

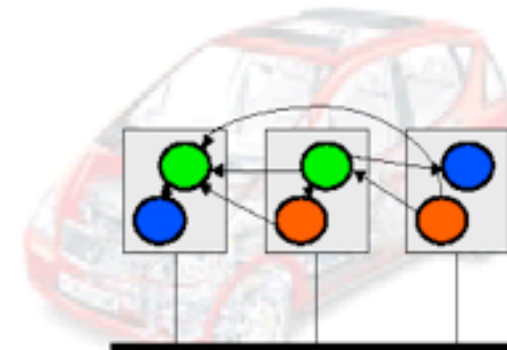
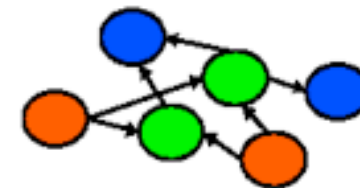
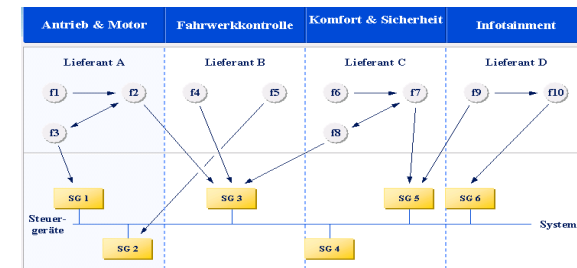


Vorlesung
Automotive Software Engineering
Teil 5 E/E-Entwicklung (1-1)
Sommersemester 2015

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	

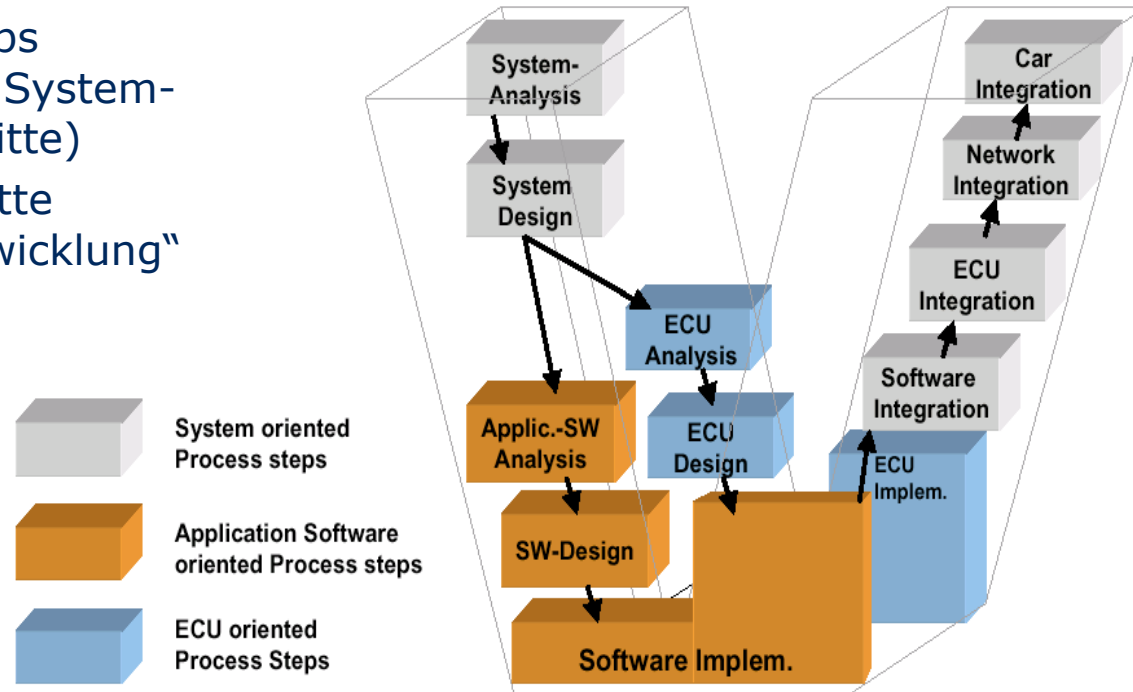


Lernziele E/E-Entwicklung

- Elektrik/Elektronik-Technologien und -Bauteile sowie Randbedingungen der Elektrik/Elektronik-Entwicklung im Automobil und deren Auswirkung auf die Software-Entwicklung kennenlernen:
 - Bussysteme im Automobil
 - Einführung
 - Architektur serieller Bussysteme
 - Serielle Bussysteme im Kfz
 - Mechatronik-Entwicklungen im Automobil
 - Grundlagen
 - X-by-Wire
 - Elektromechanische Lenkungen
 - Elektromechanische Bremsen
 - Technologien und Bauteile
 - Kabelbaum und Energiebordnetze
 - Halbleitertechnologie
 - Elektromagnetische Verträglichkeit (EMV)

E/E-Entwicklung

- Automotive Elektrik/Elektronik-Entwicklung
- Ziel:
Darstellung von Elektrik/Elektronik-Technologien und -Bauteile sowie von Randbedingungen der Elektrik/Elektronik-Entwicklung für eingebettete Systeme speziell im Automobil
- Schwerpunkt:
ECU oriented Process Steps
(mit Rückwirkung auf die System- und SW-orientierten Schritte)
- Die SW-orientierten Schritte werden in Teil 6 „SW-Entwicklung“ behandelt
- ECU?



5. E/E-Entwicklung

1. Bussysteme im Automobil
 - 1.1. Einführung
 - 1.2. Architektur serieller Bussysteme
 - 1.3. Serielle Bussysteme im Kfz
2. Mechatronik-Entwicklungen im Automobil
 - 2.1. Grundlagen
 - 2.2. X-by-Wire
 - 2.3. Elektromechanische Lenkungen
 - 2.4. Elektromechanische Bremsen
3. Technologien und Bauteile
 - 3.1. Kabelbaum und Energiebordnetze
 - 3.2. Halbleitertechnologie
 - 3.3. Elektromagnetische Verträglichkeit (EMV)

5. E/E-Entwicklung

1. Bussysteme im Automobil

- 1.1. Einführung
- 1.2. Architektur serieller Bussysteme
- 1.3. Serielle Bussysteme im Kfz

2. Mechatronik-Entwicklungen im Automobil

- 2.1. Grundlagen
- 2.2. X-by-Wire
- 2.3. Elektromechanische Lenkungen
- 2.4. Elektromechanische Bremsen

3. Technologien und Bauteile

- 3.1. Kabelbaum und Energiebordnetze
- 3.2. Halbleitertechnologie
- 3.3. Elektromagnetische Verträglichkeit (EMV)

E-Learning Plattform Vector Informatik GmbH

- https://www.vector.com/vl_index_de.html
 - Kostenlos
 - Anmeldung erforderlich
- Themen
 - Serielle Bussysteme
 - Abschnitt „Bussysteme im Automobil“ basiert hierauf teilweise
 - Einführung in CAN
 - Kapitel CAN-FD
 - Einführung in LIN
 - Einführung in FlexRay
 - (Einführung in MOST)
 - Einführung in Ethernet (geplant)
 - Einführung in AUTOSAR
 - Einführung 26262 (geplant)
- http://www.vector.com/vi_controller_area_network_de.html

Serielle Bussysteme

- 1 Einführung
- 2 Architektur serieller Bussysteme
 - 2.1 Interaktionsstruktur Einführung in CAN
 - 2.2 Topologie Einführung in LIN
 - 2.3 Adressierung und Framing Einführung in FlexRay
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
- 4 Lernzielkontrolle

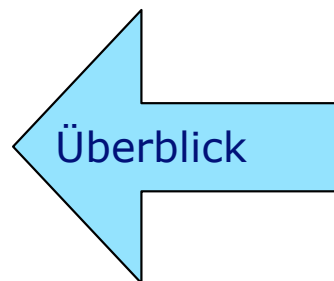
Serielle Bussysteme

- 1 Einführung
- 2 Architektur serieller Bussysteme
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
- 4 Lernzielkontrolle

Einführung in CAN

Einführung in LIN

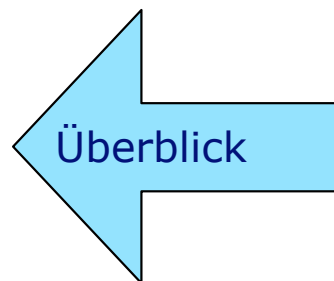
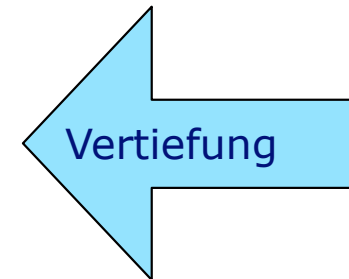
Einführung in FlexRay



Serielle Bussysteme

- 1 Einführung
- 2 Architektur serieller Bussysteme
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
- 4 Lernzielkontrolle

Einführung in CAN
Einführung in LIN
Einführung in FlexRay



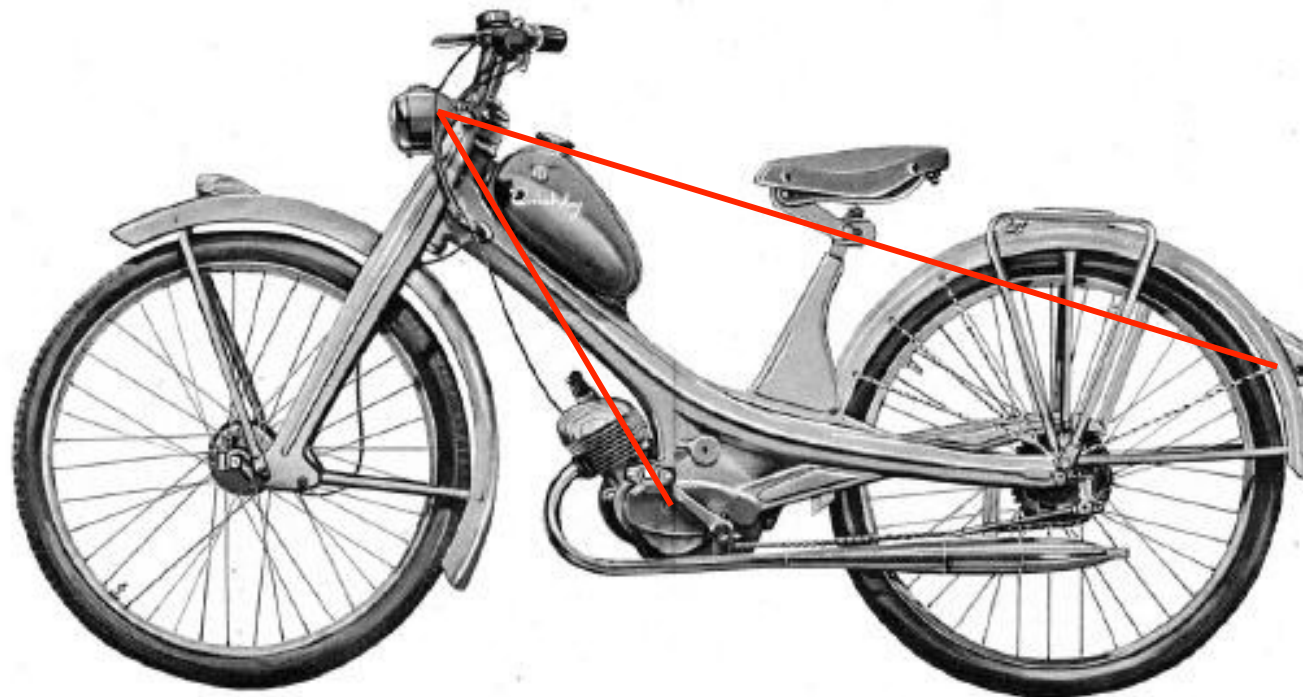
Serielle Bussysteme

- **1 Einführung**
- 2 Architektur serieller Bussysteme
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
 - 3.4 MOST
 - 3.5 Ethernet
 - 3.6 Beispiele

Geänderte Anforderungen

- Früher
 - Geringer Umfang der E/E-Systeme im Fahrzeug
 - Übertragung der Informationen über dedizierte Leitungen
 - Schalter -> SG
 - Sensor -> SG
 - SG -> Aktuator
 - SG <-> SG
- Heute
 - Ständig steigende Anzahl von Steuergeräten, Sensoren, elektromechanischen Aktuatoren und Bedienelementen
 - Gestiegener Informationsaustausch zwischen verschiedenen Steuergeräten und Teilsystemen
 - Verkabelung über einzelne Kabel stößt an Grenzen bzgl. Bauraum, Gewicht, Anschlusszahl und Kosten
- Lösung
 - Bussysteme im Automobil

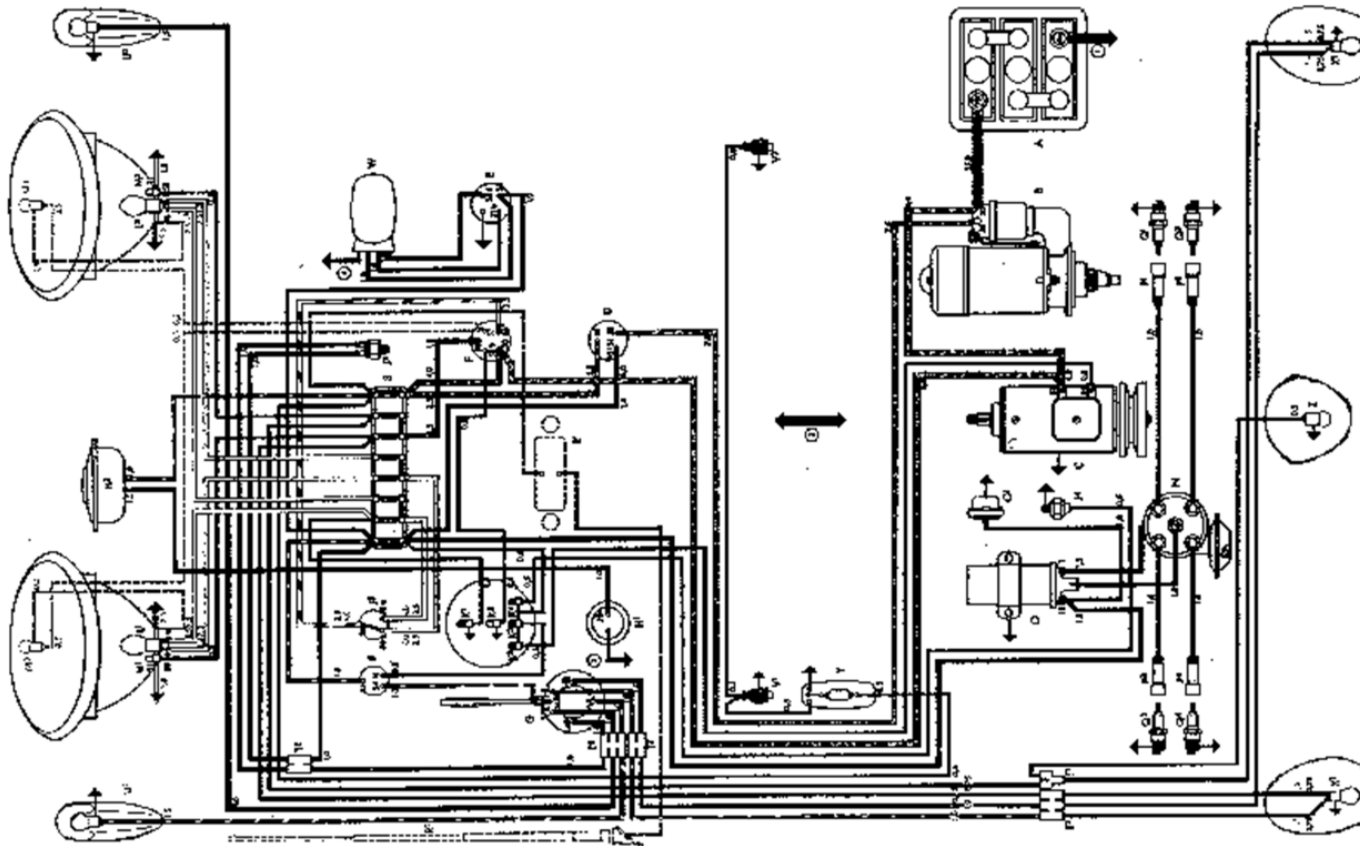






PREEvision 6

Complexity – in former times

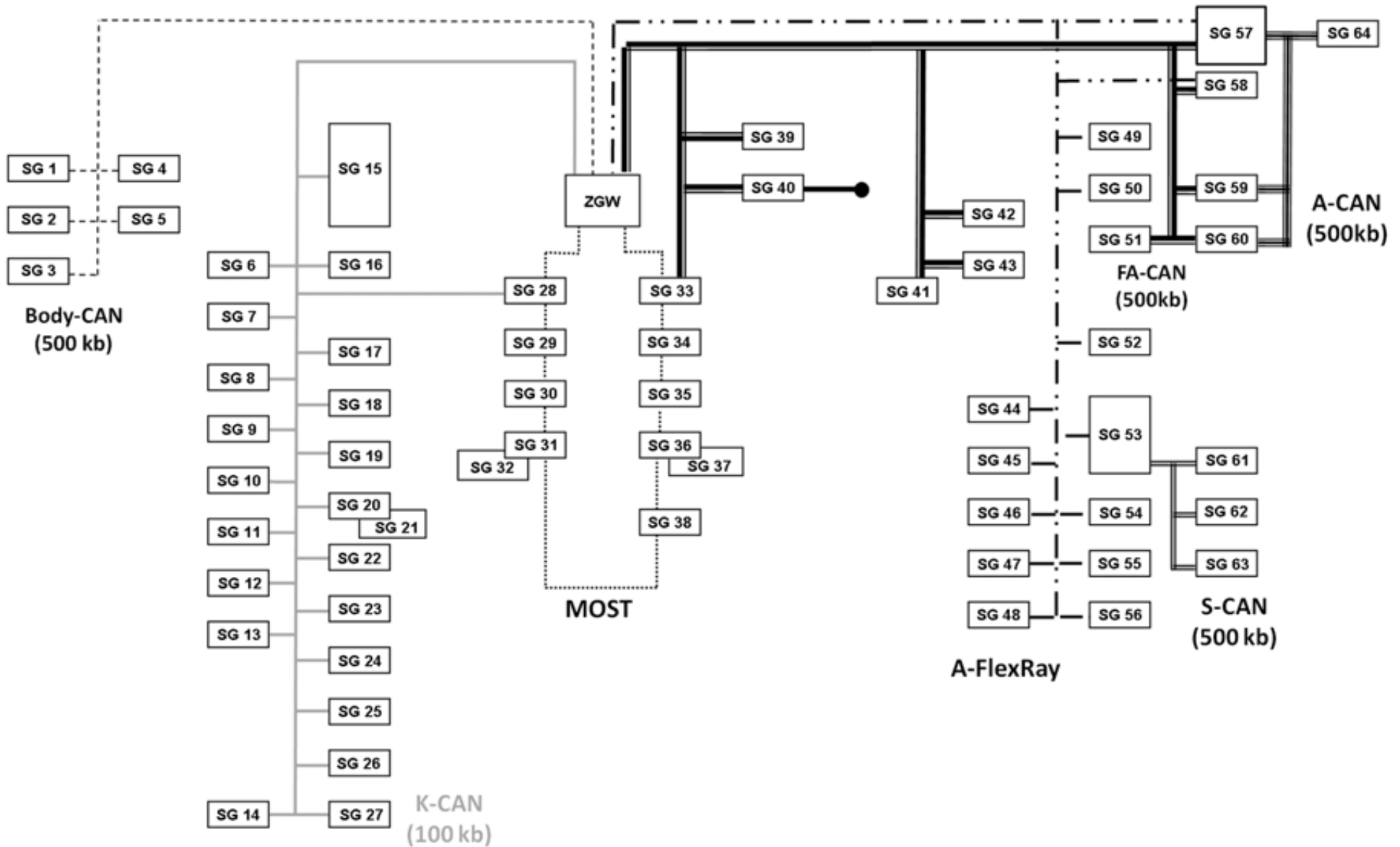


© 2013 . Vector Informatik GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector.

Slide: 6





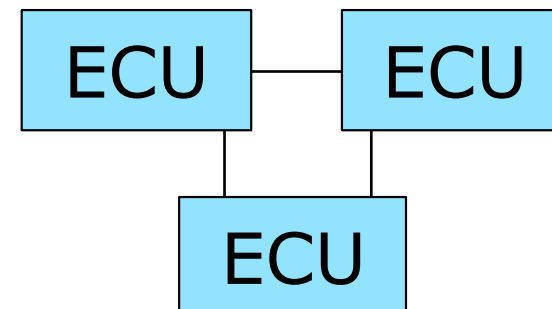
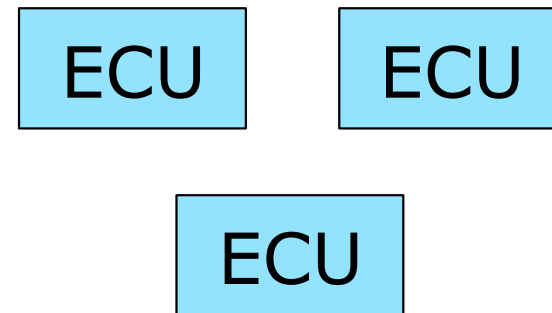


Anforderungen an Kommunikationsbusse

Hardware Layer	Topologie, Übertragungsmedium
Transferrate	Brutto- und Nettotransferrate
Zeitverhalten	Synchron (deterministisch) vs. Asynchron
Fehlererkennung	Hamming-Distanz, Busmonitoring, ACK
Ausfallsicherheit	Auswirkung ausgefallener oder störender ECUs, Redundanz
Kosten	Hardware- und Softwarekosten, Lizenzgebühren

Entwicklungsstufen der Elektronikarchitektur

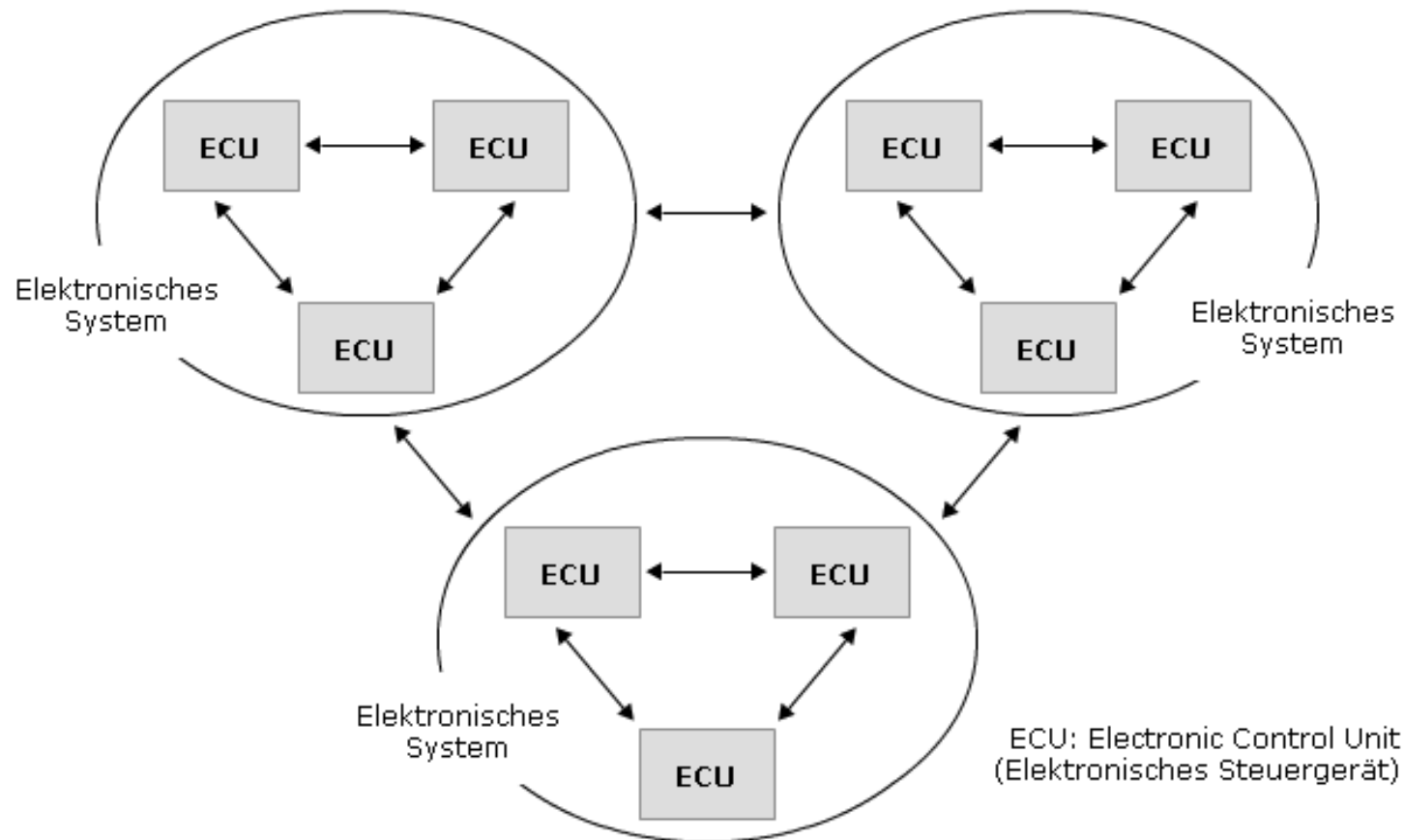
- Unabhängig voneinander arbeitende elektronische Steuergeräte
- Koordination elektronischer Steuergeräte
 - Verbesserung und Erweiterung der Fahrzeugfunktionalität.
 - Neue Funktionen durch Datenaustausch zwischen den ECUs
 - Beispiel Fahrdynamikfunktionen (ESP; ABS, ...)





