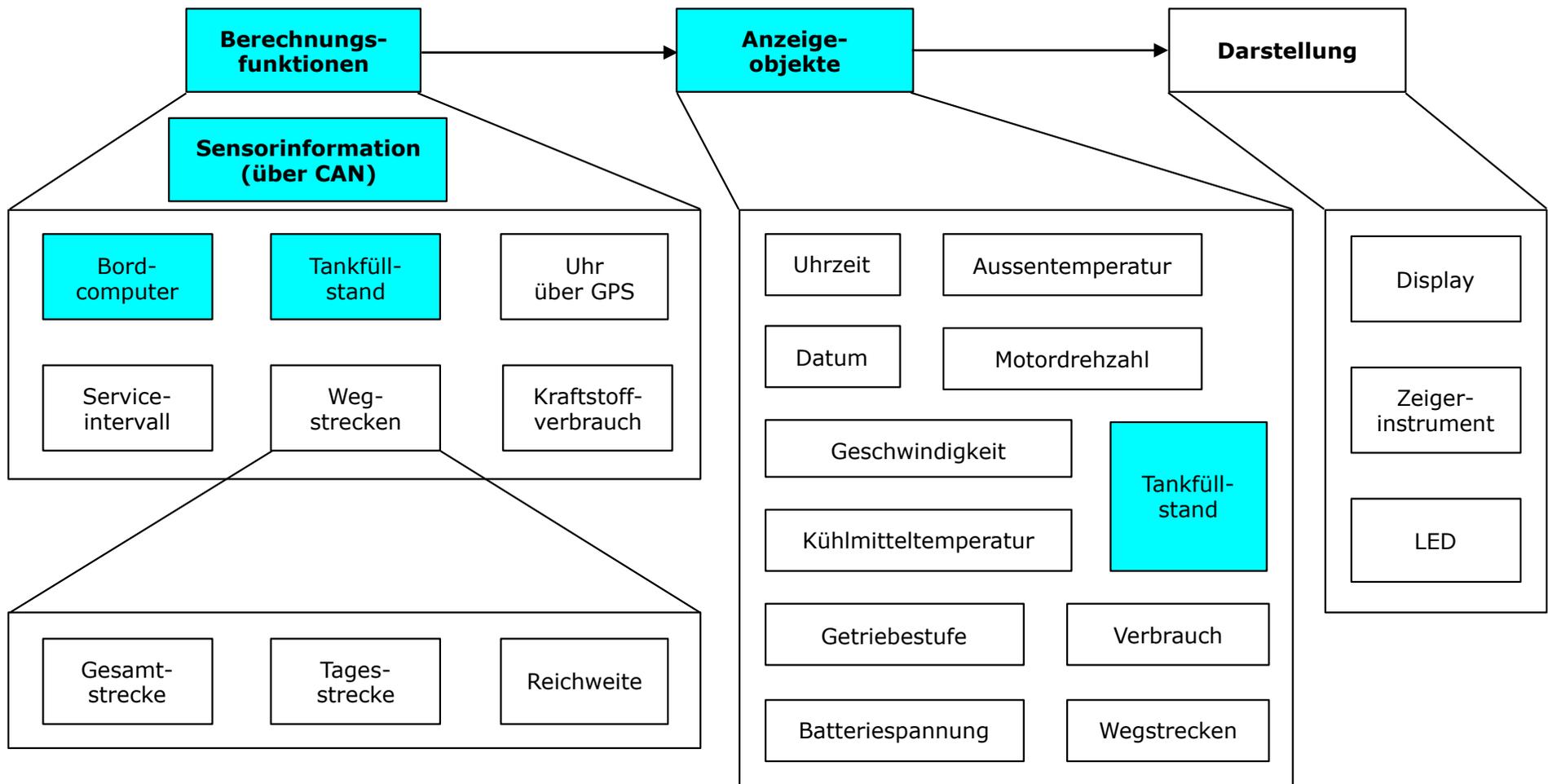
Kombiinstrument: Logische Systemarchitektur (9)



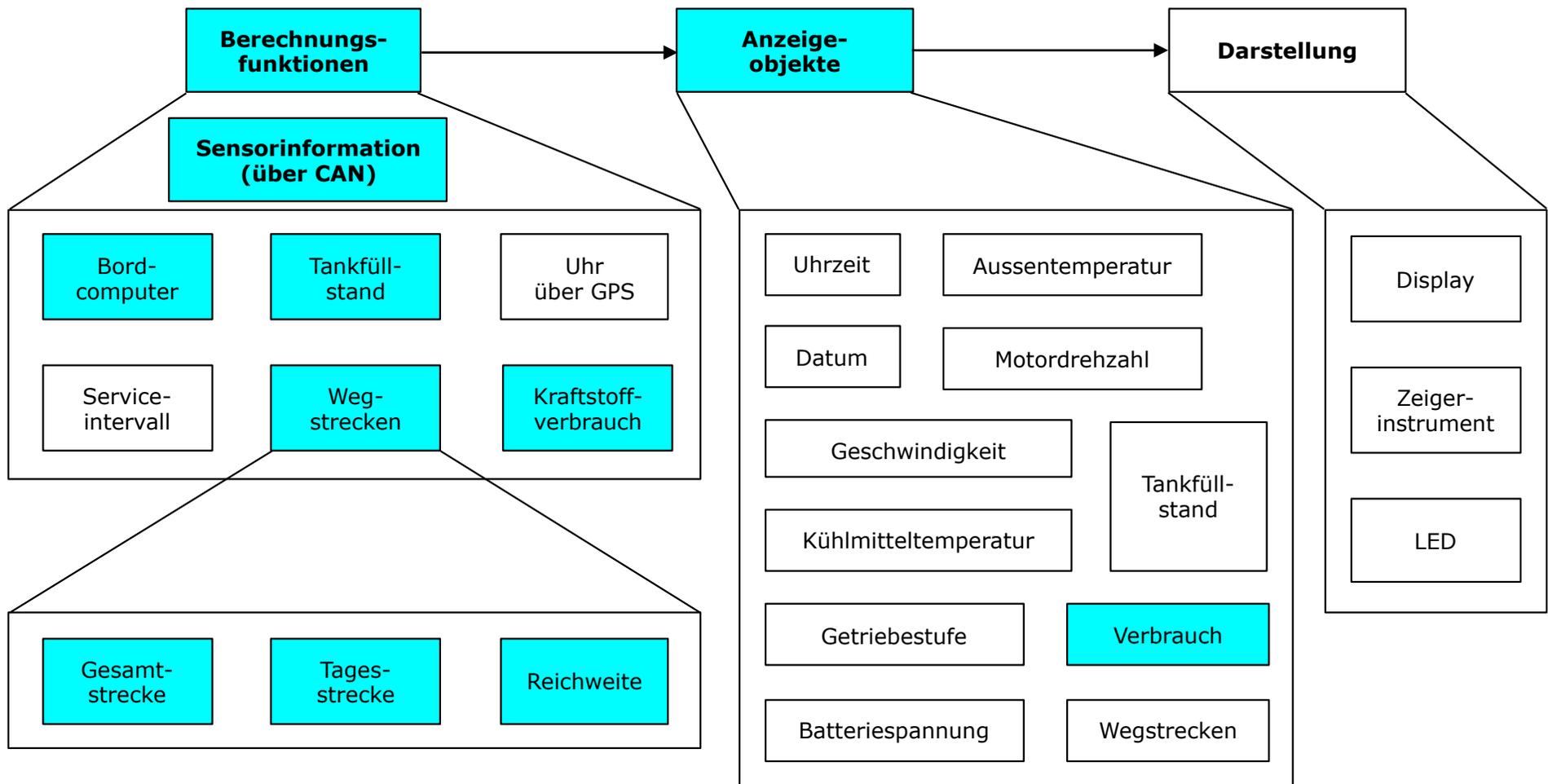
Kombiinstrument: Logische Systemarchitektur (10)



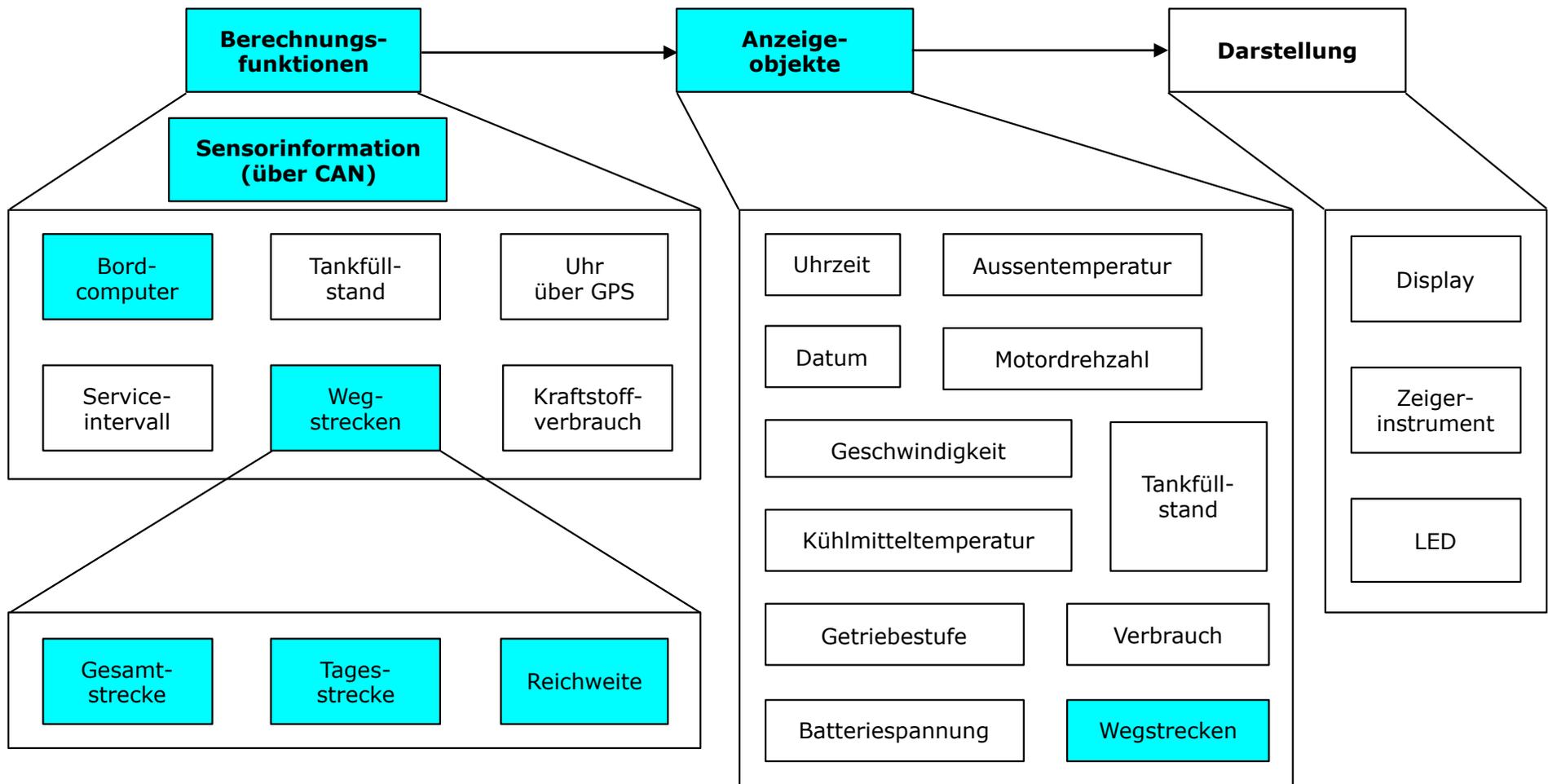
Kombiinstrument: Logische Systemarchitektur (11)



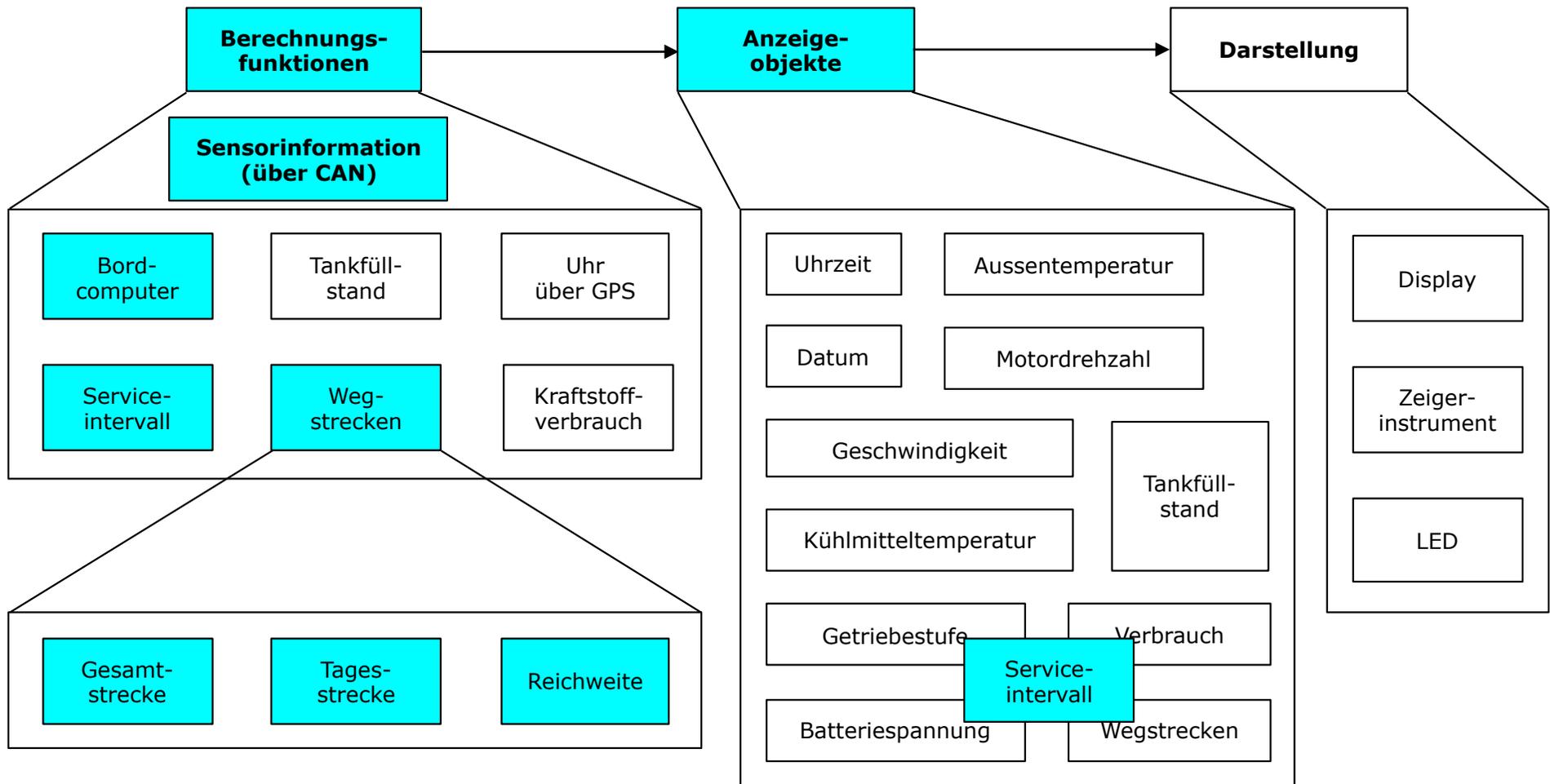
Kombiinstrument: Logische Systemarchitektur (12)



Kombiinstrument: Logische Systemarchitektur (13)



Kombiinstrument: Logische Systemarchitektur (14)



6. SW-Entwicklung / 2. Kernprozess

1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur
4. **Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur**
5. Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest

Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur

Akzeptanz- und Systemtest

Anwendungsfälle

Testergebnisse

Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur

Kalibrierung
 Integrationstest des Systems
 Integration der System-Komponenten

Testfälle

Testergebnisse

Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur

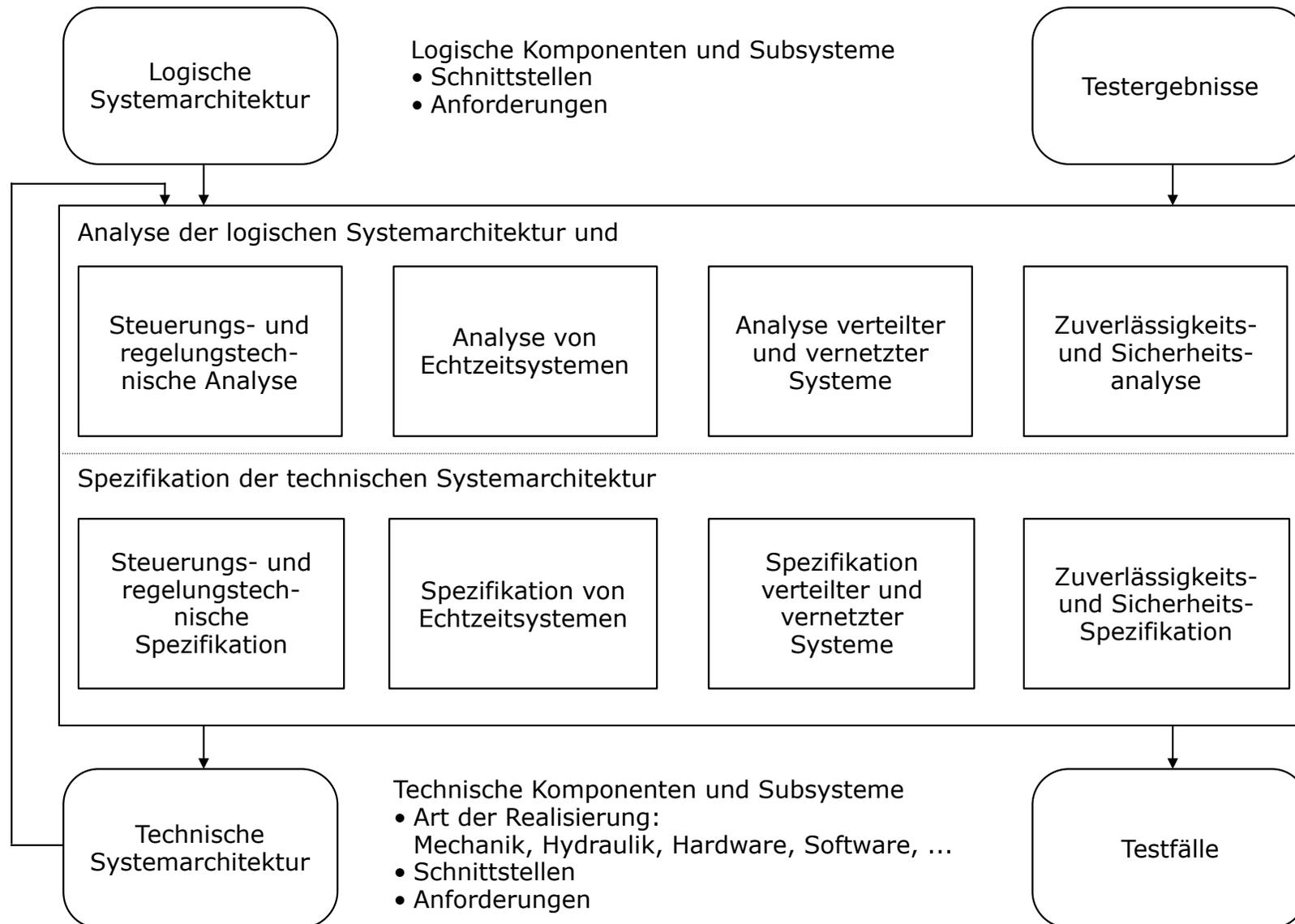
Integrationstest der Software
 Integration der Software-Komponenten

Testfälle

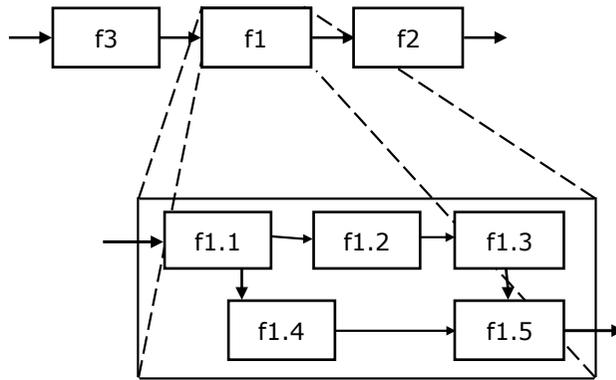
Testergebnisse

Spezifikation der Software-Komponenten
 Design und Implementierung der Software-Komponenten

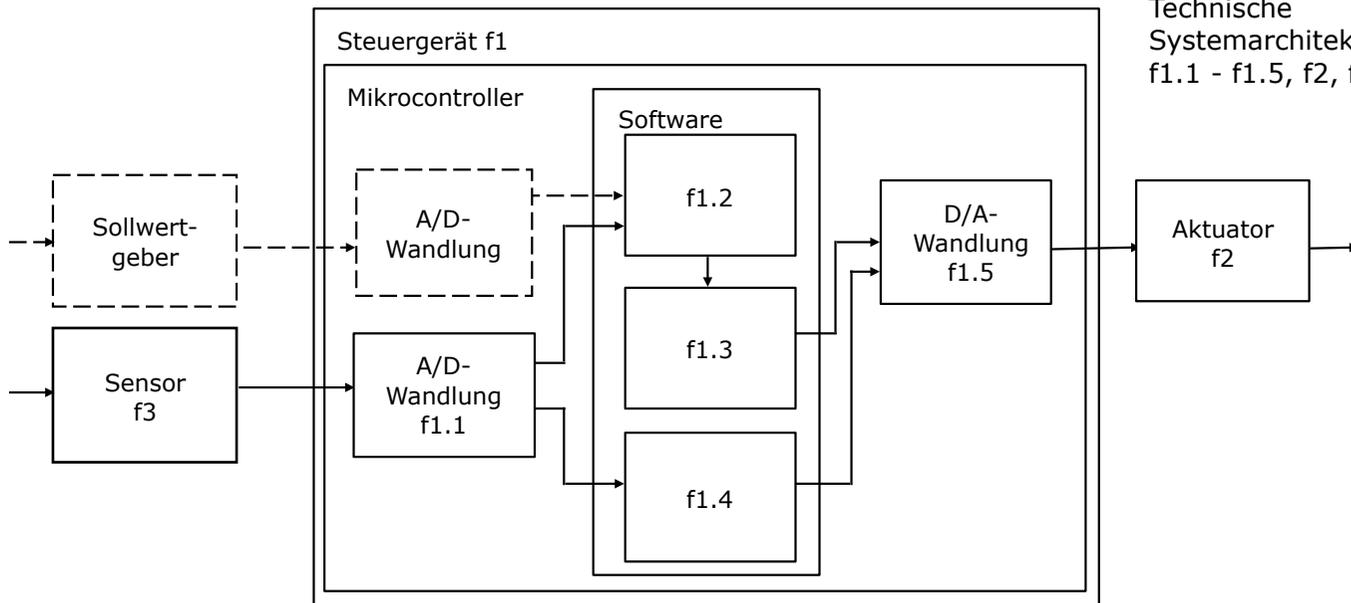
Test der Software-Komponenten



Logische und technische Systemarchitektur (1)

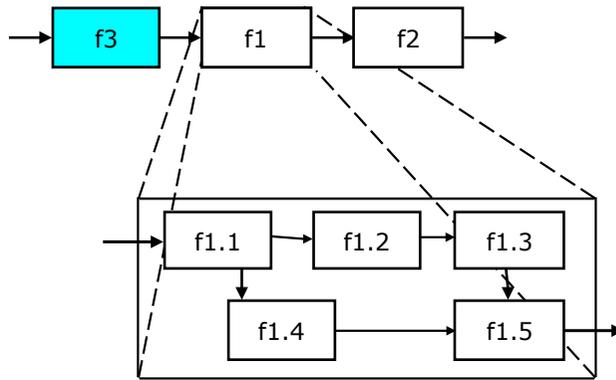


Logische Systemarchitektur für f1, f2, f3

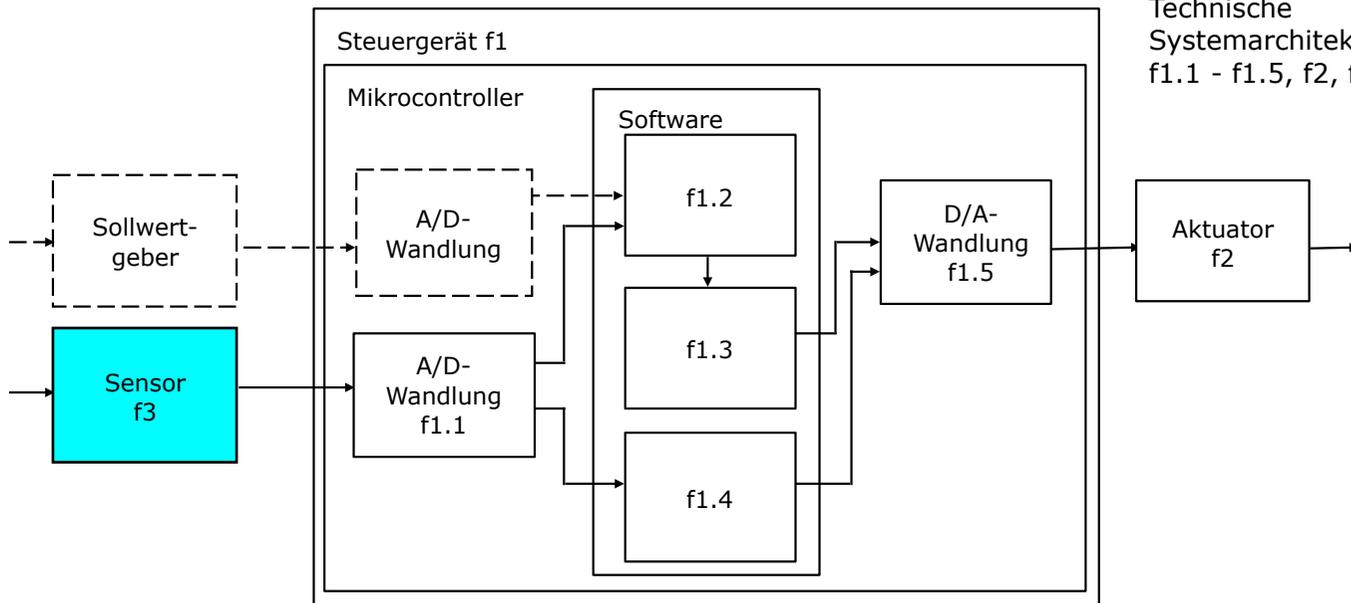


Technische Systemarchitektur für f1.1 - f1.5, f2, f3

Logische und technische Systemarchitektur (2)

