

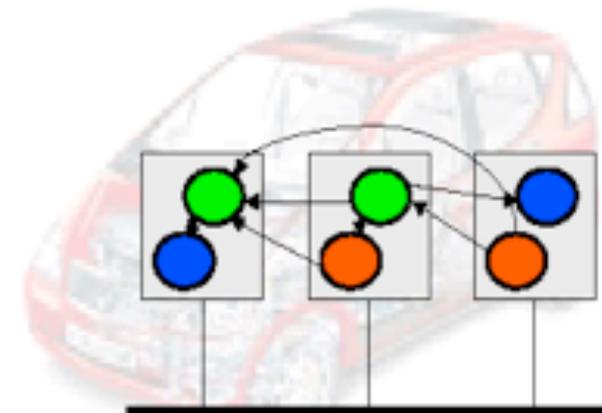
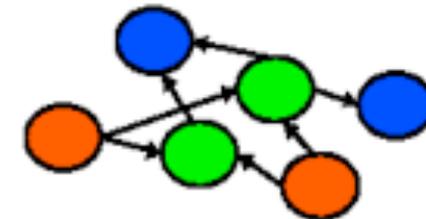
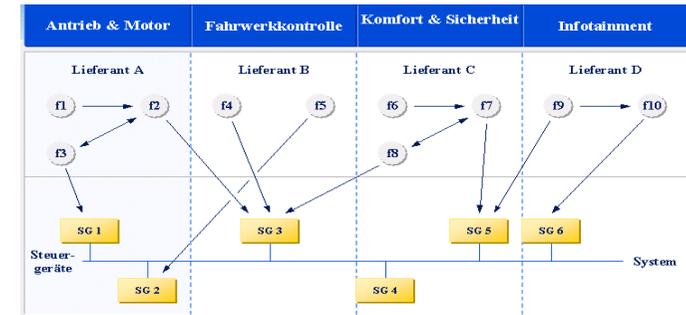
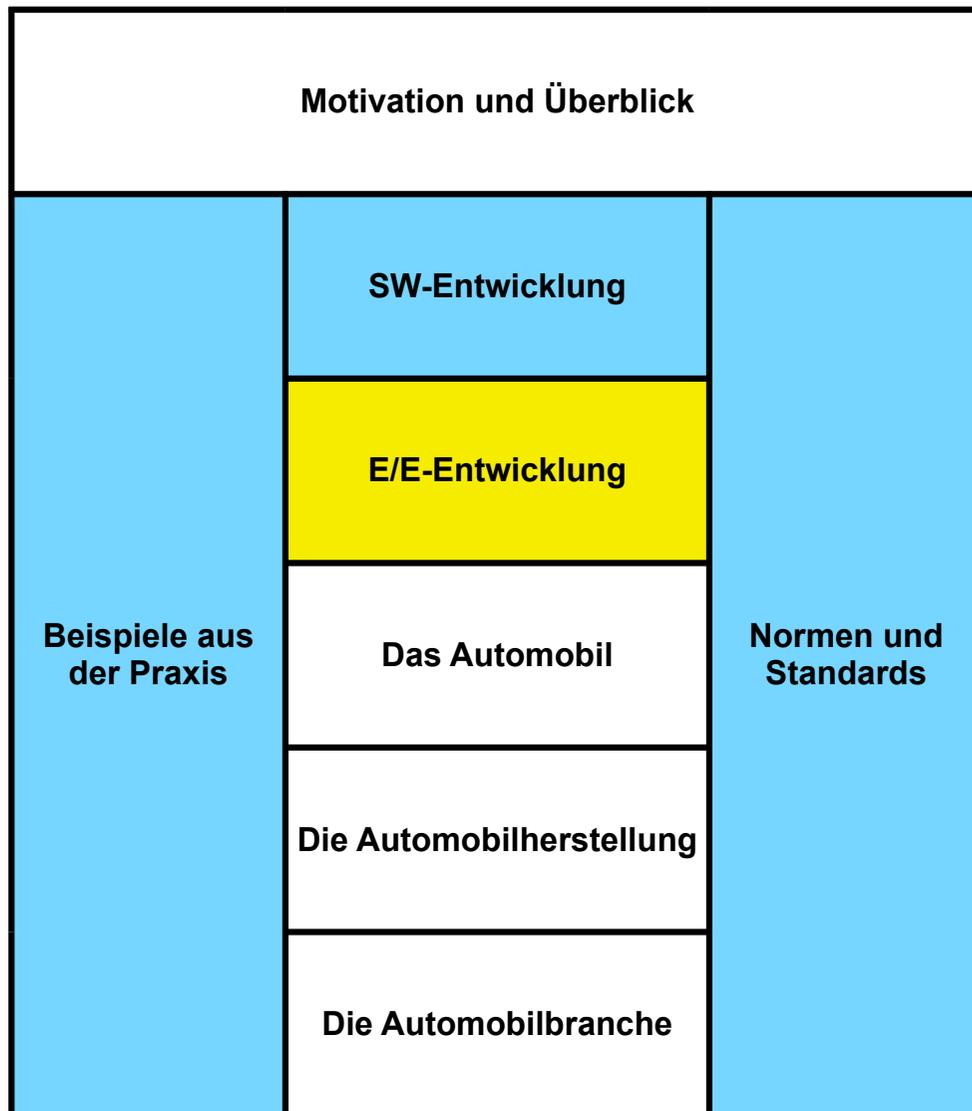
Vorlesung Automotive Software Engineering Teil 5 E/E-Entwicklung (1)