Einführung

Steuergerätekopplung



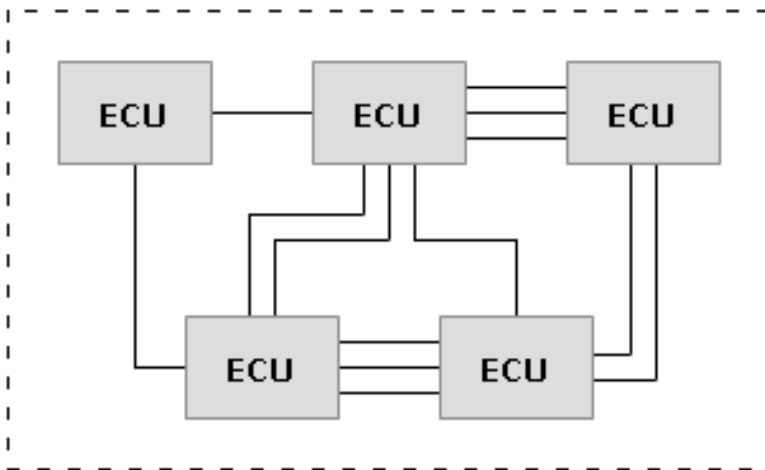
Serielle Bussysteme

- Anfangs konventionelle Vernetzung zwischen elektronischen Steuergeräten
 - Jedem zu übertragenden Signal wurde eine elektrische Leitung zugeordnet
 - Hoher Verkabelungsaufwand bei intensiver Vernetzung
 - Lösung: Bitserieller Datenaustausch von zwischen mehreren elektronischen Steuergräten über gemeinsam genutzten Kommunikationskanal (Bus)
- Heute zahlreiche serielle Bussysteme als essentieller Bestandteil moderner Elektronikarchitekturen im Kfz
 - Minimierter Verkabelungsaufwand,
 - Reduktion von
 - Kosten
 - Platzbedarf (Bauraum)
 - Gewicht
 - Fehleranfälligkeit
 - Vereinfachung von Projektierung und Installation
 - Zuverlässiger und sicherer Datenaustausch in Echtzeit
 - Hohe Flexibilität bei Änderungen und Erweiterungen.

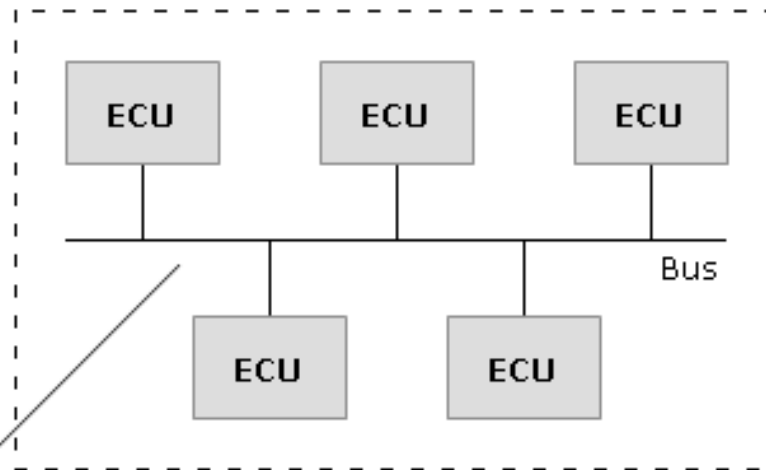


Einführung
Busvernetzung

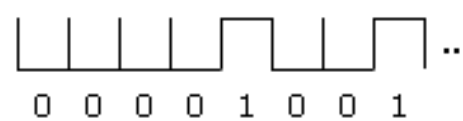
Konventionelle Vernetzung



Busvernetzung



Bitserielle
Datenübertragung

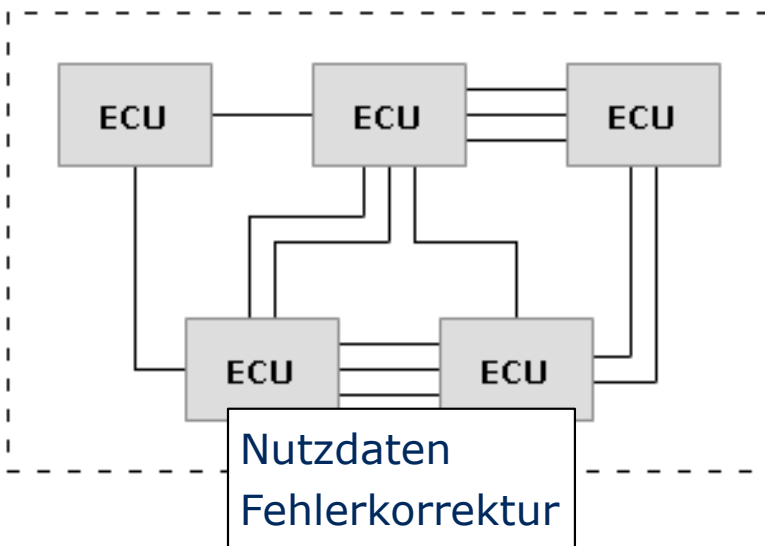


ECU: Electronic Control Unit
(Elektronisches Steuergerät)

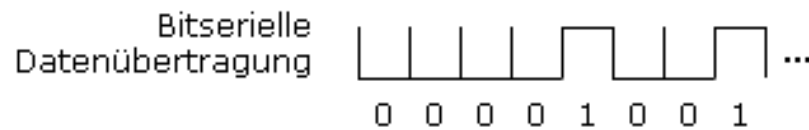
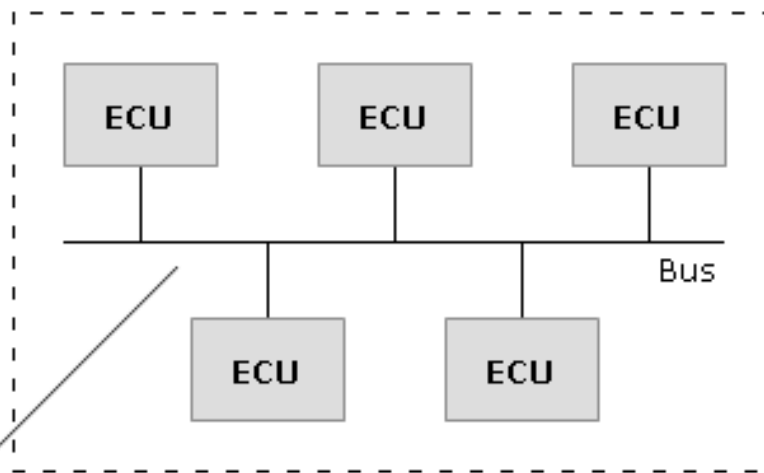


Einführung
Busvernetzung

Konventionelle Vernetzung



Busvernetzung

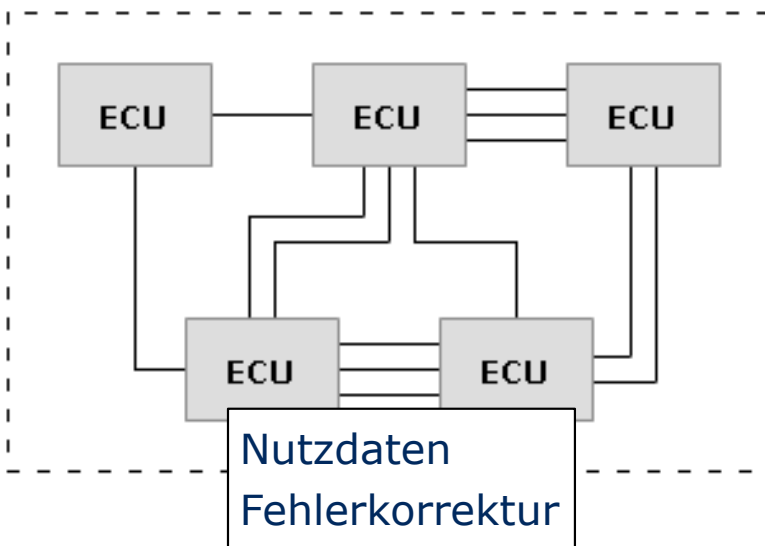


ECU: Electronic Control Unit
(Elektronisches Steuergerät)

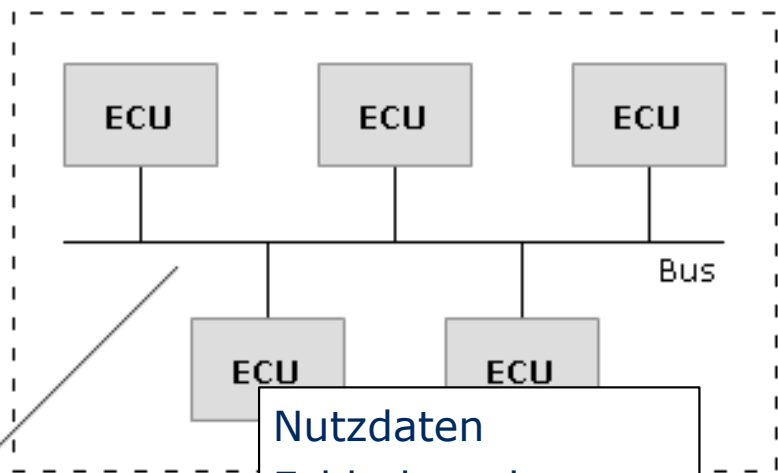


Einführung
Busvernetzung

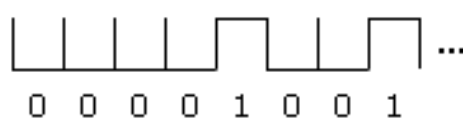
Konventionelle Vernetzung



Busvernetzung



Bitserielle
Datenübertragung

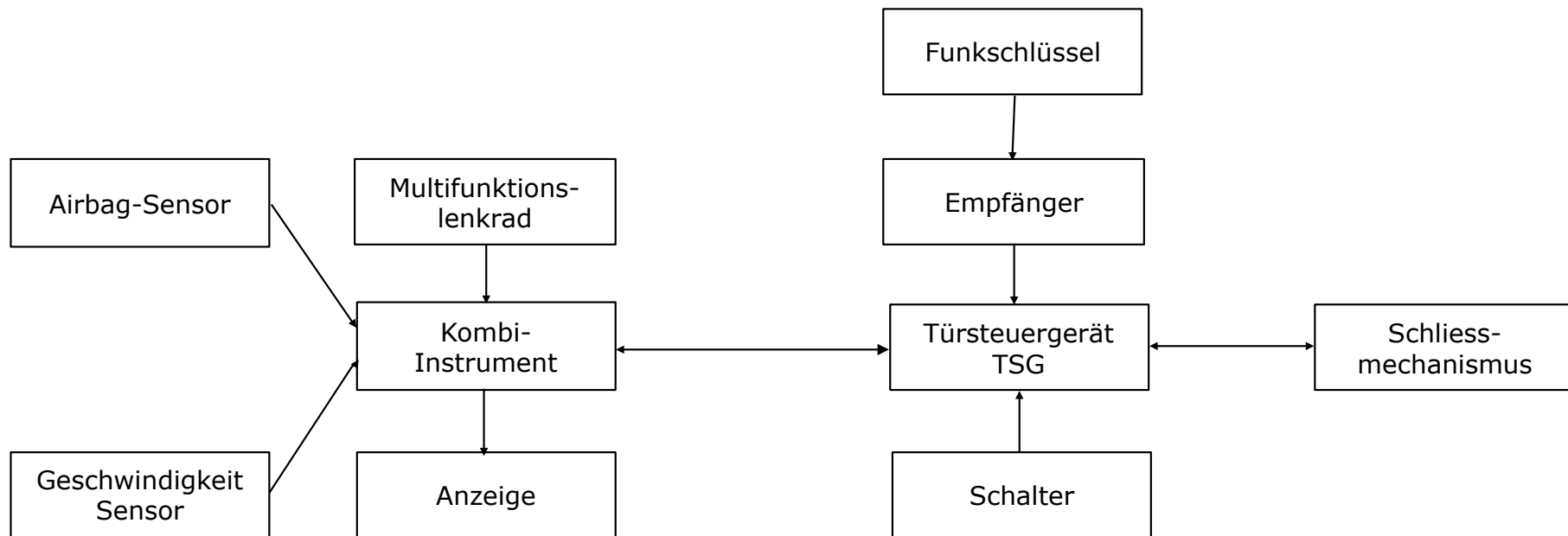


Unit
gerät)



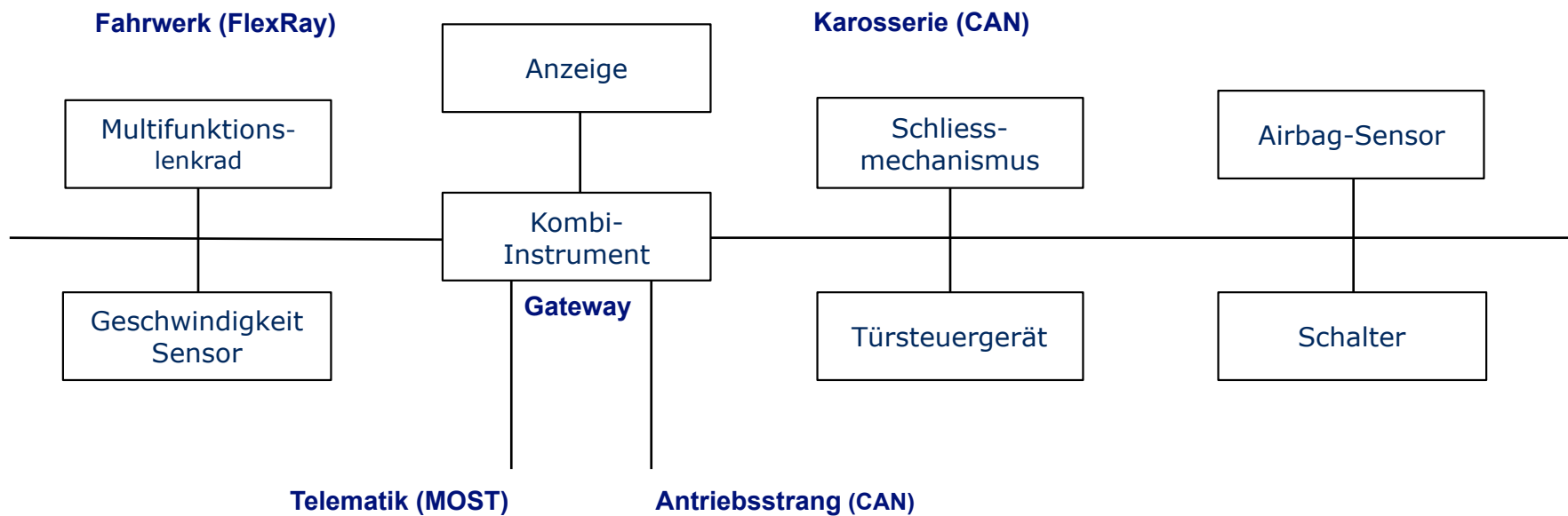
Beispiel Türsteuerung aus Teil 1

Funktionale Architektur (Ausschnitt)



Beispiel Türsteuerung aus Teil 1

Mögliche Vernetzung (Ausschnitt)



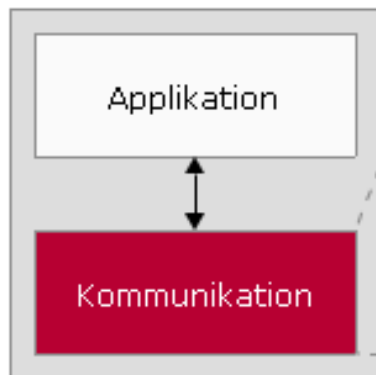
Kommunikationsmodelle

- Im einfachsten Fall werden über einen Bus zwei elektronische Steuergeräte, die sog. Busknoten, zu einem seriellen Kommunikationssystem zusammengeschlossen. Damit die Busknoten reibungslos Daten austauschen können, sind klare Kommunikationsstrukturen und -regeln unerlässlich.
- ISO/OSI-Modell
Einen wesentlichen Beitrag dazu leistet das von der ISO (International Standardization Organization) 1983 veröffentlichte ISO/OSI-Kommunikationsmodell (Open System Interconnection). Wie aus der Grafik ISO/OSI-Modell ersichtlich, gliedert dieses Modell den Kommunikationsvorgang mit seinen zahlreichen und komplexen Aufgaben in sieben überschaubare aufeinander aufbauende Schichten (Siebenschichtenmodell) und definiert die Kommunikation zwischen den Schichten.
- Dreischichtenmodell
Da für den seriellen Datenaustausch zwischen elektronischen Steuergeräten im Kfz im Wesentlichen nur die beiden unteren Schichten (Bitübertragungs- und Datensicherungsschicht) von Bedeutung sind, zudem Funktionen nicht berücksichtigter Schichten der obersten Schicht (Anwendungsschicht) zugeschlagen werden, reduziert sich das Siebenschichtenmodell auf ein Dreischichtenmodell.



Einführung
ISO/OSI-Modell

Busknoten

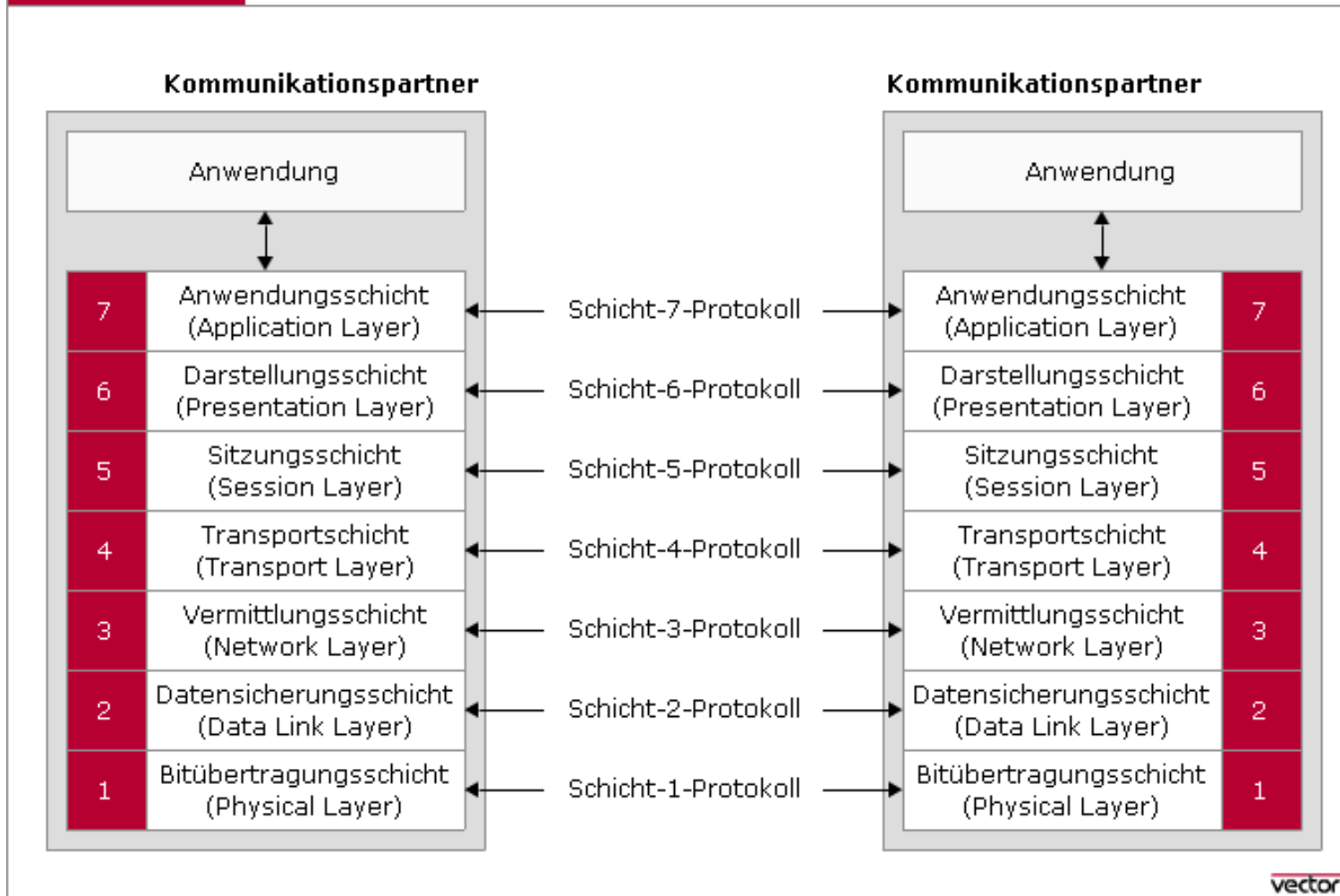


7	Anwendungsschicht (Application Layer)	Zugriff auf das Kommunikationssystem, Entkopplung Anwendung von Kommunikation
6	Darstellungsschicht (Presentation Layer)	Semantik, Kompression, Verschlüsselung
5	Sitzungsschicht (Session Layer)	Unterhalten längerer Sitzungen, Definition von Synchronisationspunkten
4	Transportschicht (Transport Layer)	Verbindungsauf- und abbau, Segmentierung, Sequenzierung, Assemblierung
3	Vermittlungsschicht (Network Layer)	Routing (Wahl des Übertragungsweges zwischen zwei Busknoten)
2	Datensicherungsschicht (Data Link Layer)	Datensicherung, Nachrichtenbildung, Buszugriff
1	Bitübertragungsschicht (Physical Layer)	Physikalische Busankopplung, Stecker, Übertragungsmedium, Leitungscodierung