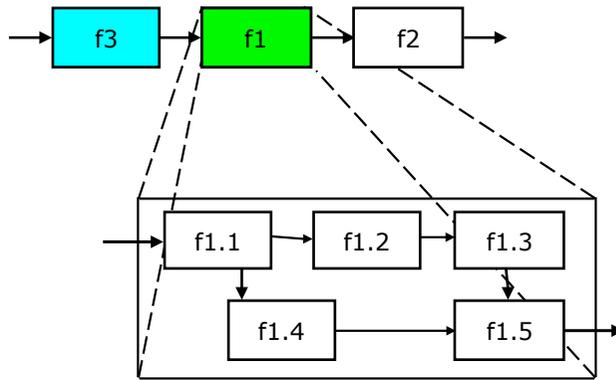


Logische
Systemarchitektur für
f1, f2, f3

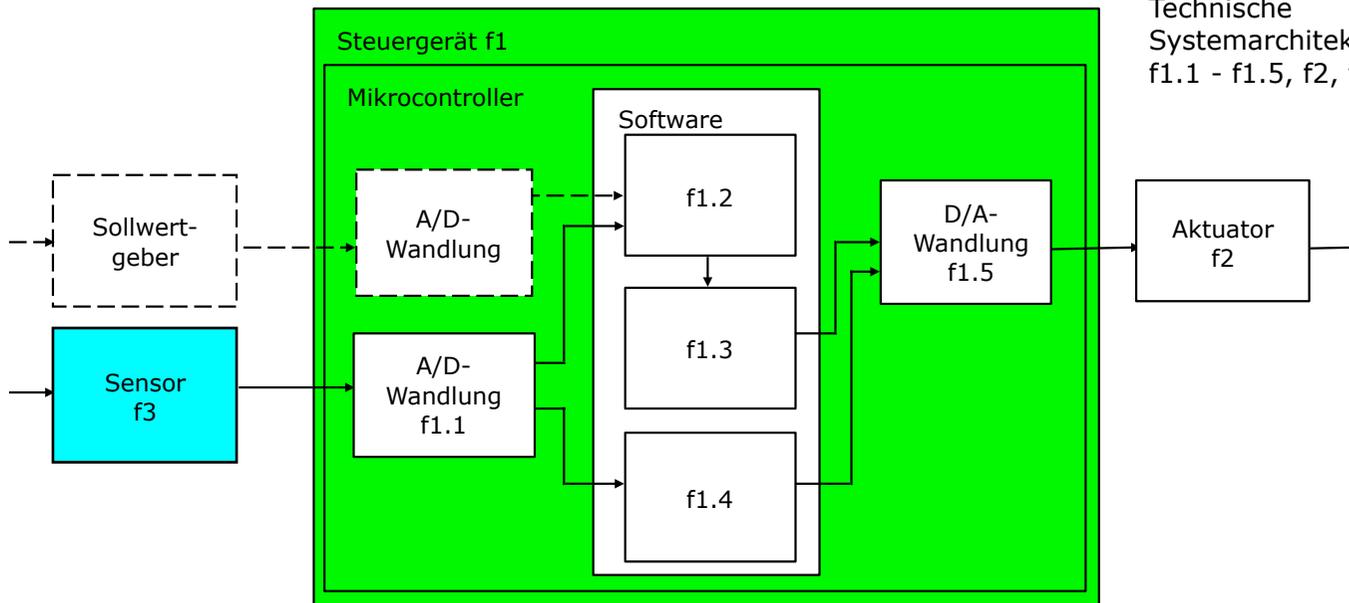


Technische
Systemarchitektur für
f1.1 - f1.5, f2, f3

Logische und technische Systemarchitektur (3)

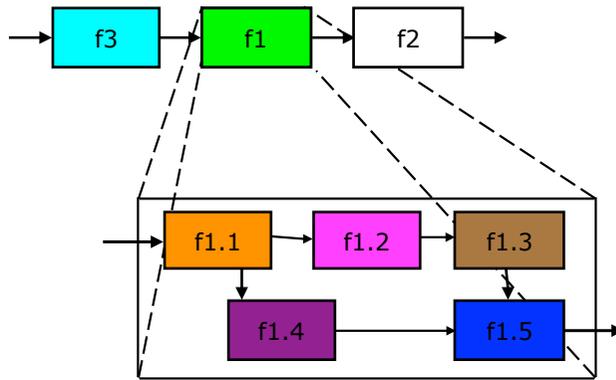


Logische Systemarchitektur für f1, f2, f3

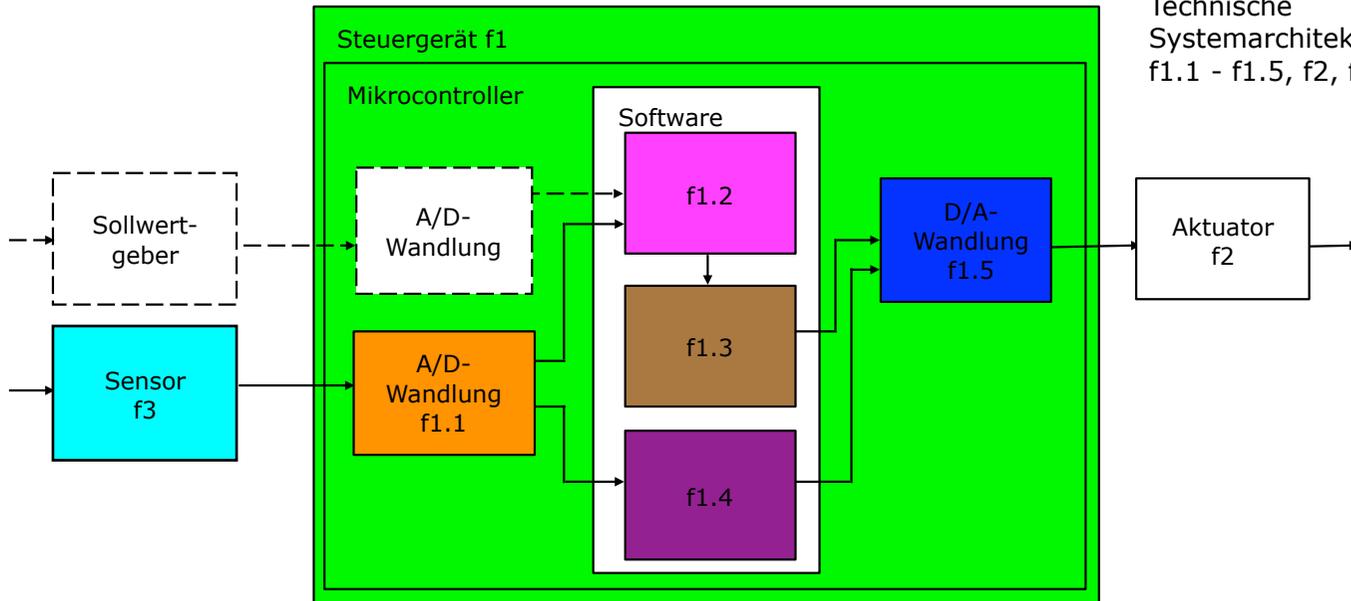


Technische Systemarchitektur für f1.1 - f1.5, f2, f3

Logische und technische Systemarchitektur (4)

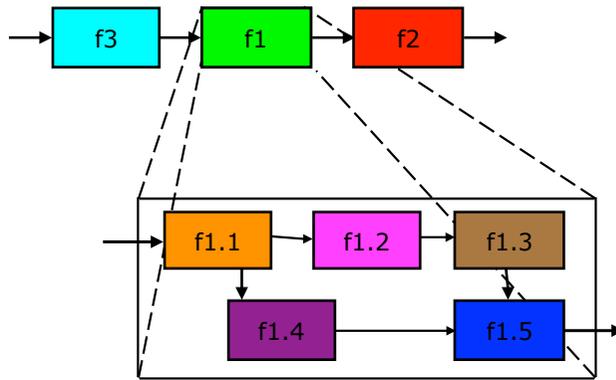


Logische Systemarchitektur für f1, f2, f3

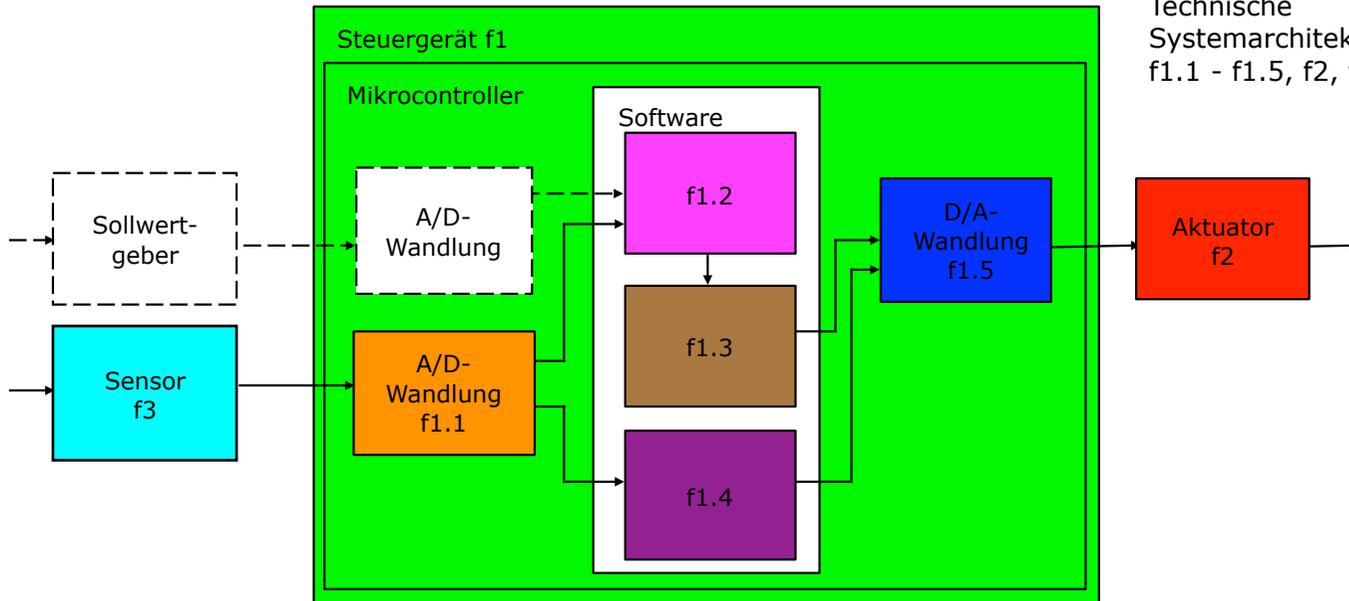


Technische Systemarchitektur für f1.1 - f1.5, f2, f3

Logische und technische Systemarchitektur (5)

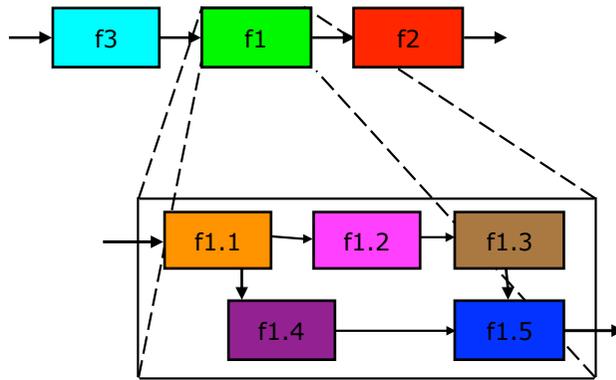


Logische Systemarchitektur für f1, f2, f3

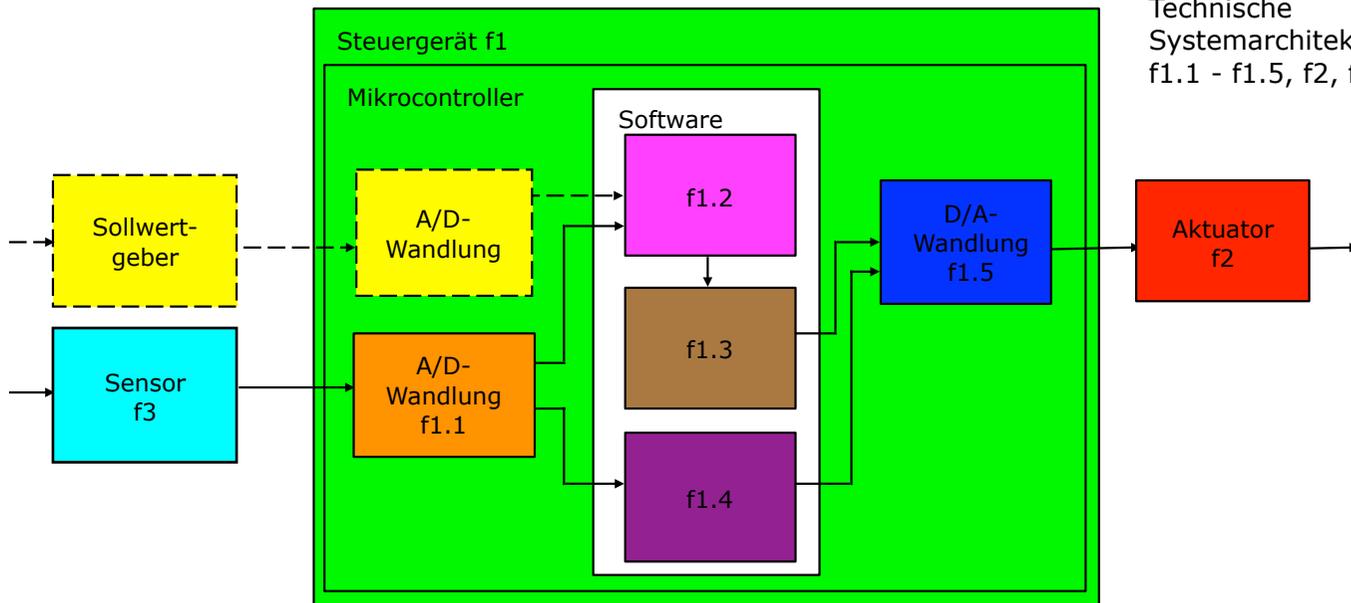


Technische Systemarchitektur für f1.1 - f1.5, f2, f3

Logische und technische Systemarchitektur (6)



Logische
Systemarchitektur für
f1, f2, f3



Technische
Systemarchitektur für
f1.1 - f1.5, f2, f3

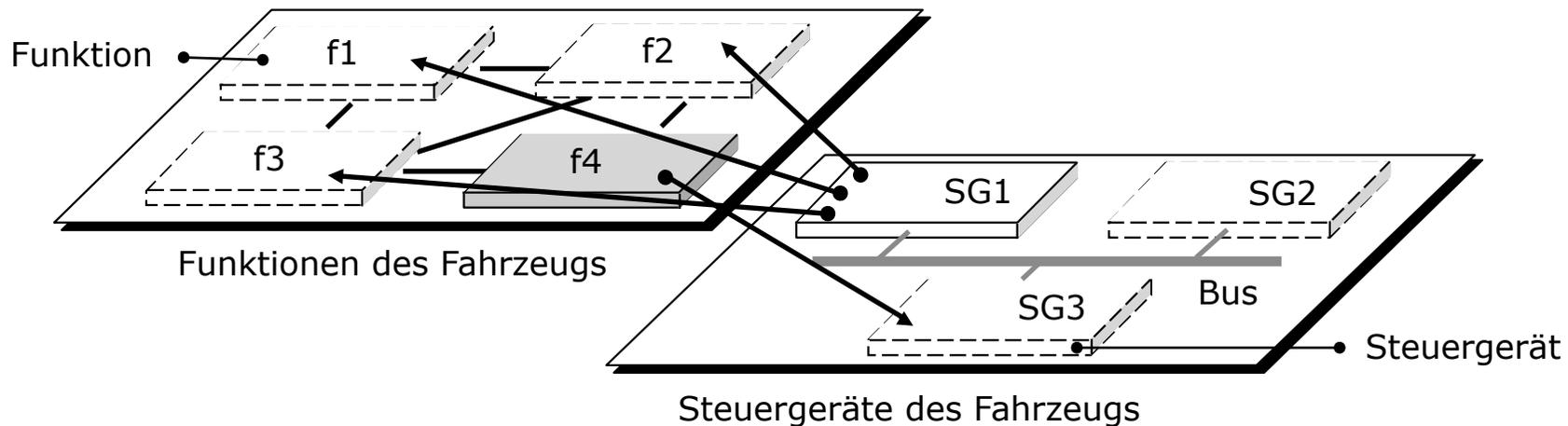
Logische und technische Systemarchitektur

- Randbedingungen für die technische Systemarchitektur
 - Standards und Entwurfsmuster
 - Abhängigkeiten zwischen verschiedenen Systemen und Komponenten
 - Ergebnisse von Machbarkeitsstudien
 - Produktions- und Serviceanforderungen
 - Änderbarkeits- und Testbarkeitsanforderungen
 - Aufwands-, Kosten- und Risikoabschätzungen
- Beispiele
 - Wiederverwendung von technischen Komponenten in verschiedenen Fahrzeugbaureihen
 - Verschiedene Fahrzeugvarianten innerhalb einer Fahrzeugbaureihe
 - Sonderausstattung versus Serienausstattung
 - Länderspezifische Ausstattungsvarianten
 - Komponentenorientierte Wiederverwendung

Randbedingungen und Zielkonflikte

- Wiederverwendung von technischen Komponenten in verschiedenen Baureihen
 - Motoren
 - Getriebe
 - Einheitliche Motor- und Getriebesteuergeräte mit unterschiedlichem Programm und Datenstand
- Verschiedene Varianten innerhalb einer Baureihe
 - Schaltgetriebe
 - Automatikgetriebe
 - Trennung von Motor- und Getriebesteuergerät
- Sonderausstattung und Serienausstattung
 - Serienausstattung
 - Realisierung auf einem Steuergerät
 - Sonderausstattung
 - Regensensor
 - Einparkhilfe
 - Elektrische Sitzverstellung
 - Separate Steuergeräte oder „Softwarefreischaltung“

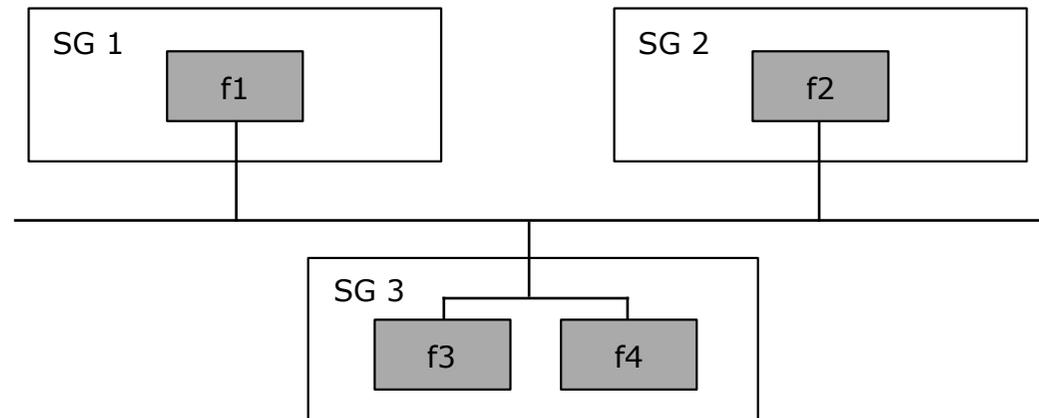
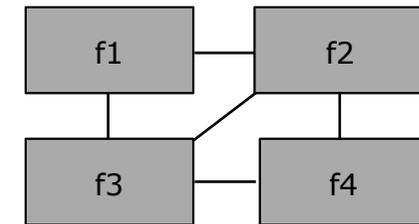
Logische und technische Systemarchitektur



- Komponentenorientierte Wiederverwendung versus funktionale Zerlegung
 - Vorgabe:
 - Wiederverwendung des Steuergerätes SG1 mit den Funktionen f1, f2, f3
 - Freiheitsgrad:
 - Zuordnung der Funktion f4 (z.B. auf SG 3)

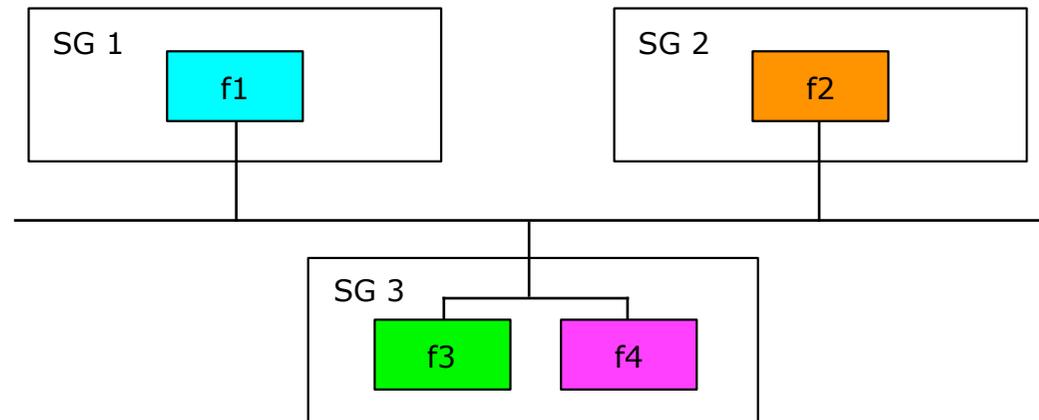
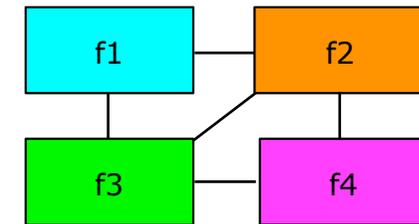
Logische und technische Systemarchitektur

- Logische Systemarchitektur
 - Funktionen
 - Logische Kommunikation
- Technische Systemarchitektur
 - Verteilung der Funktionen auf Steuergeräte
 - Kommunikation
 - Im Steuergerät
 - Zwischen Steuergeräten



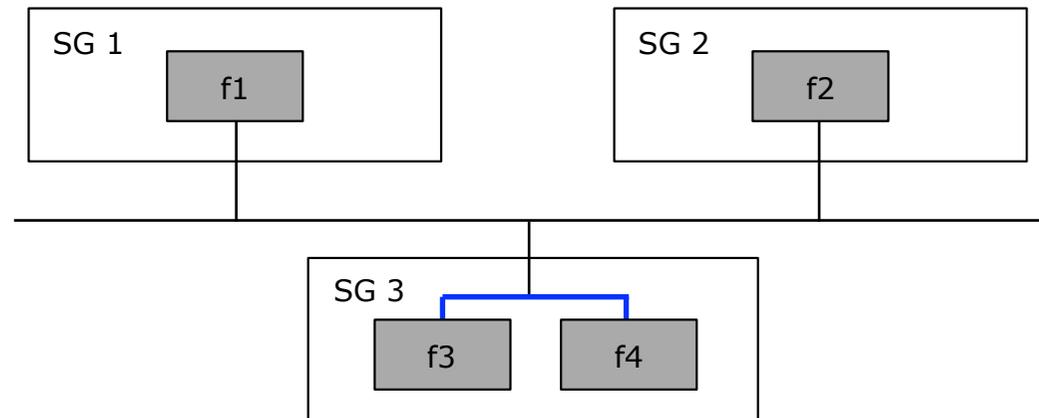
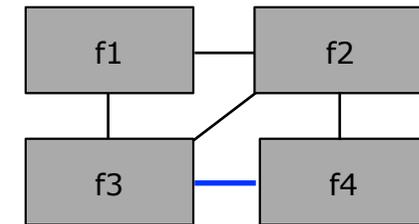
Logische und technische Systemarchitektur

- Logische Systemarchitektur
 - **Funktionen**
 - Logische Kommunikation
- Technische Systemarchitektur
 - **Verteilung der Funktionen auf Steuergeräte**
 - Kommunikation
 - Im Steuergerät
 - Zwischen Steuergeräten