Sommersemester 2015

Prof. Dr. rer. nat. Bernhard Hohlfeld

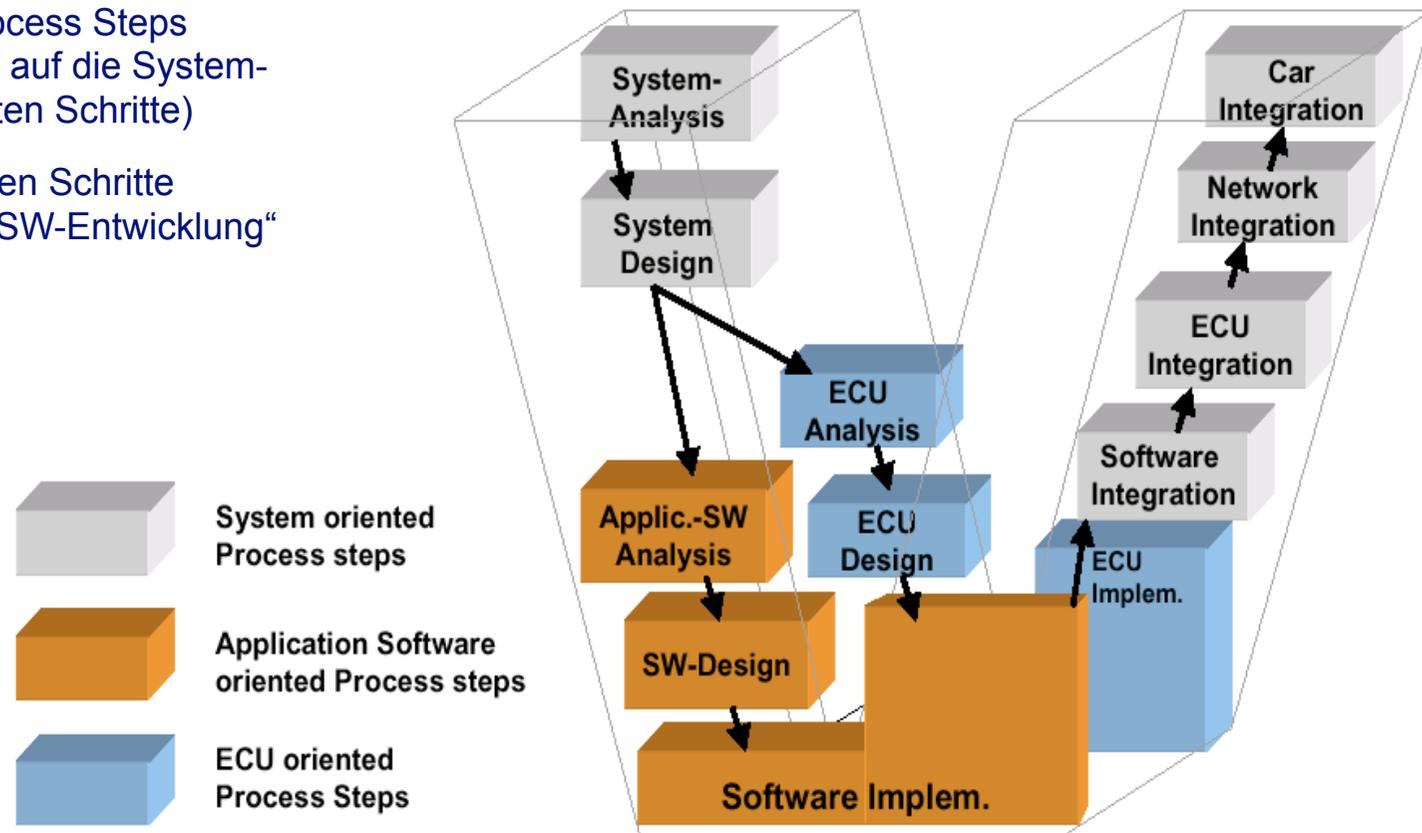
Bernhard.Hohlfeld@mailbox.tu-dresden.de

Technische Universität Dresden, Fakultät Informatik
Honorarprofessur Automotive Software Engineering



- 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)

- 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

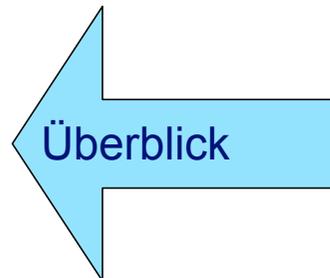
3.2. Halbleitertechnologie

3.3. Elektromagnetische Verträglichkeit (EMV)

- https://www.vector.com/vl_index_de.html
 - Kostenlos
 - Anmeldung erforderlich
- Themen
 - Serielle Bussysteme
 - Abschnitt „Bussysteme im Automobil“ basiert hierauf teilweise
 - Einführung in CAN
 - (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

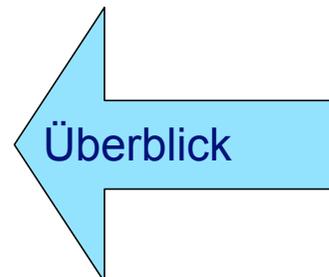
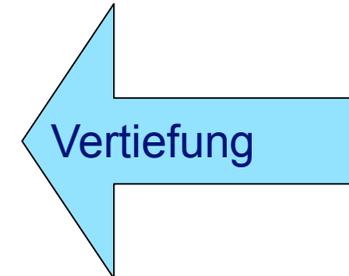
- 1 Einführung
 - 2 Architektur serieller Bussysteme
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
 - 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3: FlexRay
 - 4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

- 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



- 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



- **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**

■ 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

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

Motivation (1)