Einführung

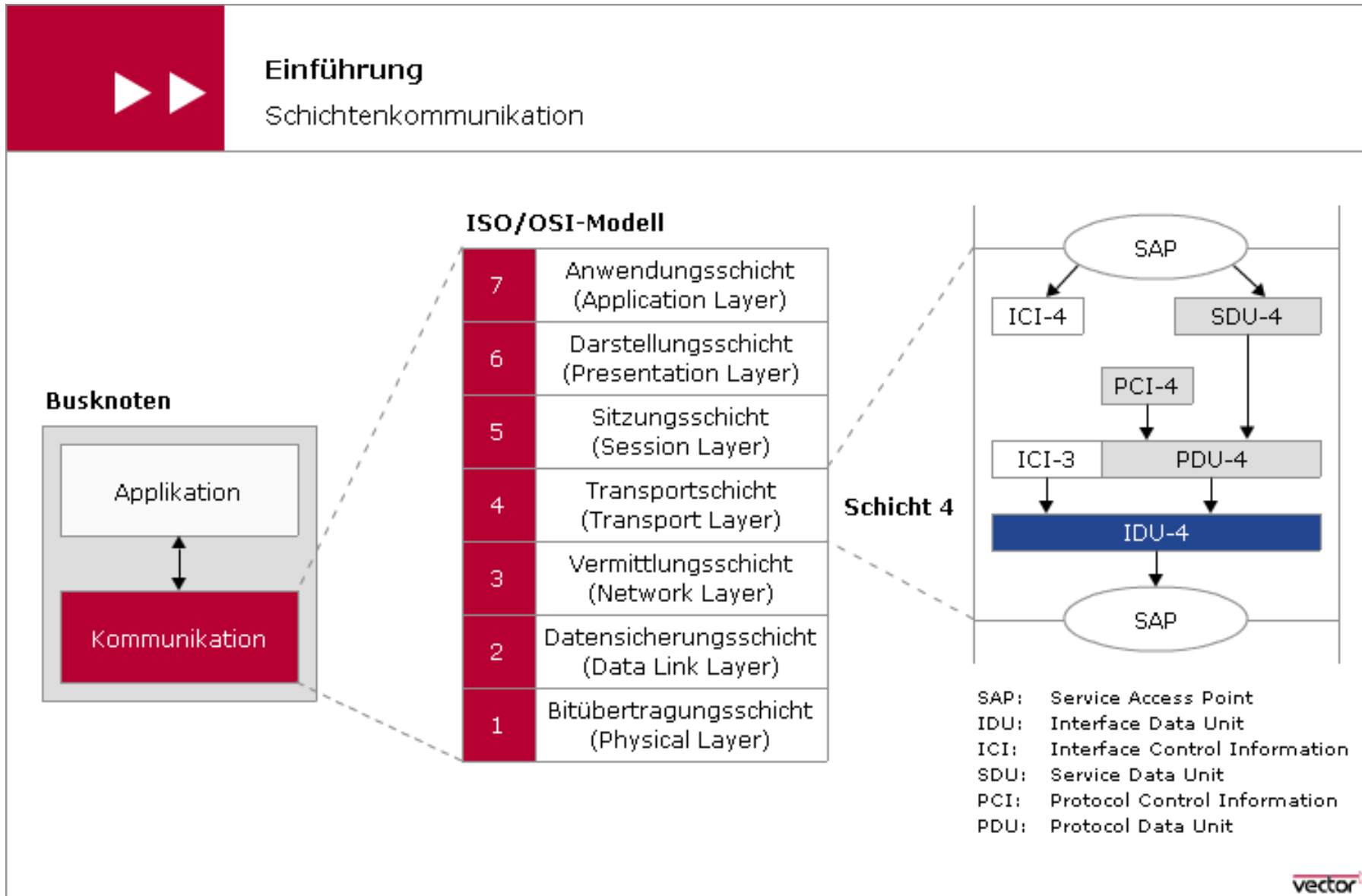
Peer-to-Peer-Kommunikation





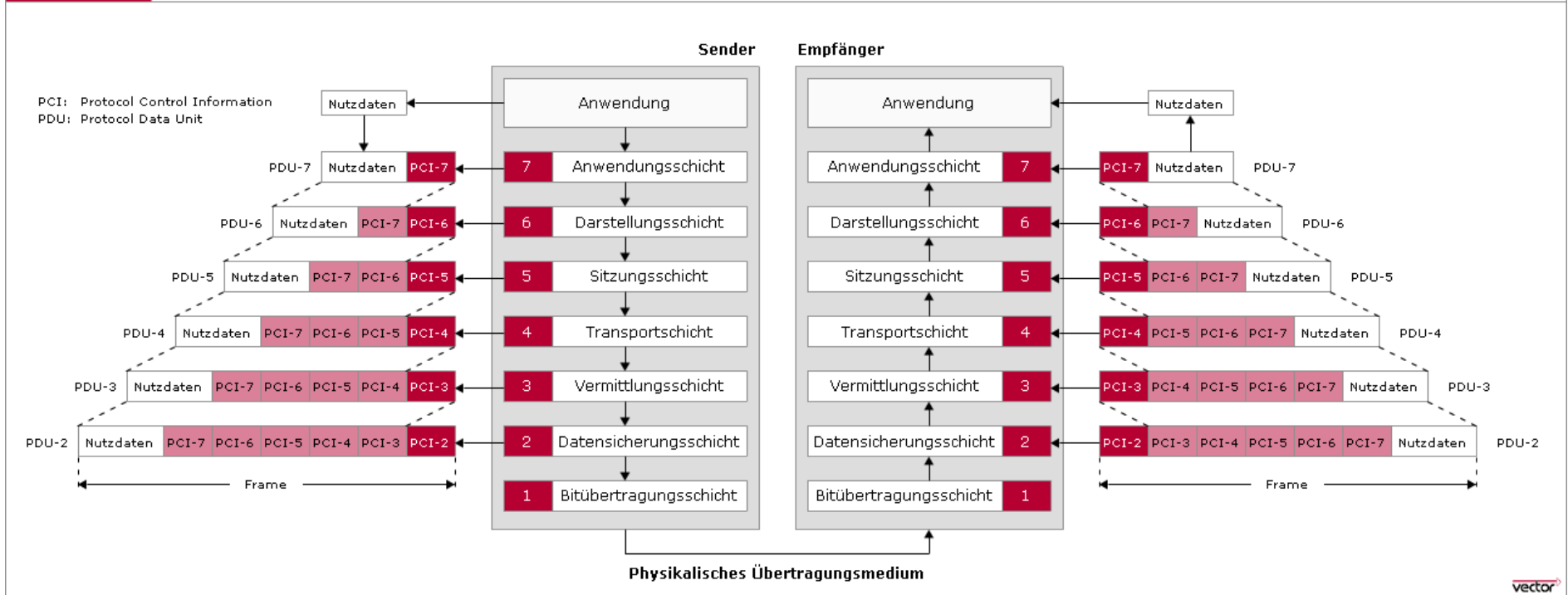
Einführung

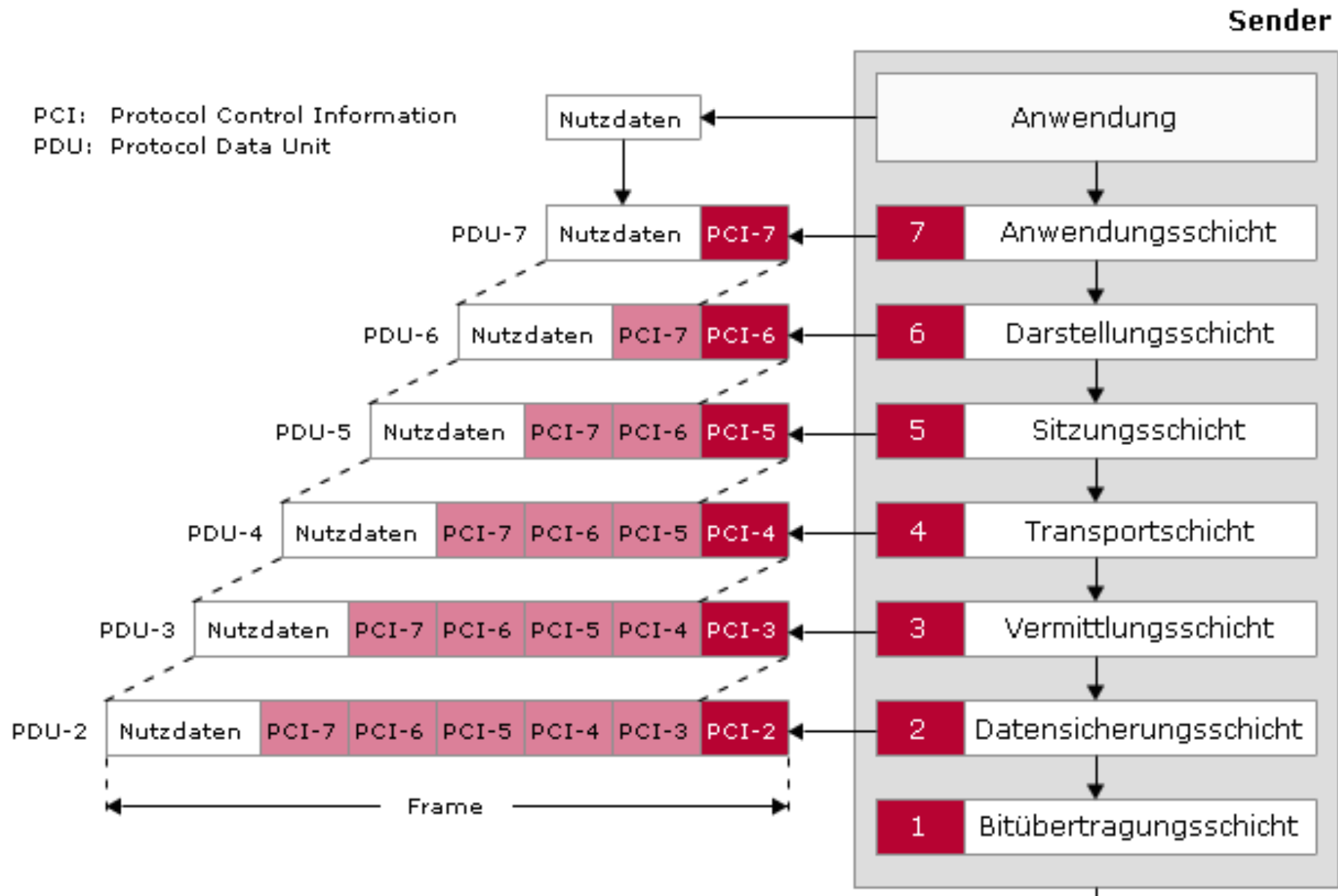
Schichtenkommunikation





Einführung
Realer Kommunikationsfluss





Dreischichtenmodell (1)

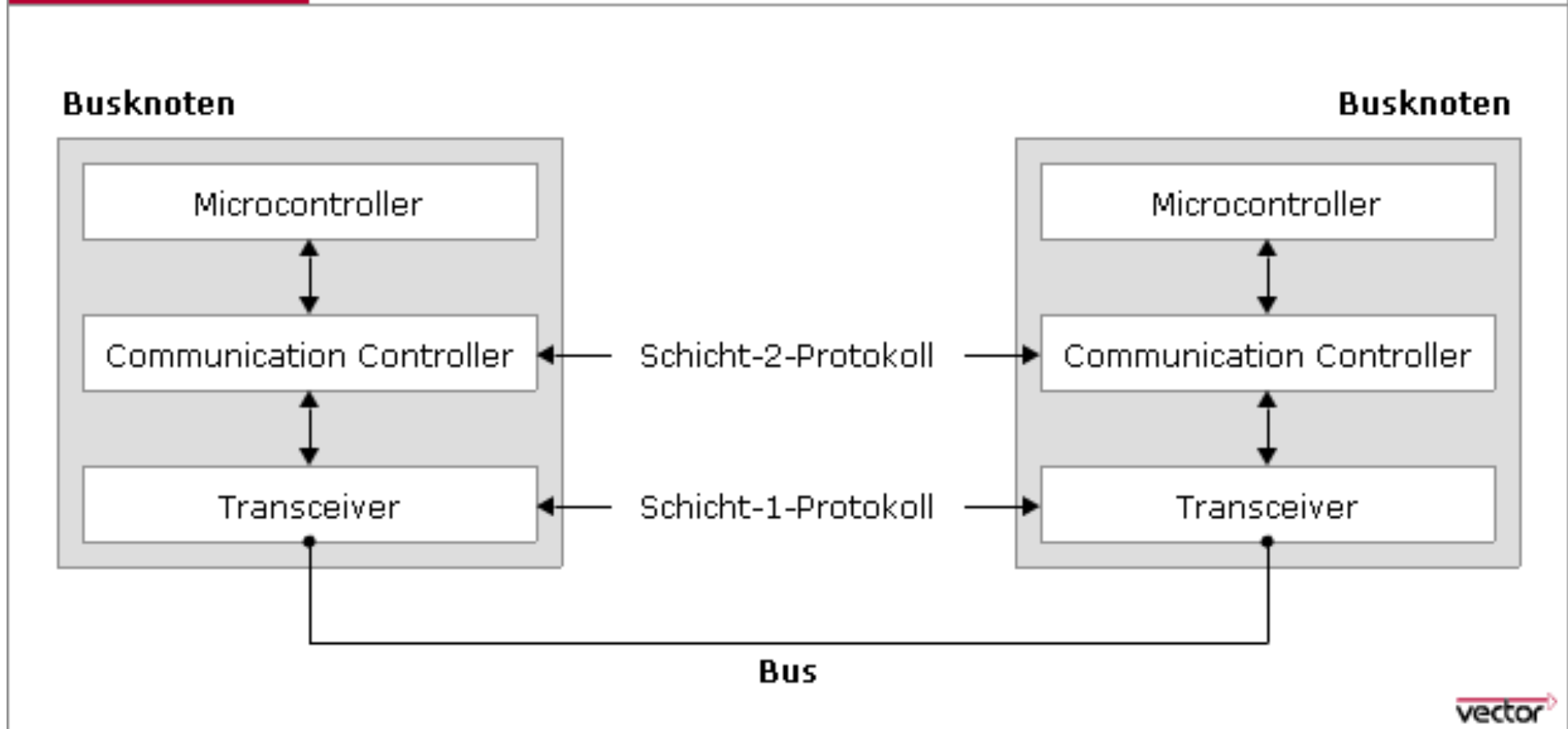
Für den seriellen Datenaustausch zwischen elektronischen Steuergeräten im Kfz sind nicht alle im ISO/OSI-Modell definierten Kommunikationsfunktionen notwendig. Für die serielle Datenkommunikation im Kfz sind prinzipiell nur die beiden unteren Schichten von Bedeutung. Dabei handelt es sich um die Datensicherungsschicht (Data Link Layer) und die Bitübertragungsschicht (Physical Layer).

Die Datensicherungsschicht nimmt folgende Aufgaben wahr: Adressierung, Nachrichtenbildung (Framing), Buszugriff, Synchronisation, Fehlererkennung und Fehlerkorrektur. Wie diese Aufgaben ausgeführt werden, definiert das Schicht-2-Protokoll. Üblicherweise übernimmt die Ausführung der Schicht-2-Aufgaben ein Kommunikationscontroller.

Die Bitübertragungsschicht enthält die Beschreibung der physikalischen Busankopplung und Vereinbarungen zur physikalischen Signalübertragung. Üblicherweise realisiert man die physikalische Busankopplung mit Hilfe eines Transceivers. Die wesentlichste Aufgabe eines Transceivers besteht beim Senden darin, die vom Kommunikationscontroller empfangenen Daten in die im Schicht-1-Protokoll definierten Spannungspegel umzusetzen. Umgekehrt übersetzt der Transceiver vom Bus empfangene Spannungspegel in für den Kommunikationscontroller handhabbare Eingangssignale.



Einführung Busknotenarchitektur



Dreischichtenmodell (2)

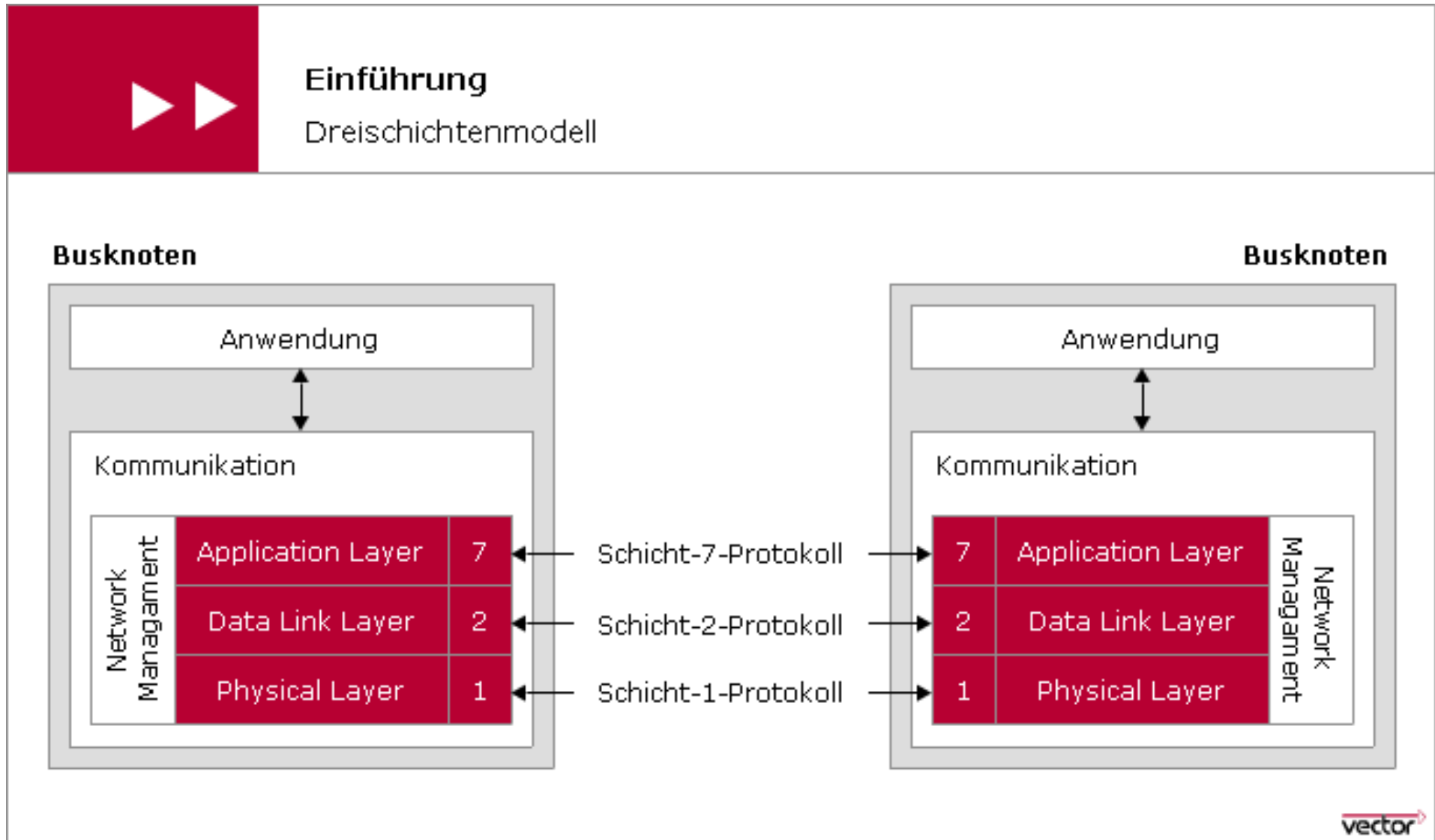
Die Reduktion des sog. Kommunikationsstacks auf zwei Schichten vereinfacht die Handhabung der Kommunikation und sorgt zudem für eine schnellere Kommunikation, was der Erfüllung anspruchsvollen Zeitbedingungen entgegen kommt. Funktionen nicht berücksichtigter Schichten werden üblicherweise der obersten Schicht (Anwendungsschicht) zugeschlagen. Der elektronischen Datenkommunikation im Kfz liegt also prinzipiell ein Dreischichtenmodell zugrunde.

Mit der Definition einer Anwendungsschicht verlagert sich die Schnittstelle zwischen der Anwendung und der Kommunikation von der Datensicherungsschicht zur Anwendungsschicht. Quer zu den drei Schichten liegt das Netzwerkmanagement, das schichtenübergreifend organisatorische Funktionen übernimmt, wie beispielsweise die Netzwerkinitialisierung sowie das Schlafenlegen und Aufwecken des Netzwerkes.