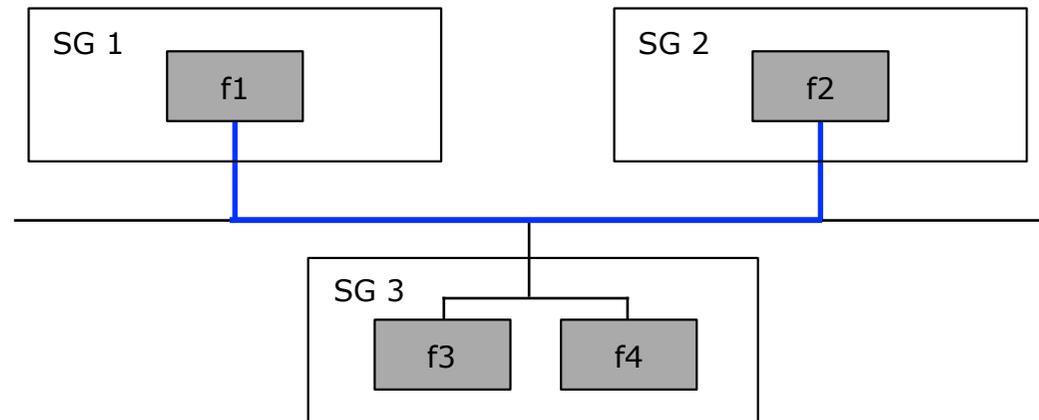
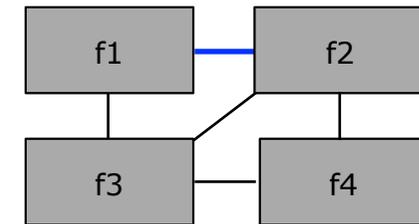
Logische und technische Systemarchitektur

- Logische Systemarchitektur
 - Funktionen
 - **Logische Kommunikation**
- Technische Systemarchitektur
 - Verteilung der Funktionen auf Steuergeräte
 - Kommunikation
 - **Im Steuergerät**
 - Zwischen Steuergeräten



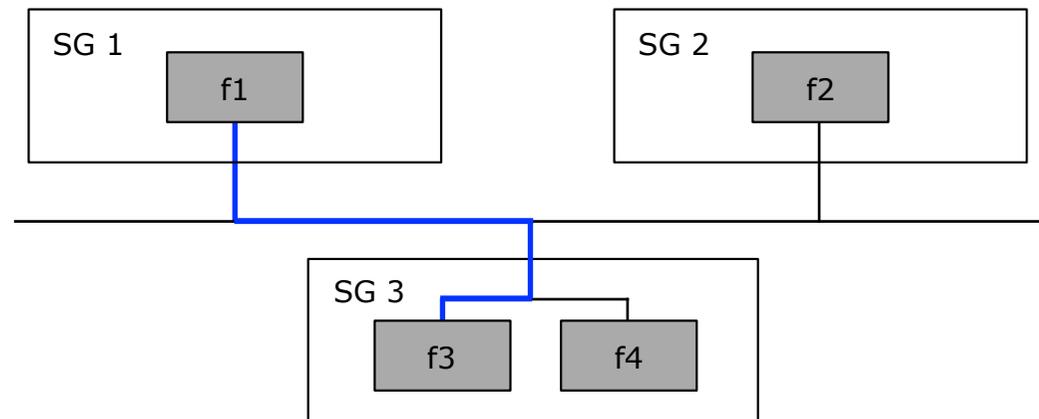
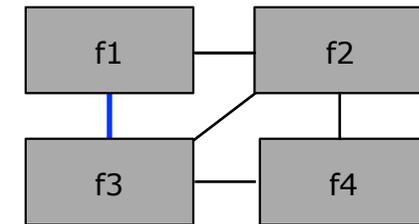
Logische und technische Systemarchitektur

- Logische Systemarchitektur
 - Funktionen
 - **Logische Kommunikation**
- Technische Systemarchitektur
 - Verteilung der Funktionen auf Steuergeräte
 - Kommunikation
 - Im Steuergerät
 - **Zwischen Steuergeräten**



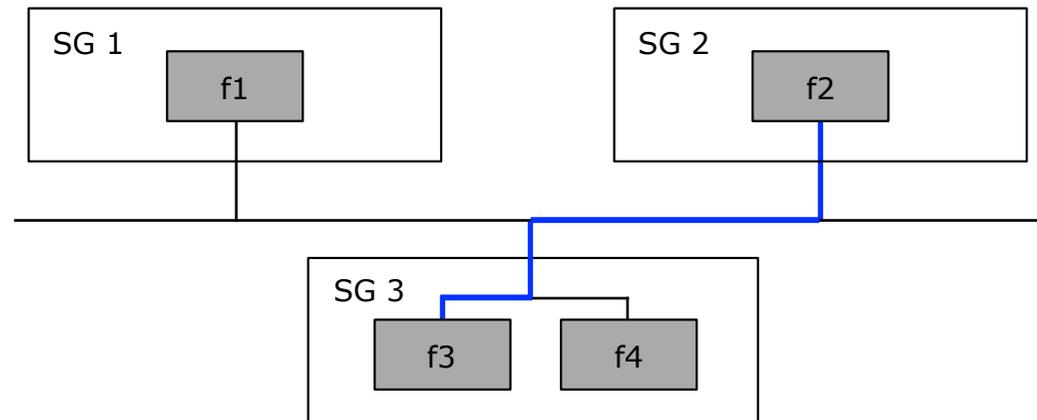
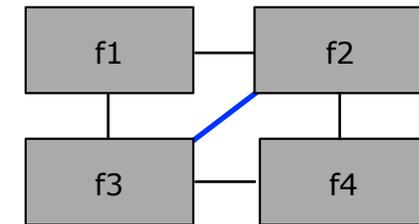
Logische und technische Systemarchitektur

- Logische Systemarchitektur
 - Funktionen
 - **Logische Kommunikation**
- Technische Systemarchitektur
 - Verteilung der Funktionen auf Steuergeräte
 - Kommunikation
 - Im Steuergerät
 - **Zwischen Steuergeräten**



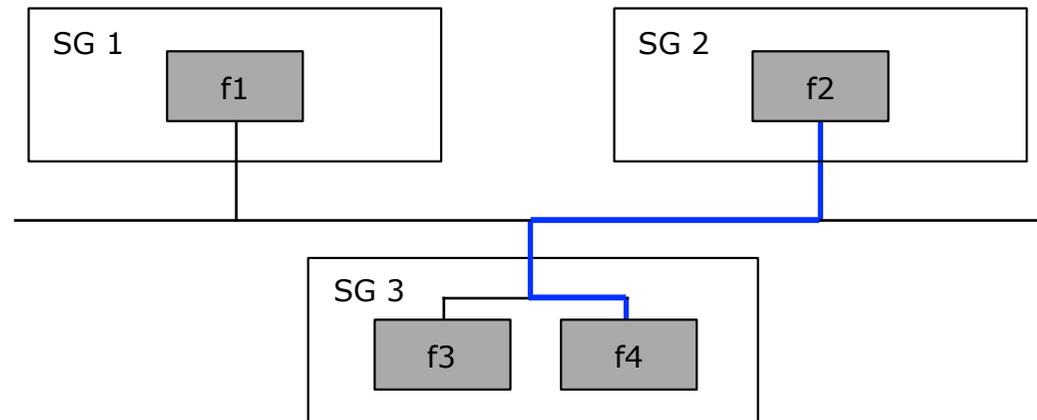
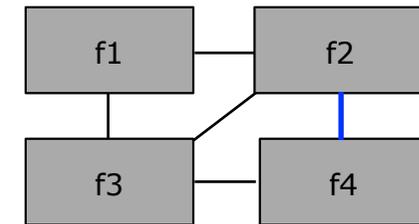
Logische und technische Systemarchitektur

- Logische Systemarchitektur
 - Funktionen
 - **Logische Kommunikation**
- Technische Systemarchitektur
 - Verteilung der Funktionen auf Steuergeräte
 - Kommunikation
 - Im Steuergerät
 - **Zwischen Steuergeräten**



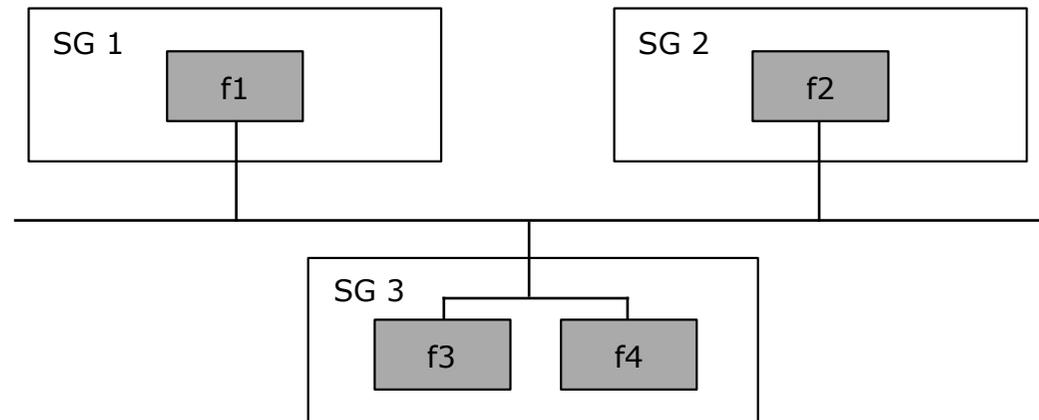
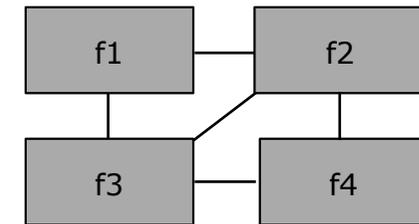
Logische und technische Systemarchitektur

- Logische Systemarchitektur
 - Funktionen
 - **Logische Kommunikation**
- Technische Systemarchitektur
 - Verteilung der Funktionen auf Steuergeräte
 - Kommunikation
 - Im Steuergerät
 - **Zwischen Steuergeräten**



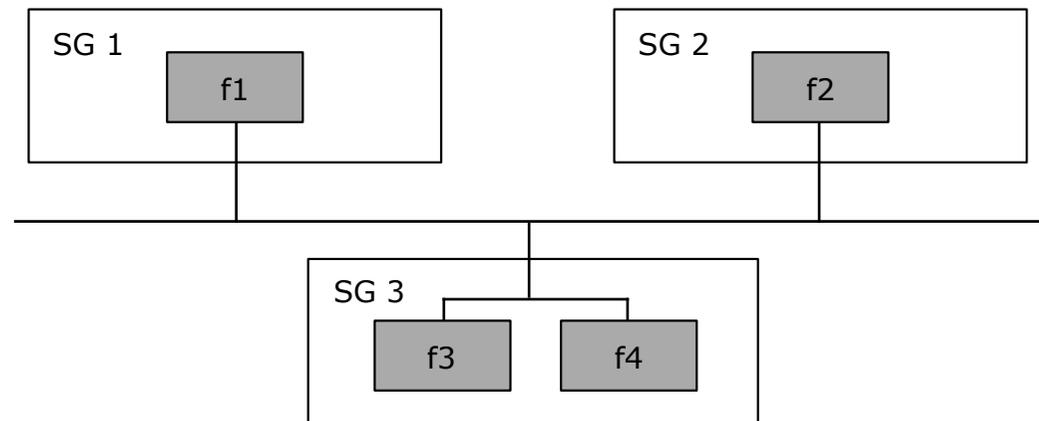
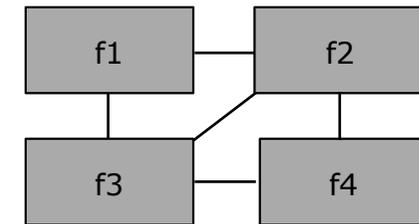
Logische und technische Systemarchitektur

- Logische Systemarchitektur
 - Funktionen
 - Logische Kommunikation
- Technische Systemarchitektur
 - Verteilung der Funktionen auf Steuergeräte
 - Kommunikation
 - Im Steuergerät
 - Zwischen Steuergeräten
- siehe Teil 7 Normen und Standards

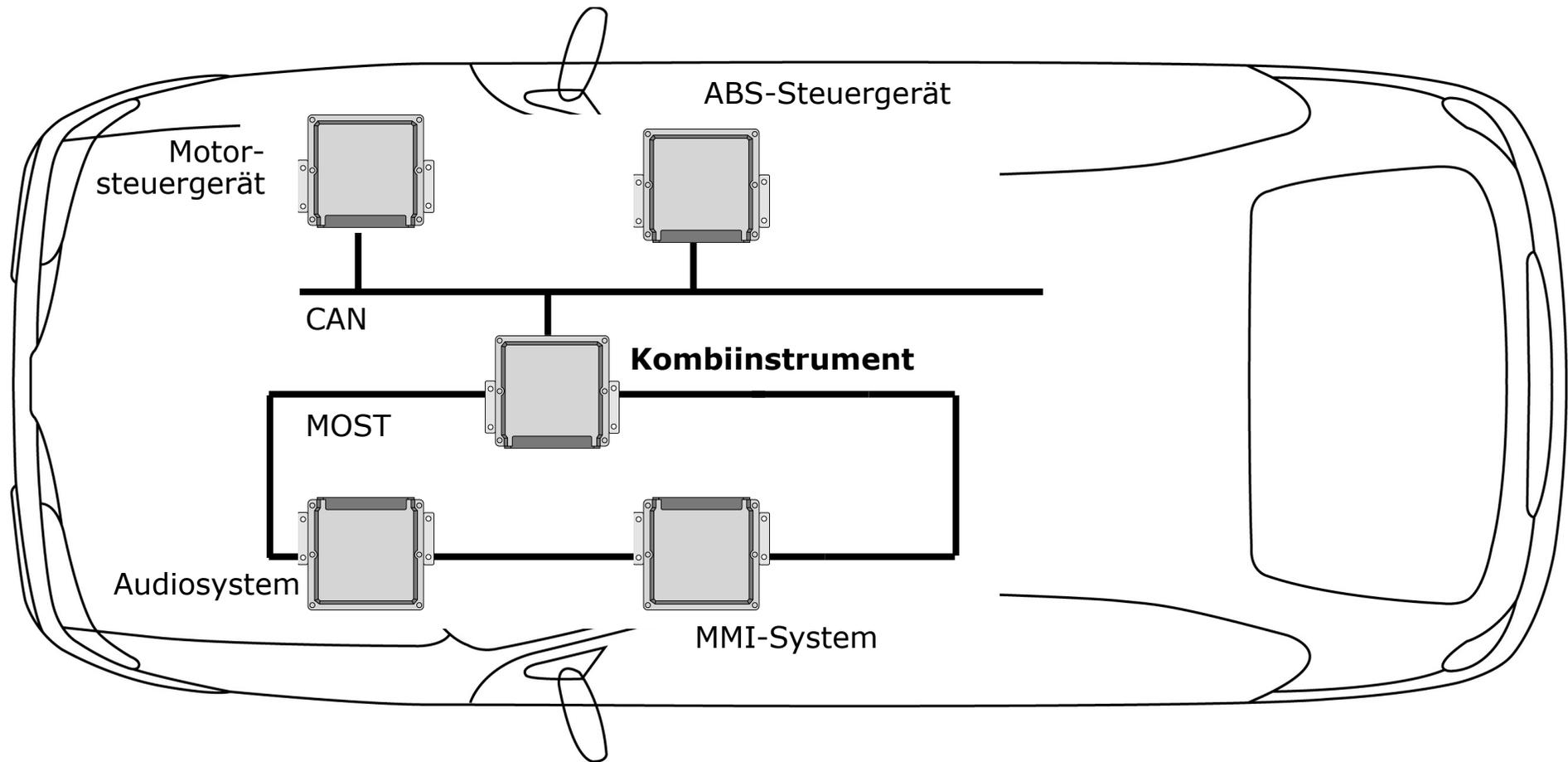


Logische und technische Systemarchitektur

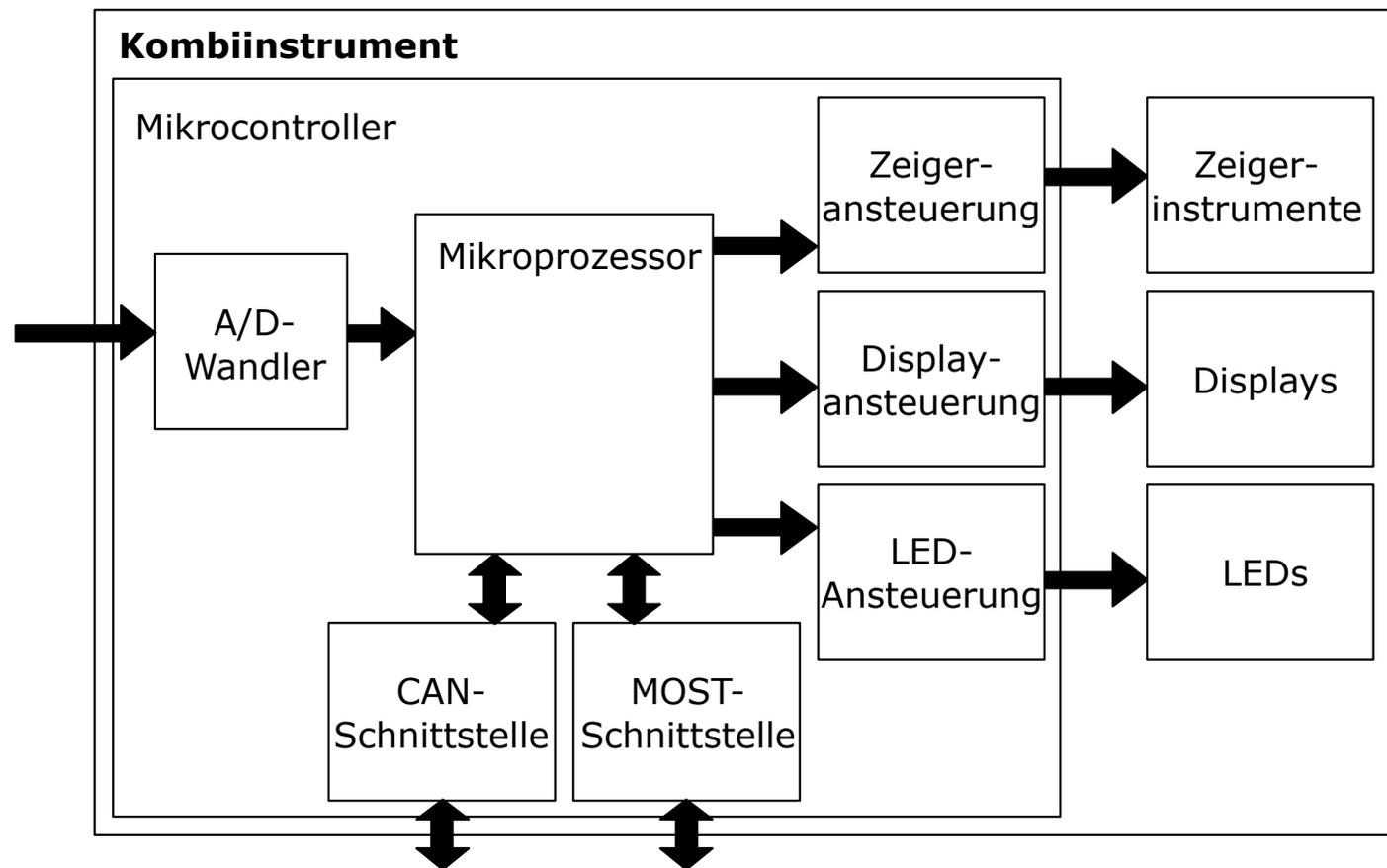
- Logische Systemarchitektur
 - Funktionen
 - Logische Kommunikation
- Technische Systemarchitektur
 - Verteilung der Funktionen auf Steuergeräte
 - Kommunikation
 - Im Steuergerät
 - Zwischen Steuergeräten
- siehe Teil 7 Normen und Standards