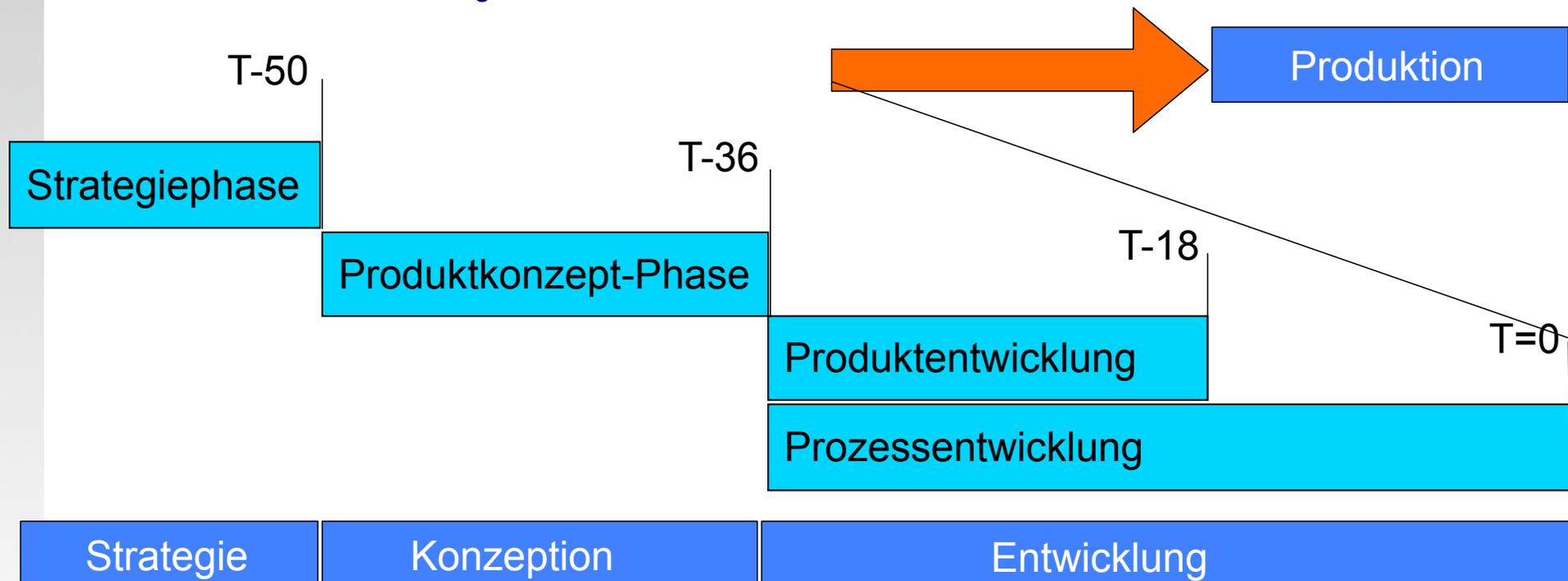


- Seit ca. 1990 intensive Elektronifizierung des Automobils
Das Automobil wird zu einem High-Tech-Produkt
- Treibende Kräfte
 - Immer anspruchsvollere Kundenwünsche an Automobil
 - Sicherheit
 - Komfort
 - Immer strengere Vorgaben des Gesetzgebers
 - Abgasemission
 - Verbrauch
 - Sicherheit
 - Wettbewerbs- und Kostendruck führen zu kontinuierlichem Innovationsdruck
 - siehe auch Teil 3 Die Automobilherstellung

Phasen der Fahrzeugentwicklung (Schematisch)

Anstösse / Treibende Kräfte

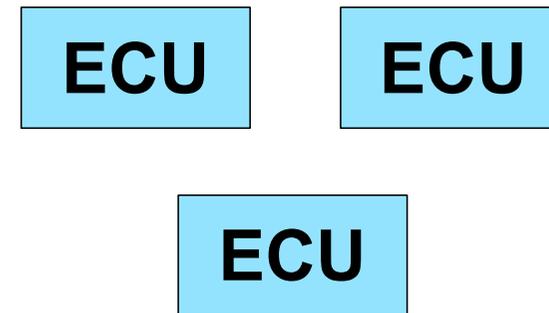
- Kundenwünsche
- Technischer Fortschritt
- Wettbewerbsdruck
- Gesetzliche Bestimmungen



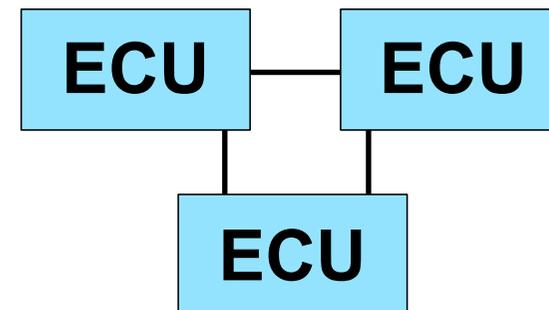
Motivation (2)

- Entwicklungsstufen der Elektronikarchitektur

- Unabhängig voneinander arbeitende elektronische Steuergeräte



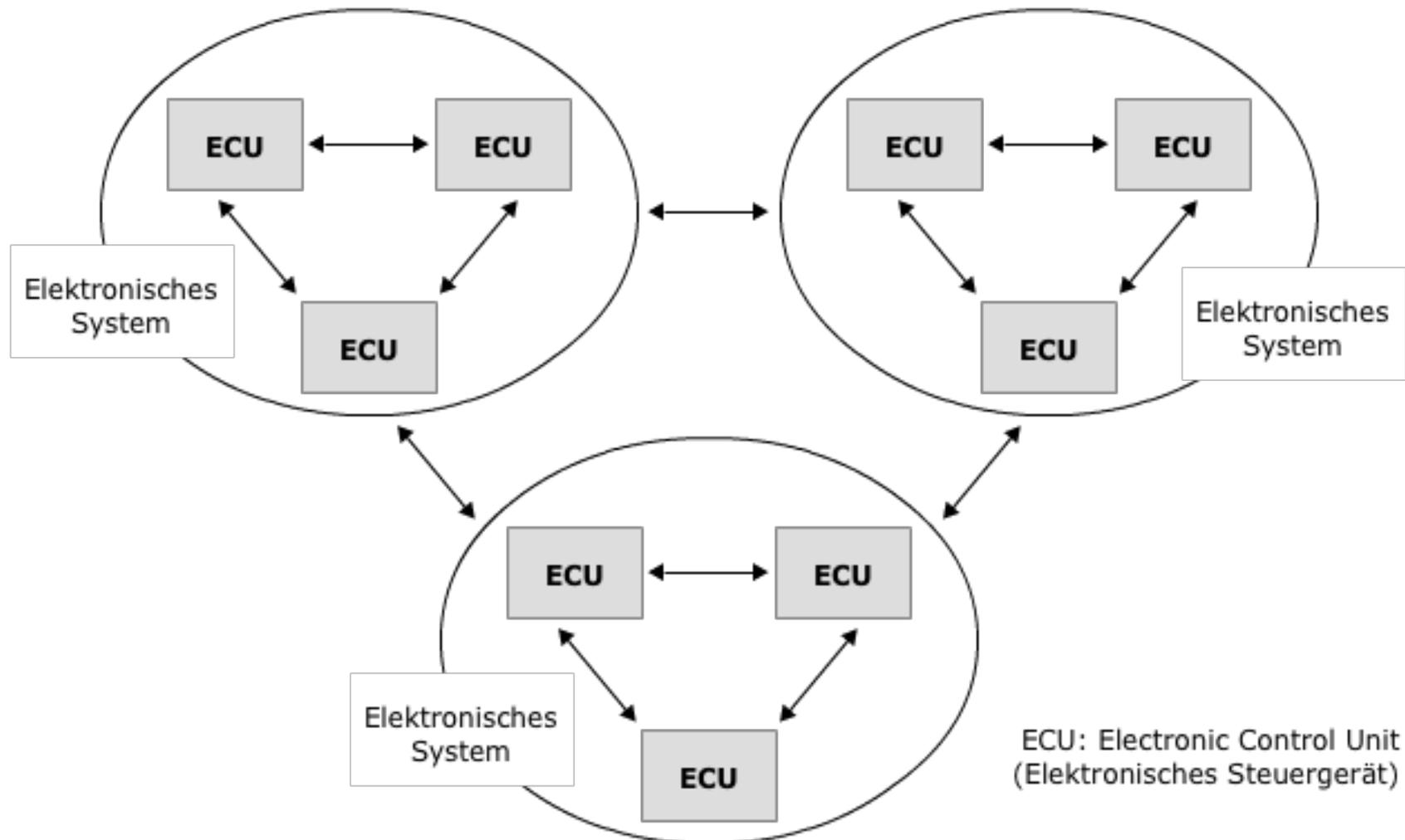
- Koordination elektronischer Steuergeräte
Verbesserung und Erweiterung der Fahrzeugfunktionalität.
- Neue elektronische Funktionen durch regen Datenaustausch zwischen den elektronischen Steuergeräten
- Beispiel Fahrdynamikfunktionen (ESP; ABS, ...)