Einführung

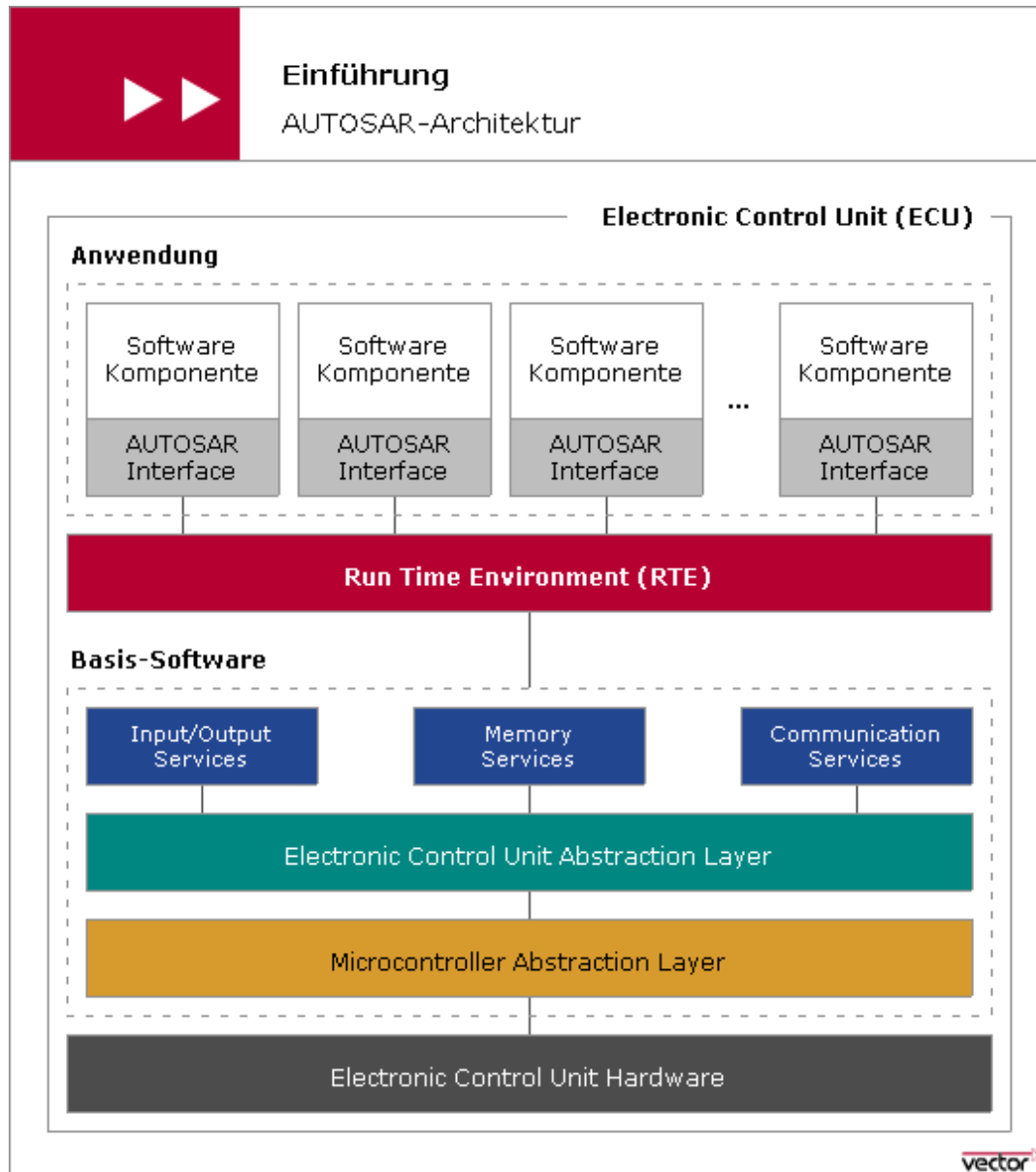
Dreischichtenmodell





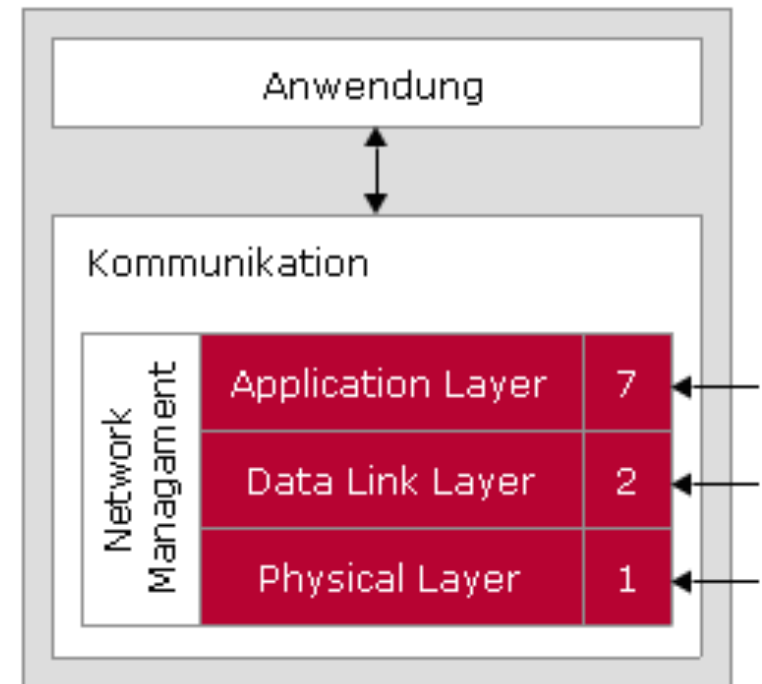
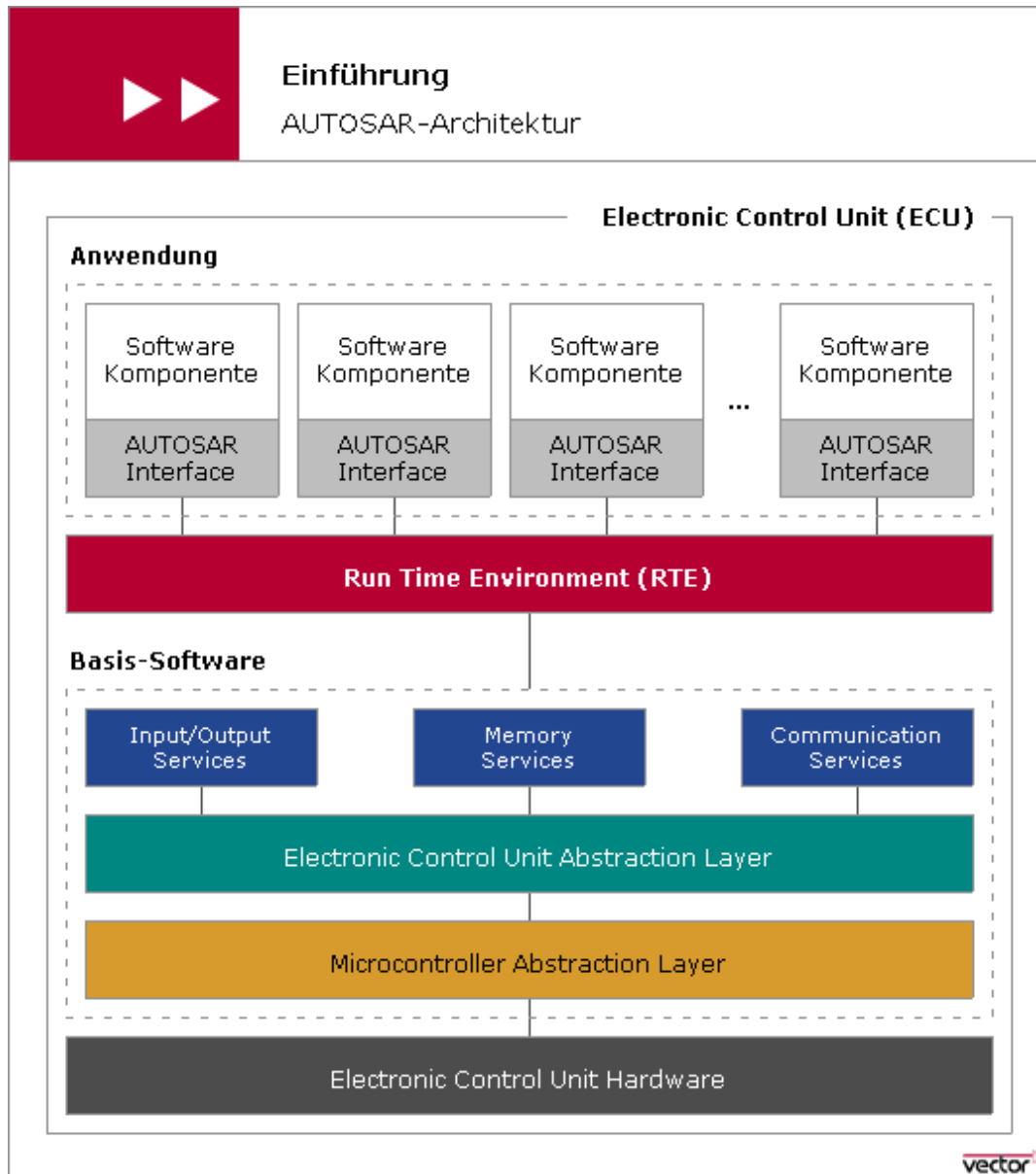
Einführung

AUTOSAR-Architektur



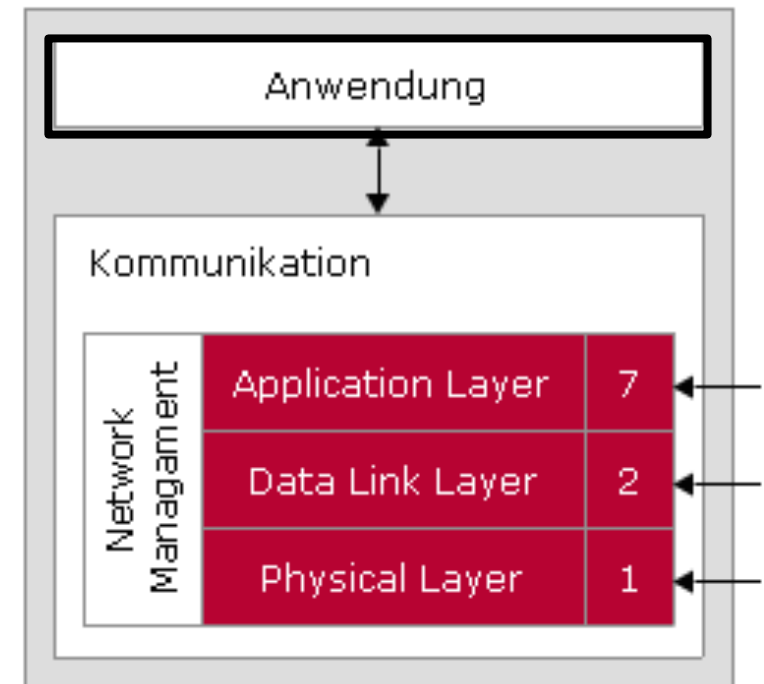
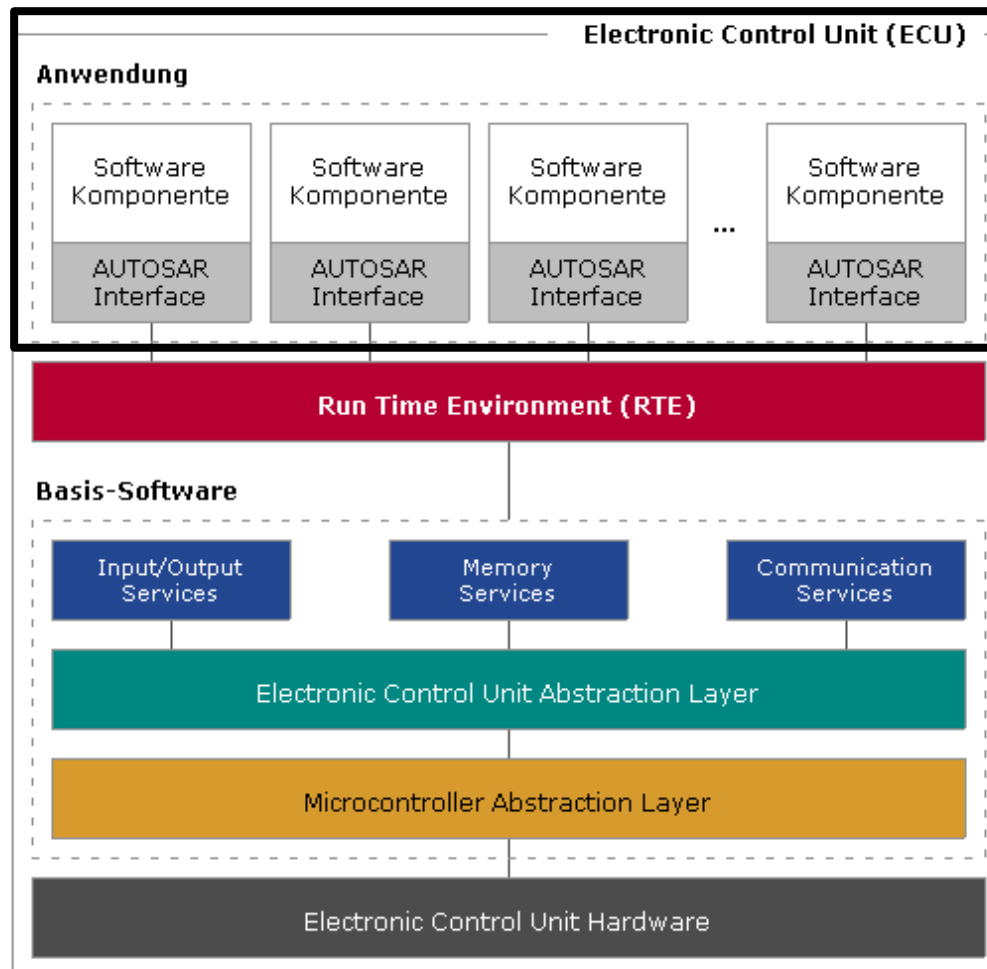


Einführung
AUTOSAR-Architektur



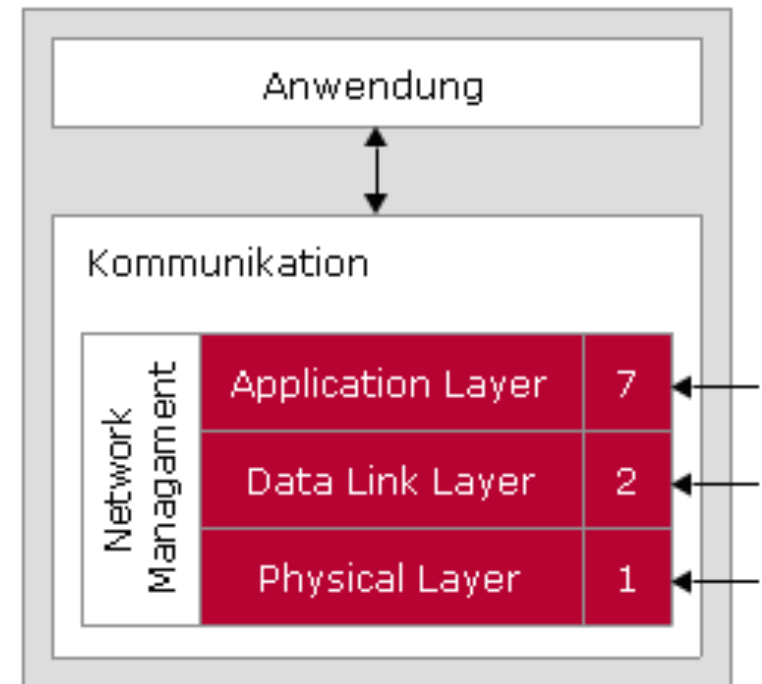
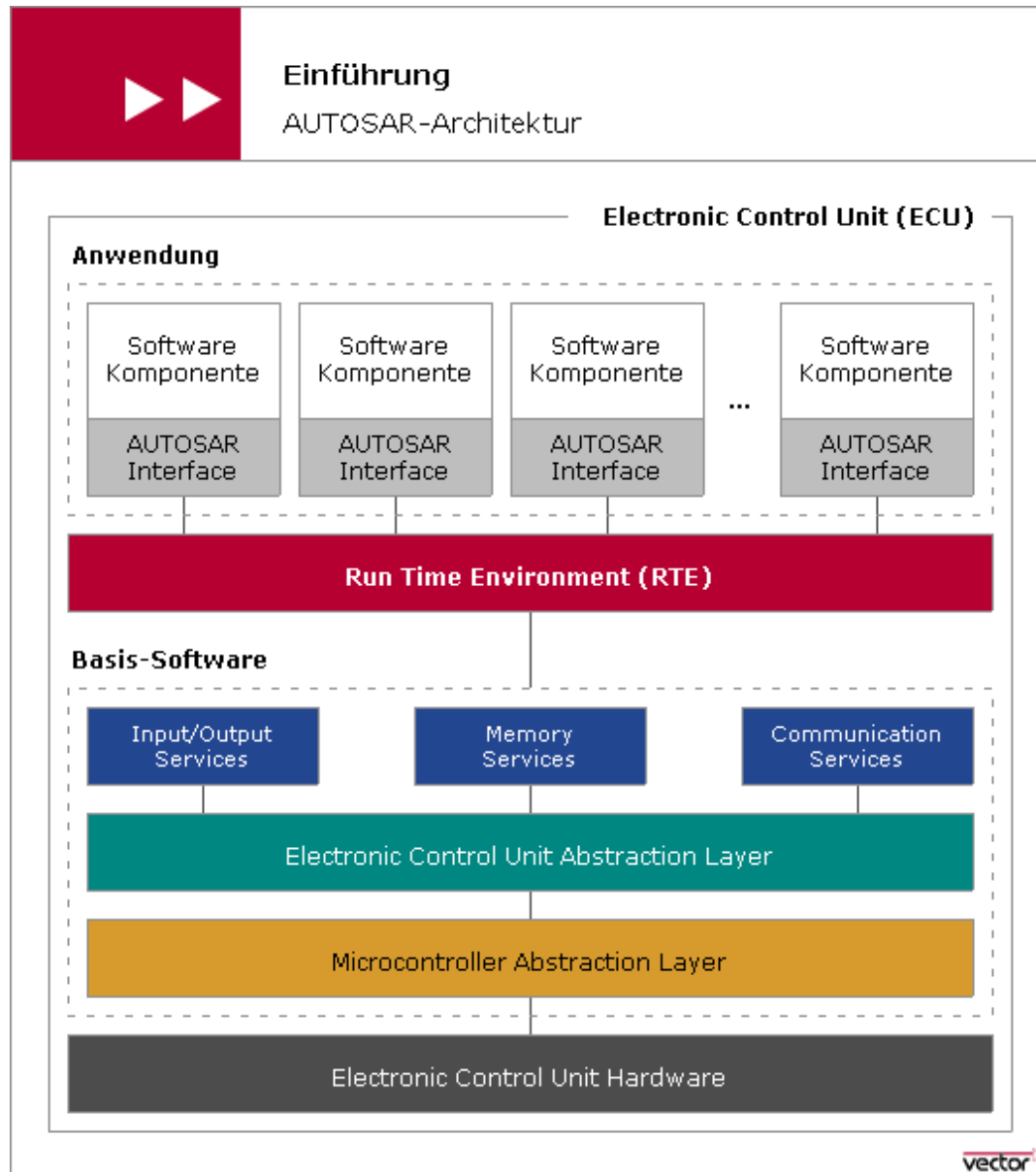


Einführung
AUTOSAR-Architektur





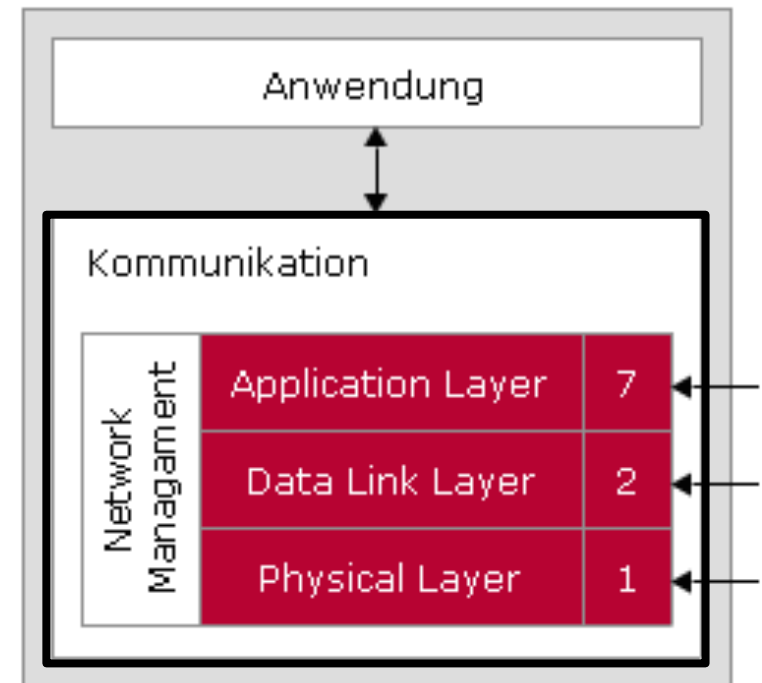
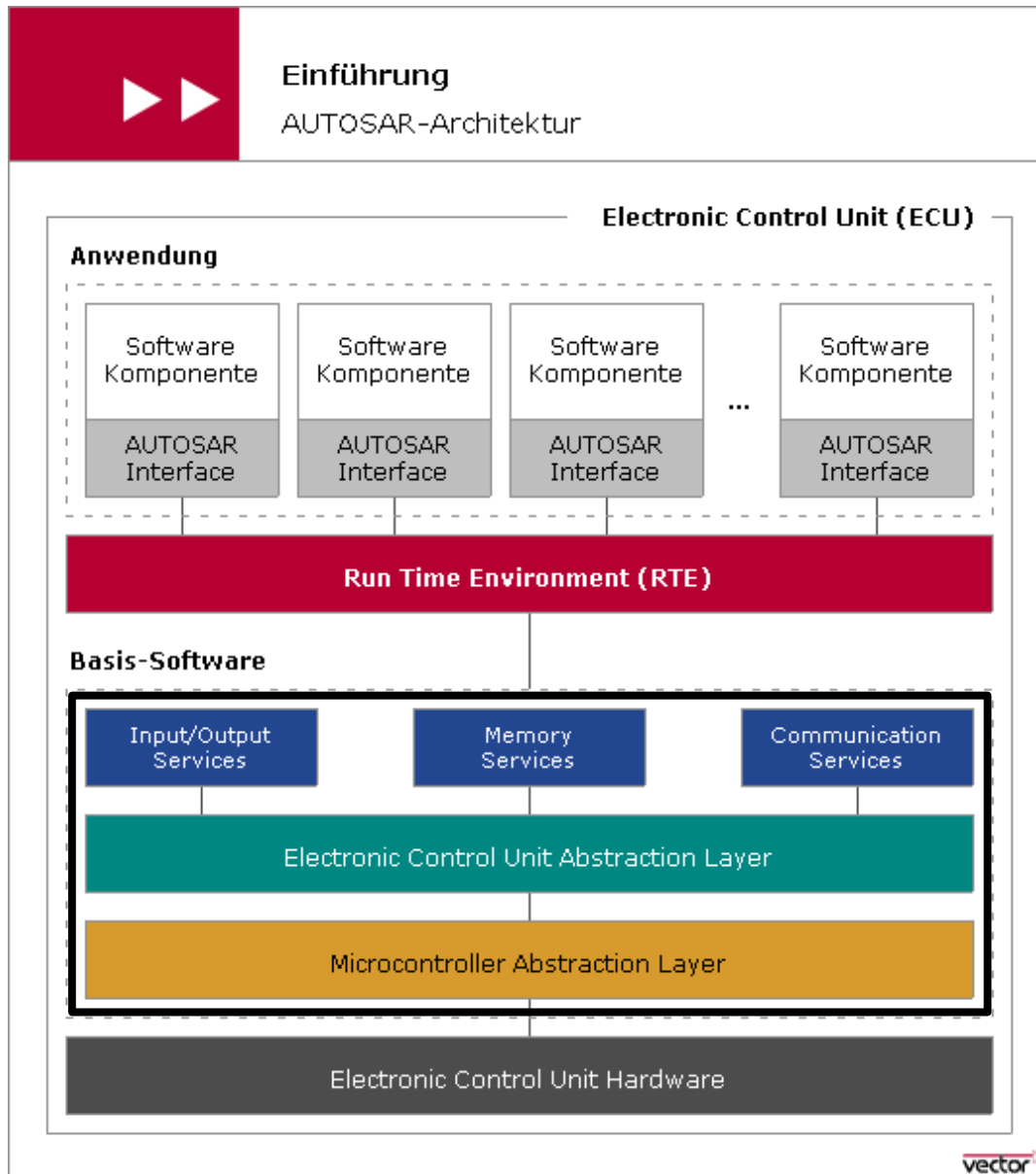
Einführung
AUTOSAR-Architektur





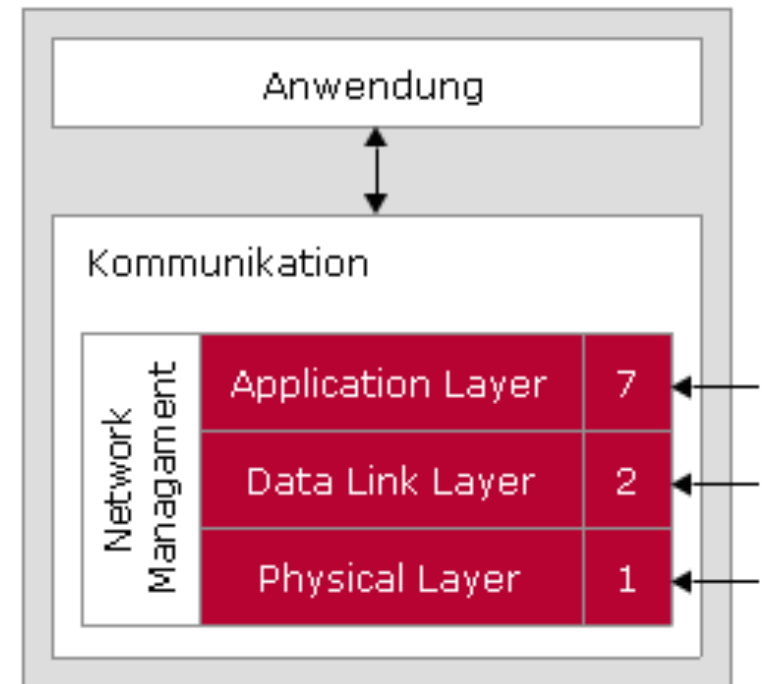
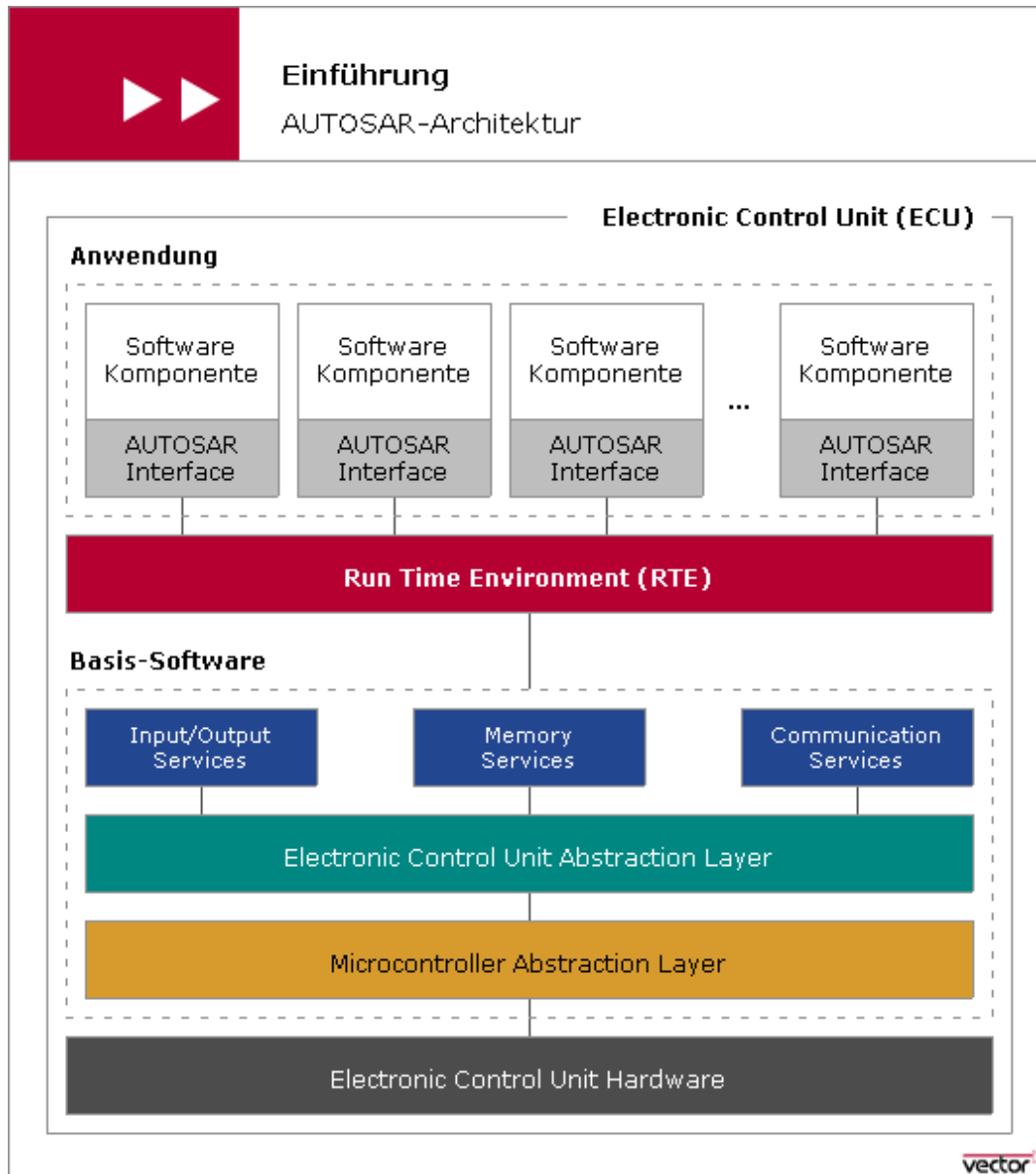
Einführung