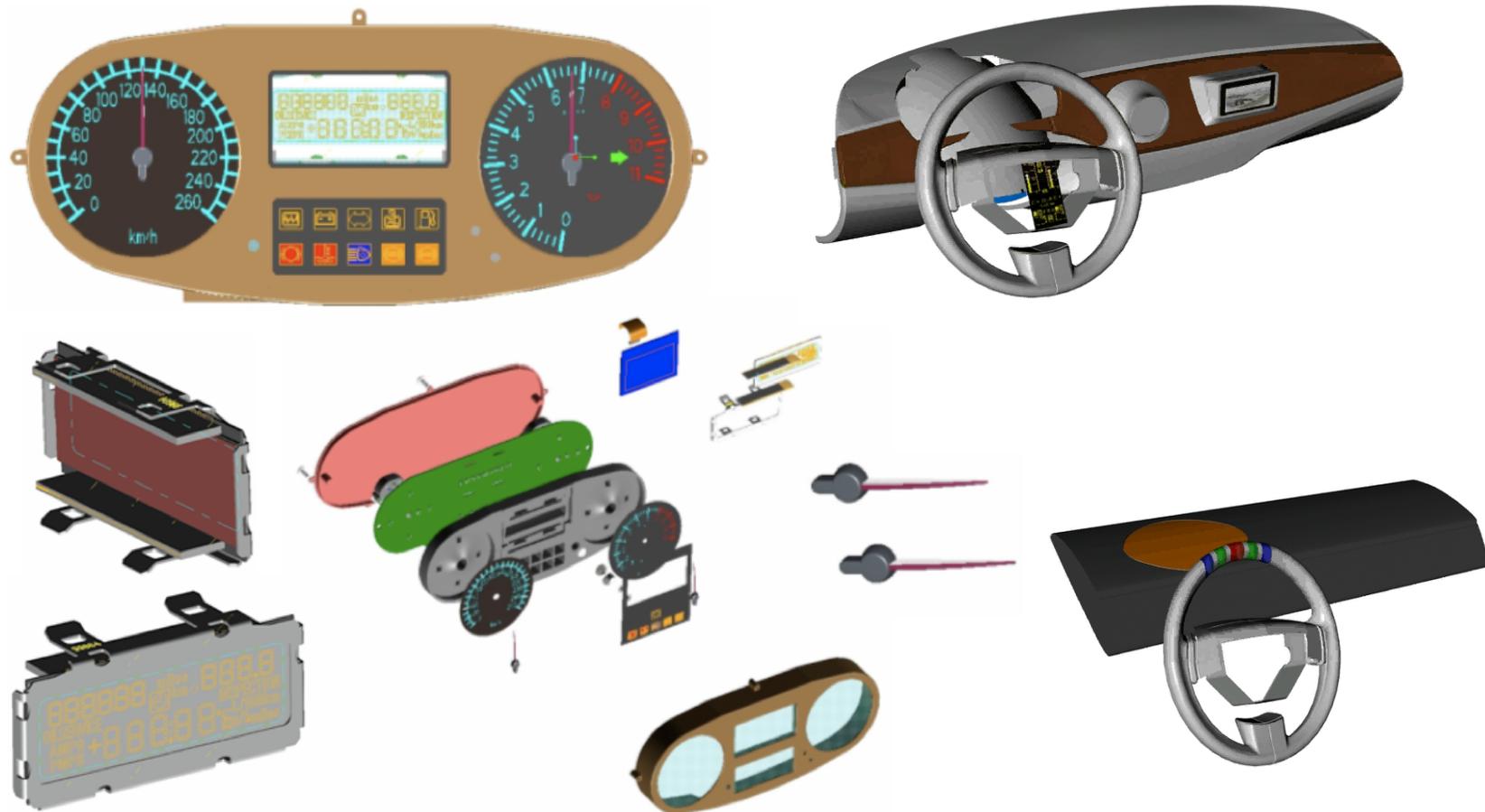
Kombiinstrument im Steuergerätenetzwerk

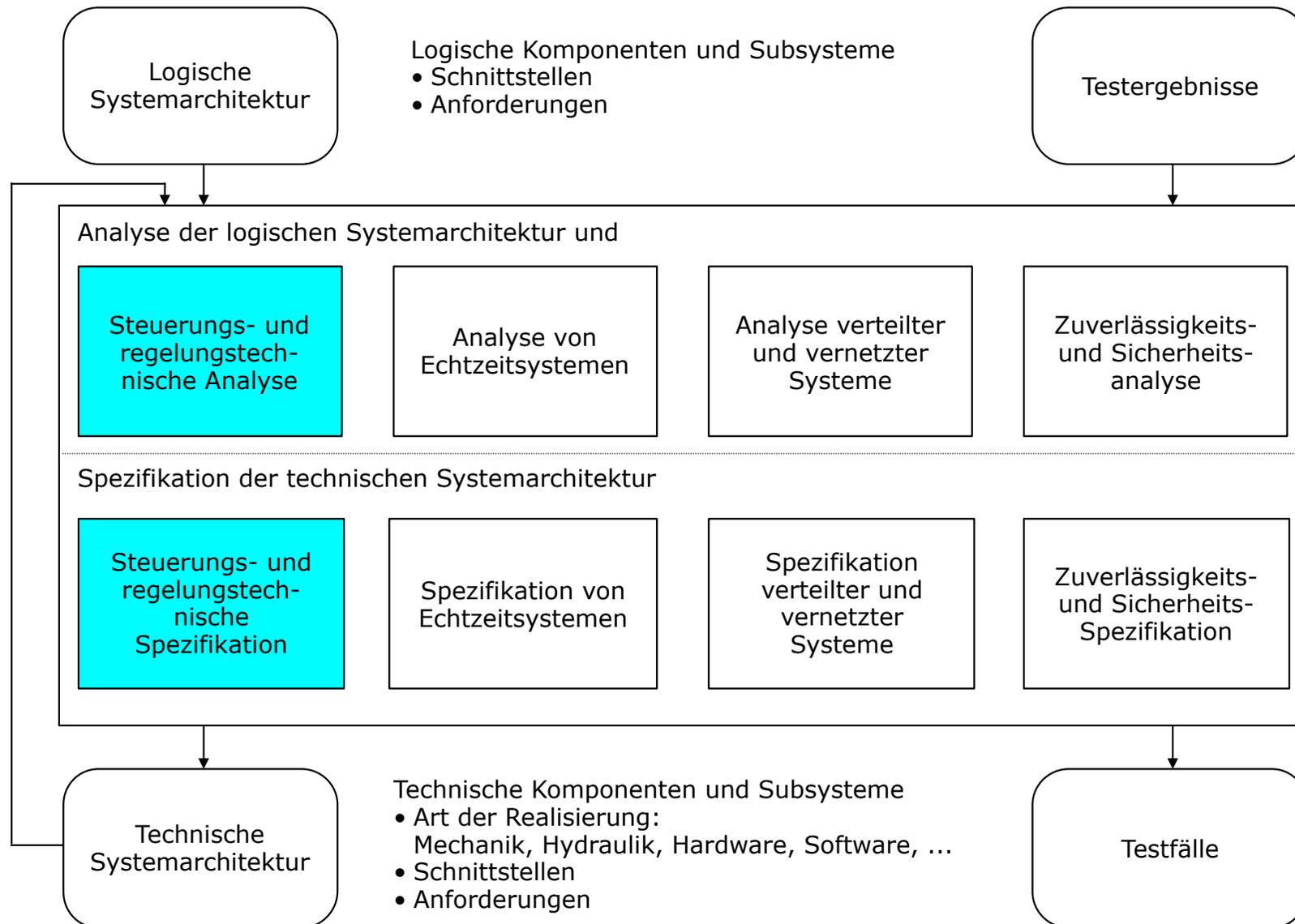


Hardware des Kombiinstrument

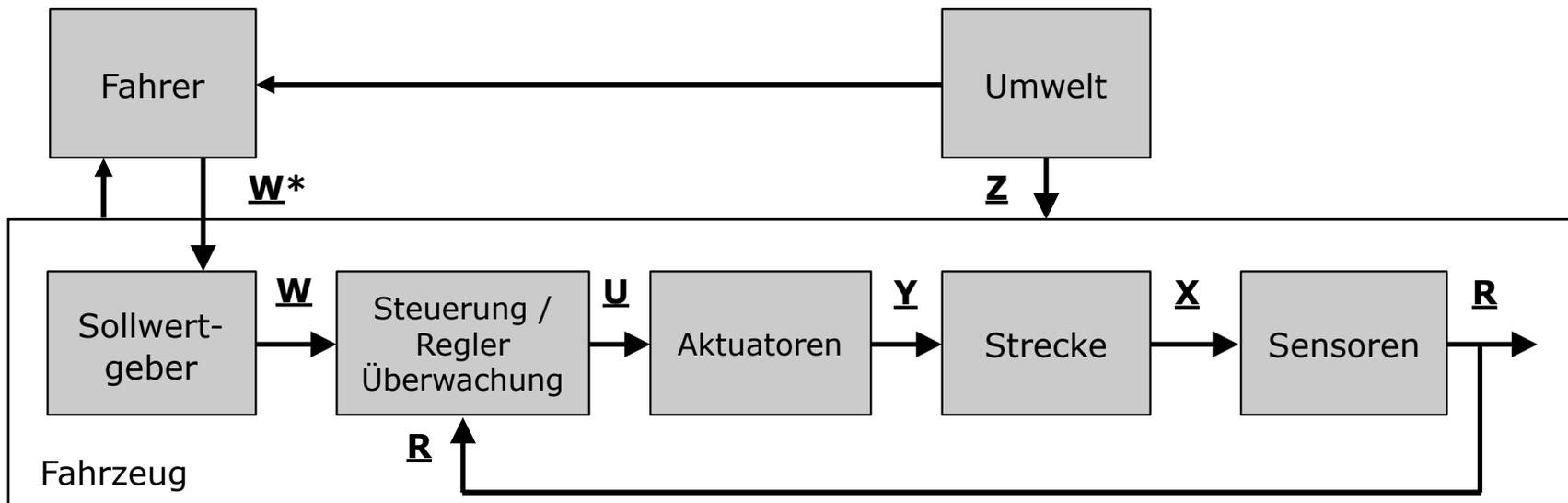


Kombiinstrument

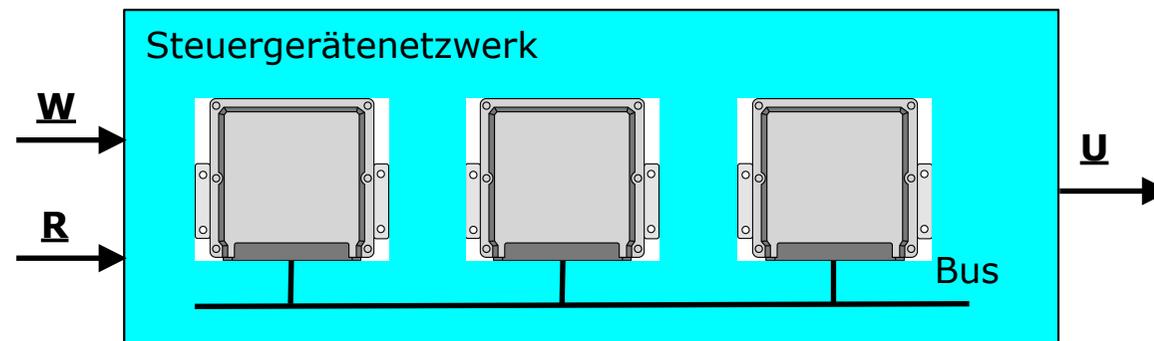
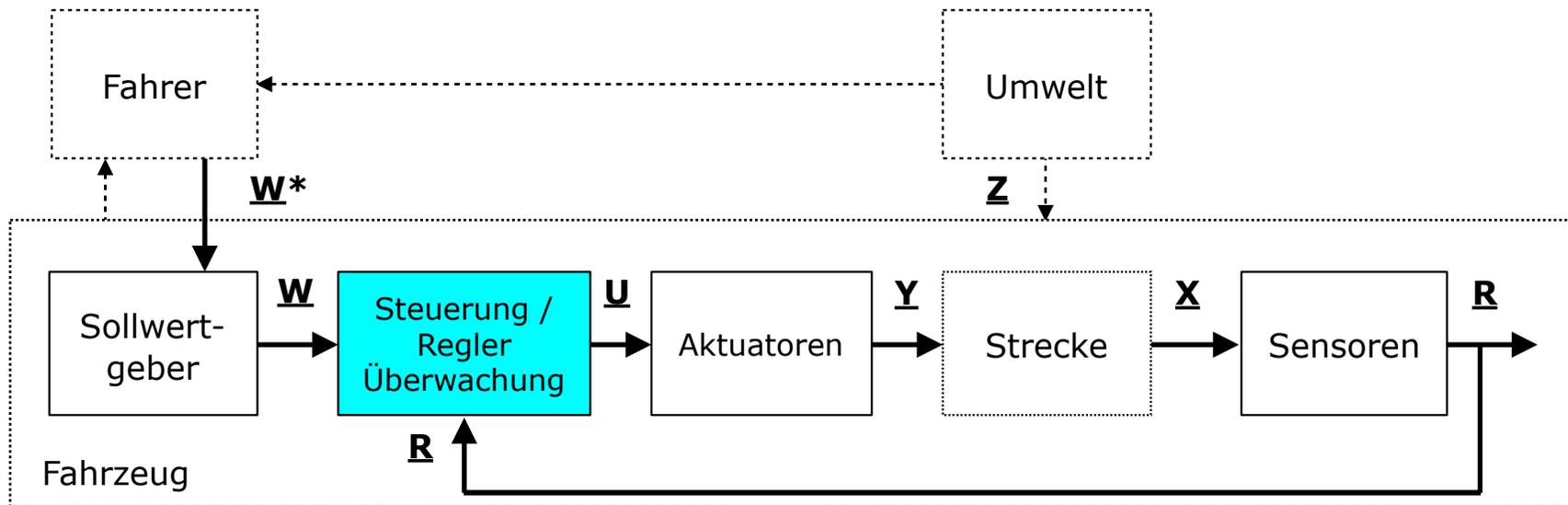


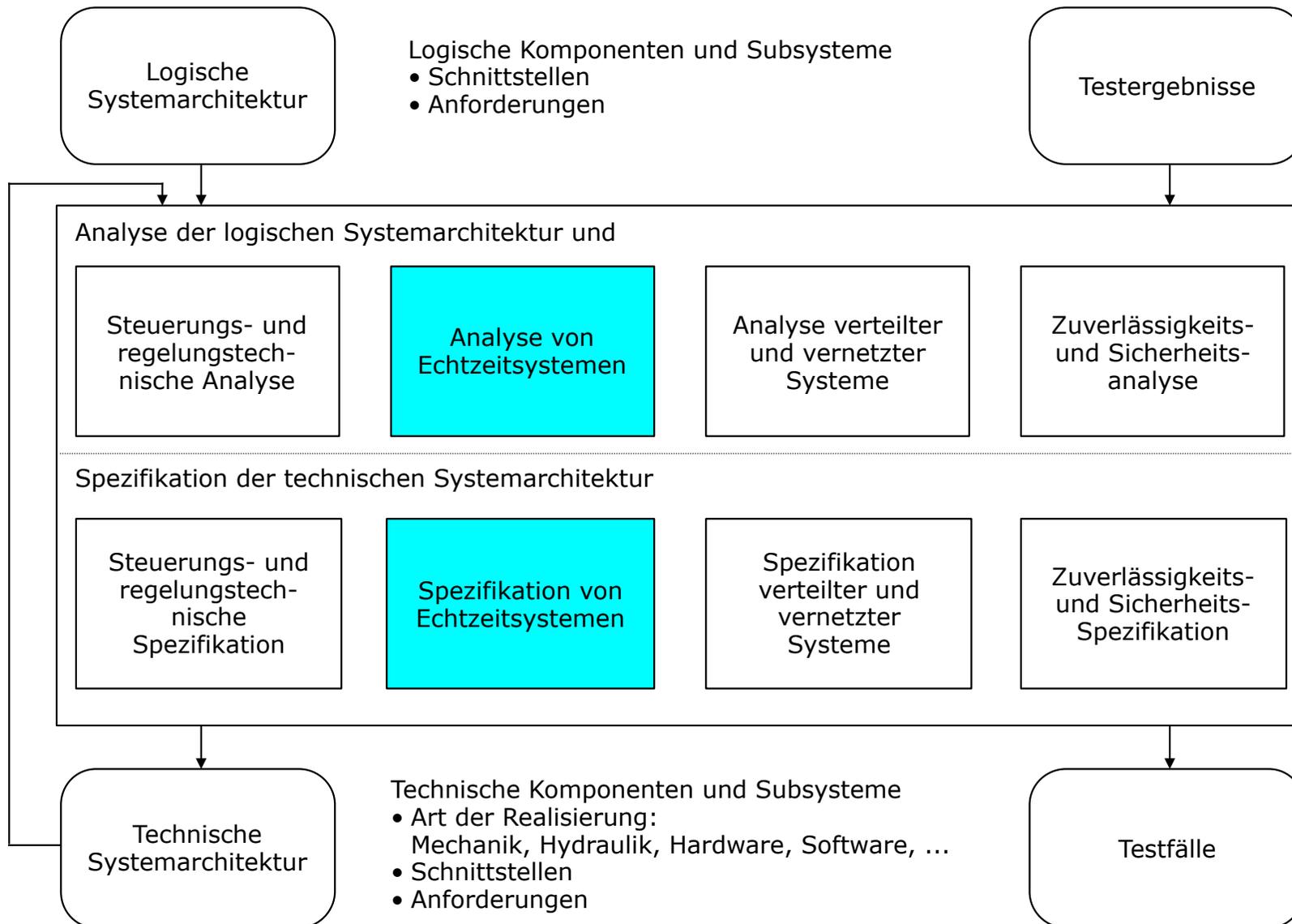


Steuerungs- und Regelungstechnik



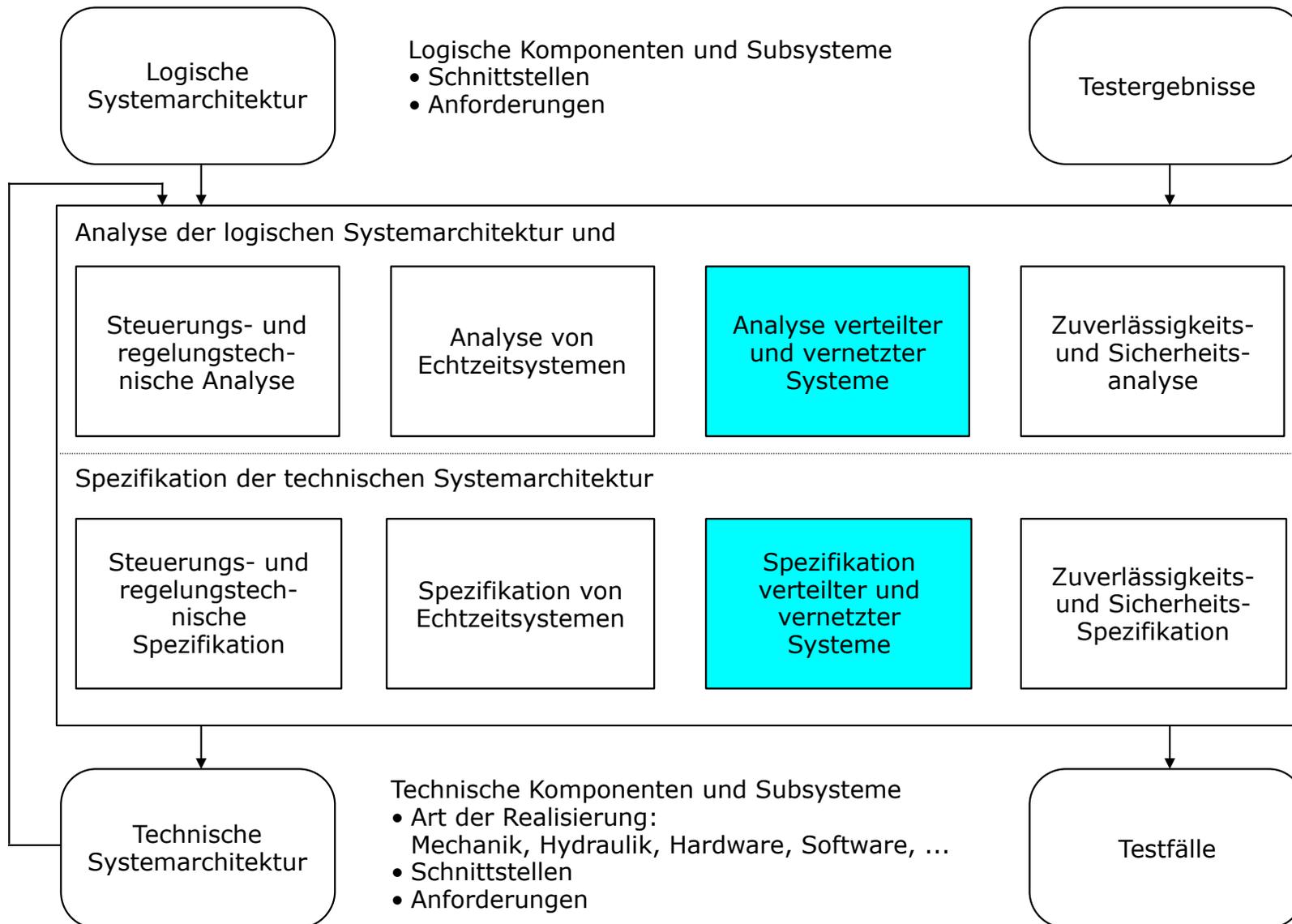
Steuerungs- und Regelungstechnik

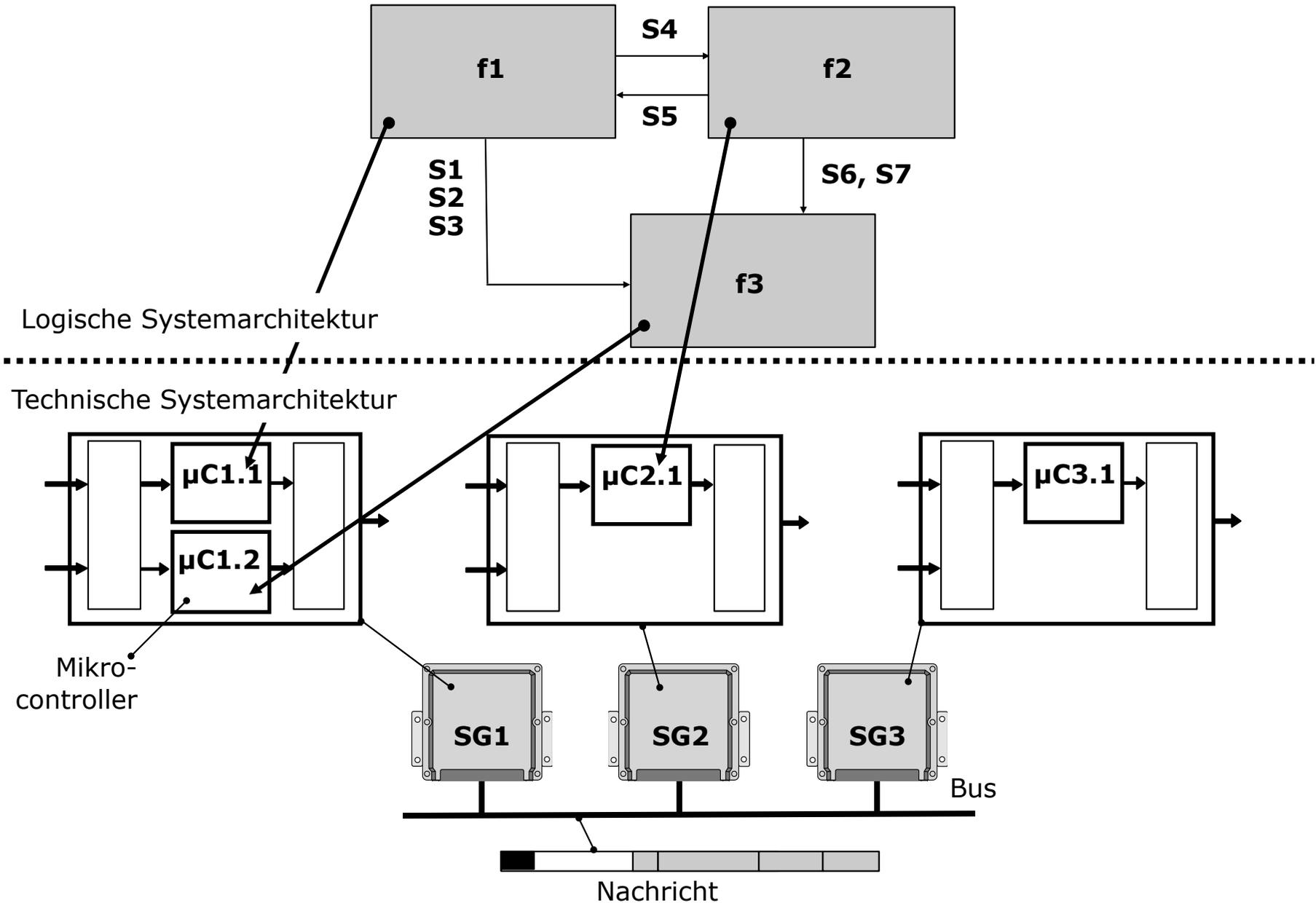




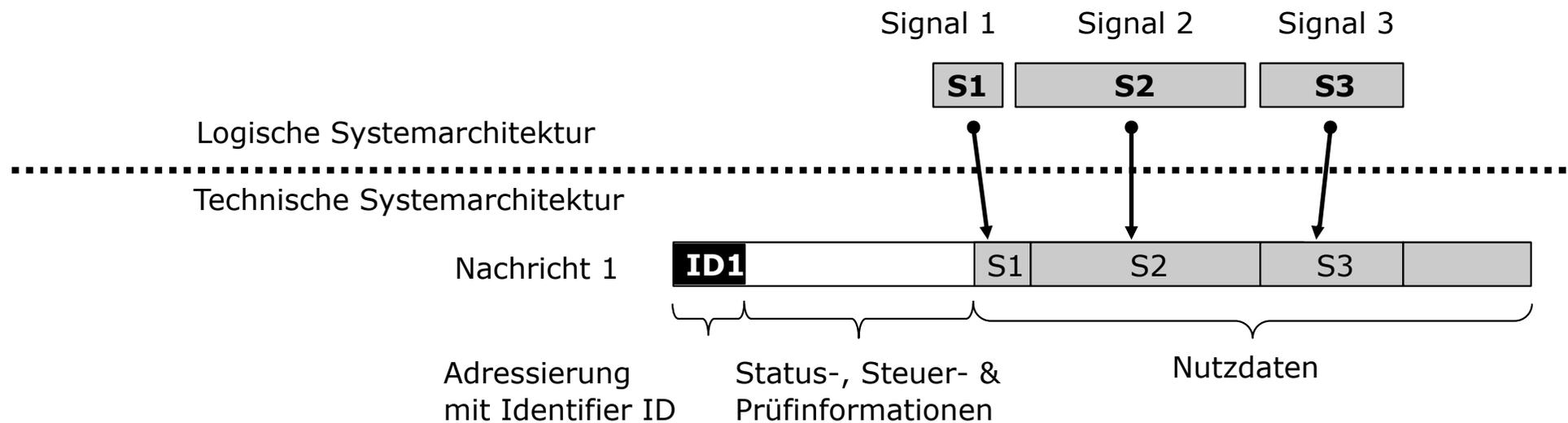
Analyse und Spezifikation von Echtzeitsystemen

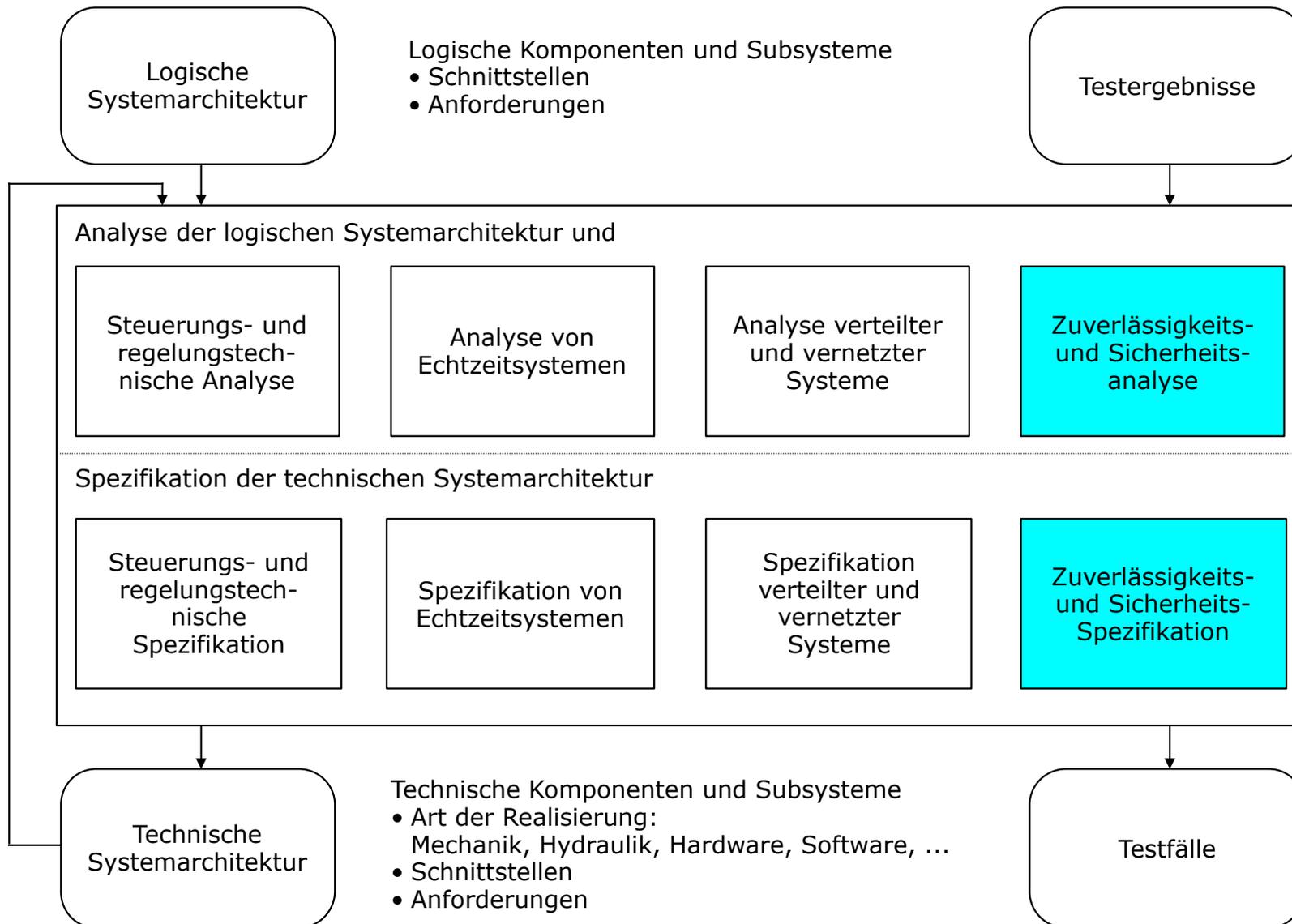
- Bei Analyse und Spezifikation steuerungs- und regelungstechnischer Systeme werden Anforderungen bezüglich der Abstrakte festgelegt.
- Abstrakten bilden die Basis für Echtzeitanforderungen an Software-Funktionen.
- Bei verteilten, vernetzten Systemen ergeben sich aus den Abstrakten auch die Echtzeitanforderungen an das Kommunikationssystem.
- Realisierbarkeit wird mit geeigneten Methoden analysiert
 - Bewertung technischer Realisierungsalternativen
 - Eventuell Korrektur der Konfiguration des Echtzeitsystems
- Betrachtet werden
 - Anwendungssoftware
 - Basissoftware
 - Kommunikationssystem
- Vorlesungen Echtzeitsysteme
 - Prof. Schürr, TU Darmstadt
 - Prof. Härtig, TU Dresden