Einführung

Steuergerätekopplung

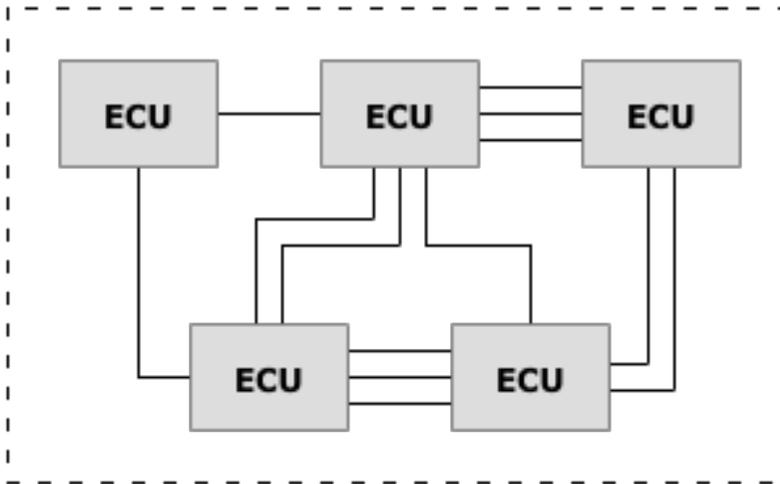


- Anfangs konventionelle Vernetzung zwischen elektronischen Steuergeräten
 - Jedem zu übertragenden Signal wurde eine elektrische Leitung zugeordnet
 - Hoher Verkabelungsaufwand bei intensiver Vernetzung
 - Lösung: Bitserieller Datenaustausch von zwischen mehreren elektronischen Steuergeräten über gemeinsam 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.

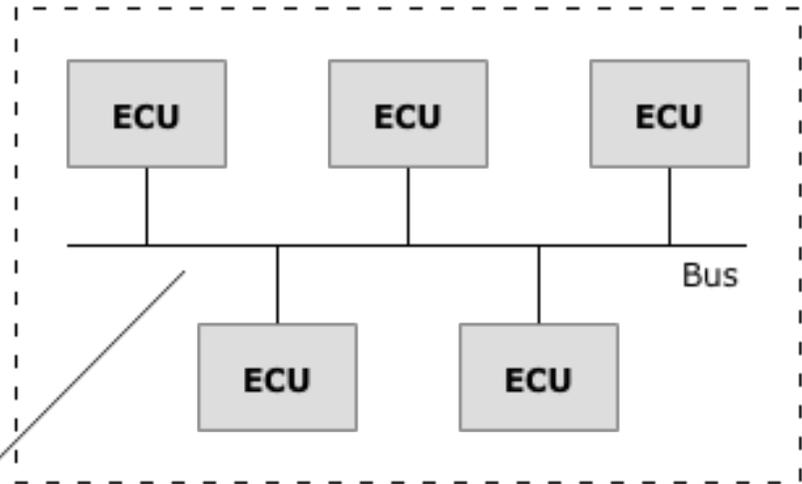


Einführung Busvernetzung

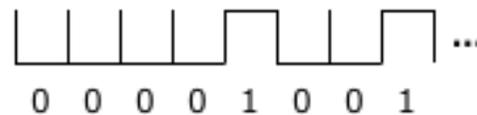
Konventionelle Vernetzung



Busvernetzung



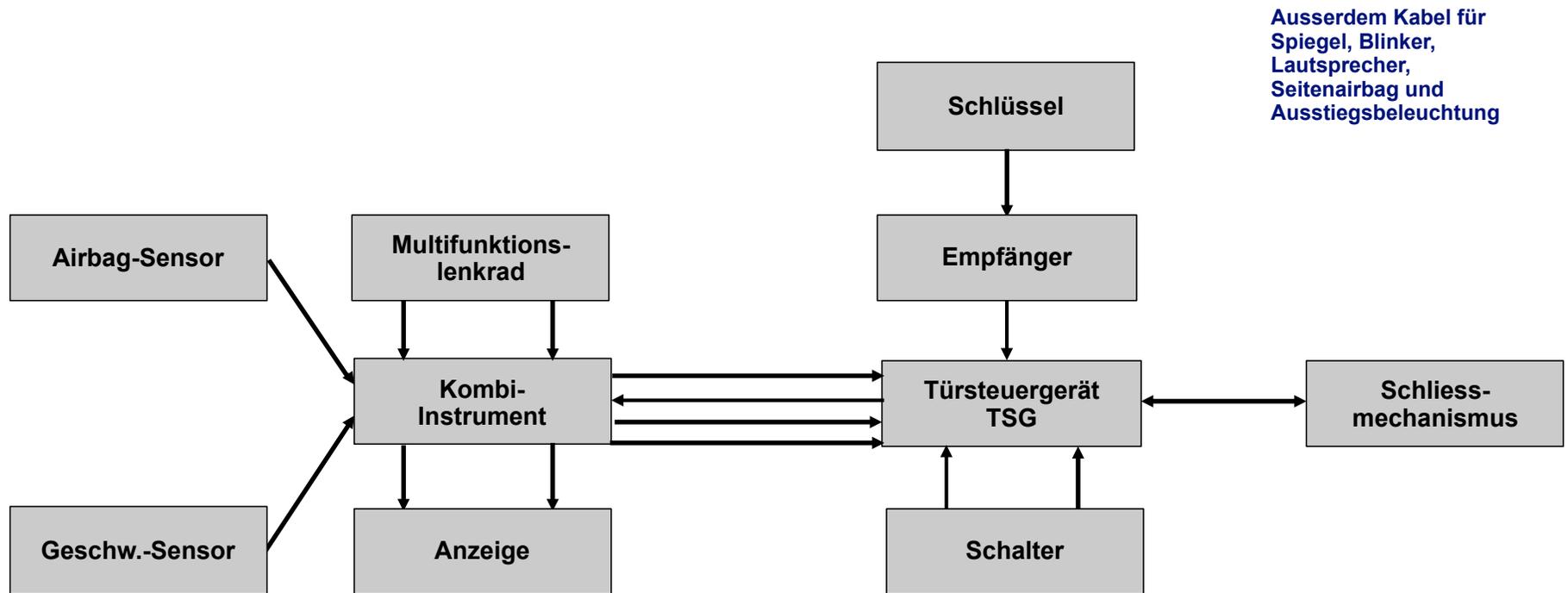
Bitserielle
Datenübertragung



ECU: Electronic Control Unit
(Elektronisches Steuergerät)