AUTOSAR-Architektur





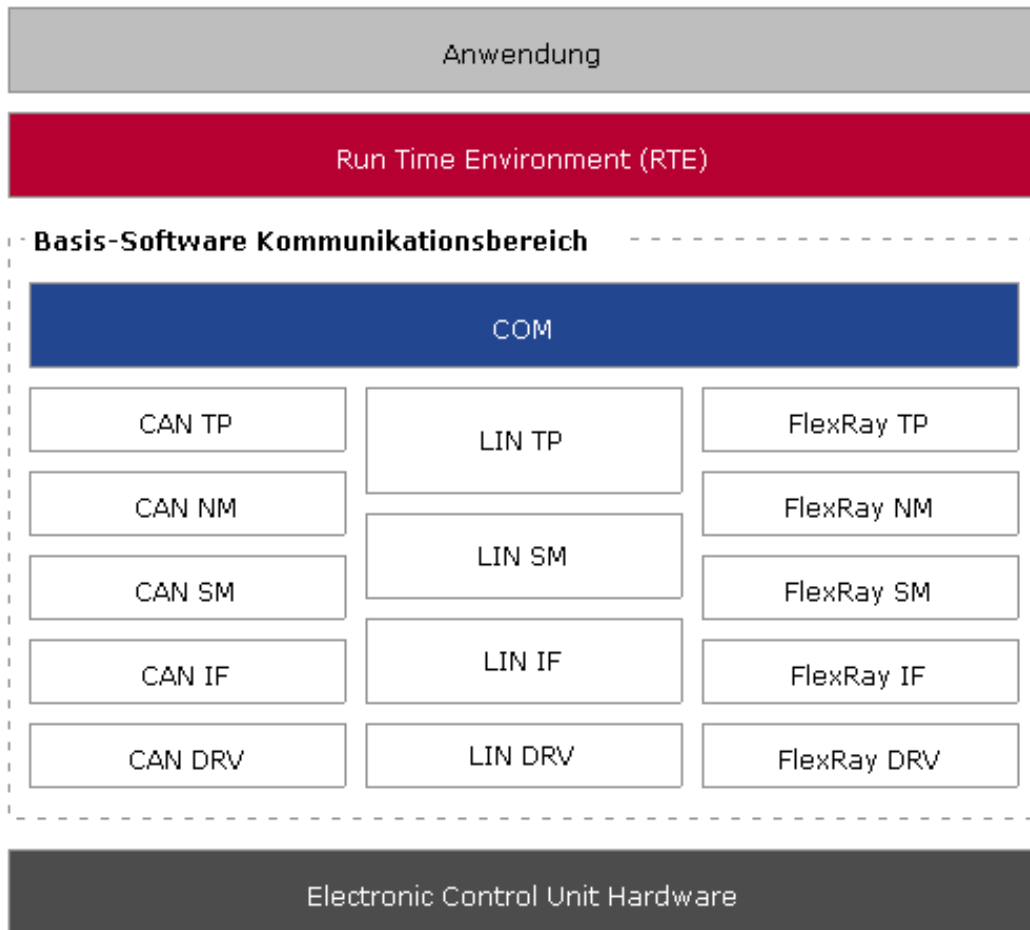
Einführung
AUTOSAR-Architektur





Einführung

AUTOSAR-Architektur



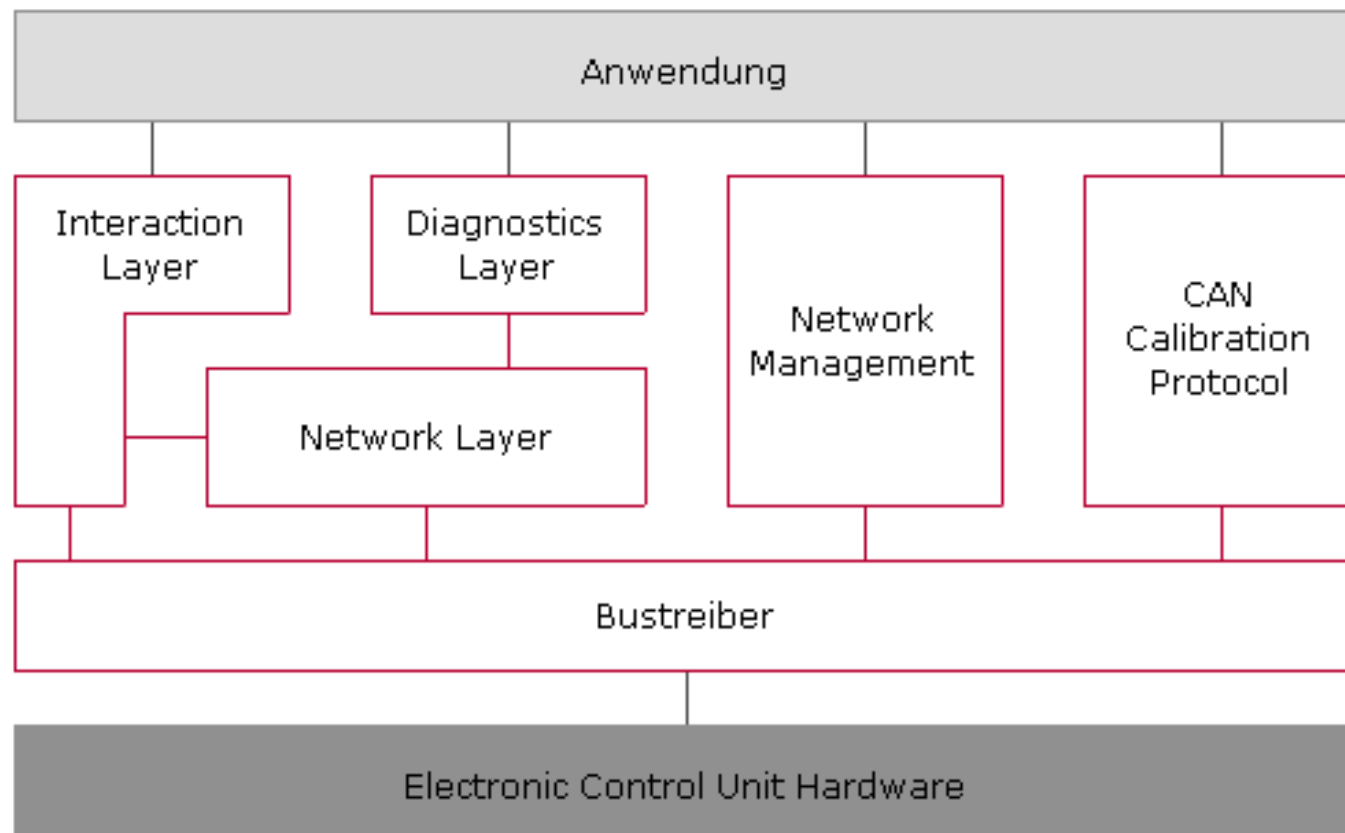
COM: Communication
 TP: Transportprotokoll
 NM: Netzmanagement
 SM: Statusmanagement
 IF: Interface
 DRV: Driver





Einführung

ISO/OSEK-Architektur



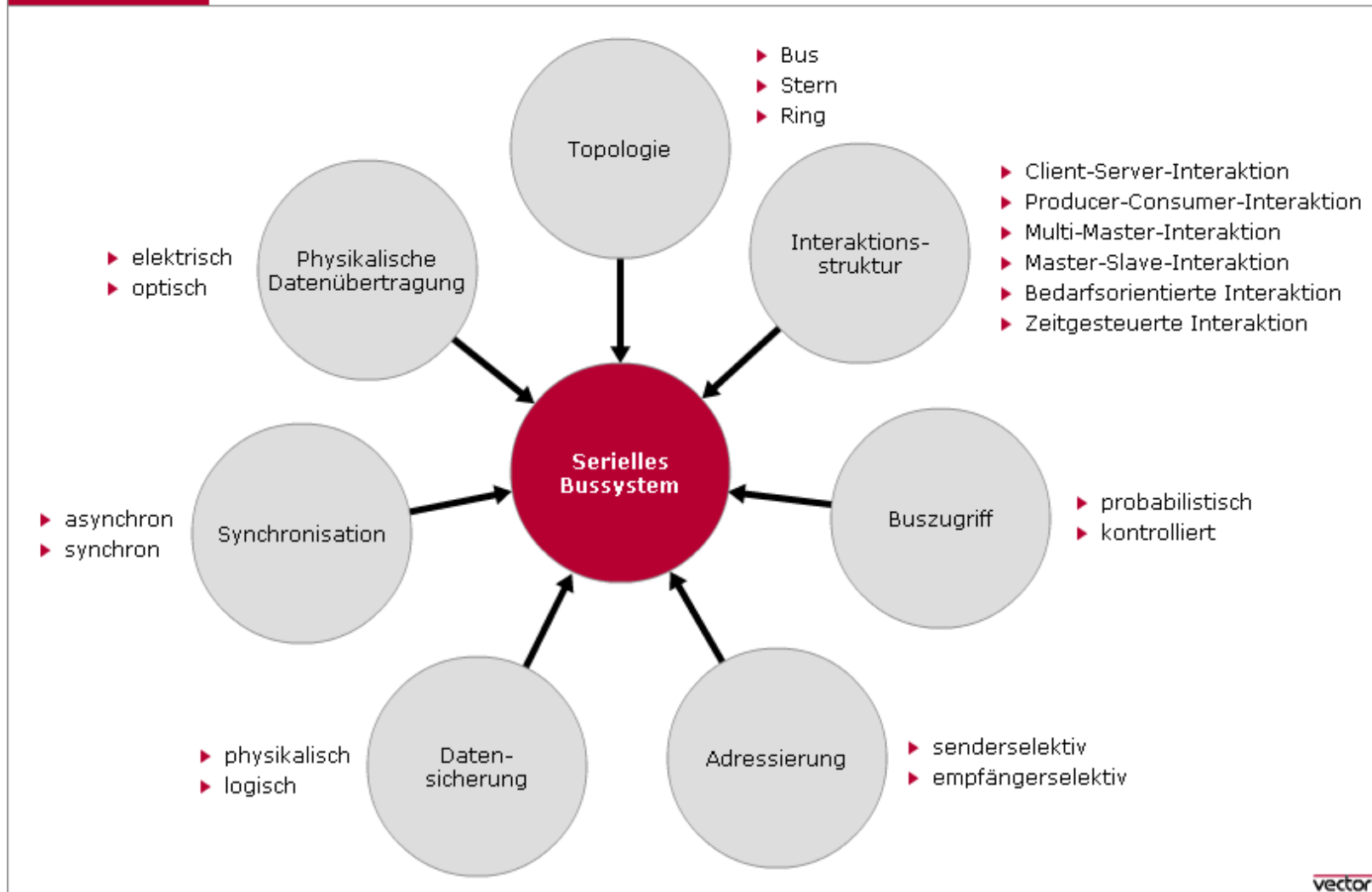
Serielle Bussysteme

- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
 - 3.4 MOST
 - 3.5 Ethernet
 - 3.6 Beispiele



Einführung

Komponenten serieller Bussysteme

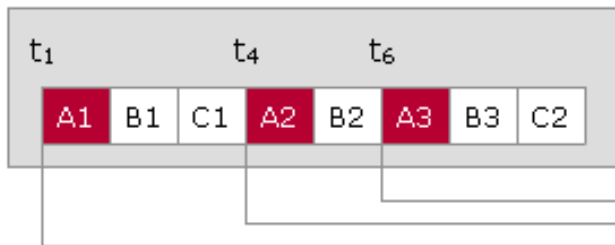




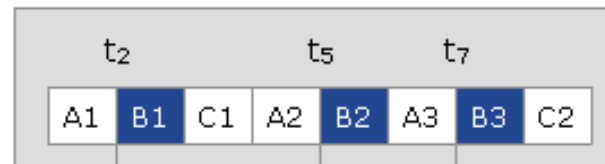
Architektur serieller Bussysteme

Systemintegration

Busknoten A

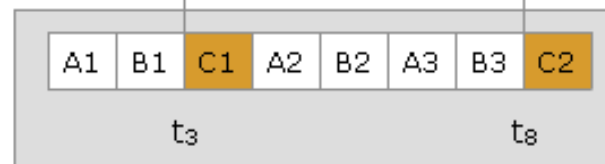
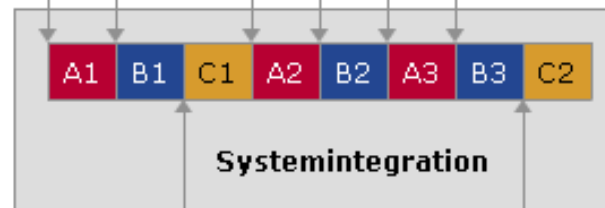


Busknoten B



Kommunikationsablaufplan

t ₁	Zeitschlitz 1	Botschaft A1	Busknoten A
t ₂	Zeitschlitz 2	Botschaft B1	Busknoten B
t ₃	Zeitschlitz 3	Botschaft C1	Busknoten C
t ₄	Zeitschlitz 4	Botschaft A2	Busknoten A
t ₅	Zeitschlitz 5	Botschaft B2	Busknoten B
t ₆	Zeitschlitz 6	Botschaft A3	Busknoten A
t ₇	Zeitschlitz 7	Botschaft B3	Busknoten B
t ₈	Zeitschlitz 8	Botschaft C2	Busknoten C



Busknoten C

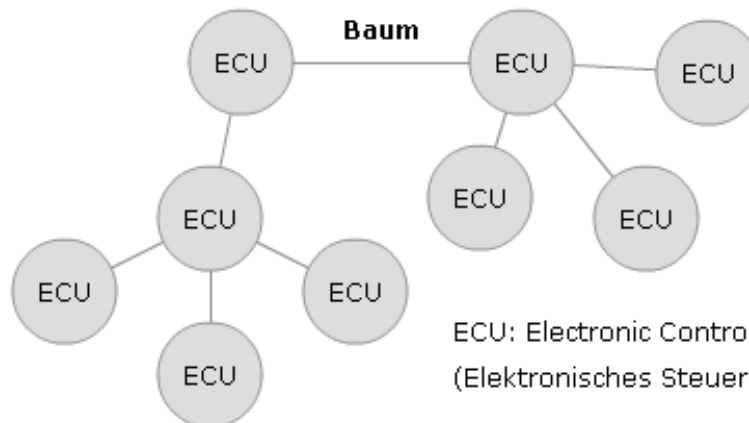
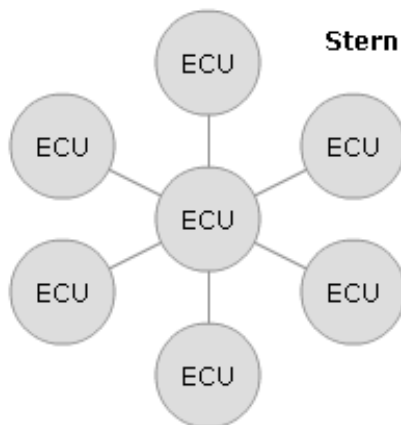
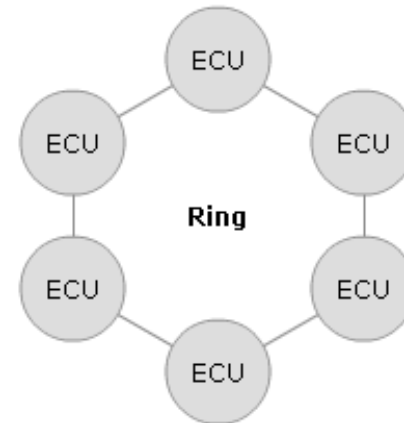
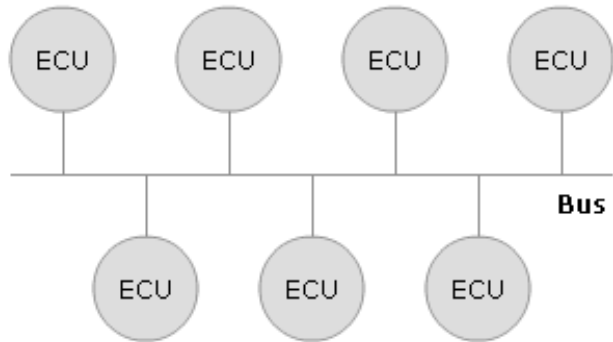
Serielle Bussysteme

- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - **2.2 Topologie**
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
 - 3.4 MOST
 - 3.5 Ethernet
 - 3.6 Beispiele



Architektur serieller Bussysteme

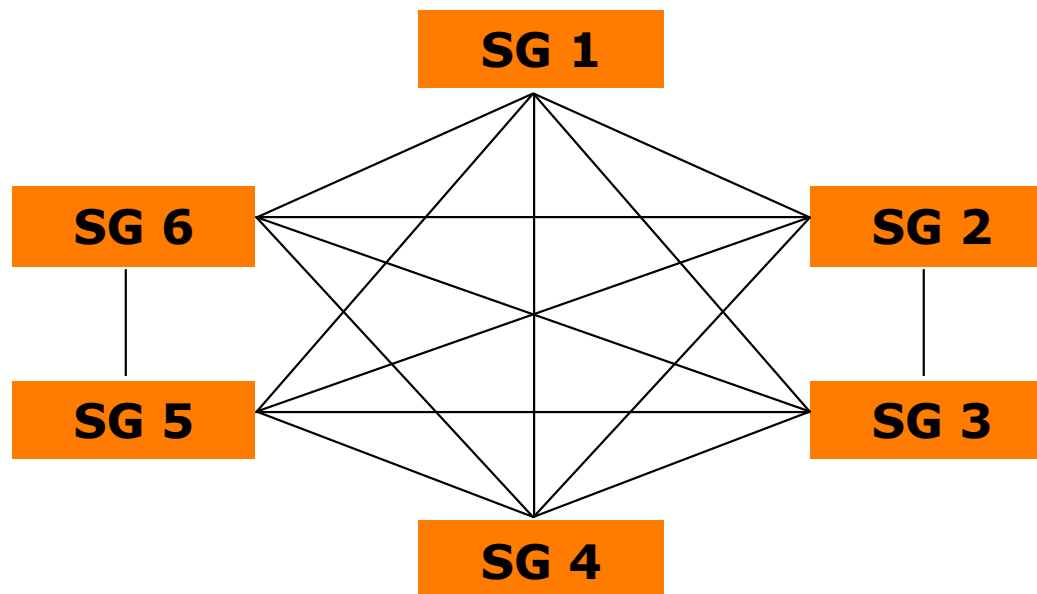
Topologien



ECU: Electronic Control Unit
(Elektronisches Steuergerät)

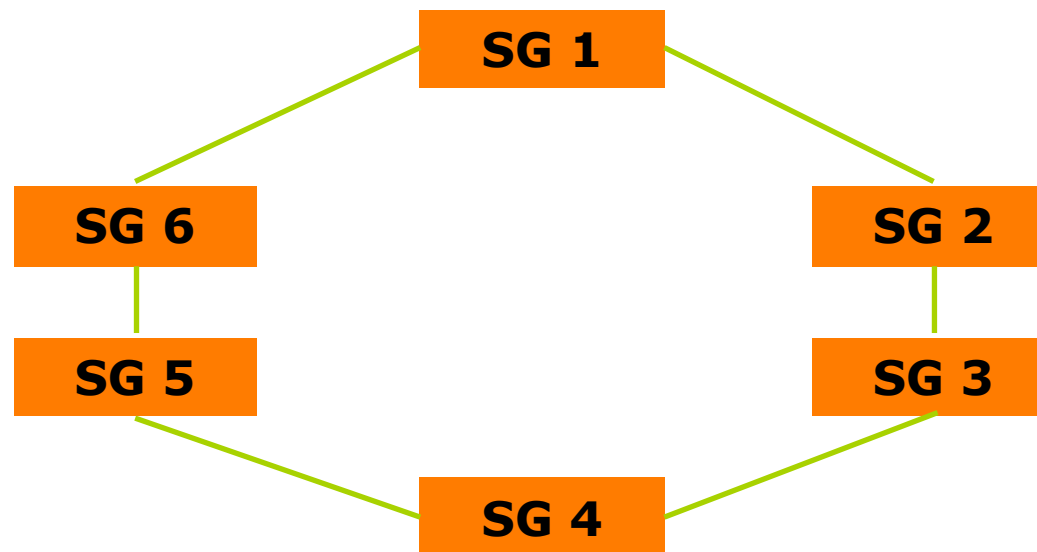
Topologien - Direkte Verbindung

- Bei direkter Verbindung sind für 6 SGe 15 Leitungen und 30 Transceiver notwendig
- Keine Probleme mit Deterministik
- Datenmanagement im SG sehr aufwändig
 - Anwendung lediglich für Point-to-Point-Verbindung
 - Nicht benutzbar für große Systeme (Gewicht, Kosten, Datenmanagement)



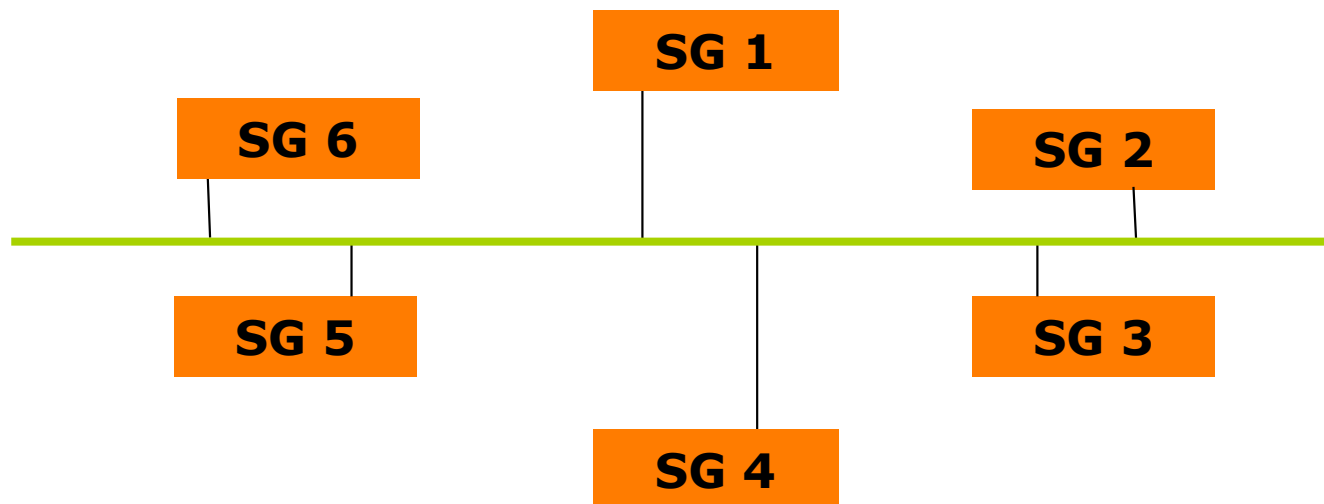
Topologien - Ring

- Im Ring sind für 6 SGe 6 Leitungen und 12 Single Ended Transceiver notwendig
- Protokollstruktur z.B. durch Tokens, gewährleistet hohe Performance, aber Einbrüche der Übertragung bei Verlust des Tokens
- Busmaster notwendig, aber Backup durch beliebiges anderes SG möglich
- Gewährleistung der Kommunikation bei Ausfall der Transceiver eines SG schwierig
 - Hervorragend für Breitband-Datendienste (Infotainment)
 - Nicht geeignet für sicherheitskritische Anwendungen