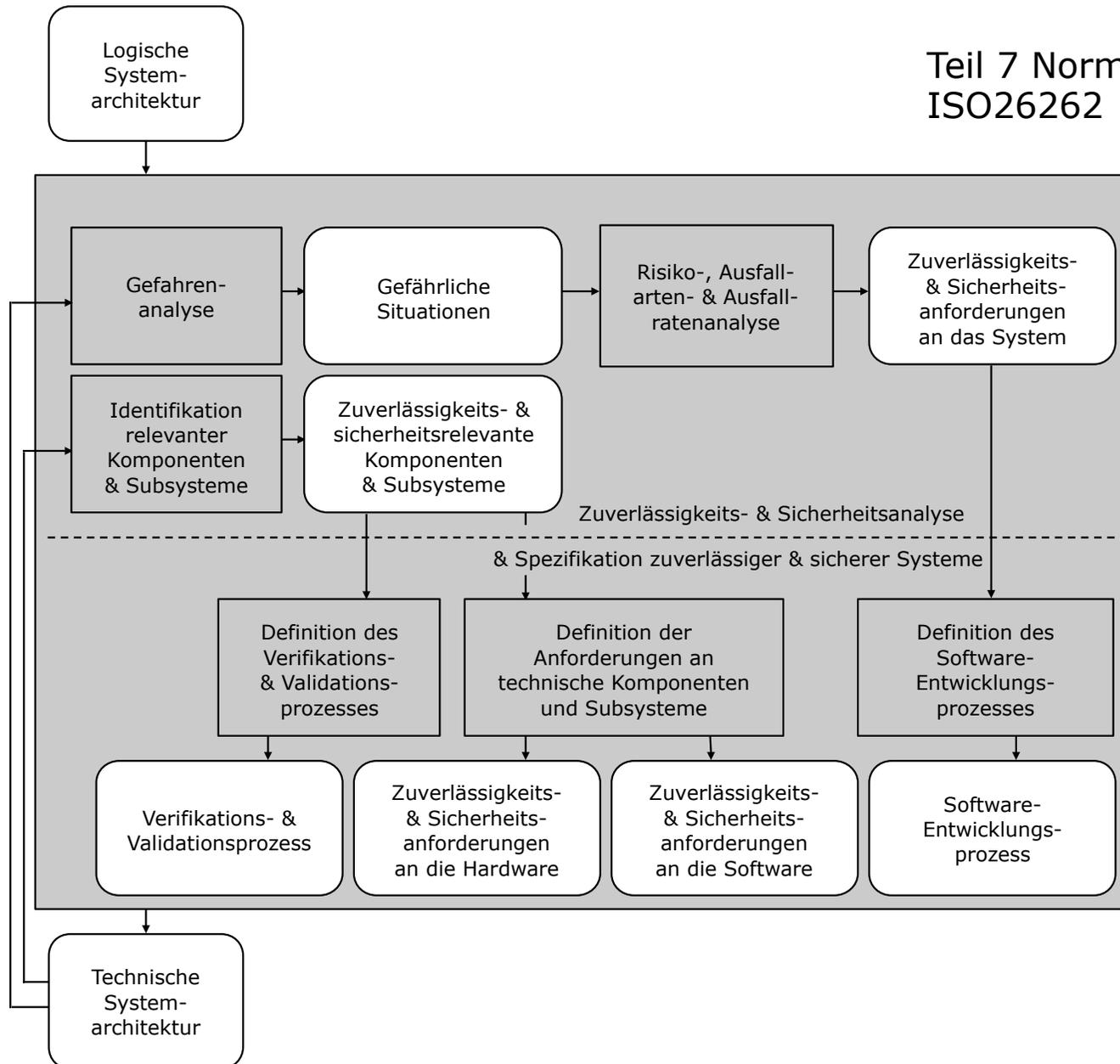


Zuordnung von Signalen zu Nachrichten





Teil 7 Normen und Standards ISO26262



6. SW-Entwicklung / 2. Kernprozess

1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur
4. Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur
5. **Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur**
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest

Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur

Akzeptanz- und Systemtest

Anwendungsfälle

Testergebnisse

Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur

Kalibrierung
 Integrationstest des Systems
 Integration der System-Komponenten

Testfälle

Testergebnisse

Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur

Integrationstest der Software
 Integration der Software-Komponenten

Testfälle

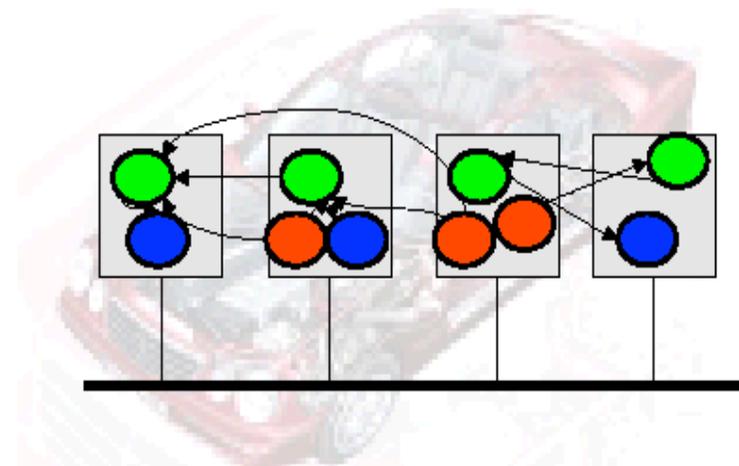
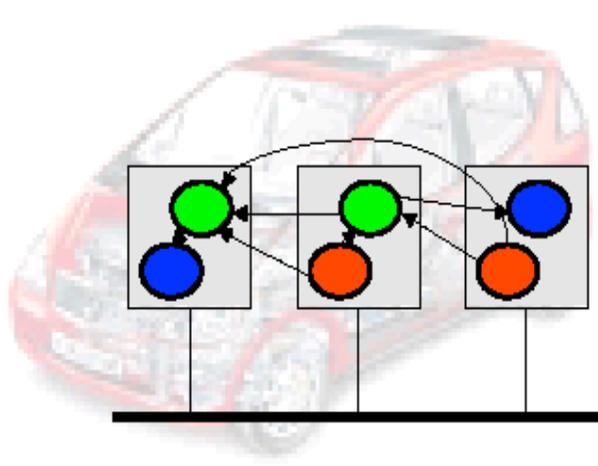
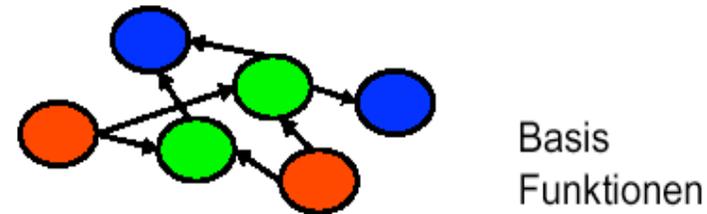
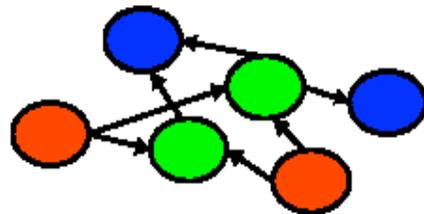
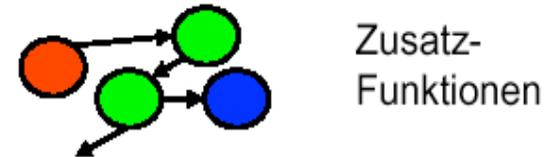
Testergebnisse

Spezifikation der Software-Komponenten
 Design und Implementierung der Software-Komponenten

Test der Software-Komponenten

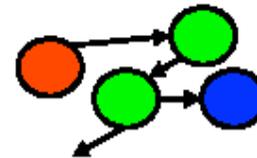
Software Architekturen und Wiederverwendung

Wiederverwendung der Entwürfe
 Wiederverwendung Validation and Verification
 Wiederverwendung Software
 Maximale Nutzung Hardware

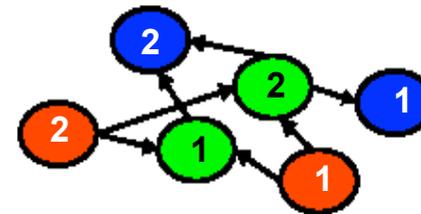
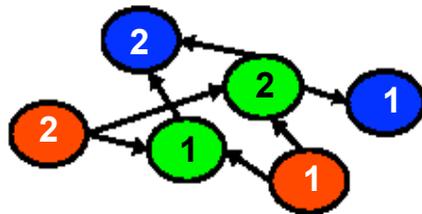


Software Architekturen und Wiederverwendung

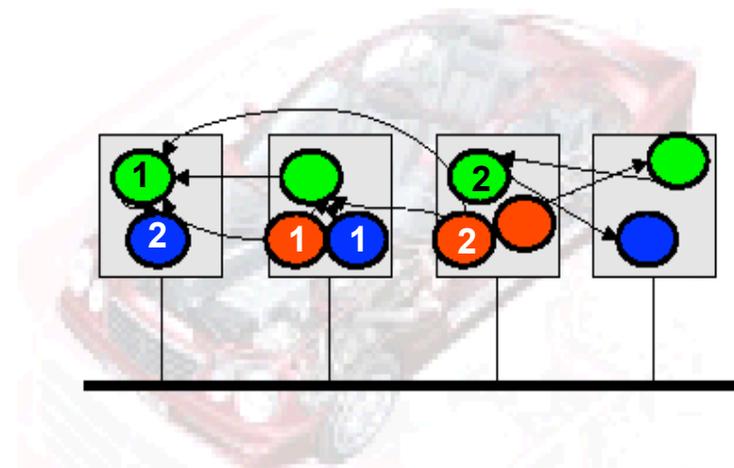
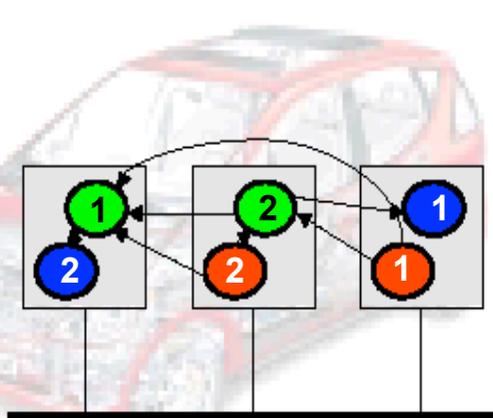
Wiederverwendung der Entwürfe
 Wiederverwendung Validation and Verification
 Wiederverwendung Software
 Maximale Nutzung Hardware



Zusatz-
Funktionen

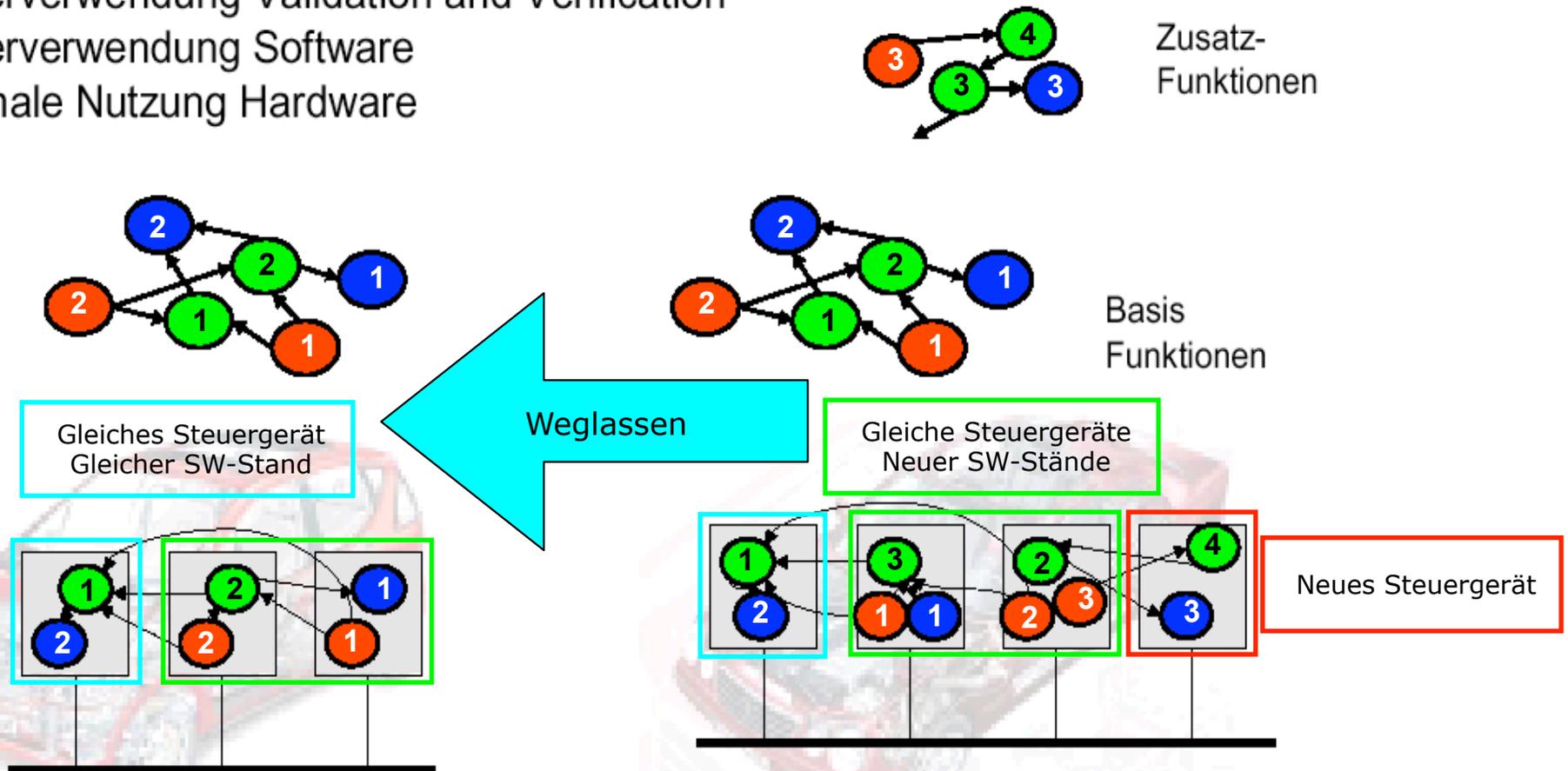


Basis
Funktionen



Software Architekturen und Wiederverwendung

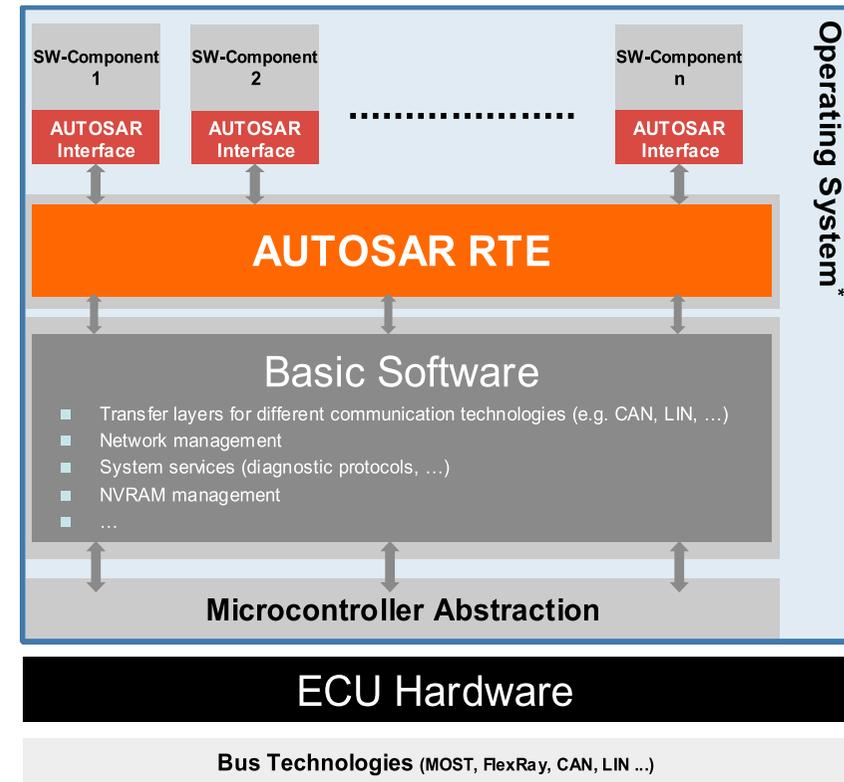
Wiederverwendung der Entwürfe
 Wiederverwendung Validation and Verification
 Wiederverwendung Software
 Maximale Nutzung Hardware



AUTOSAR Komponenten

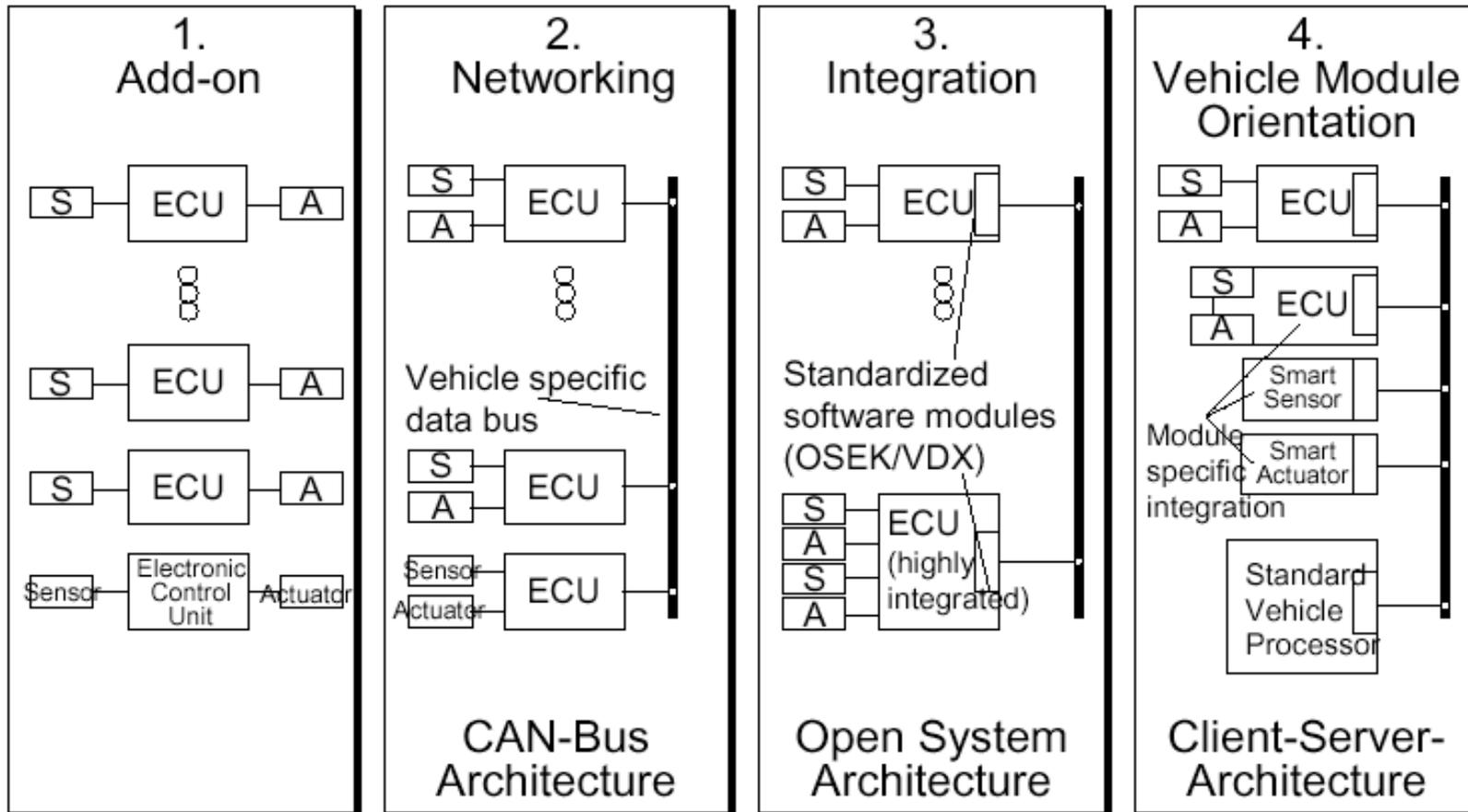
- Automotive Open System Architecture:
- Standardized, openly disclosed interfaces
- HW independent SW layer
- Transferability of functions
- Redundancy activation

- AUTOSAR RTE:
by specifying interfaces and their communication mechanisms, the applications are decoupled from the underlying HW and Basic SW, enabling the realization of Standard Library Functions

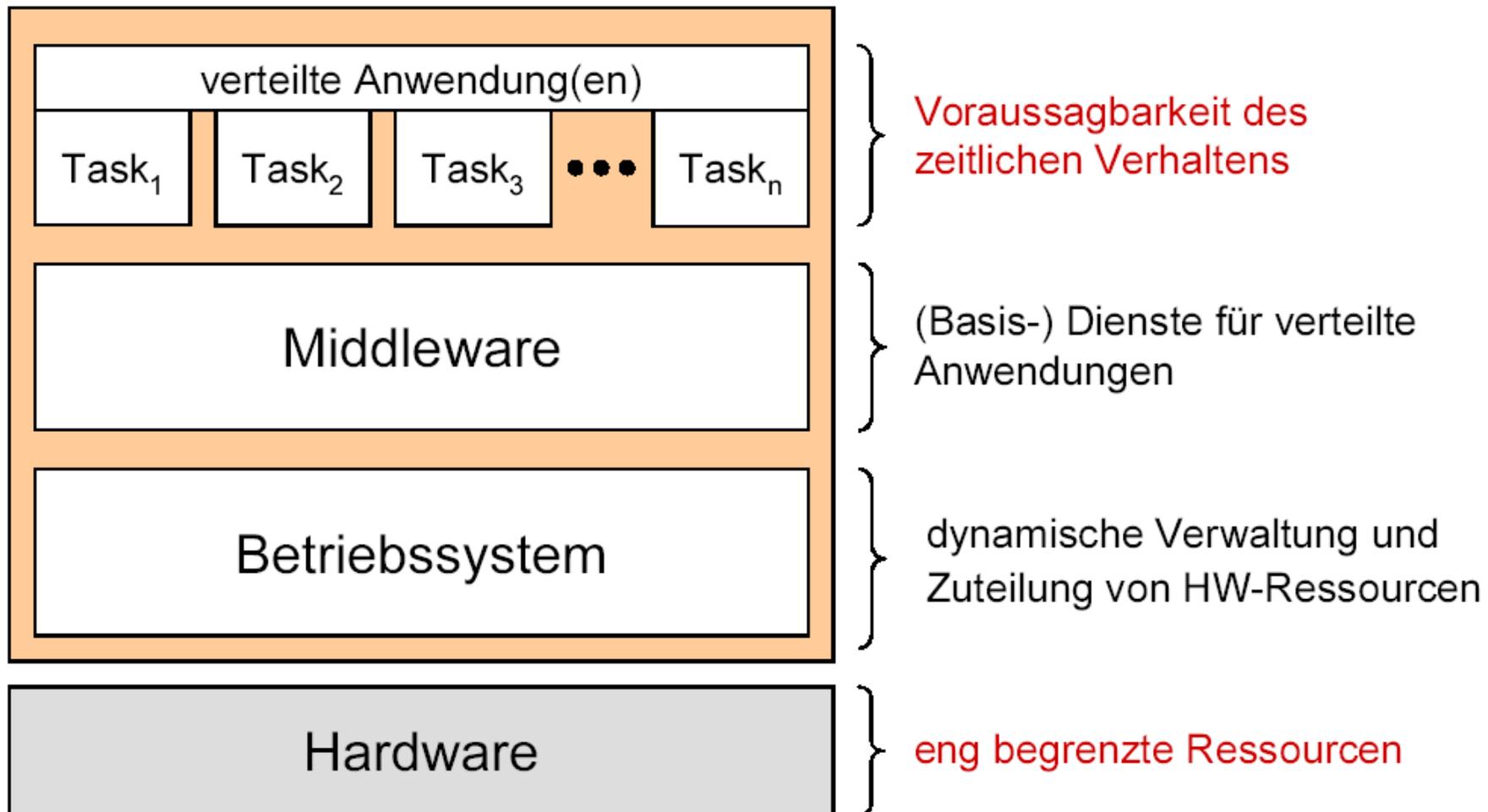


Evolution der Software Architekturen

Die Evolution der Elektrik/Elektronik Systemarchitekturen führen zu offenen Systemarchitekturen mit modularer Softwarearchitektur



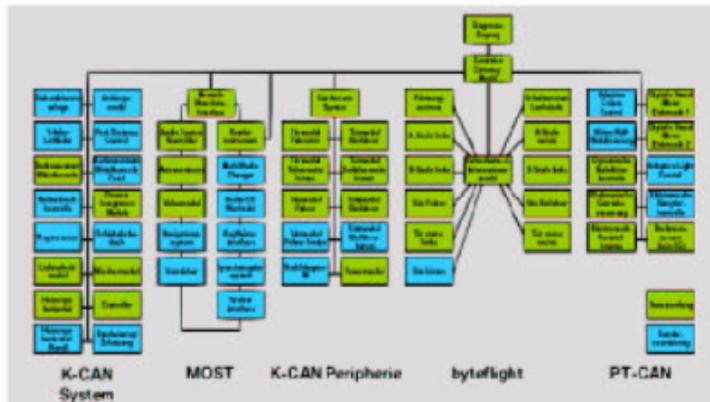
Was ist Softwarearchitektur?