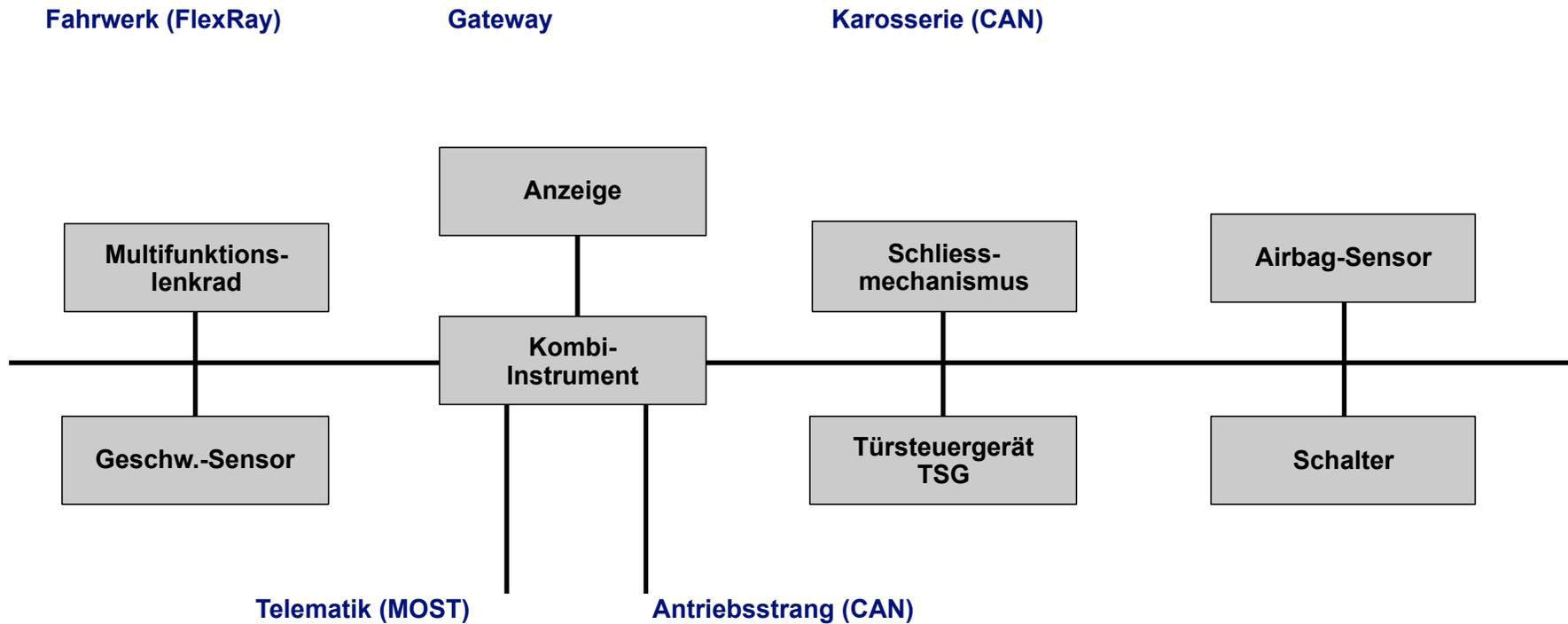


Beispiel Türsteuerung aus Teil 1 Motivation und Überblick



Funktionale Architektur (Ausschnitt)

Beispiel Türsteuerung aus Teil 1 Motivation und Überblick



Mögliche Vernetzung (Ausschnitt)

- 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

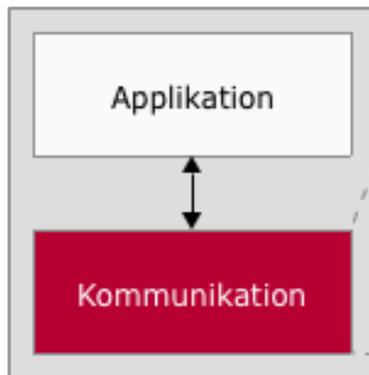
Da für den seriellen Datenaustausch zwischen elektronischen Steuergeräten im Kfz im Wesentlichen nur die beiden unteren Schichten (Bitübertragungs- und Datensicherungsschicht) von Bedeutung sind, zudem Funktionen nicht berücksichtigter Schichten der obersten Schicht (Anwendungsschicht) zugeschlagen werden, reduziert sich das Siebenschichtenmodell auf ein Dreischichtenmodell.