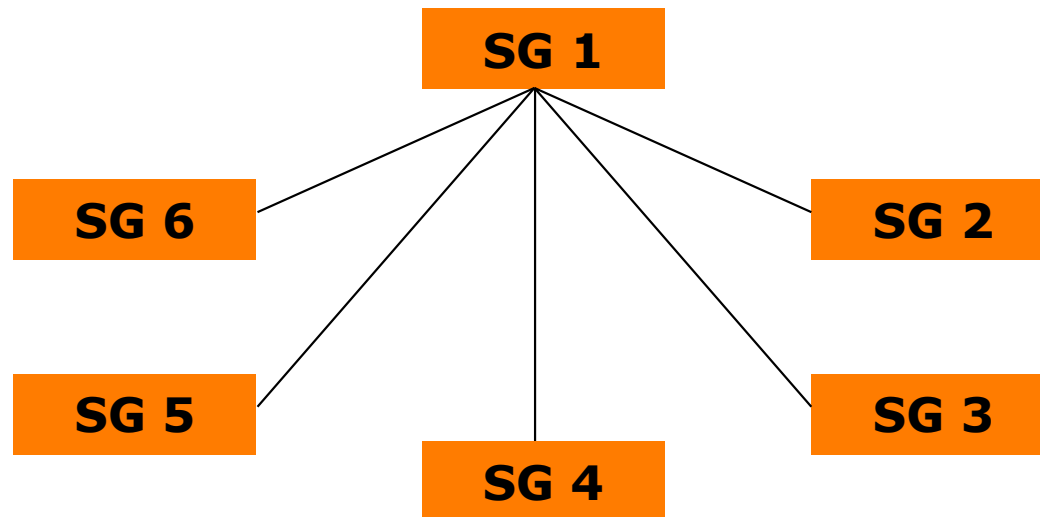
Topologien - Bus

- Beim Bus ist für 6 SGe „eine große Leitung“ und 6 bidirektionale Transceiver notwendig
- Aushandlung der Kommunikation über nichtdeterministische Arbitrierung oder aufwändige und bandbreitenintensive Protokolle
- Ausfall eines SGe beeinträchtigt Kommunikation nicht
- Kostengünstig und (bei niedriger Bandbreite) unkompliziert
 - Einsatz für sicherheitskritische Systeme erfordert spezielles Protokoll
 - Nettobandbreite ungeeignet für datenintensive Dienste



Topologien - Stern

- Im Stern sind für 6 SGe 5 Leitungen und 10 bidirektionale Transceiver notwendig
- Zentrales SG kann effektiv Netzmanagement betreiben (Fehlerbehandlung etc.)
- Mit zentralem SG steht und fällt das System, ansonsten extrem ausfallsicher!
- Protokolle können je nach Anforderung als Point-to-Point oder Bus gefahren werden
 - Je nach Protokoll für alle Anwendungen geeignet
 - Abhängigkeit von zentralem SG erfordert spezielle Maßnahmen: Ausfallsicherheit



Serielle Bussysteme

- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - **2.3 Adressierung und Framing**
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
 - 3.4 MOST
 - 3.5 Ethernet
 - 3.6 Beispiele

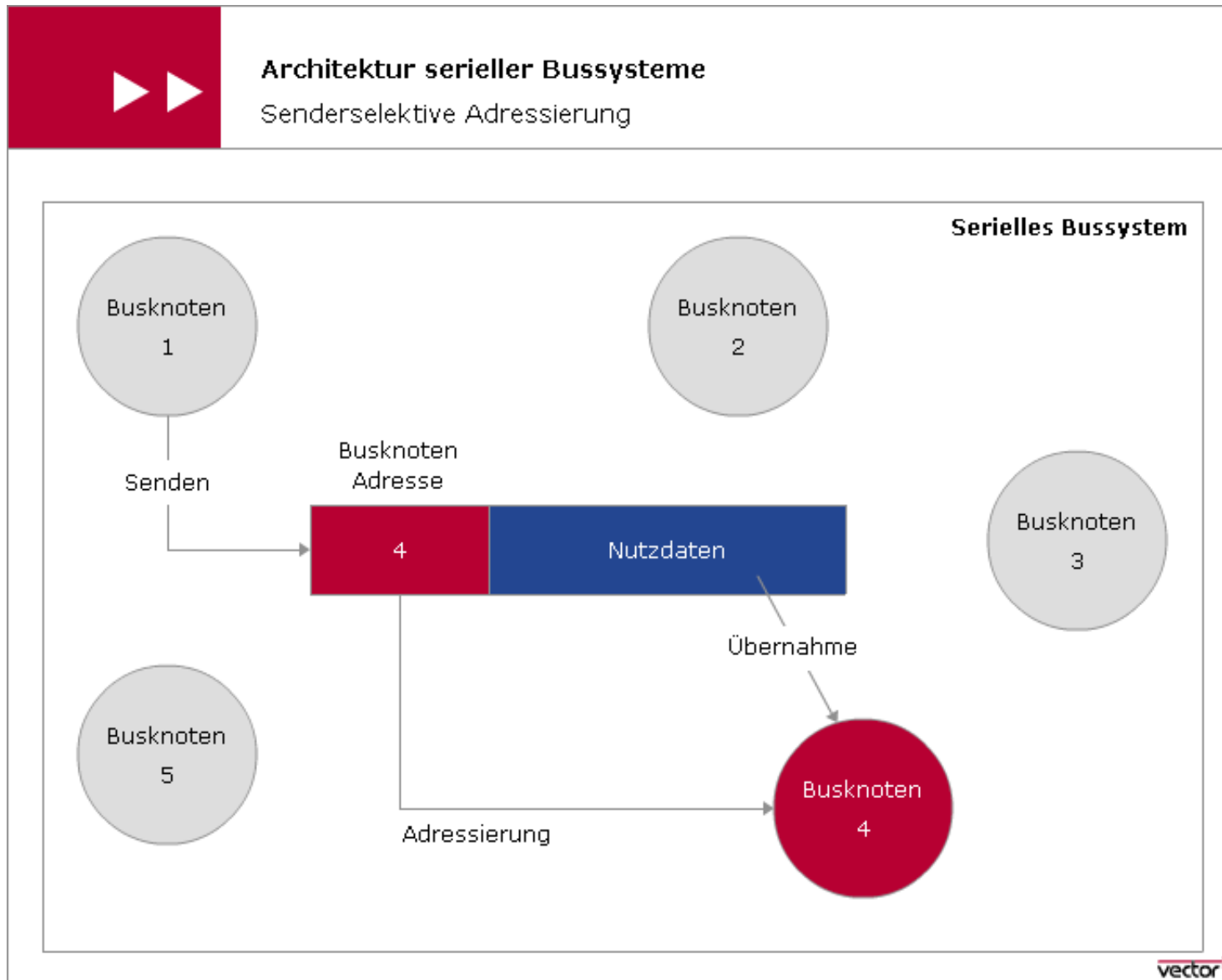
Adressierung (1)

- Für einen reibungslosen Datenaustausch zwischen den Busknoten eines seriellen Bussystems ist die Herstellung einer eindeutigen Zuordnung zwischen Nutzdaten und Busknoten unentbehrlich. Man unterscheidet die senderselektive von der empfängerselektiven Zuordnung bzw. Adressierung.
- Im Falle einer senderselektiven Adressierung bestimmt der Sender den gewünschten Empfänger über eine eindeutige Busknotenadresse. Zum Multi- oder Broadcasting müssen Sonderadressen vereinbart werden. Die senderselektive Adressierung stellt die Voraussetzung für einen verbindungsorientierten Datenaustausch (one-to-one-communication) dar. Die Grafik Senderselektive Adressierung zeigt, wie innerhalb eines seriellen Bussystems mit senderselektiver Adressierung ein Busknoten einen anderen Busknoten direkt über die Busknotenadresse anspricht.



Architektur serieller Bussysteme

Senderselektive Adressierung



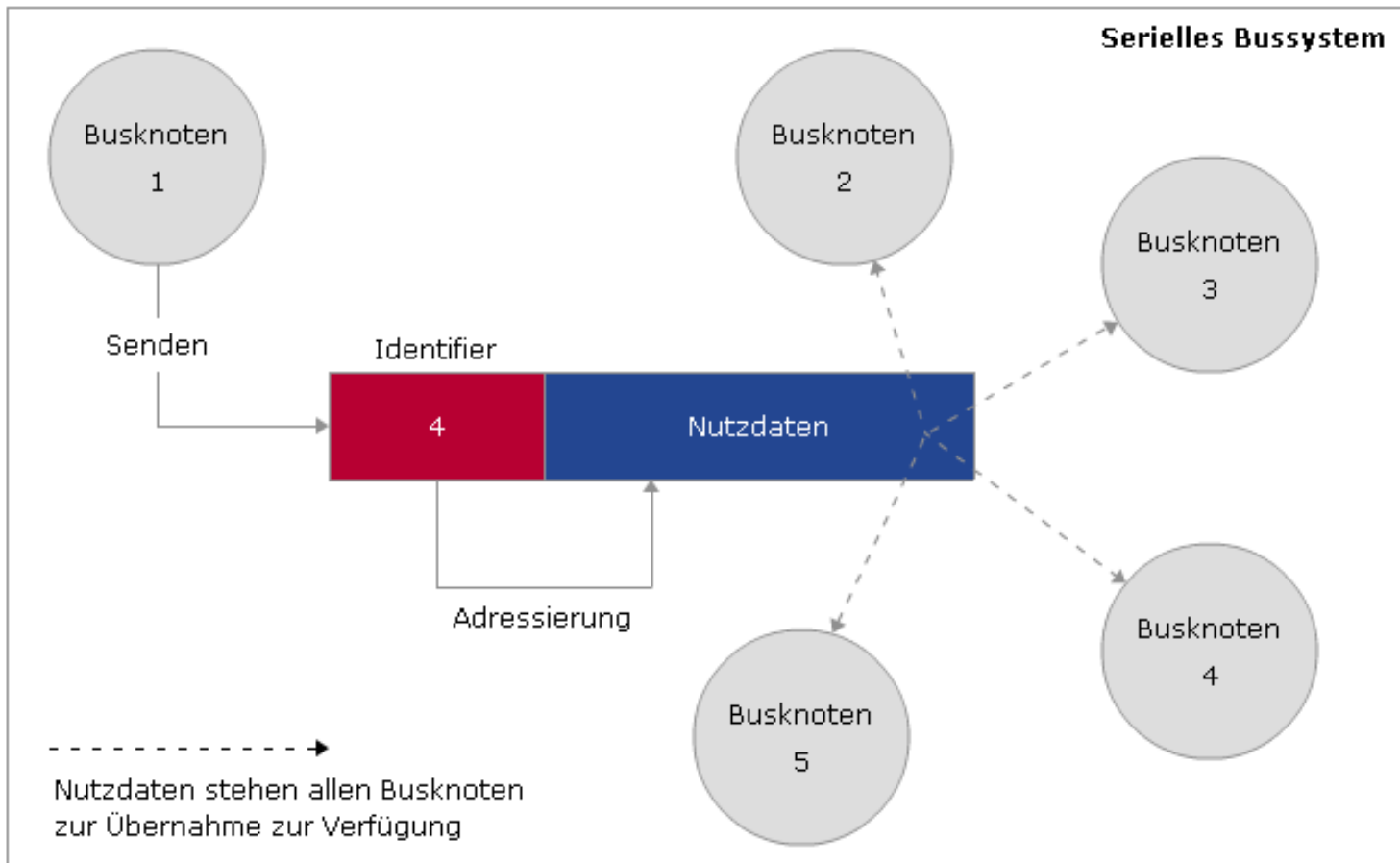
Adressierung (2)

- Im Falle der empfängerselektiven Adressierung werden nicht die Busknoten, sondern die zu versendenden Nutzdaten mit einer Kennung (Identifier) versehen. Dadurch stehen alle Nutzdaten prinzipiell jedem Busknoten zur Übernahme zur Verfügung (Broadcasting). Jeder Busknoten hat die Aufgabe, die für ihn relevanten Nutzdaten aus dem Datenstrom zu filtern. Die empfängerselektive Adressierung zeichnet sich durch eine hohe Konfigurationsflexibilität aus und ermöglicht beispielsweise durch die Mehrfachausnutzung von Sensoren Kosten einzusparen. Zum Einsatz kommt die empfängerselektive Adressierung sowohl bei CAN als auch bei LIN und FlexRay.
- Die Grafik Empfängerselektive Adressierung zeigt, wie innerhalb eines seriellen Bussystems mit empfängerselektiver Adressierung ein Busknoten Nutzdaten allen anderen Busknoten zum Empfang zur Verfügung stellt. Über den Identifier entscheiden die Busknoten über den Empfang: entweder selektieren alle Busknoten die angebotenen Nutzdaten (one-to-all-communication) oder nur ein Teil der Busknoten (one-to-many-communication). Es ist nicht üblich, dass keiner der Busknoten angebotene Nutzdaten selektiert.



Architektur serieller Bussysteme

Empfängerselektive Adressierung



Serielle Bussysteme

- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - **2.5 Datensicherung**
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
 - 3.4 MOST
 - 3.5 Ethernet
 - 3.6 Beispiele

Fehlersicherheit und Codierung - Verfahren

- Parity:
 - XOR aller Bits der Nachricht wird als Paritybit angehängt
 - Bandbreitenbedarf: 1 Bit unabhängig von der Wortlänge
 - Hammingdistanz: 2 -> erkennt nur einfache Fehler
- Checksum:
 - Quersumme aller Bits des Wortes wird angehängt
 - Bandbreitenbedarf: $\log_2(n)$ Bits
 - Empfindlich gegen systematische Fehler (Zahlendreher etc.)
- Zyklische Codes:
 - Verwendet nichtlineare Eigenschaften bestimmter mathematischer Gruppen (Ideale) zur Optimierung des Verhältnisses von Hammingdistanz zu Bandbreitenbedarf
 - Unempfindlich gegen die meisten statistischen und systematischen Fehler

Fehlersicherheit und Codierung - Anforderungen

Anwendung	Verlust von Datenpaketen	Verzögerung
Sprache / Telefon	Unkritisch	Kritisch
email	Kritisch	Unkritisch
Motorsteuerung	Kritisch	Kritisch
Steer by Wire	Sehr kritisch	Sehr kritisch



Architektur serieller Bussysteme

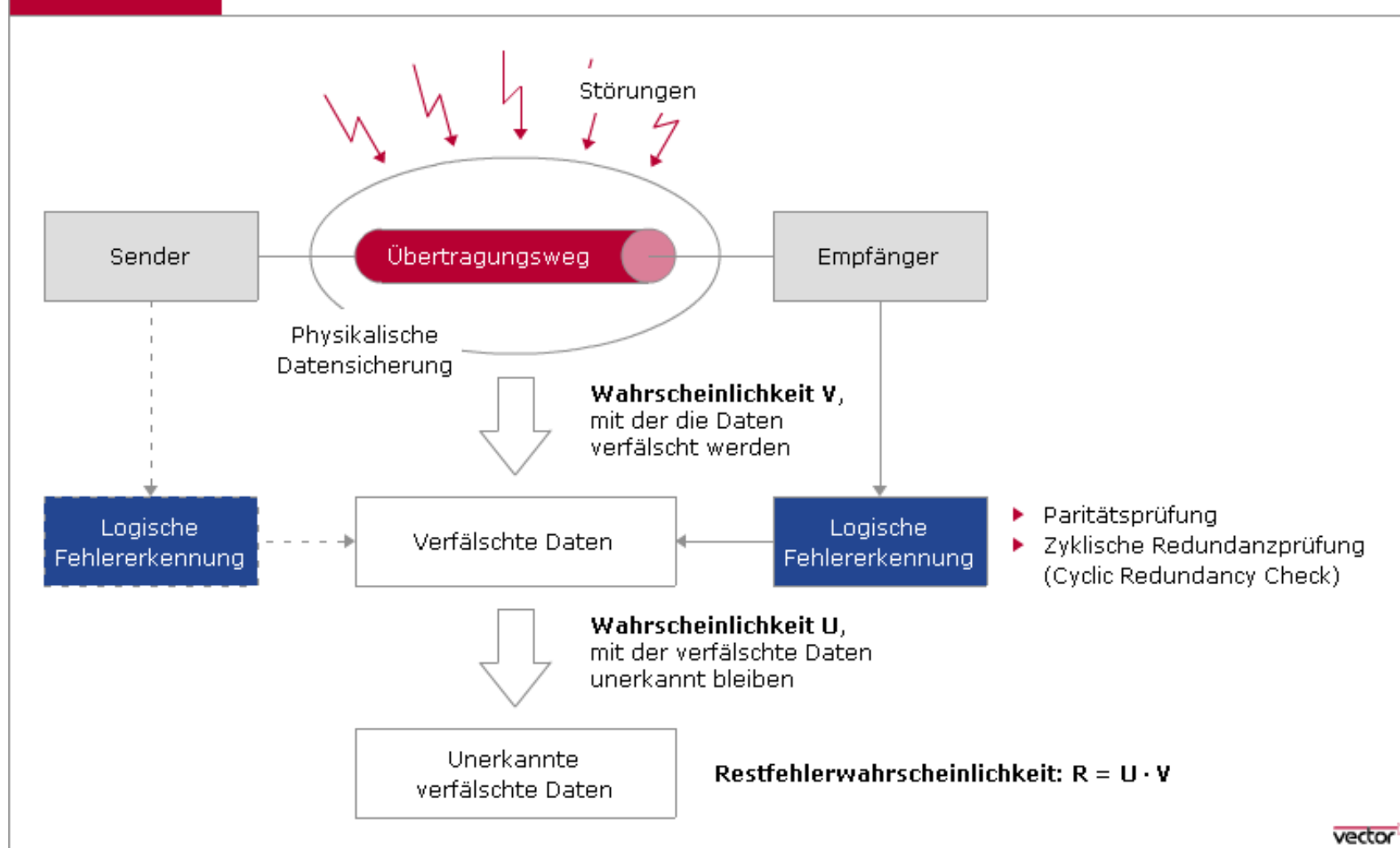
Fehlererkennungsmethoden

Serielle Kommunikationstechnologie	Einsatzgebiet	Sicherheitsanforderungen	Fehlererkennungsmethode
LIN (Local Interconnect Network)	Komfort	eher gering	Paritätsprüfung, Prüfsumme
CAN (Controller Area Network)	Komfort, Antrieb, Fahrwerk	hoch	CRC (Cyclic Redundancy Check)
FlexRay	Fahrwerk, Backbone	hoch - sehr hoch	CRC (Cyclic Redundancy Check)
MOST (Media Oriented System Transport)	Infotainment	gering	Paritätsprüfung, CRC (Cyclic Redundancy Check)



Architektur serieller Bussysteme

Prinzip der Datensicherung



Paritätsprüfung (1)

- Eine einfache Methode, Übertragungsfehler aufzudecken, besteht darin, einen fehlererkennenden Code zu verwenden. Dieser basiert darauf, dass den Nutzdaten zusätzliche, also redundante Bits (Prüfbits) hinzugefügt werden. Durch die Vergrößerung der Datenmenge gibt es dann neben gültigen Codewörtern auch ungültige Codewörter. Je größer die Distanz - auch als Hamming-Distanz bezeichnet - zwischen den gültigen und ungültigen Codewörtern, desto mehr Bitfehler kann der Empfänger erkennen.
- Bei einer Hamming-Distanz von beispielsweise 2 kann der Empfänger einen Bitfehler erkennen. Der Empfänger kann zwei Bitfehler erkennen, wenn die Hamming-Distanz 3 beträgt. Zwar nimmt die Fehlererkennungsfähigkeit mit steigender Hamming-Distanz zu. Gleichzeitig nimmt aber die Übertragungseffizienz ab und die Wahrscheinlichkeit steigt, dass eines der Prüfbits gestört ist. Schließlich muss in Abhängigkeit des Einsatzgebietes ein Kompromiss zwischen Übertragungseffizienz und Fehlererkennungsfähigkeit gefunden werden.

Paritätsprüfung (2)

- Bei der Paritätsprüfung wird einem vergleichsweise kleinen Datenpaket, üblicherweise acht Bits, ein zusätzliches Bit (Paritätsbit) angehängt, so dass die Summe der logischen Einsen gerade (Even Parity) oder ungerade (Odd Parity) ist. Mithilfe der Paritätsprüfung können auf jeden Fall Einzelbitfehler erkannt werden, was einer Hamming-Distanz von zwei entspricht. Zwei Fehler können nicht erkannt werden, da ein gültiges Codewort durch zwei Bitfehler in ein anderes gültiges Codewort überführt wird.
- Mithilfe der Grafik Paritätsprüfung können Sie nachvollziehen, wie die Paritätsprüfung angewendet wird, beispielsweise an einem drei Bit breiten Datenfeld. Möglich sind acht Codewörter. Den einzelnen Codewörtern wird ein Paritätsbit hinzugefügt, so dass die Summe der logischen Einsen gerade ist (Even Parity). Der Umfang an Codewörter verdoppelt sich: bei der einen Hälfte handelt es sich um gültige, bei der anderen Hälfte um ungültige Codewörter.