Was ist Softwarearchitektur?

- Architektur beschreibt den Aufbau eines Systems aus seinen Komponenten und deren Wechselwirkungen
- Architektur gibt Systemen Struktur und Gestalt
- Architektur strukturiert komplexe Systeme mit vielen Komponenten und macht sie überschaubar

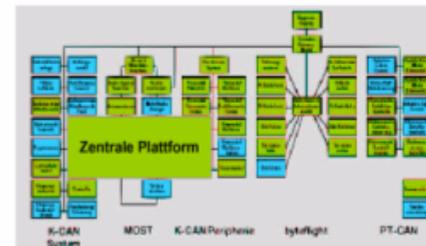
Heutige Elektroniksysteme bestehen aus hochgradig vernetzten embedded Komponenten für typischerweise jeweils eine Funktion



Zukünftig wird das bestehende System durch *Zentrale Verarbeitungseinheiten* ergänzt.

Diese Komponenten besitzen eine „offene“ Softwarearchitektur mit

- Erweiterbarkeit und Updatefähigkeit
- Implementierung neuer Funktionalitäten
- Übergreifende Fahrzeugfunktionen



- Die Fahrzeug Software Architektur beschreibt den im Rahmen einer definierten Zielvorstellung besten Aufbau eines Fahrzeug Software Systems

Wozu Softwarearchitektur?

- Software Architektur macht die Komplexität zukünftiger Software Systeme mit vielen wechselwirkenden Funktionen durchschaubar und beherrschbar
 - ... und begünstigt damit den Bau zuverlässiger Systeme
- Software Architektur zeigt mehrfach verwendbare Komponenten eines Software Systems auf
 - ... und schafft damit eine Grundlage für baureihenübergreifende Software Systeme
- Software Architektur zeigt Wechselwirkungen und Schnittstellen zwischen den Teilen des Software Systems auf und beschreibt diese
 - ... und schafft damit u.a. die Grundlage für die Plug&Play-Fähigkeit zukünftiger Architekturkomponenten
- Software Architektur liefert die Grundlage für
 - flexible,
 - updatefähige und
 - erweiterbare
- Software Systeme im Fahrzeug.

Trends E/E-Komponenten: Verteilte Systeme (1)

- Gestern
 - Klassische Bedienelemente (Schalter, Potis) oder mechanisch betätigte Schalter (Verteiler) steuern direkt die Funktionen bzw. die Leistungstreiber an
 - Eindeutige Zuordnung von Funktion und Bedienung



Trends E/E-Komponenten: Verteilte Systeme (2)

- Heute
 - Vielzahl von Funktionen wird durch teilweise vernetzte Steuergeräte angeboten
 - Ansteuerung der Steuergeräte durch dedizierte Bedienelemente, Steuergeräte treiben Lasten
 - Probleme
 - Vielzahl von Bedienelementen
 - Explosion der Steuergeräte-Anzahl

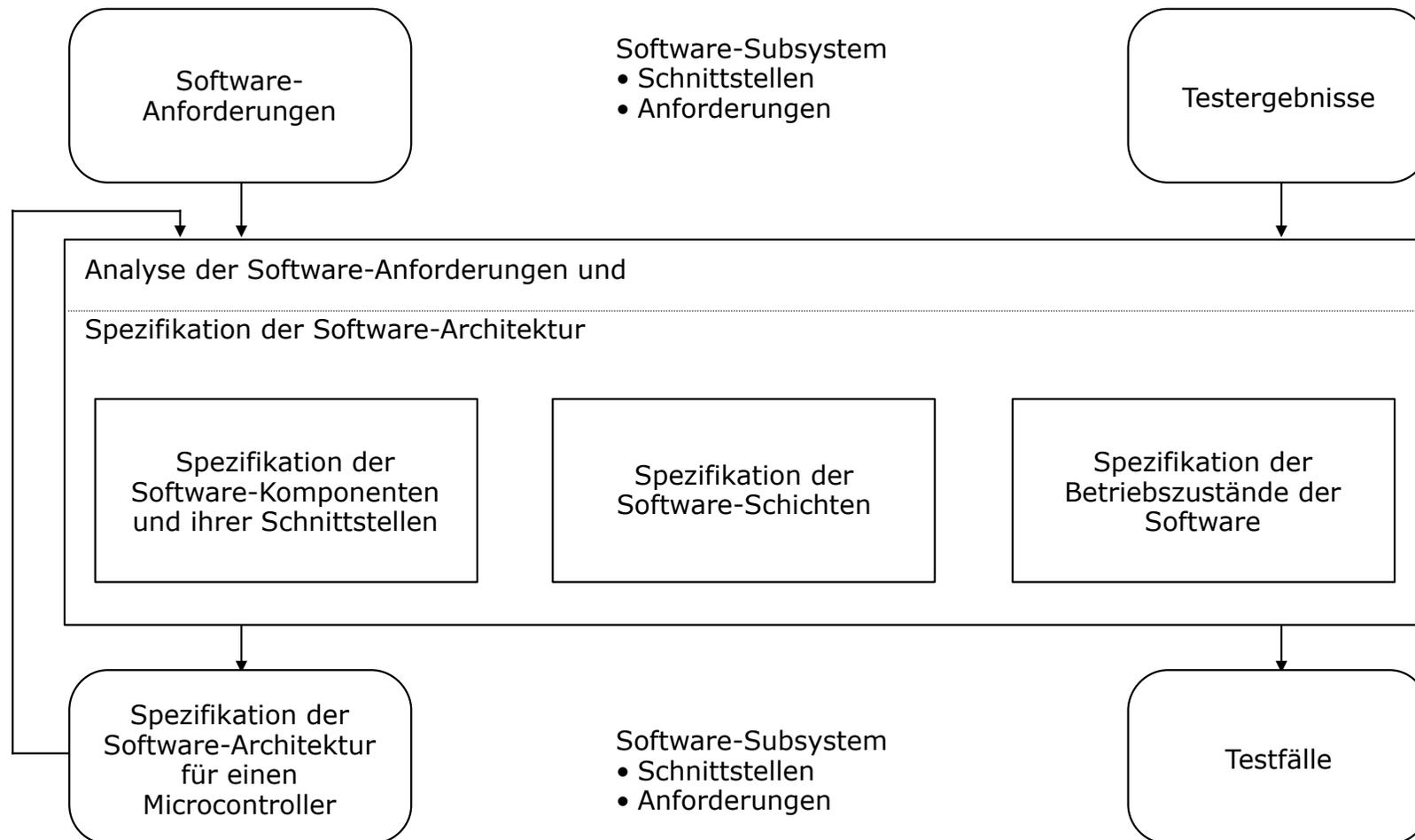


Trends E/E-Komponenten: Verteilte Systeme (3)

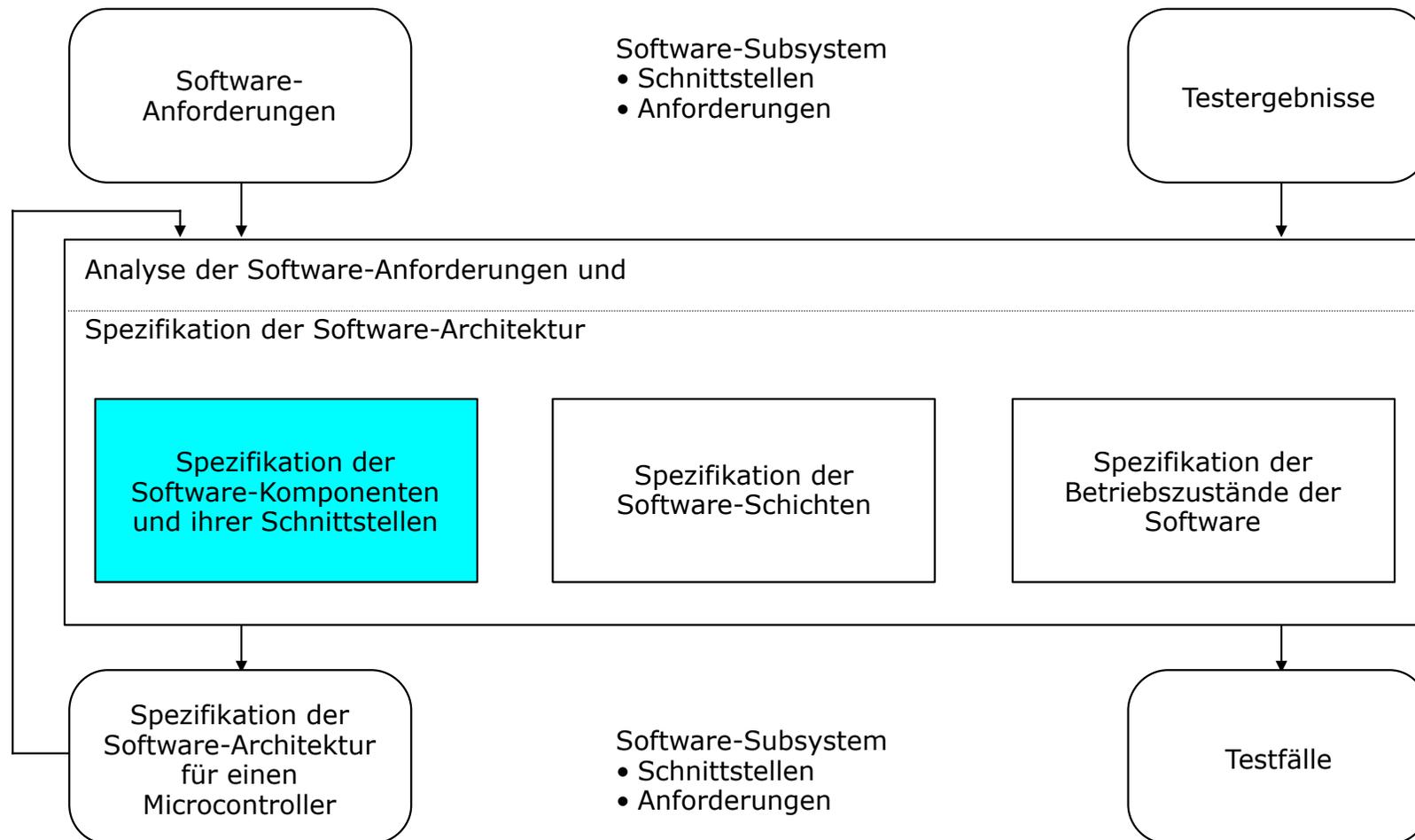
- Morgen
 - Systeme übernehmen Funktionen unabhängig vom Einbauort
 - Kommunikation über vernetzte Bussysteme
 - Einheitliches MMI wird durch Head Unit des Infotainments bereitgestellt und steuert viele Systeme an



Software-Anforderungen und Software-Architektur



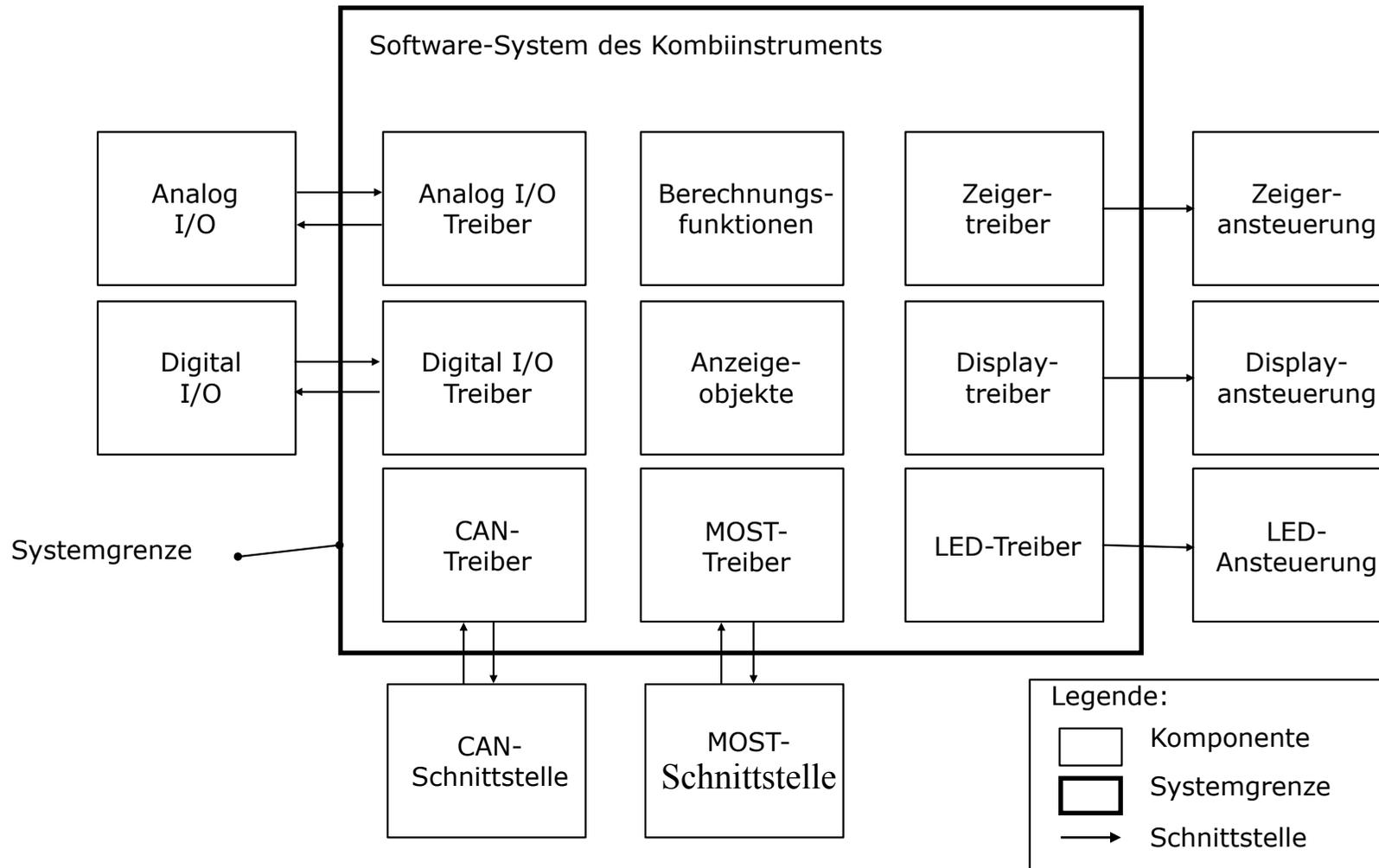
Software-Anforderungen und Software-Architektur



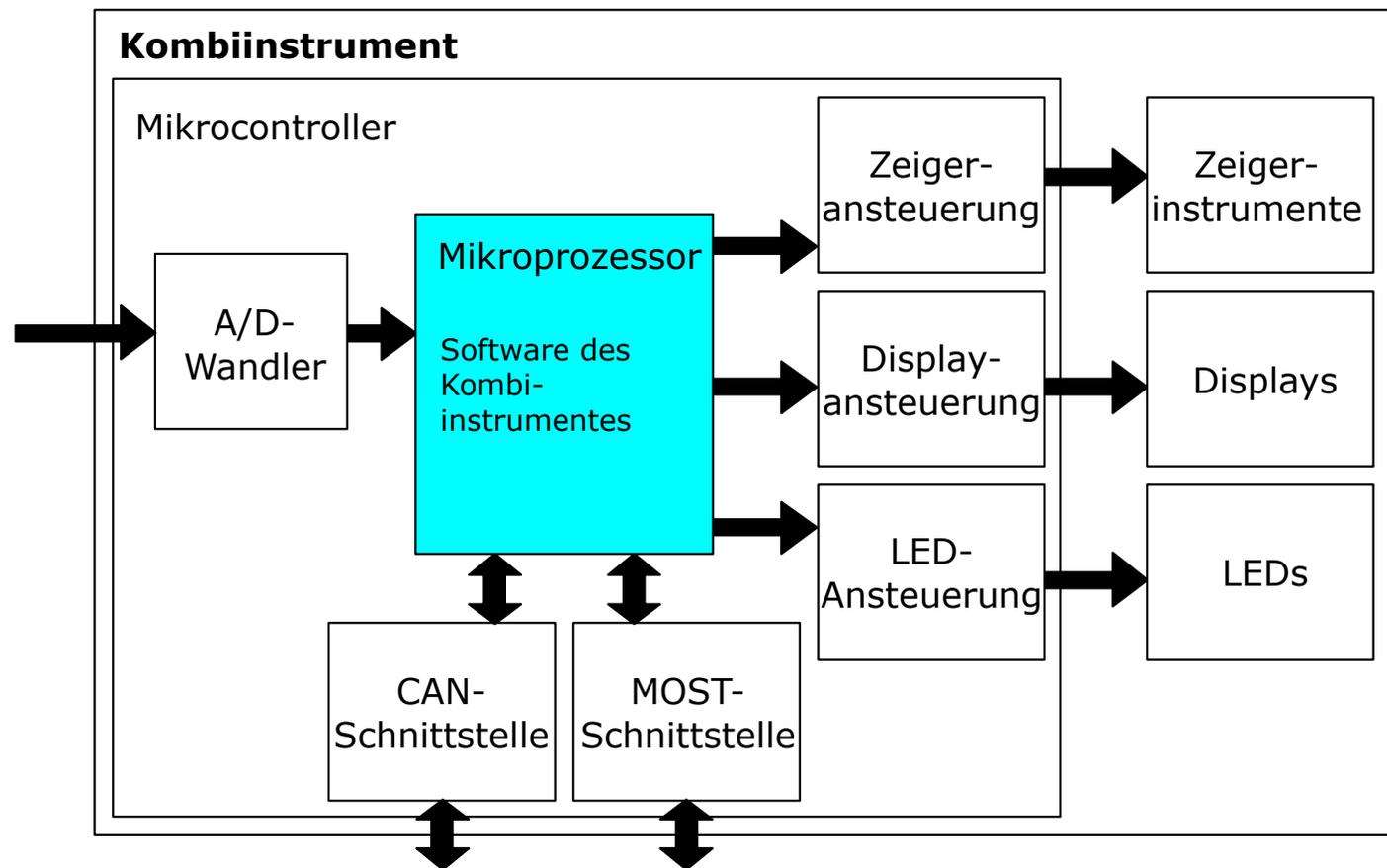
Software-Anforderungen und Software-Architektur

- Datenfluss und Kontrollfluss
 - Programmierung
 - Dateninformation
 - Kontroll- oder Steuerinformation
 - Software-Schnittstellen
 - Datenschnittstellen
 - Kontroll- oder Steuerschnittstellen
 - Beispiel
 - Inhalt einer CAN-Nachricht: Dateninformation
 - Eintreffen einer CAN-Nachricht: Kontrollinformation
- Spezifikation der On-Board-Schnittstellen eines Steuergerätes
 - Sollwertgeber
 - Sensoren
 - Aktuatoren
 - Andere Steuergeräte

Software-Schnittstellen des Kombiinstrumentes



Hardware des Kombiinstruments

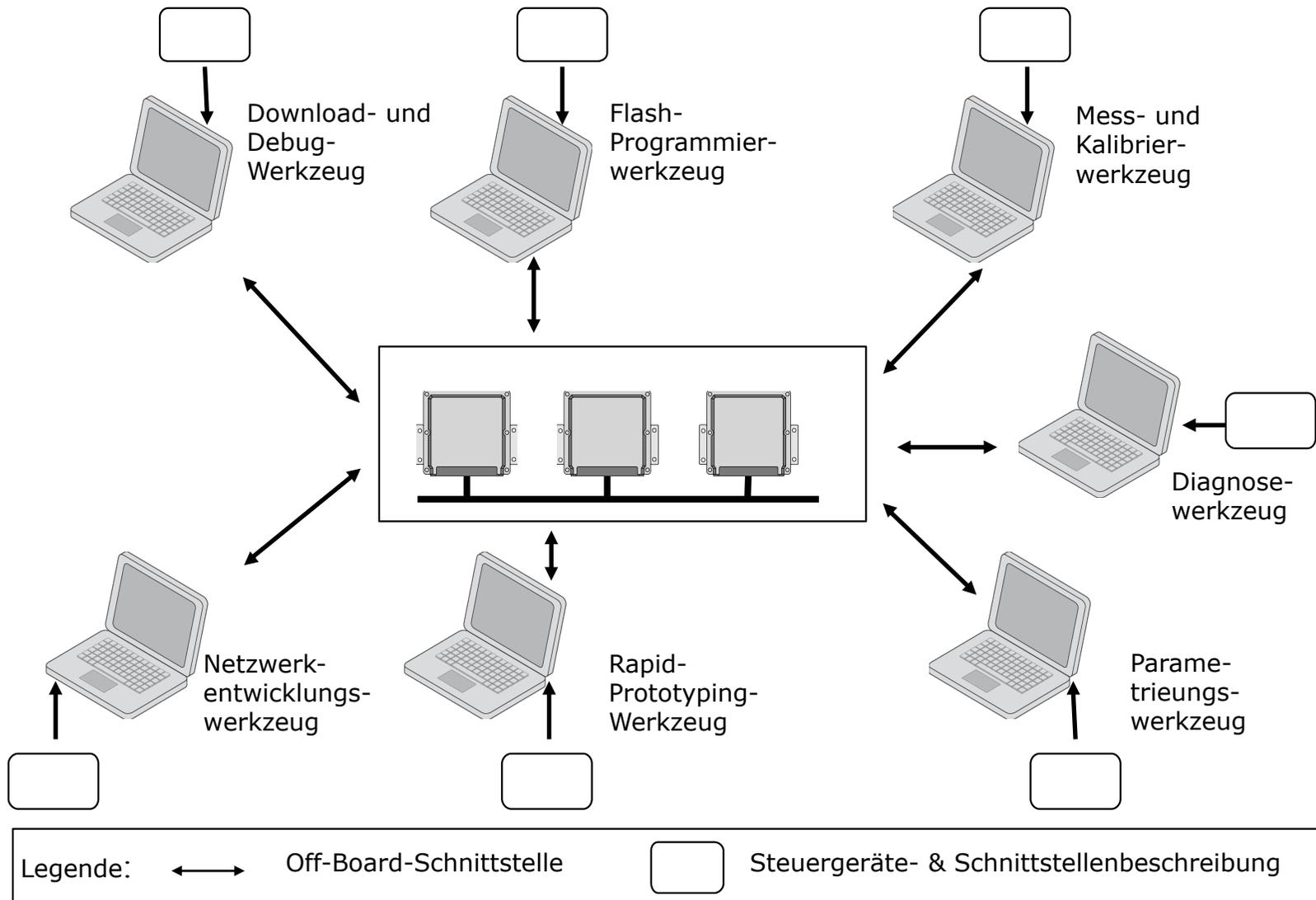


Spezifikation der Off-Board-Schnittstellen

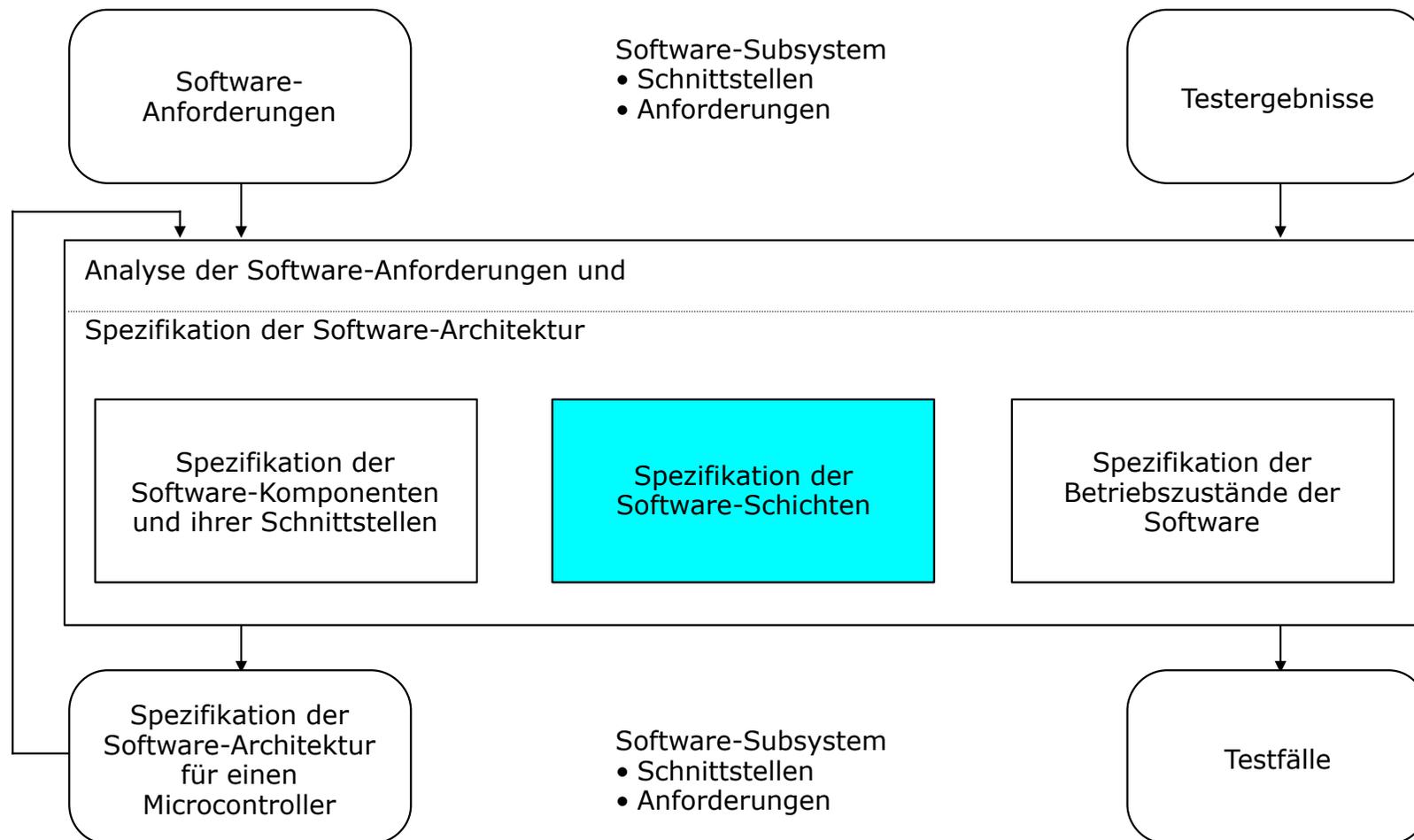
- Die Software-Architektur muss alle Schnittstellen unterstützen, die für den Off-Board-Betrieb des Steuergerätes in Entwicklung, Produktion und Betrieb notwendig sind
- Verschiedene Hardware- und Software-Varianten
 - Modifizierte Off-Board-Schnittstellen
- Verfahren für
 - Messen
 - Kalibrieren
 - Diagnose
 - Flash-Programmierung
- Beschreibungsdateien für Off-Board-Schnittstellen