Einführung

ISO/OSI-Modell

Busknoten

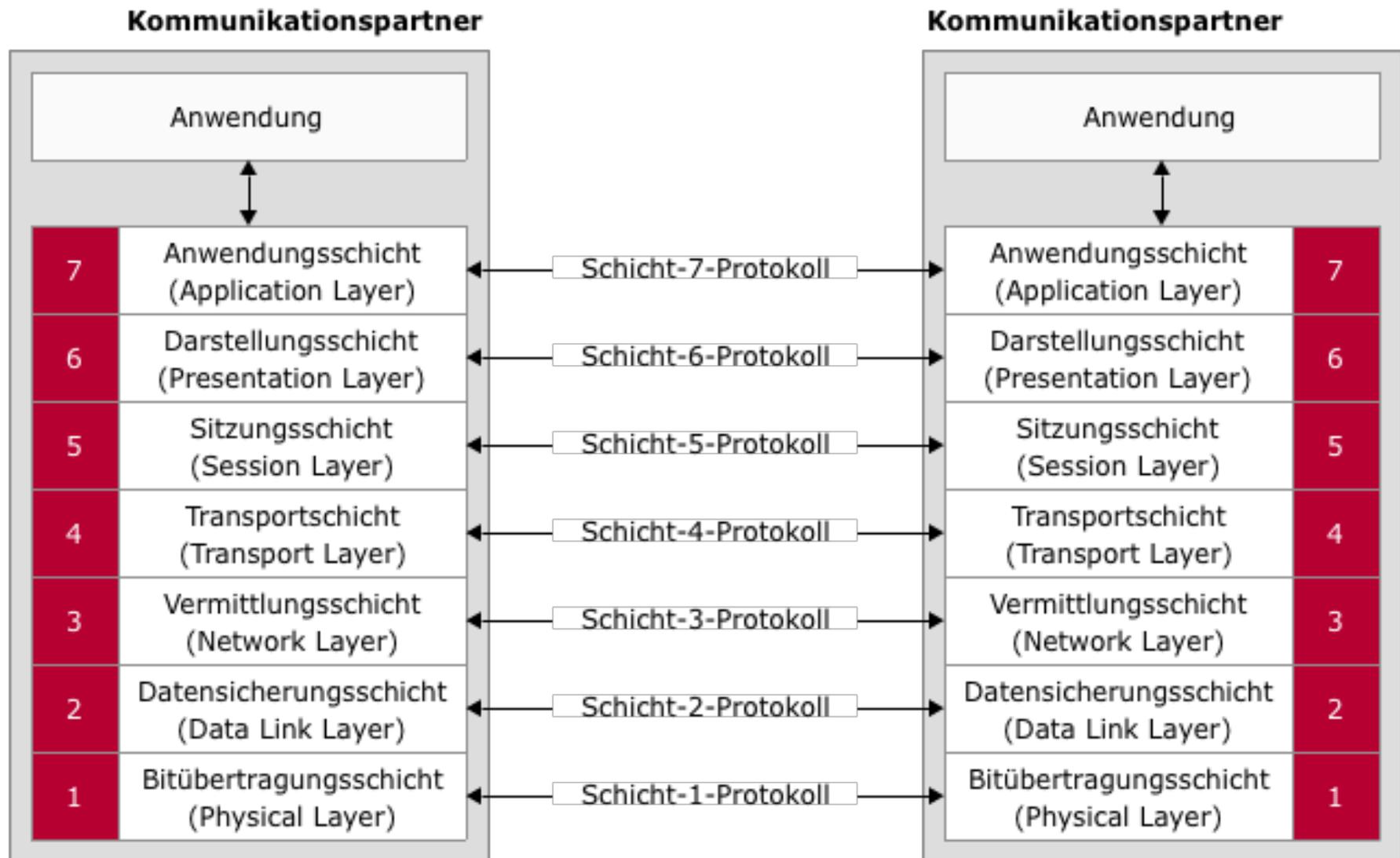


7	Anwendungsschicht (Application Layer)	Zugriff auf das Kommunikationssystem, Entkopplung Anwendung von Kommunikation
6	Darstellungsschicht (Presentation Layer)	Semantik, Kompression, Verschlüsselung
5	Sitzungsschicht (Session Layer)	Unterhalten längerer Sitzungen, Definition von Synchronisationspunkten
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