Architektur serieller Bussysteme

Paritätsprüfung

Datenfeld

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



Gültige Codewörter

0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



Ungültige Codewörter

0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

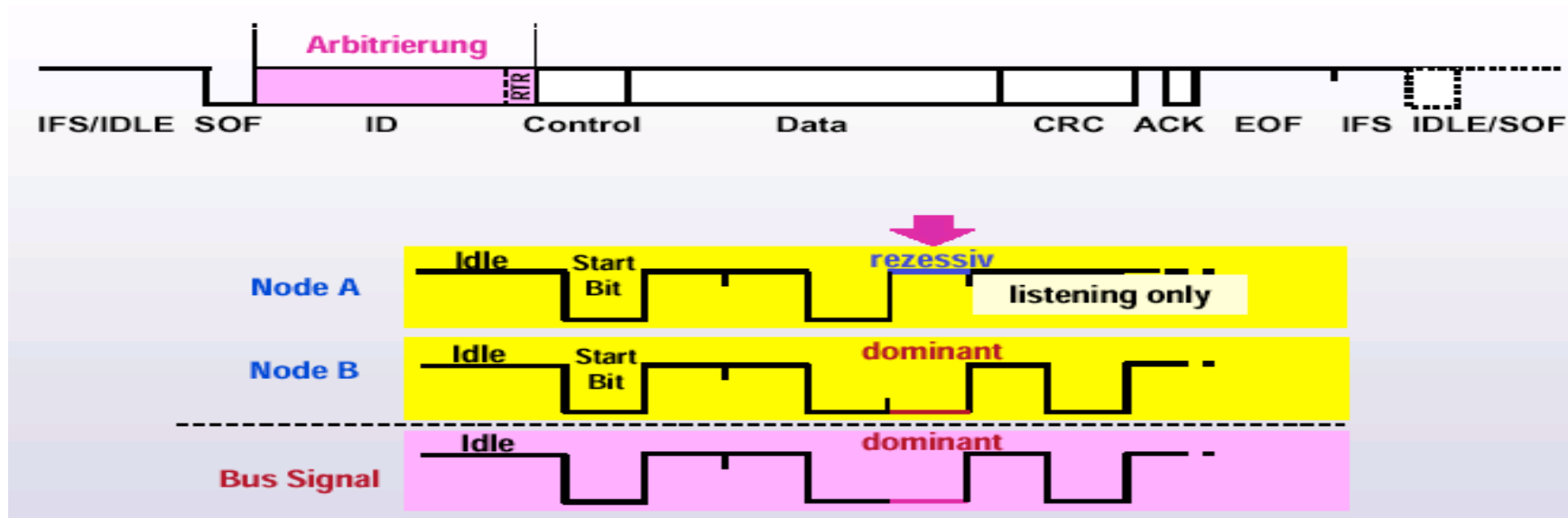
Paritätsbit
(Even Parity)

Serielle Bussysteme

- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - **2.6 Synchronisation**
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
 - 3.4 MOST
 - 3.5 Ethernet
 - 3.6 Beispiele

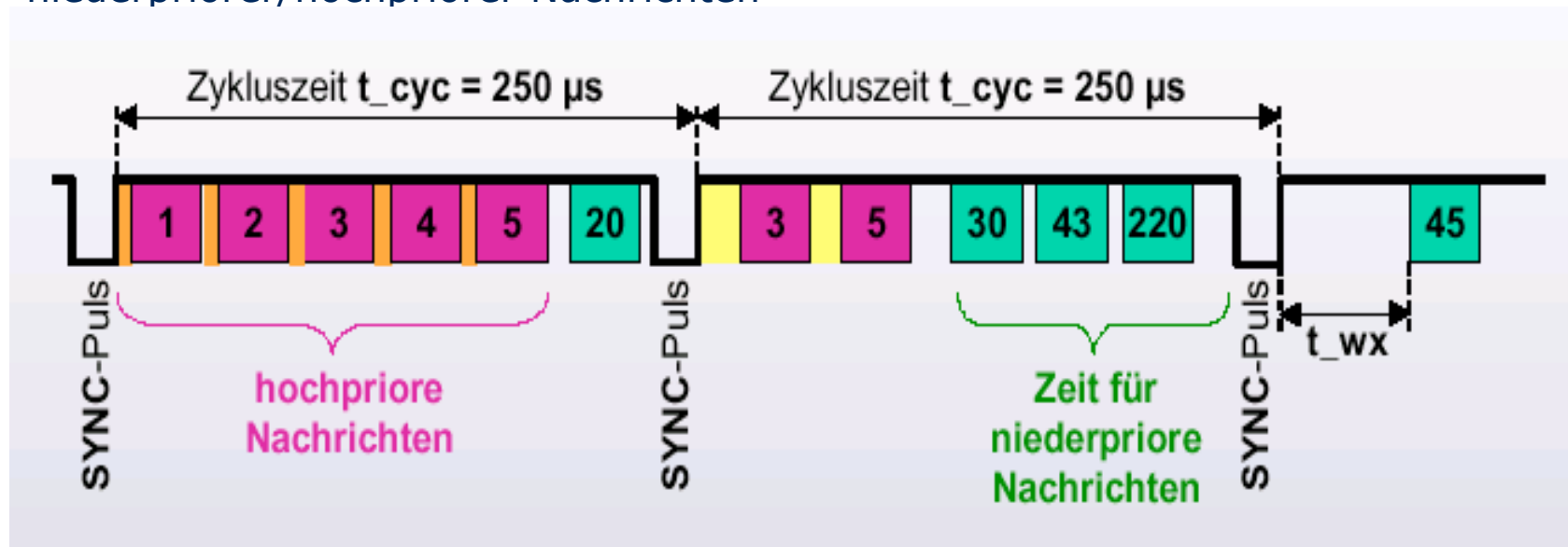
Zeitverhalten - asynchron

- Bei inaktivem Bus (idle) kann jeder Teilnehmer durch Senden des dominanten Start-Bit Kommunikation beginnen
- Beginnen 2 oder mehr Teilnehmer gleichzeitig einen Sendeversuch, wird über das ID-Feld eine Arbitrierung durchgeführt
 - Flexible Bandbreitennutzung
 - Versenden von Nachrichten in bestimmter Zeit kann nicht garantiert werden



Zeitverhalten - synchron

- Ein Busmaster synchronisiert alle Teilnehmer durch Senden eines Synchronisationssignals (SYNC-Puls)
- Innerhalb eines Zyklus gibt es für bestimmte (hochpriore) Nachrichten-IDs festgelegte Zeitfenster, in denen deren Übertragung garantiert wird.
- Freibleibende Zeit kann für asynchrone (niederpriore) Nachrichten genutzt werden
 - Deterministisches Verhalten und flexible Bandbreitennutzung
 - Bedarf für dedizierten Busmaster, inflexible Struktur für die Auslegung niederpriorer/hochpriorer Nachrichten



Asynchrone Übertragung (1)

- Die Grundvoraussetzung für einen reibungslosen Datenaustausch sind auf Bitebene synchronisierte Kommunikationspartner. Die Herstellung der Synchronisation auf Bitebene kann entweder asynchron oder synchron erfolgen. Deswegen unterscheidet man auch zwischen der asynchronen und der synchronen Datenübertragung.
- In einem seriellen Bussystem mit asynchroner Datenübertragung herrscht zwischen Kommunikationspartnern nur während der Übertragung weniger Bits Synchronität. Hergestellt wird die Synchronität durch ein Startzeichen. Ein Stoppzeichen signalisiert das Ende der Datenübertragung. Im Rahmen der Übertragung längerer Datenblöcke muss sich der Empfänger immer wieder aufs Neue mit dem Sender synchronisieren (Start-Stopp-Verfahren).
- In der Praxis verwendet man üblicherweise das UART-Format (Universal Asynchronous Receiver Transmitter): acht Nutzbits sind von einem Startbit (logische Null) und zwei Stoppbits (logische Eins) eingerahmt. Sichert werden muss lediglich, dass der Empfänger bis zum Ende der Übertragung eines UART-Zeichens den Sendetakt halten kann, um sämtliche Bits korrekt abtasten zu können.

Asynchrone Übertragung (2)

- Da dies mit vergleichsweise einfachen elektronischen Schaltungen und Taktgeneratoren möglich ist, eröffnet die asynchrone Datenübertragung das Potenzial, auch in Bereichen mit hoher Kostensensibilität seriellen Datenaustausch zu realisieren. So geschehen im Kfz zur Übertragung sicherheitsunkritischer Signale im Komfortbereich. Dort kommt heute das auf der asynchronen Datenübertragung basierende serielle Bussystem LIN zum Einsatz.



Architektur serieller Bussysteme

UART-Format



UART: Universal Asynchronous Receiver Transmitter

LSB: Least Significant Bit

MSB: Most Significant Bit

Synchrone Übertragung (1)

- Im Gegensatz zur asynchronen Datenübertragung arbeiten die Kommunikationspartner bei der synchronen Datenübertragung mit Taktgebern derselben Taktfrequenz und sind für die gesamte Dauer einer Datenübertragung synchron (Gleichlauf). Weil in der Praxis trotzdem kleine Frequenzabweichungen auftreten können, muss den Empfängern während der Datenübertragung Nachsynchronisation ermöglicht werden.
- Während man dies bei zeichenorientierter Datenübertragung mithilfe besonderer Zeichen (Sync-Zeichen) bewerkstelligt, erreicht man es bei bitorientierter Datenübertragung durch die Codierung des Bitstroms. Das Ziel besteht darin, den Bitstrom mit zusätzlichen Flanken auszustatten.
- Entweder erfolgt dies durch die Codierung der beiden logischen Pegel „0“ und „1“, dazu verwendet man sog. selbsttaktende Codes, wie beispielsweise den Manchester Code, oder durch die Definition eines zusätzlichen Codeelements wie dies beispielsweise bei FlexRay der Fall ist: jedem Nutzbyte ist das Codeelement BSS (Byte Start Sequence) vorangestellt. Es setzt sich aus einer logischen Eins und einer logischen Null zusammen und erzeugt so stets einen Flankenwechsel.

Synchrone Übertragung (2)

- Eine andere Methode, Flankenwechsel zu erzeugen, besteht darin, die Sender zu zwingen, nach einer definierten Anzahl homogener Bits jeweils ein komplementäres Bit zu übertragen. Dieses Bit bezeichnet man als Stopfbit (Stuffbit) und die Methode als Bitstuffing. Eingesetzt wird diese Methode beispielsweise bei CAN: nach spätestens fünf homogenen Bits überträgt ein Sender immer ein komplementäres Bit.



Architektur serieller Bussysteme

Manchester Codierung

Logischer Bitpegel



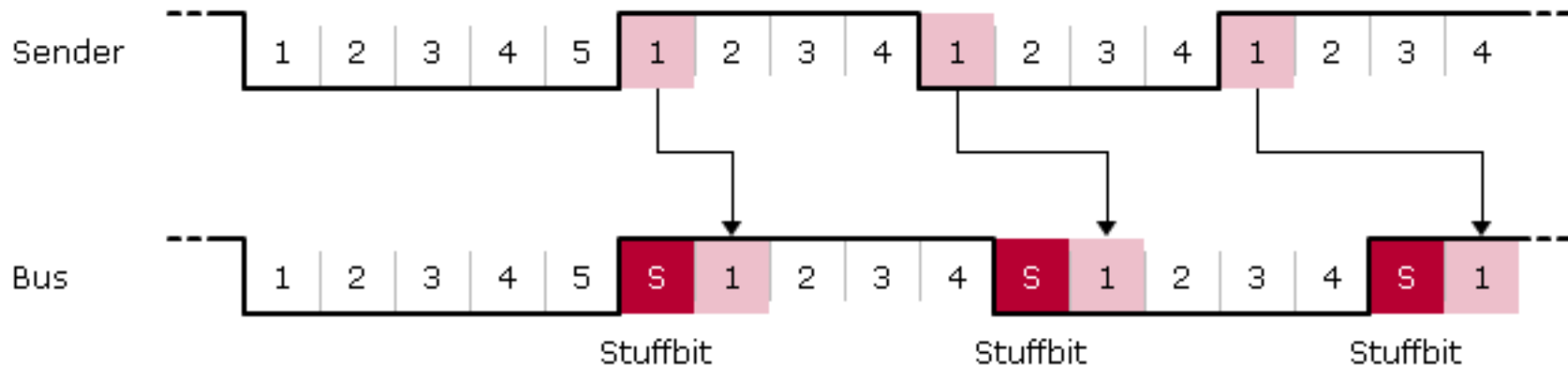
Manchester Codierung





Architektur serieller Bussysteme

Bitstuffing



Serielle Bussysteme

- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - **2.7 Physikalische Übertragung**
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
 - 3.4 MOST
 - 3.5 Ethernet
 - 3.6 Beispiele

Übertragungsmedien (1)

- Unentbehrlich für die Übertragung von Signalen ist eine physikalische Übertragungsstrecke. Diese kann entweder leitungsgebunden oder –ungebunden realisiert sein. Im Kontext der seriellen Datenübertragung zwischen elektronischen Steuergeräten im Kfz hat man es hauptsächlich mit leitungsgebundenen Übertragungsstrecken in Form von elektrischen Leitungen zu tun.
- Zwar dämpfen und verzerren elektrische Leitungen Signale auf ihrem Weg vom Sender zum Empfänger. Jedoch sind diese Phänomene bei den im Kfz üblichen geringen Netzausdehnungen und Datenraten handhabbar und beeinflussen kaum die Signalintegrität. Allerdings sind elektrische Leitungen sehr stöempfindlich: trotz elektromagnetisch verträglichem Design können galvanische, kapazitive und induktive Kopplungen nicht ausgeschlossen werden. Zudem trägt die Übertragung digitaler Signale aufgrund von Emissionen zur elektromagnetischen Unverträglichkeit bei, dies umso mehr je steiler die Flanken der Bits sind.

Übertragungsmedien (2)

- In Abhängigkeit der Anforderungen an die Übertragungssicherheit und Übertragungskosten kommen zur Verbesserung der Störfestigkeit und zur Reduzierung der Emissionen unterschiedliche Methoden zum Einsatz. Dazu zählen die Abschirmung der elektrischen Leitung und die symmetrische Signalübertragung. Wegen den höheren Kosten verzichtet man üblicherweise auf die Abschirmung, und oft auch auf die symmetrische Signalübertragung. Letzteres dort wo die Anforderungen an die Übertragungssicherheit nicht allzu hoch sind, wie beispielsweise im Komfortbereich.
- In sicherheitskritischen Einsatzgebieten hingegen, wie zum Beispiel der Antrieb und das Fahrwerk, kommt man wegen der notwendigen Übertragungssicherheit nicht ohne symmetrische Signalübertragung aus. Deshalb erfolgt sowohl in CAN- als auch in FlexRay-Netzwerken die serielle Datenkommunikation symmetrisch unter Einsatz verdrahteter Zweidrahtleitungen.

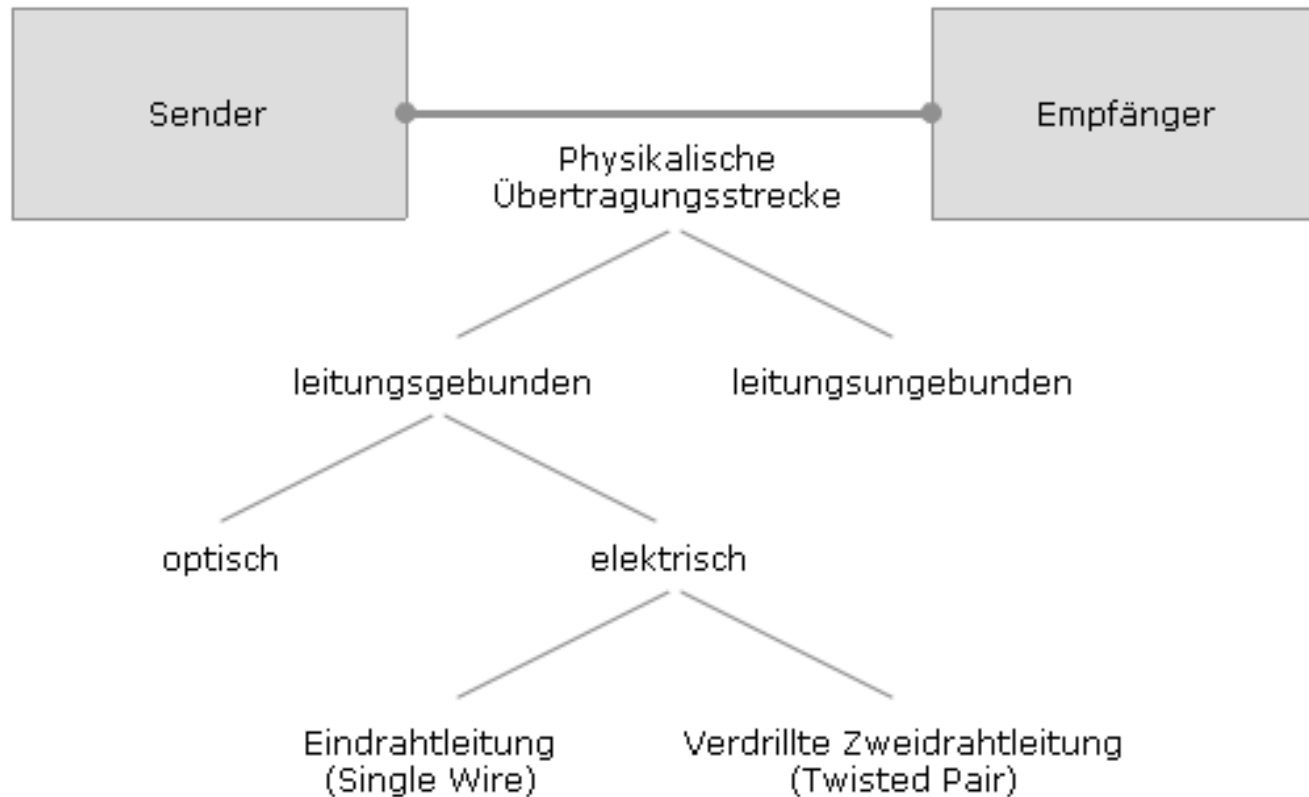
Übertragungsmedien (3)

- Optische Medien zeichnen sich zwar im Vergleich mit verdrehten Zweidrahtleitungen durch eine deutlich bessere Störfestigkeit und wesentlich geringere Emissionen aus. Allerdings verhindert der Kostenaufwand und auch die schlechtere Handhabbarkeit den breiten Einsatz im Kfz. Derzeit spielen optische Medien lediglich im Infotainmentbereich, vor allem in Oberklassenmodellen, eine Rolle.
- Die MOST-Technologie (Media Oriented System Transport), die im Infotainment in der Variante MOST 50 zur seriellen Übertragung von Audio- und Videosignalen zum Einsatz kommt, basiert auf einer polymeroptischen Faser (POF) und bietet eine Bandbreite von rund 50 MBit/s. Verläuft die Elektronifizierung des Kfz weiterhin so dynamisch, werden optische Übertragungsmedien auch für sicherheitskritische Einsatzgebiete zunehmend interessant. Dies nicht nur wegen der Gewährleistung deutlich höherer Datenraten, sondern auch aufgrund einer wesentlich besseren elektromagnetischen Verträglichkeit.



Architektur serieller Bussysteme

Physikalische Übertragungsstrecke



Asymmetrische Übertragung (1)

- Die kostengünstigste Art, elektrische Signale zwischen elektronischen Steuergeräten im Kfz zu übertragen, ist die asymmetrische Übertragung. Zur Signalübertragung ist lediglich eine Signalleitung erforderlich, an die die Kommunikationsteilnehmer passiv angekoppelt werden. Man unterscheidet zwischen High-Pegel (logische Eins) und Low-Pegel (logische Null). Der Low-Pegel entspricht üblicherweise der Masse.
- Allerdings steht dem Vorteil, nur eine Leitung und eine Signalquelle zu benötigen, der Nachteil einer vergleichsweise geringen Störfestigkeit gegenüber. Zudem kann sich die Signalübertragung über ihr generiertes elektromagnetisches Feld störend auf benachbarte elektrische Systeme oder Leitungen auswirken. Die Emission wird bei höheren Frequenzen groß, weil die Signalleitung als Antenne wirkt.
- Offensichtlich kann die asymmetrische Signalübertragung im Kfz nicht ohne Maßnahmen zur Verbesserung der elektromagnetischen Verträglichkeit eingesetzt werden. Man muss einerseits die Emissionen elektromagnetischer Störfelder verringern und andererseits den Störabstand erhöhen, darf jedoch dabei die Kosten nicht erhöhen.

Asymmetrische Übertragung (2)

- Gelöst wird dieses Dilemma im Kfz durch die LIN-Technologie (Local Interconnect Network). Diese serielle Kommunikationstechnologie wurde speziell dafür entwickelt, die Vorteile der seriellen Datenkommunikation auch für jene Einsatzbereiche im Kfz zu nutzen, die erheblichem Kostendruck ausgesetzt sind. Darunter fällt vor allem der Komfortbereich. Dieser zeichnet sich durch eine rasant steigende Anzahl elektronischer Funktionen aus, deren Realisierung ohne den regen Datenaustausch zwischen elektronischen Steuergeräten, Sensoren und Aktoren nicht möglich wäre.
- Die LIN-Technologie verzichtet zur Herstellung einer höheren Störfestigkeit und zur Reduzierung der Emission elektromagnetischer Felder auf eine teure Abschirmung. Einen ausreichenden Störabstand bekommt man durch vergleichsweise weit auseinander liegende Signalpegel (Masse und Batteriespannung). Der Emission elektromagnetischer Felder begegnet die LIN-Technologie mit einer niedrigen Datenrate (maximal 20 KBit/s) und einem dadurch weniger steilen Anstieg der Signalflanken.

Asymmetrische Übertragung (3)

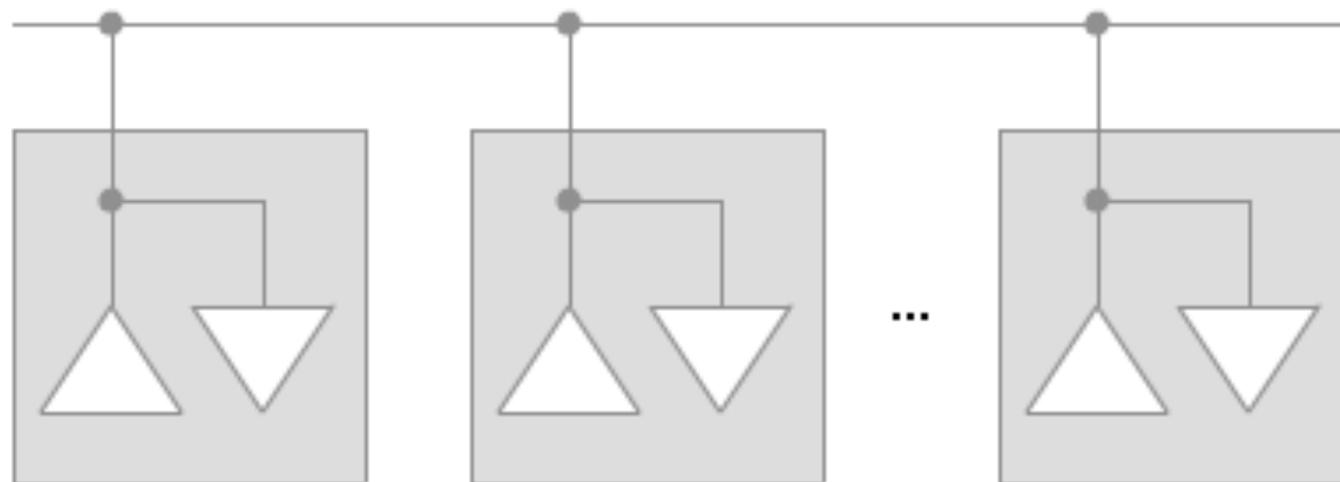
- Anhand der Grafik LIN-Signalspezifikationen können die Signalspezifikationen für Sender und Empfänger in einem LIN-Netzwerk studiert werden. Ein Sender im LIN-Netzwerk muss mindestens 80% der Batteriespannung für einen High-Pegel treiben. Will der Sender ein Low-Pegel übertragen, darf der Spannungspegel 20% der Batteriespannung nicht überschreiten. Ein Pegel unterhalb von 40% der Batteriespannung wird vom Empfänger als einen Low-Pegel. Als High-Pegel interpretieren Empfänger Pegel oberhalb von 60% der Batteriespannung.



Architektur serieller Bussysteme

Asymmetrische Signalübertragung

Signalleitung



Sender/
Empfänger

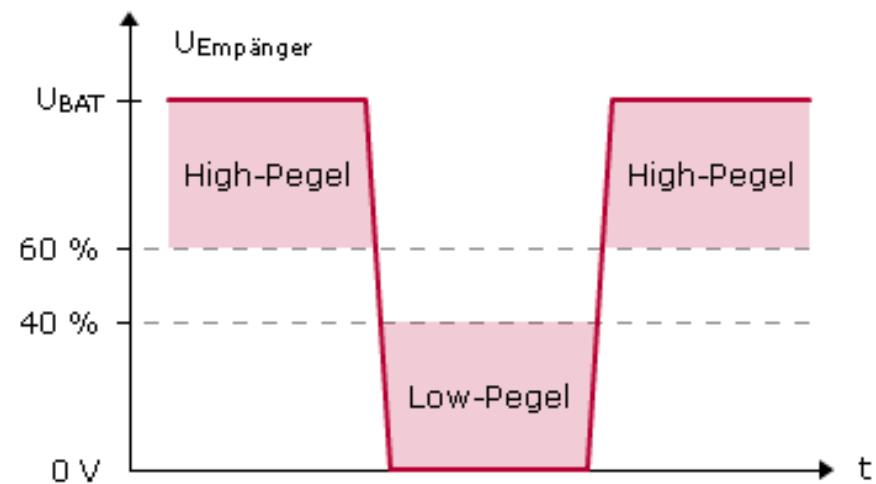
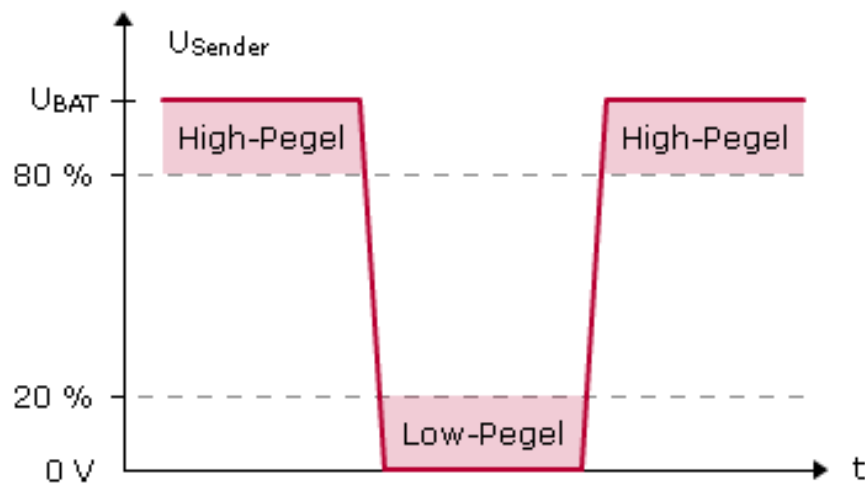
Sender/
Empfänger

Sender/
Empfänger



Architektur serieller Bussysteme

LIN-Signalspezifikation



Symmetrische Übertragung (1)

- Bei der symmetrischen Signalübertragung werden die Signale als Spannungsdifferenzen über zwei Leitungen übertragen. Das Leitungspaar setzt sich dabei aus einer nicht invertierten und einer invertierten Signalleitung zusammen. Aus der Differenz dieser beiden Signale rekonstruieren die Empfänger das ursprüngliche Datensignal. Dies hat den Vorteil, dass sich Gleichtaktstörungen nicht auf die Übertragung auswirken.
- Um induktiven Kopplungen und somit dem Problem des Nebensprechens begegnen zu können, werden die beiden Signalleitungen verdreht. Dadurch wird die für die induktive Kopplung wirksame Fläche verkleinert und im Idealfall zeigen die einzelnen magnetischen Felder in jedem Teilstück in entgegen gesetzte Richtungen, was dazu führt, dass sich die induktiven Einflüsse gegenseitig aufheben.
- In der Konsequenz stellt die symmetrische Signalübertragung mittels Twisted-Pair eine hohe Störfestigkeit sicher, und dies bei vergleichsweise geringen Spannungsdifferenzen bzw. geringem Leistungsbedarf. Zudem zeichnet sich ein Twisted-Pair durch eine deutlich herabgesetzte Emission aus.