- siehe auch 9. Integration der Software-Komponenten

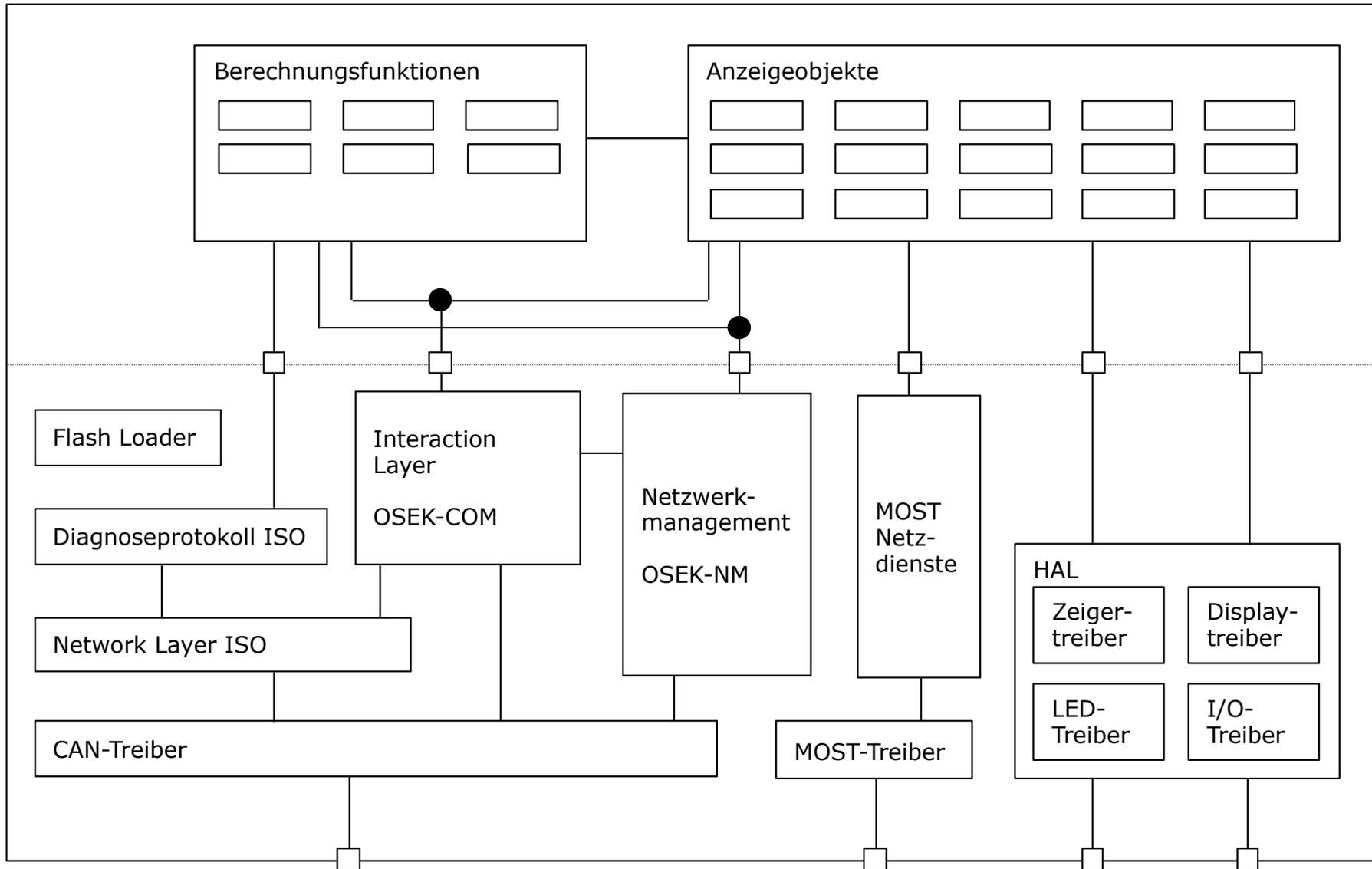
Off-Board-Schnittstellen eines Steuergerätes



Software-Anforderungen und Software-Architektur



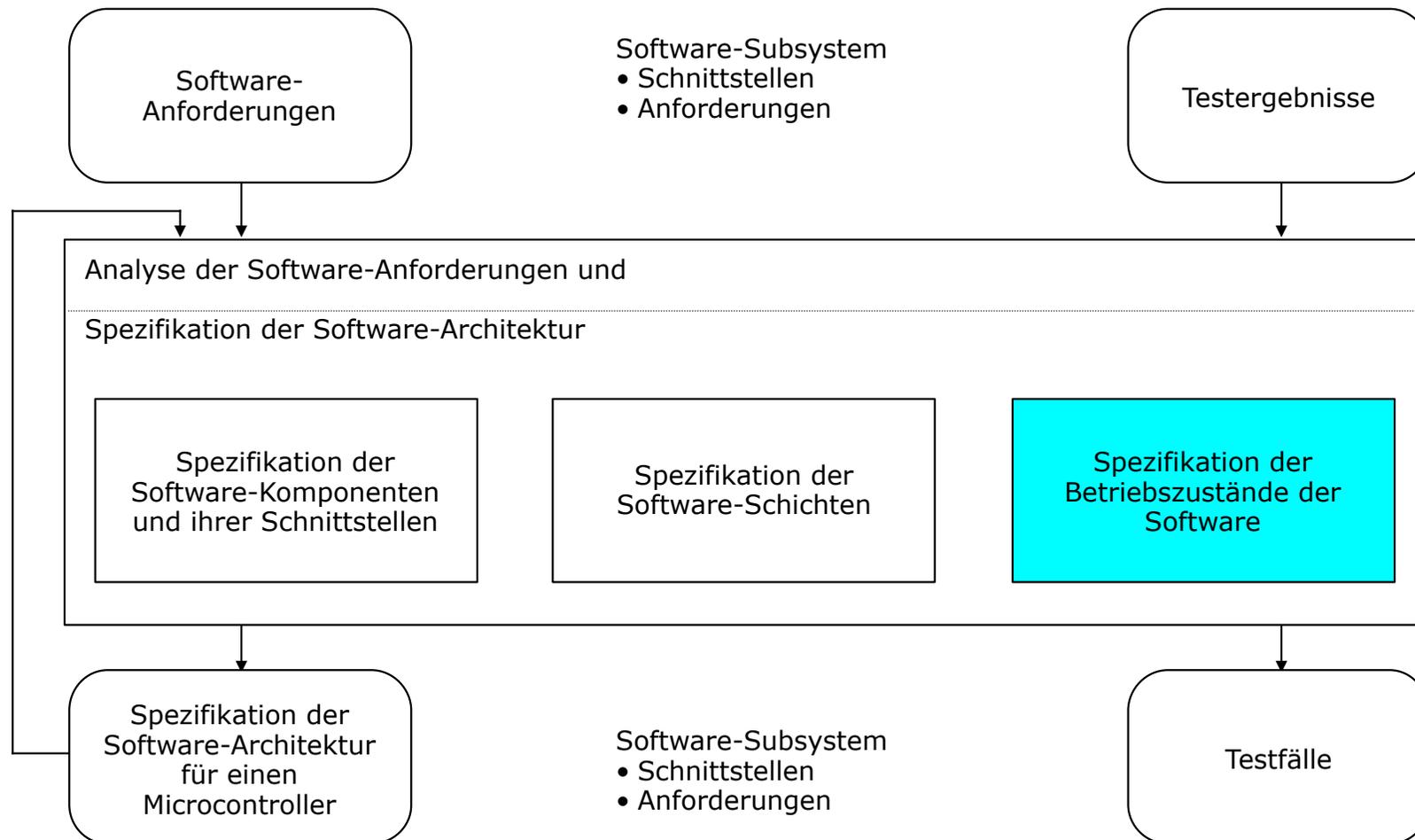
Softwarearchitektur des Kombiinstrumentes



Anwendungs-Software

Plattform-/Basis-Software
Betriebssystem OSEK-OS

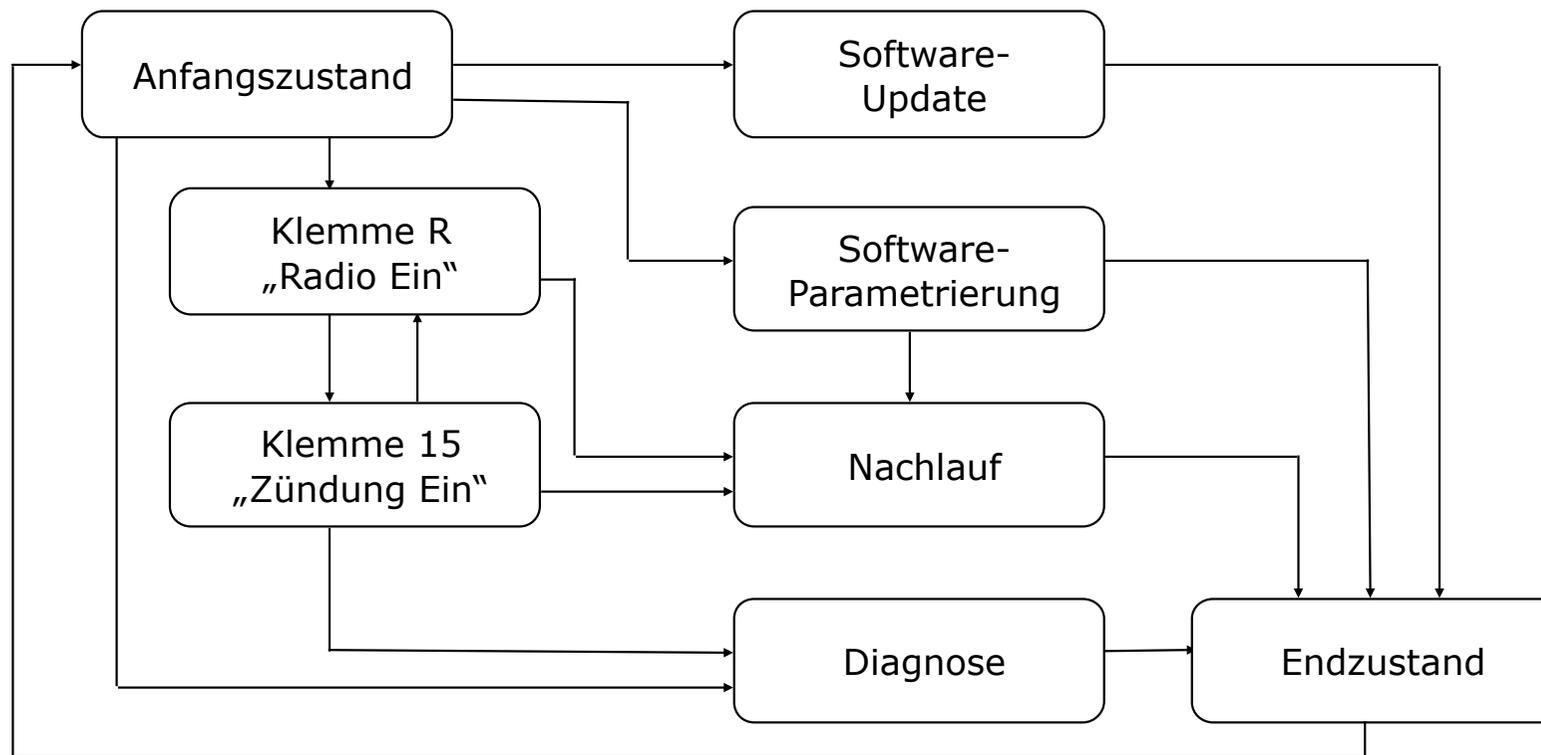
Software-Anforderungen und Software-Architektur



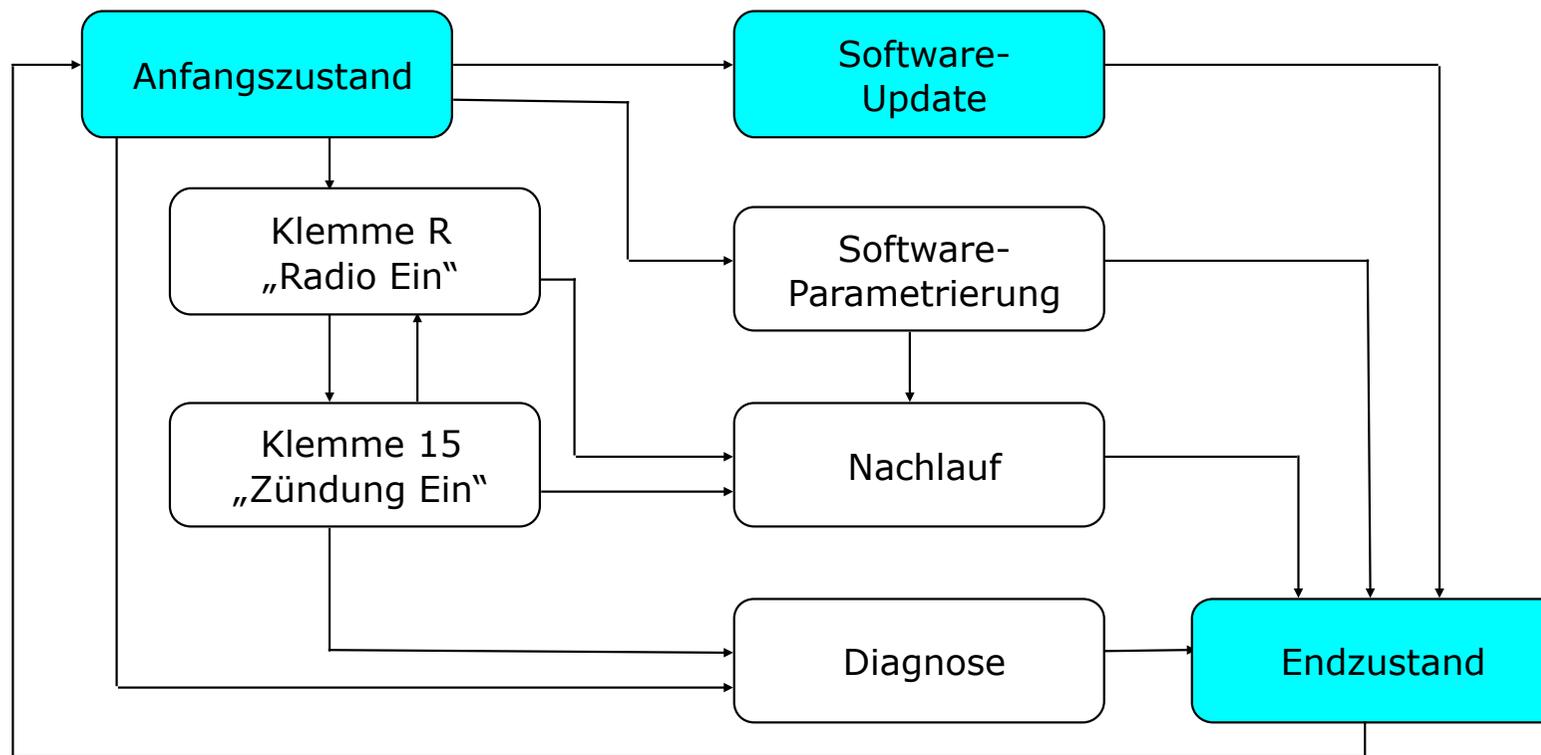
Betriebszustände

- Normalbetrieb
 - Betriebszustand für Diagnose
 - Betriebszustand für Update
 - Notlauf
-
- Spezifikation durch Zustandsautomaten