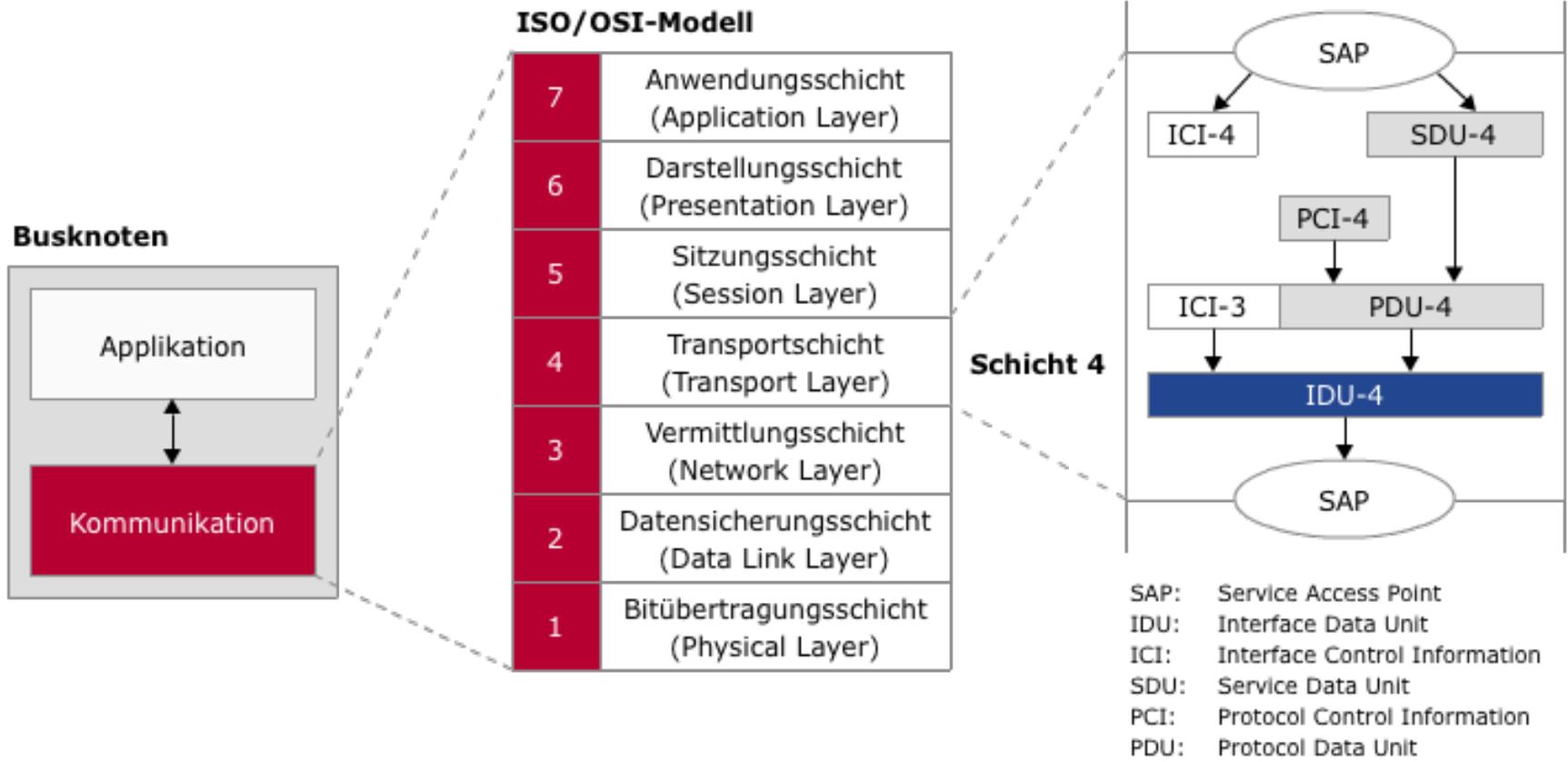
Peer-to-Peer-Kommunikation





Einführung

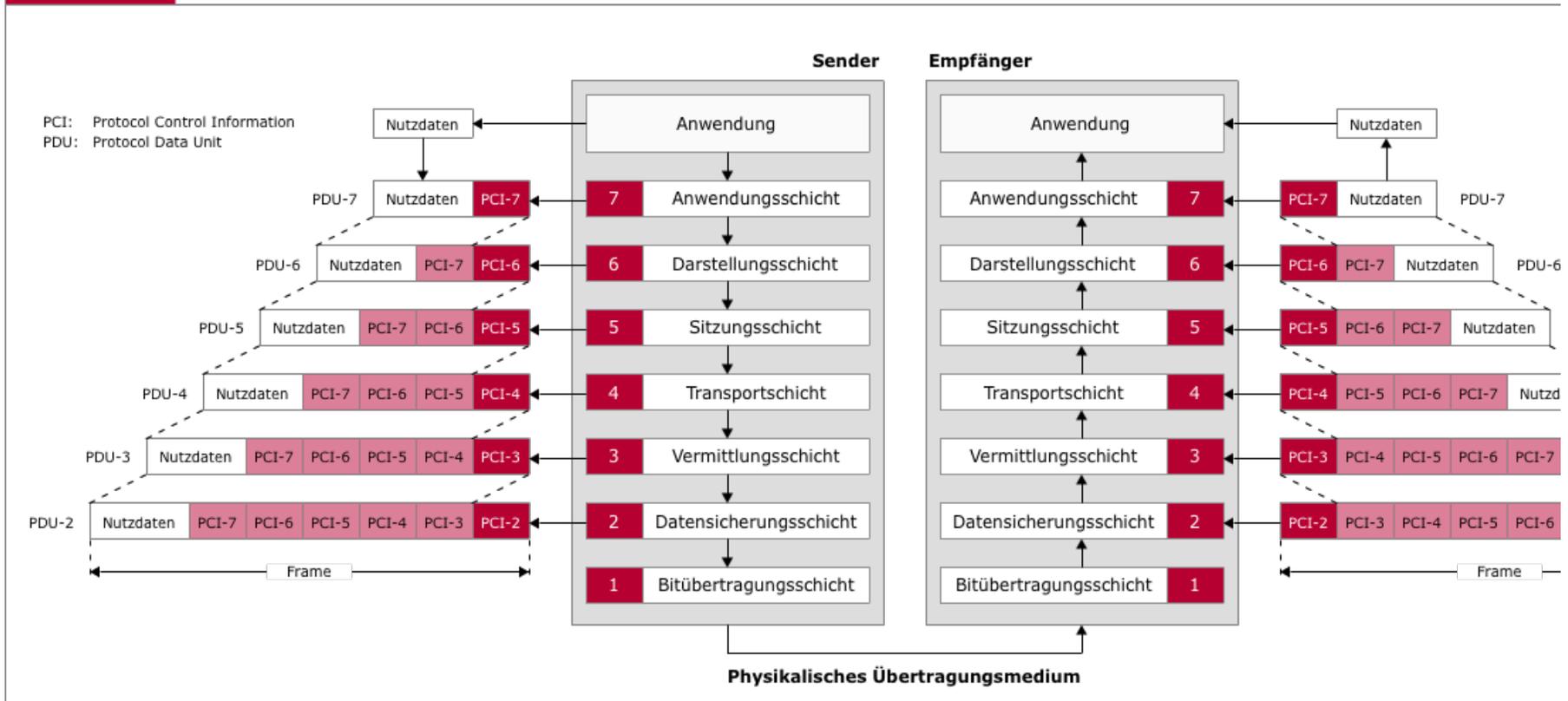
Schichtenkommunikation

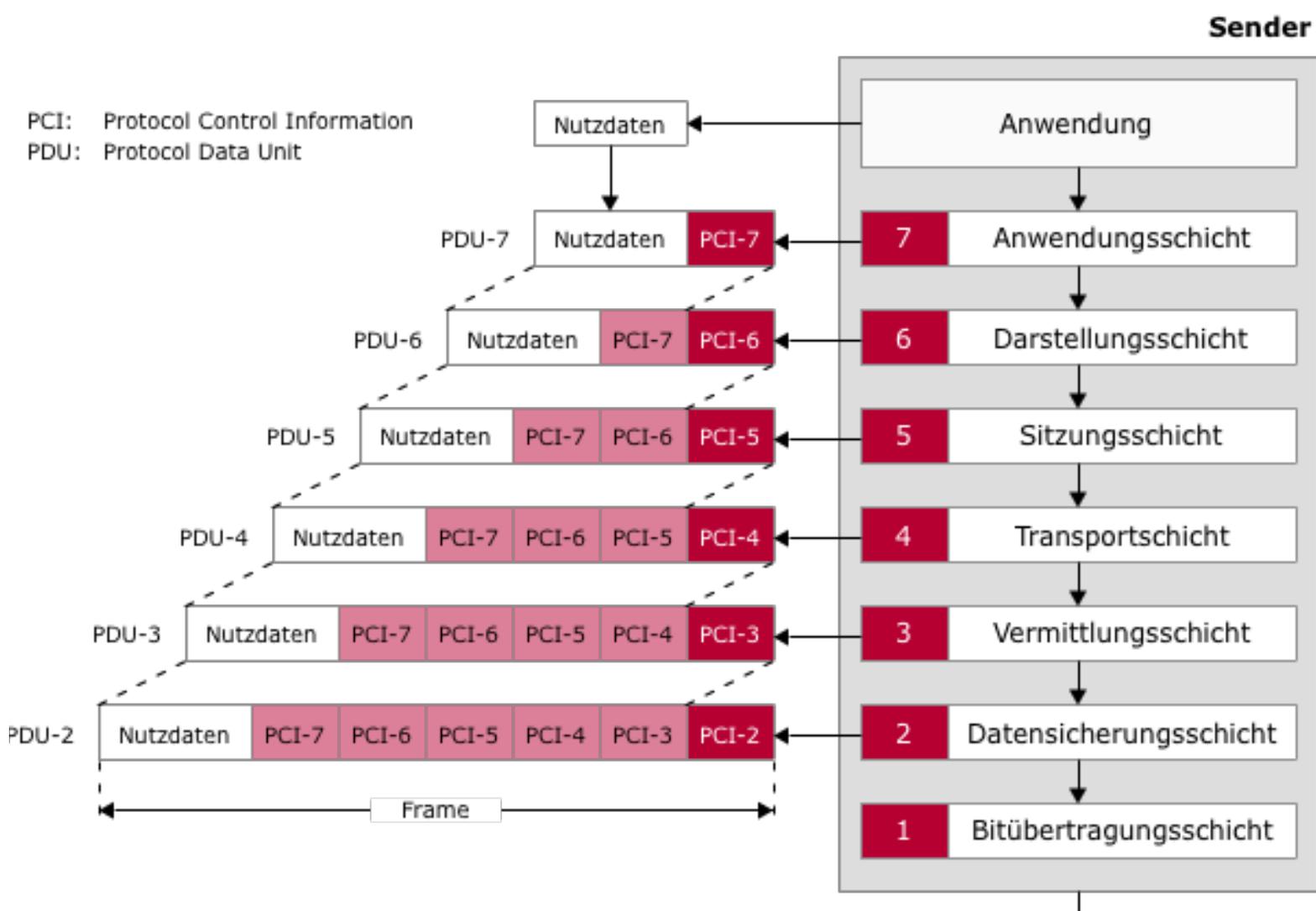




Einführung

Realer Kommunikationsfluss



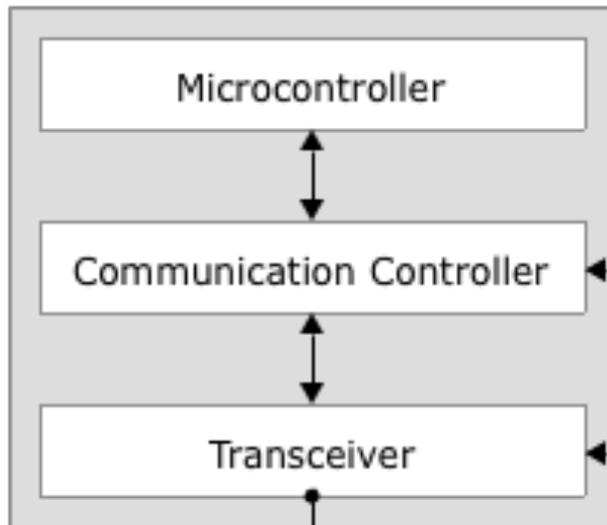




Einführung

Busnotenarchitektur