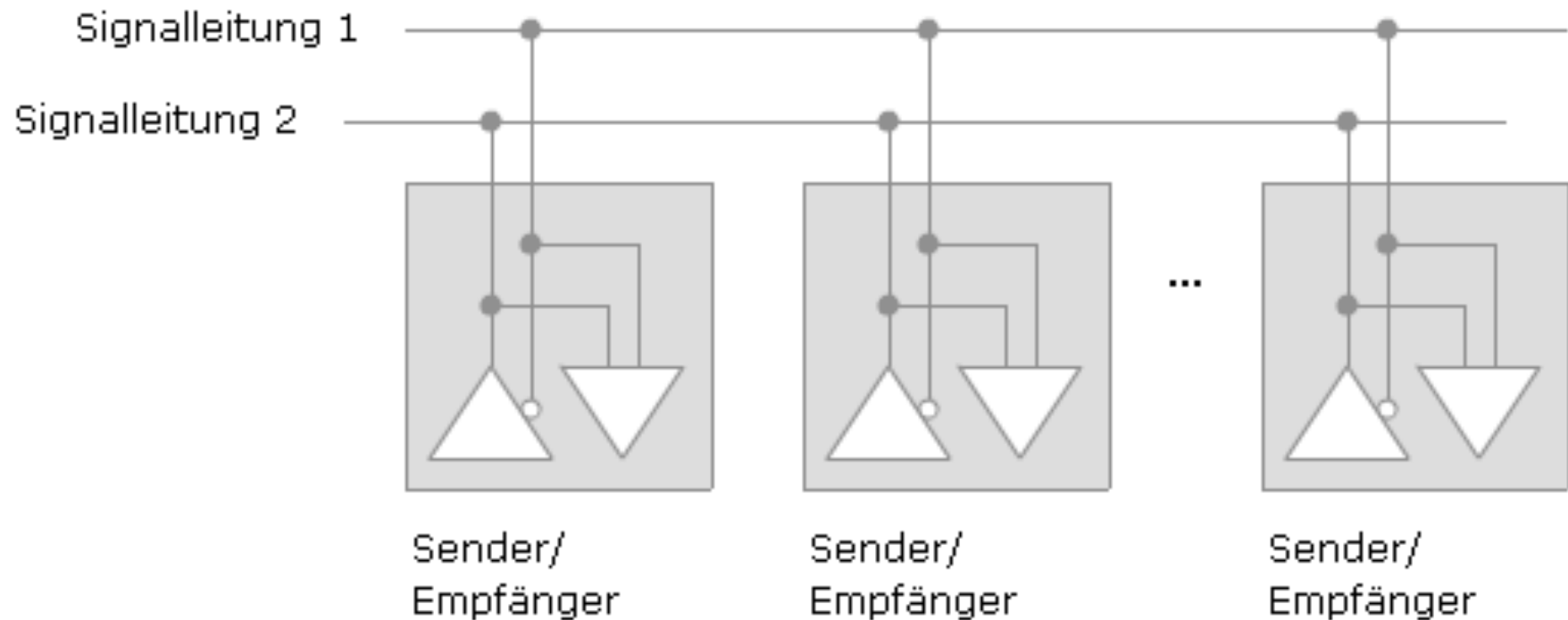
Symmetrische Übertragung (2)

- Diese Vorteile begründen den weit verbreiteten Einsatz der symmetrischen Signalübertragung unter Verwendung einer verdrehten Zweidrahtleitung im Kfz. Sowohl der CAN-Technologie als auch der FlexRay-Technologie liegt eine solche physikalische Signalübertragung zugrunde, um den hohen Anforderungen an die Übertragungssicherheit entsprechen zu können.
- Mithilfe der Grafik CAN-Signalspezifikationen (High-Speed) können die Signalspezifikationen für Sender und Empfänger in einem CAN-High-Speed-Netzwerk (Datenraten bis 1 MBit/s) studiert werden. Ein High-Pegel (logische Eins) entspricht einer Differenzspannung von Null Volt. Eine Differenzspannung von 2 Volt entspricht dem Low-Pegel (logische Null). Empfänger werten eine Differenzspannung von mehr als 0,9 Volt als Low-Pegel. Unterhalb von 0,5 Volt deuten die Empfänger die Differenzspannung als High-Pegel.



Architektur serieller Bussysteme

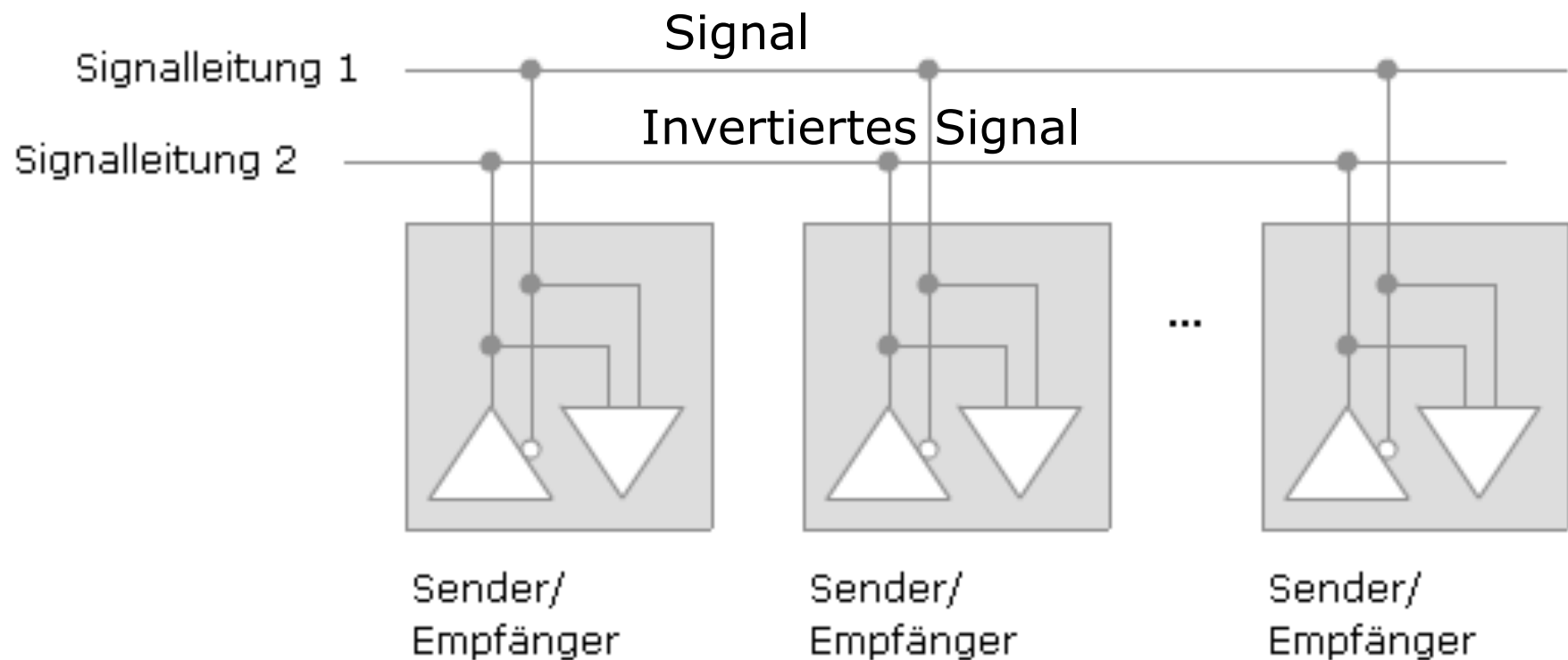
Symmetrische Signalübertragung





Architektur serieller Bussysteme

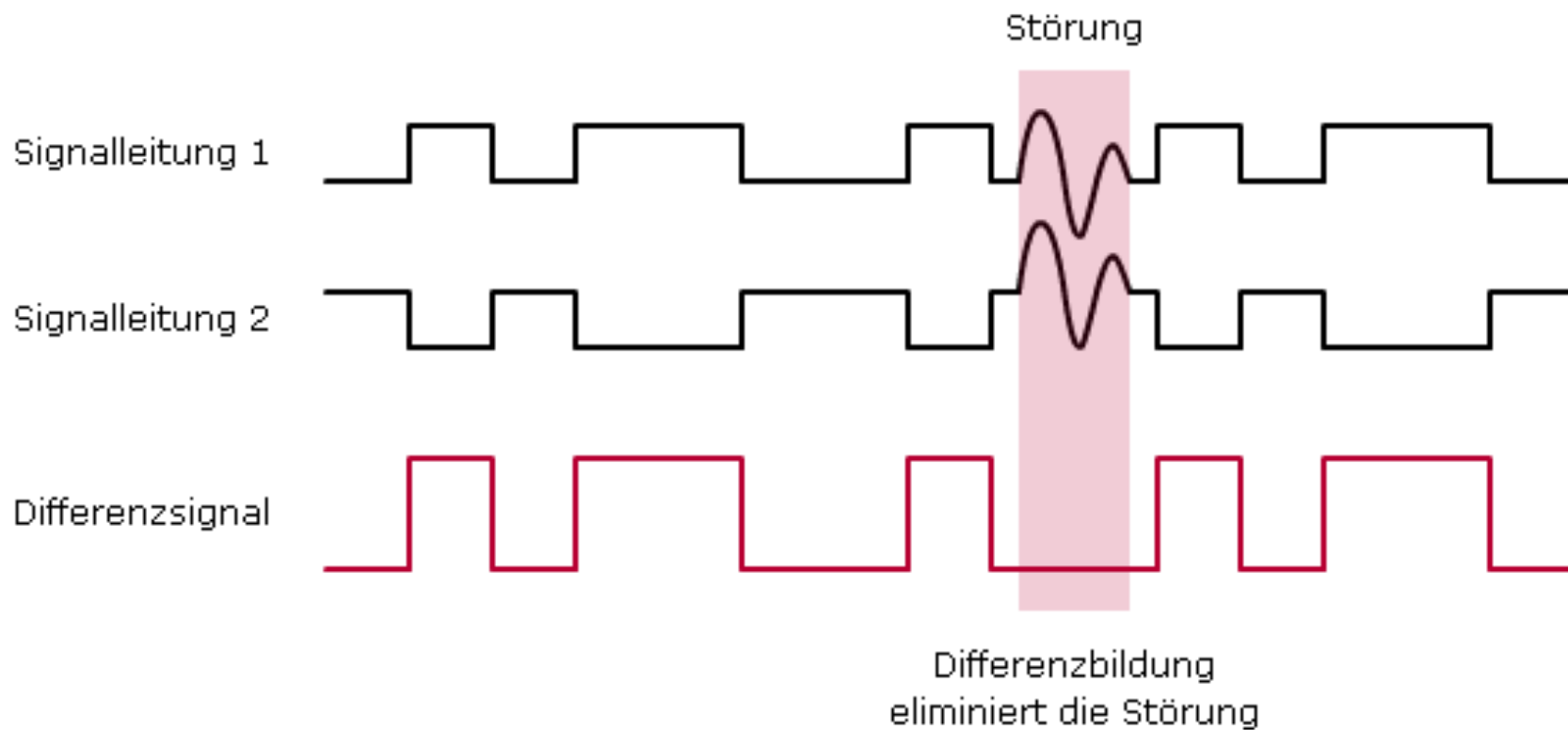
Symmetrische Signalübertragung





Architektur serieller Bussysteme

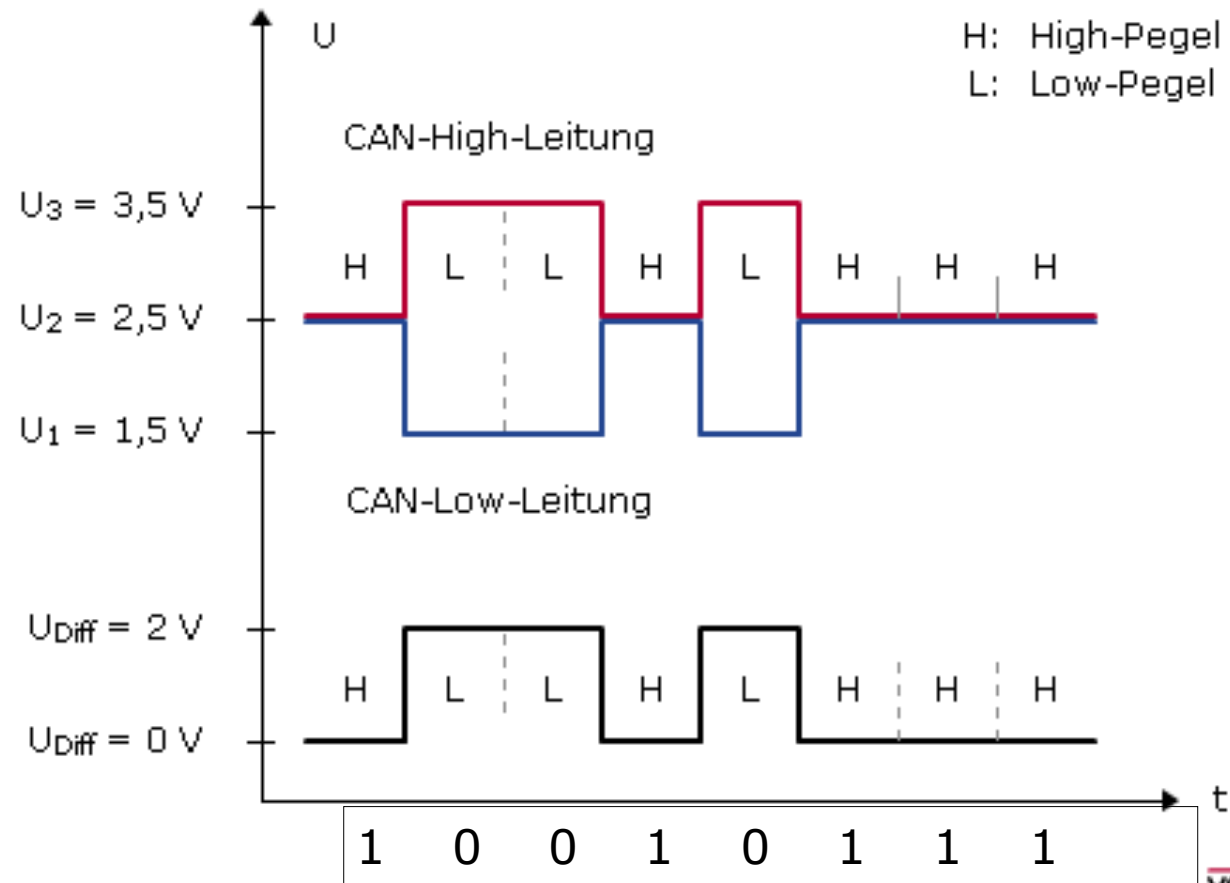
Differenzbildung





Architektur serieller Bussysteme

CAN-Signalspezifikation (High-Speed)



> 0,9 Volt

< 0,5 Volt



Architektur serieller Bussysteme

Fehlererkennungsmethoden

Serielle Kommunikationstechnologie	Einsatzgebiet	Sicherheitsanforderungen	Fehlererkennungsmethode
LIN (Local Interconnect Network)	Komfort	eher gering	Paritätsprüfung, Prüfsumme
CAN (Controller Area Network)	Komfort, Antrieb, Fahrwerk	hoch	CRC (Cyclic Redundancy Check)
FlexRay	Fahrwerk, Backbone	hoch - sehr hoch	CRC (Cyclic Redundancy Check)
MOST (Media Oriented System Transport)	Infotainment	gering	Paritätsprüfung, CRC (Cyclic Redundancy Check)



Architektur serieller Bussysteme

Übertragungsmedien serieller Bussysteme

Serielle Kommunikationstechnologie	Einsatzgebiet	Sicherheitsanforderungen	Bandbreitenanforderung	Übertragungsmedium	Maximale Datenrate
LIN (Local Interconnect Network)	Komfort	eher gering	gering	Eindrahtleitung ohne Schirmung	20 KBit/s
CAN (Controller Area Network)	Komfort, Antrieb, Fahrwerk	hoch	hoch	Verdrillte Zweidrahtleitung ohne Schirmung	CAN-Low-Speed: 125 KBit/s CAN-High-Speed: 1 MBit/s
FlexRay	Fahrwerk, Backbone	hoch - sehr hoch	hoch	Verdrillte Zweidrahtleitung ohne Schirmung	2 Kanäle, pro Kanal: 10 MBit/s
MOST 50 (Media Oriented System Transport)	Infotainment	gering	sehr hoch	POF (Polymer Optische Faser)	117 Streaming-Kanäle à 384 KBit/s

Klassifikation von Bussystemen

Quelle: BOSCH: Kraftfahrtechnisches Taschenbuch, Vieweg+Teubner, 28. Auflage, 2014.

Klasse	Übertragungsraten	Anwendung	Vertreter
Klasse A	Geringe Datenraten (bis 10 kBit/s)	Vernetzung von Aktoren und Sensoren	LIN, PSI5
Klasse B	Mittlere Datenraten (bis 125 kBit/s)	Komplexe Mechanismen zur Fehlerbehandlung, Vernetzung von Steuergeräten im Komfortbereich	Lowspeed-CAN
Klasse C	Hohe Datenraten (bis 1 MBit/s)	Echtzeitanforderungen, Vernetzung von Steuergeräten im Antriebs- und Fahrwerksbereich	Highspeed-CAN
Klasse C+	Sehr hohe Datenraten (bis 10 MBit/s)	Echtzeitanforderungen, (Sicherheitsanforderungen,) Vernetzung von Steuergeräten im Antriebs- und Fahrwerksbereich	FlexRay
Klasse D	Sehr hohe Datenraten (ab 10 MBit/s)	Vernetzung von Steuergeräten im Telematik- und Multimediabereich	MOST, Ethernet

Klassifikation von Bussystemen - Ergänzungen (gelb)

Klasse	Übertragungsraten	Anwendung	Vertreter
Klasse A	Geringe Datenraten (bis 10 kBit/s)	Vernetzung von Aktoren und Sensoren	LIN, PSI5
Klasse B	Mittlere Datenraten (bis 125 kBit/s)	Komplexe Mechanismen zur Fehlerbehandlung, Vernetzung von Steuergeräten im Komfortbereich	Lowspeed-CAN
Klasse C	Hohe Datenraten (bis 1 MBit/s)	Echtzeitanforderungen, Vernetzung von Steuergeräten im Antriebs- und Fahrwerksbereich	Highspeed-CAN
	bis zu 15 MBit/s	Wie Klasse C, zusätzlich - Mehr ECUs / CAN - Schnellere Kommunikation über lange CAN	CAN FD (Flexible Data Rate)
Klasse C+	Sehr hohe Datenraten (bis 10 MBit/s)	Echtzeitanforderungen, (Sicherheitsanforderungen,) Vernetzung von Steuergeräten im Antriebs- und Fahrwerksbereich	FlexRay
Klasse D	Sehr hohe Datenraten (ab 10 MBit/s)	Vernetzung von Steuergeräten im Telematik- und Multimediabereich, MOST verliert an Bedeutung	MOST, Ethernet


Vergleich: Bussysteme im Kfz und Funknetze

Übertragungs- rate	Übertragungs- sicherheit	Übertragungs- strecke	Bussysteme	Funknetze
Gering	Gering	Kurz		
Mittel - hoch	Hoch	Mittel		



Vergleich: Bussysteme im Kfz und Funknetze

Übertragungs- rate	Übertragungs- sicherheit	Übertragungs- strecke	Bussysteme	Funknetze
Gering	Gering	Kurz		
Mittel - hoch	Hoch	Mittel		



Vergleich: Bussysteme im Kfz und Funknetze

Übertragungs- rate	Übertragungs- sicherheit	Übertragungs- strecke	Bussysteme	Funknetze
Gering	Gering	Kurz		
Mittel - hoch	Hoch	Mittel		

Vergleich: Bussysteme im Kfz und Funknetze

Übertragungs- rate	Übertragungs- sicherheit	Übertragungs- strecke	Bussysteme	Funknetze
Gering	Gering	Kurz		
Mittel - hoch	Hoch	Mittel		

Vergleich: Bussysteme im Kfz und Funknetze

Übertragungs- rate	Übertragungs- sicherheit	Übertragungs- strecke	Bussysteme	Funknetze
Gering	Gering	Kurz		
Mittel - hoch	Hoch	Mittel		

Bussysteme im KFZ - Überblick

Quelle: Vector Informatik GmbH

- Weitere Informationen: Overview of Current Automotive Protocols (Vector Informatik GmbH)

Bussystem	Beschreibung	Anwendungsgebiet
CAN	CAN (Controller Area Network) wurde von der Robert Bosch GmbH Anfang der achtziger Jahre entwickelt und 1994 international genormt (ISO 11898). CAN wurde speziell für den schnellen seriellen Datenaustausch zwischen elektronischen Steuergeräten in Kraftfahrzeugen entwickelt. Daneben wird CAN auch für die Realisierung industrieller Mikrocontroller-Netzwerke eingesetzt.	Kfz-Technik, Automatisierungstechnik
LIN	LIN (Local Interconnect Network) wurde speziell für die kostengünstige Kommunikation intelligenter Sensoren und Aktuatoren in Kraftfahrzeugen entwickelt. Charakteristisch für LIN-Bussysteme ist: <ul style="list-style-type: none"> > Master/Slave-Architektur > zeitgesteuerte Datenübertragung > Single-Wire-Datenübertragung mit max. 20kBaude > im Protokoll eingebauter Synchronisationsmechanismus (keine teuren Quarze nötig) 	Kfz-Technik (Innenraumbus, z.B. Vernetzung innerhalb eines Sitzes)
MOST	MOST (Media Oriented Systems Transport) wurde speziell für die Übertragung von Multimediadaten im Kraftfahrzeug entwickelt. Charakteristisch für MOST Bussysteme ist: <ul style="list-style-type: none"> > optische Datenübertragung bis zu 25Mbit/s > Ringstruktur des Busses > Verwendung des genormten XML-Funktionskatalog > Plug&Play-Fähigkeit 	Kfz-Technik (Multimedia-Anwendungen; Vernetzung von Infotainmentgeräten wie Tuner, DVD-Wechsler, etc.)
FlexRay	FlexRay ist als Bussystem für alle sicherheitskritischen Anwendungen sowie zur Übertragung großer Datenmengen im Kraftfahrzeug konzipiert. Charakteristisch für FlexRay-Bussysteme ist: <ul style="list-style-type: none"> > Datenübertragung bis zu 10Mbit/s > redundante Ausführung aller Netzteile 	Kfz-Technik (z.B. sicherheitsrelevante Anwendungen, Brake-by-Wire)