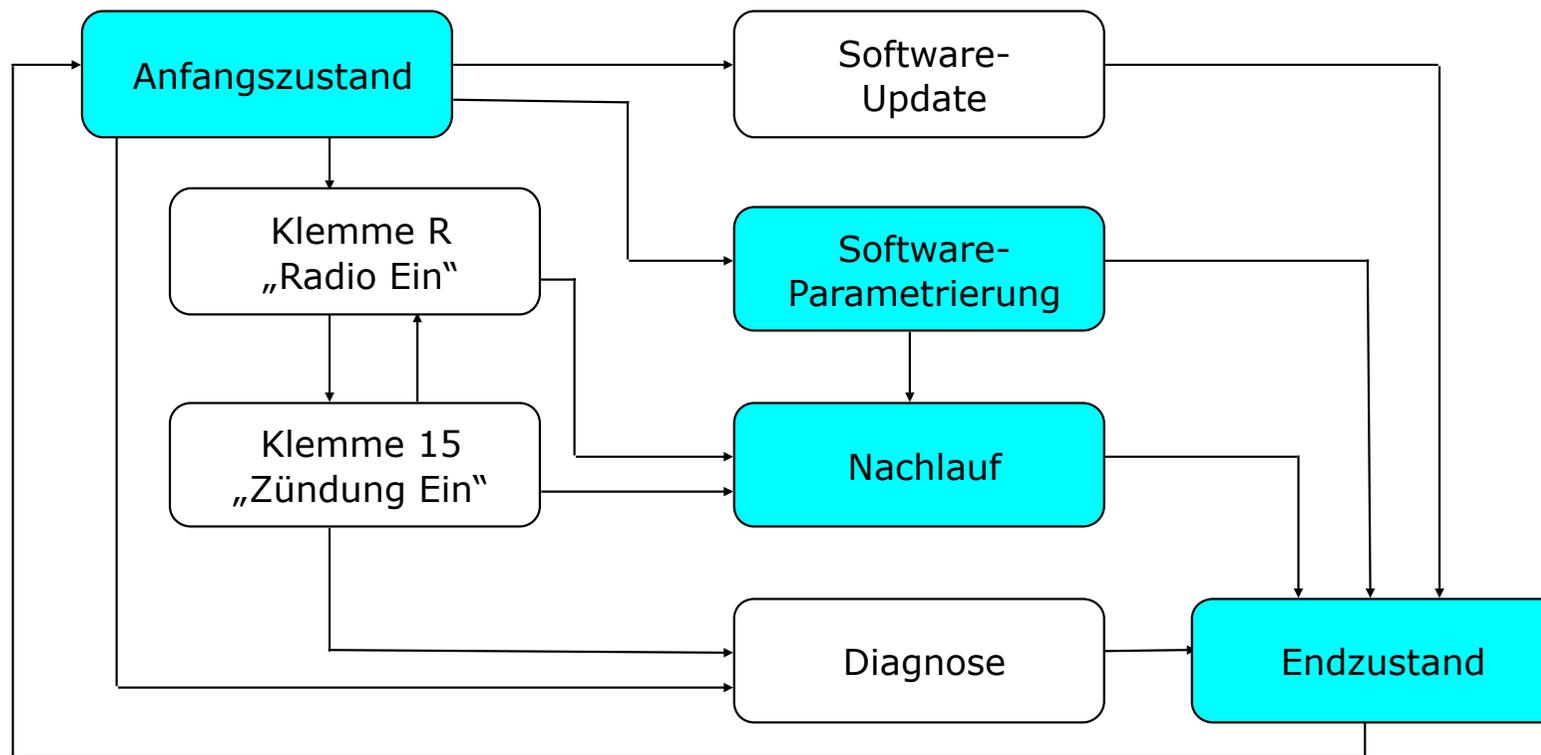
Betriebszustände des Kombiinstrumentes



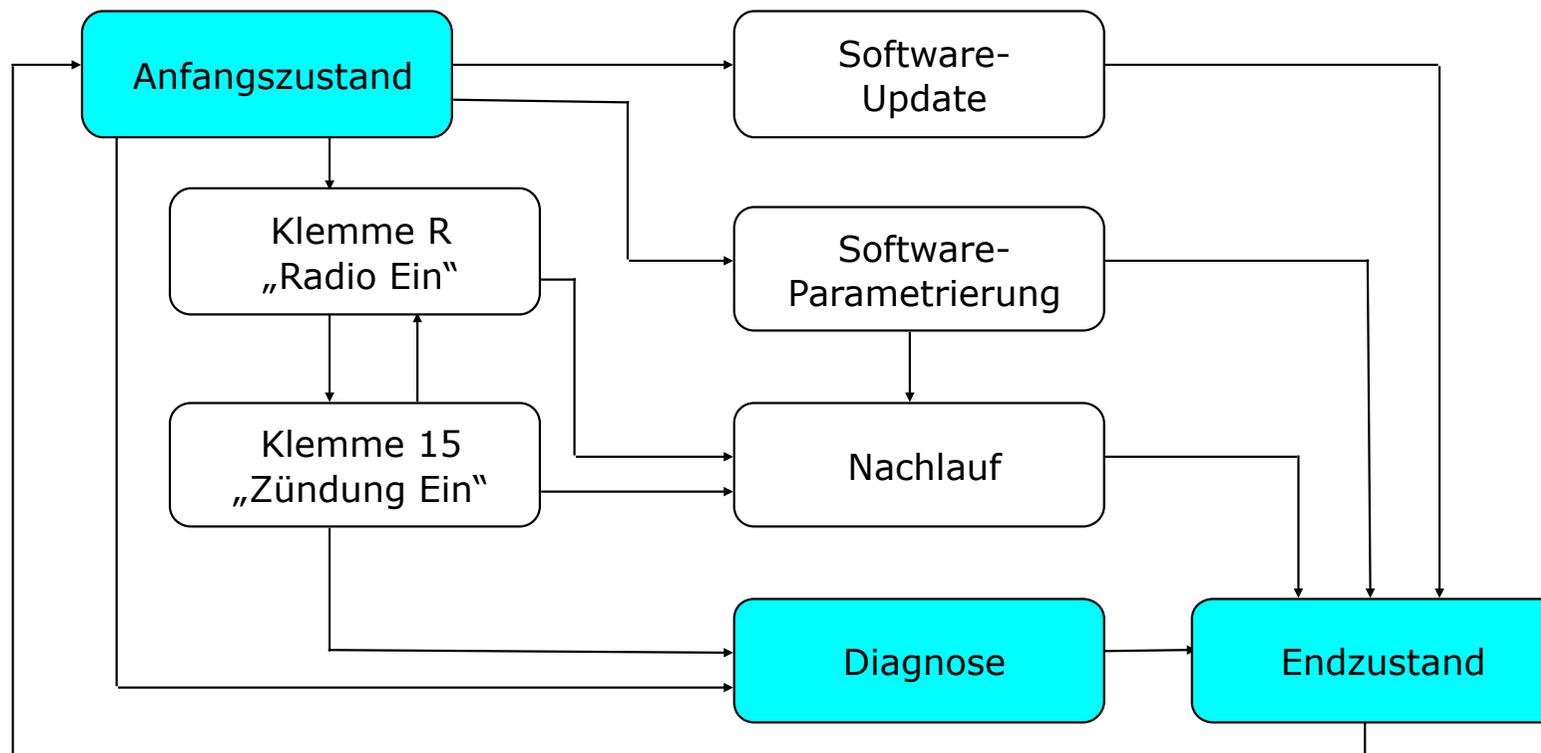
Betriebszustände des Kombiinstrumentes



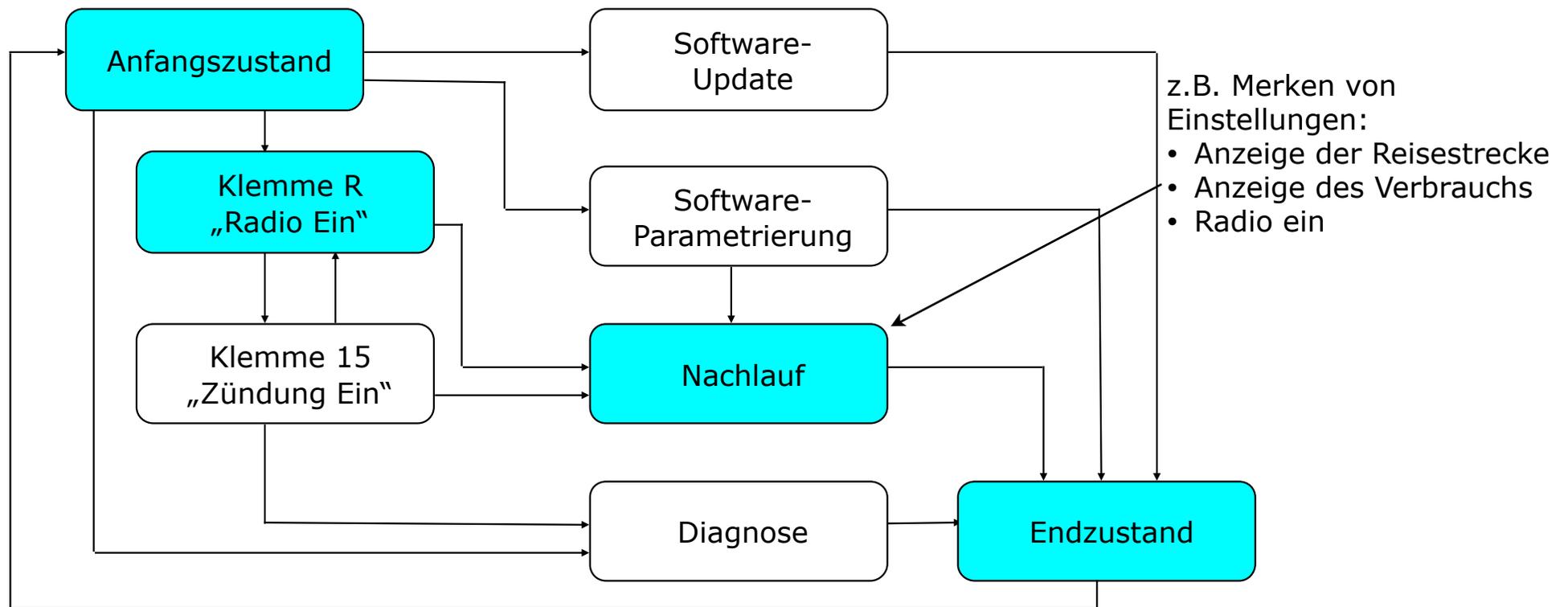
Betriebszustände des Kombiinstrumentes



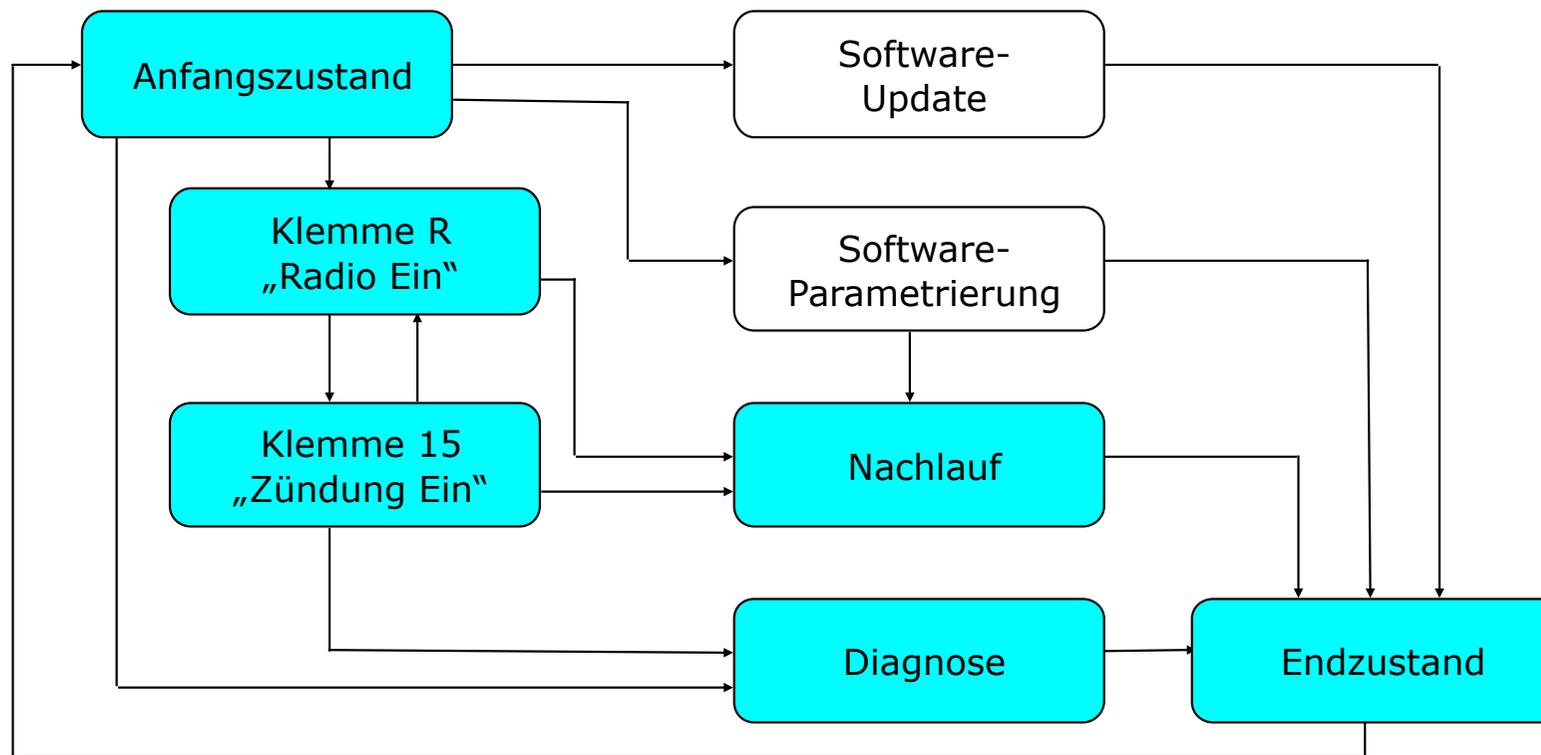
Betriebszustände des Kombiinstrumentes



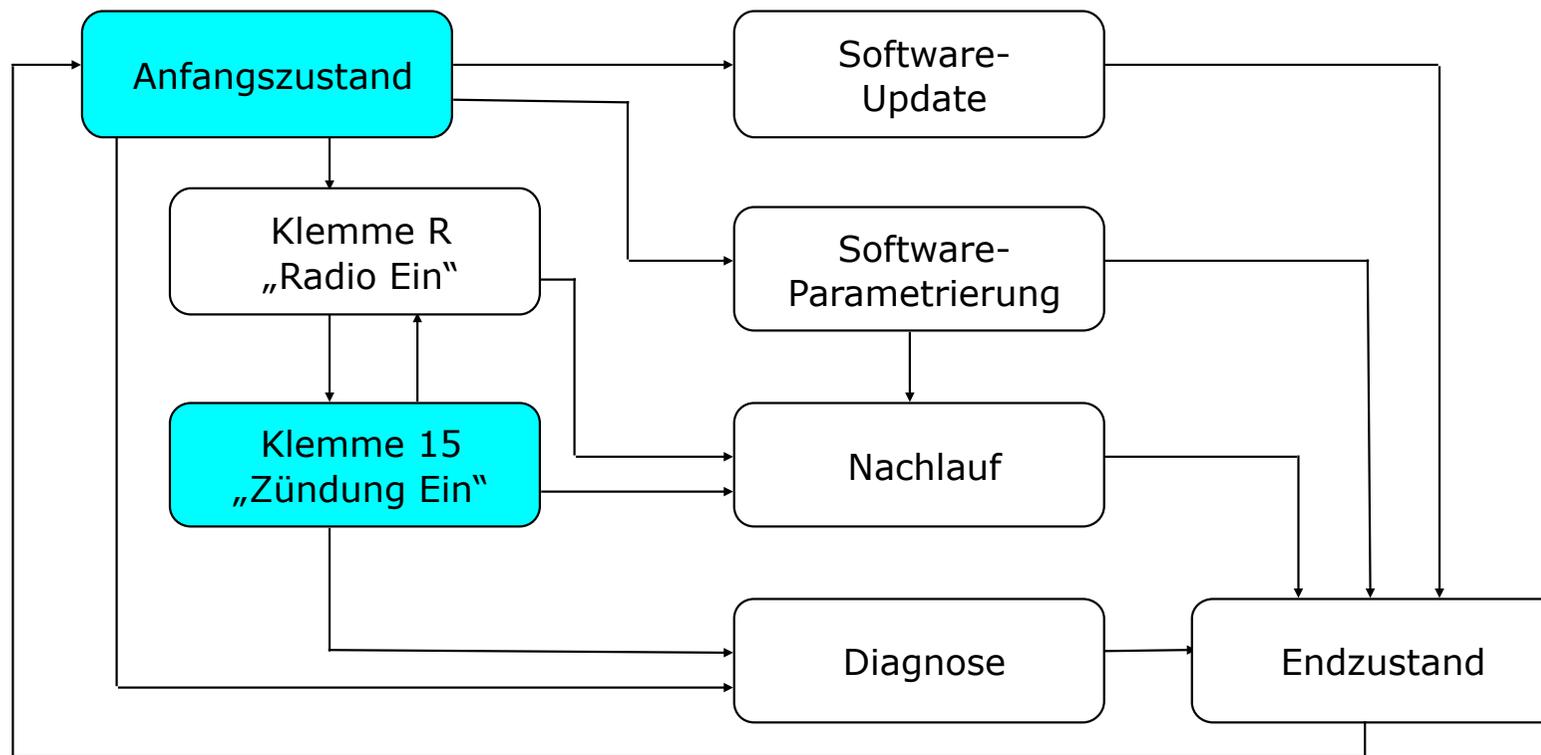
Betriebszustände des Kombiinstrumentes



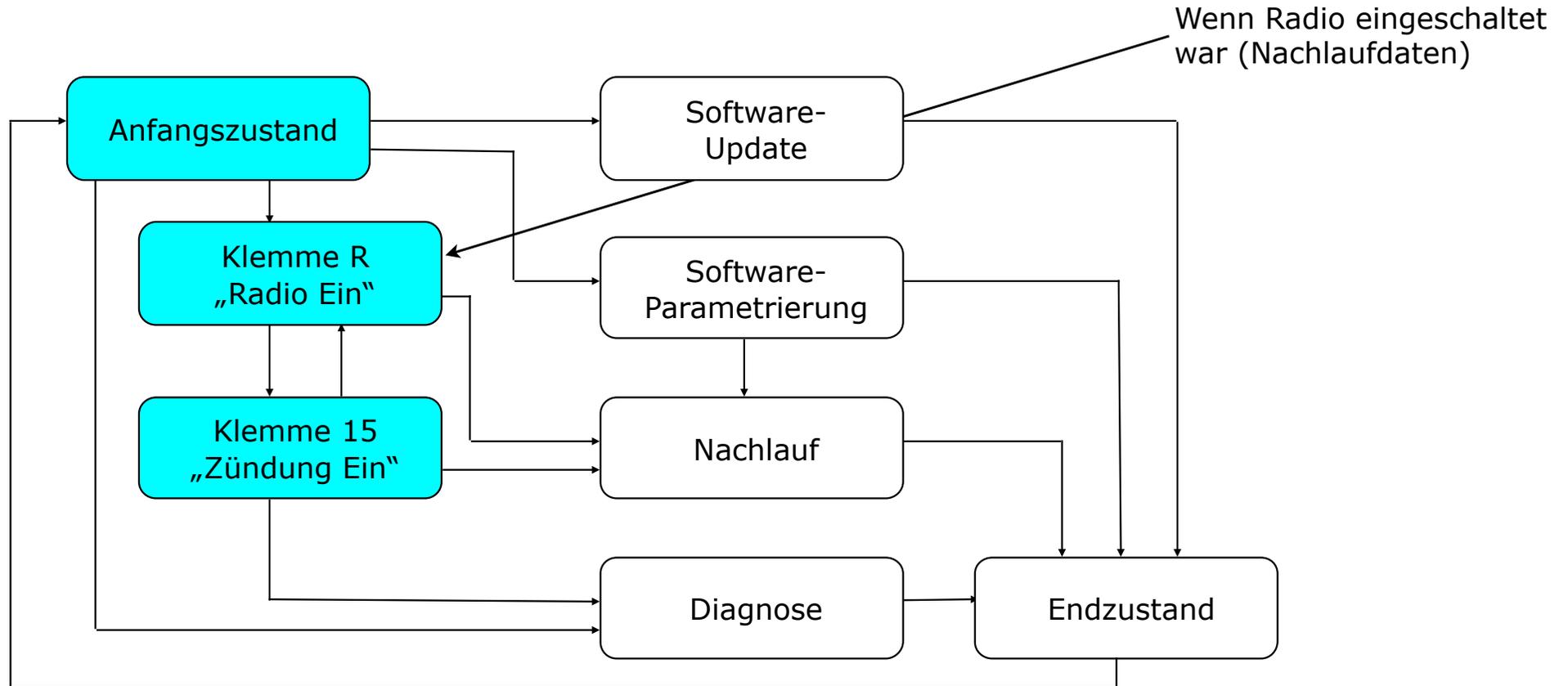
Betriebszustände des Kombiinstrumentes



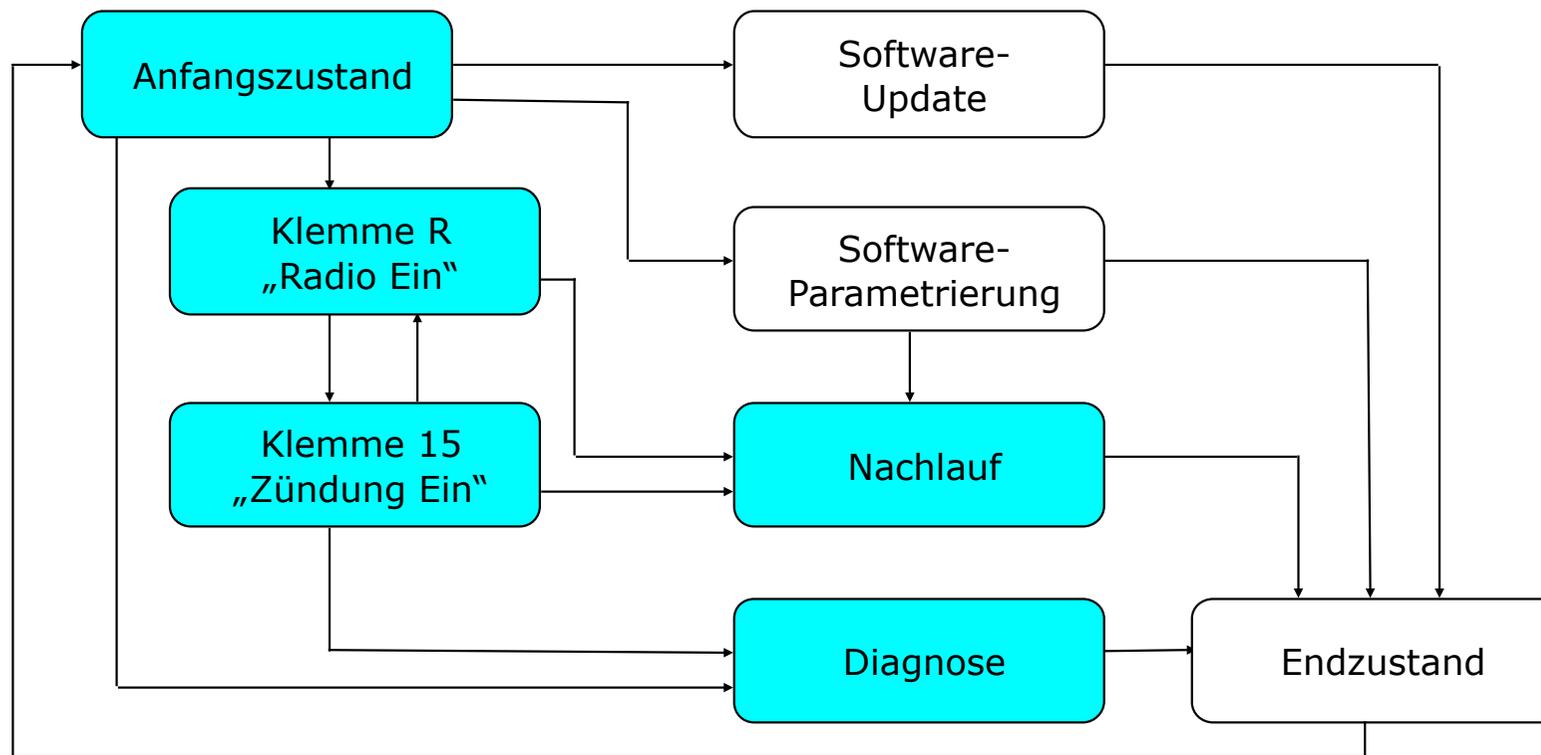
Betriebszustände des Kombiinstrumentes



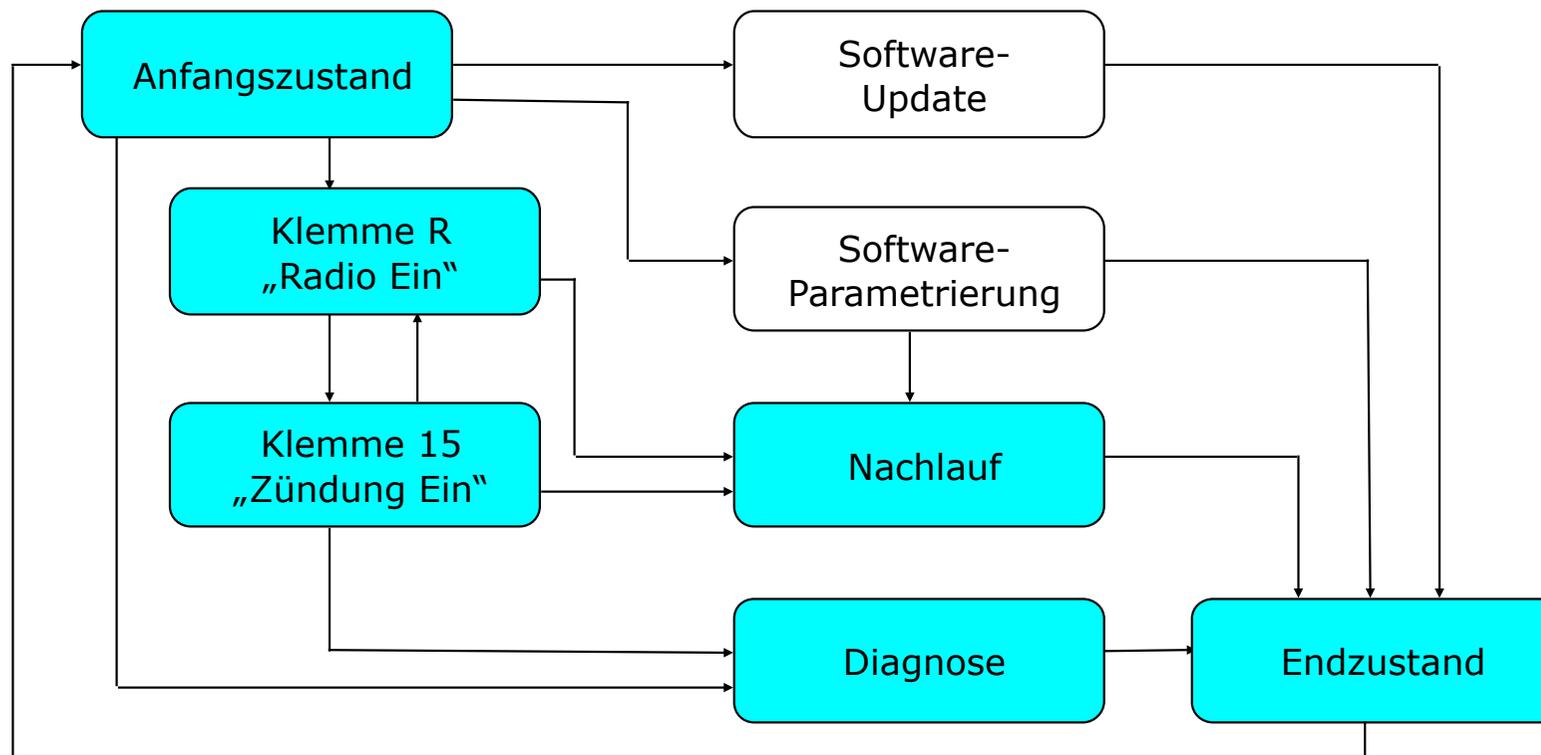
Betriebszustände des Kombiinstrumentes



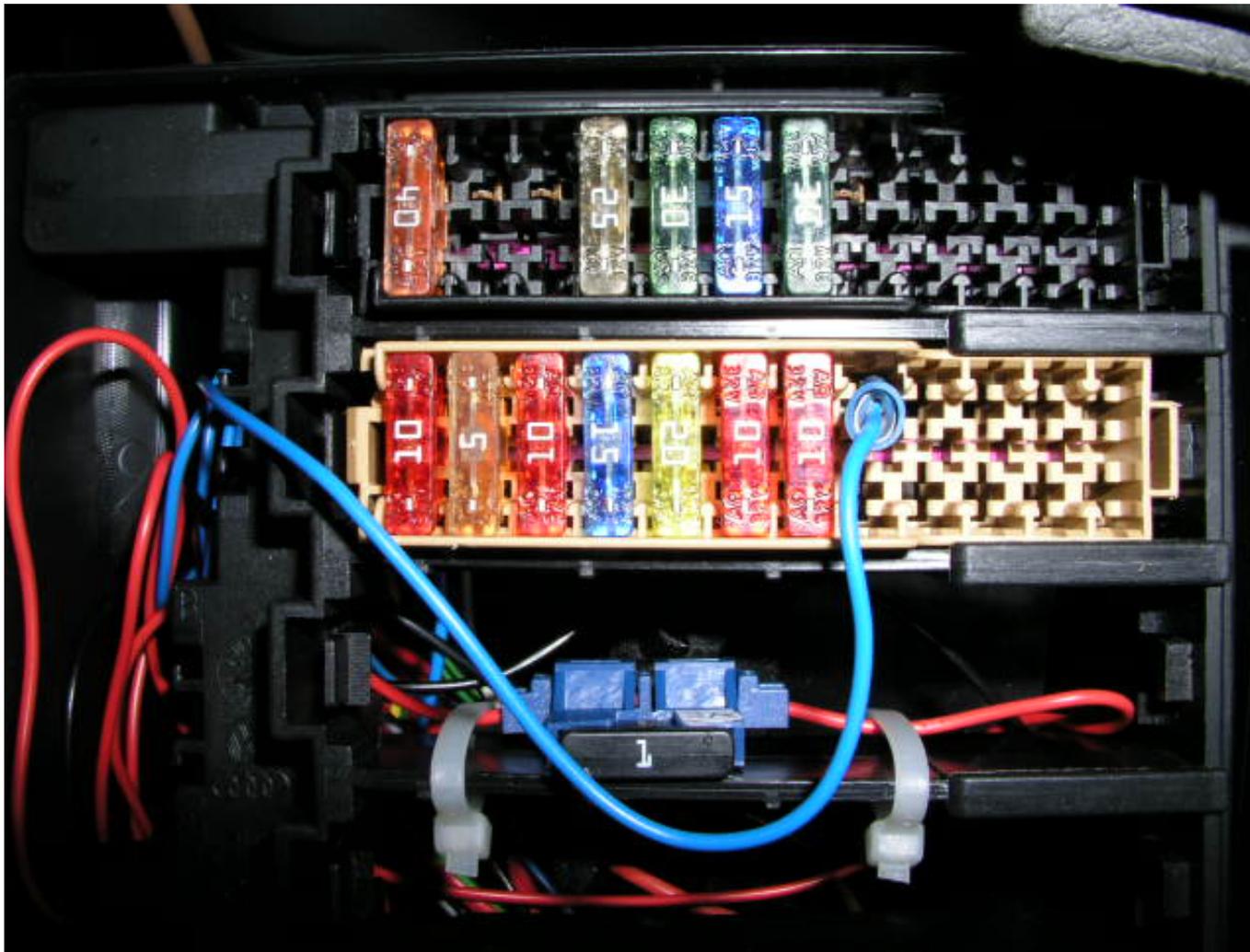
Betriebszustände des Kombiinstrumentes



Betriebszustände des Kombiinstrumentes



Klemme 15



Klemmenbezeichnung

- Eine Klemmenbezeichnung ist im Maschinen- und Anlagenbau sowie in der Kraftfahrzeugelektrik eine Hilfszahl oder ein Buchstabe, um das Anschließen von Leitungen, aber auch die Fehlerdiagnose mithilfe eines Schaltplans in Kraftfahrzeugen, Automatisierungsanlagen oder Schaltschränken zu vereinfachen. Sie ist an oder nahe bei der Klemme angebracht und oft auch an der dort anzuschließenden bzw. angeschlossenen Leitung.
- DIN 72552 Klemmenbezeichnungen in Kraftfahrzeugen
- Quelle: <http://de.wikipedia.org/wiki/Klemmenbezeichnung>