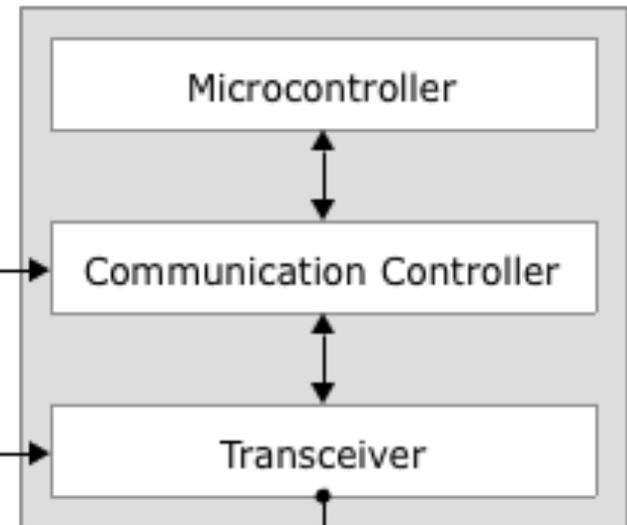
Busknoten



Schicht-2-Protokoll

Schicht-1-Protokoll

Busknoten



Bus

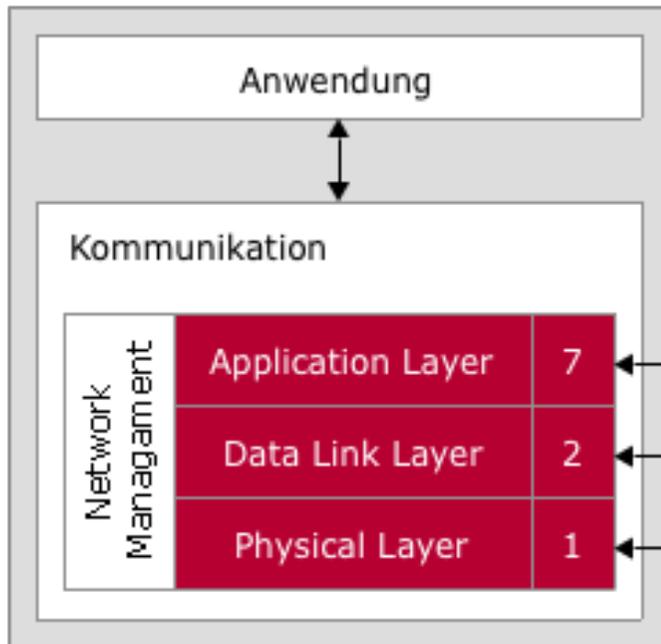




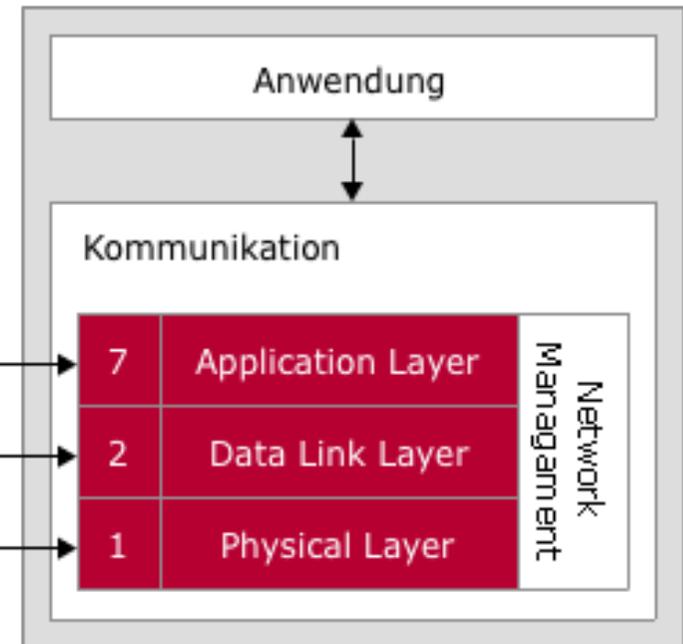
Einführung

Dreischichtenmodell

Busknoten



Busknoten



Schicht-7-Protokoll

Schicht-2-Protokoll

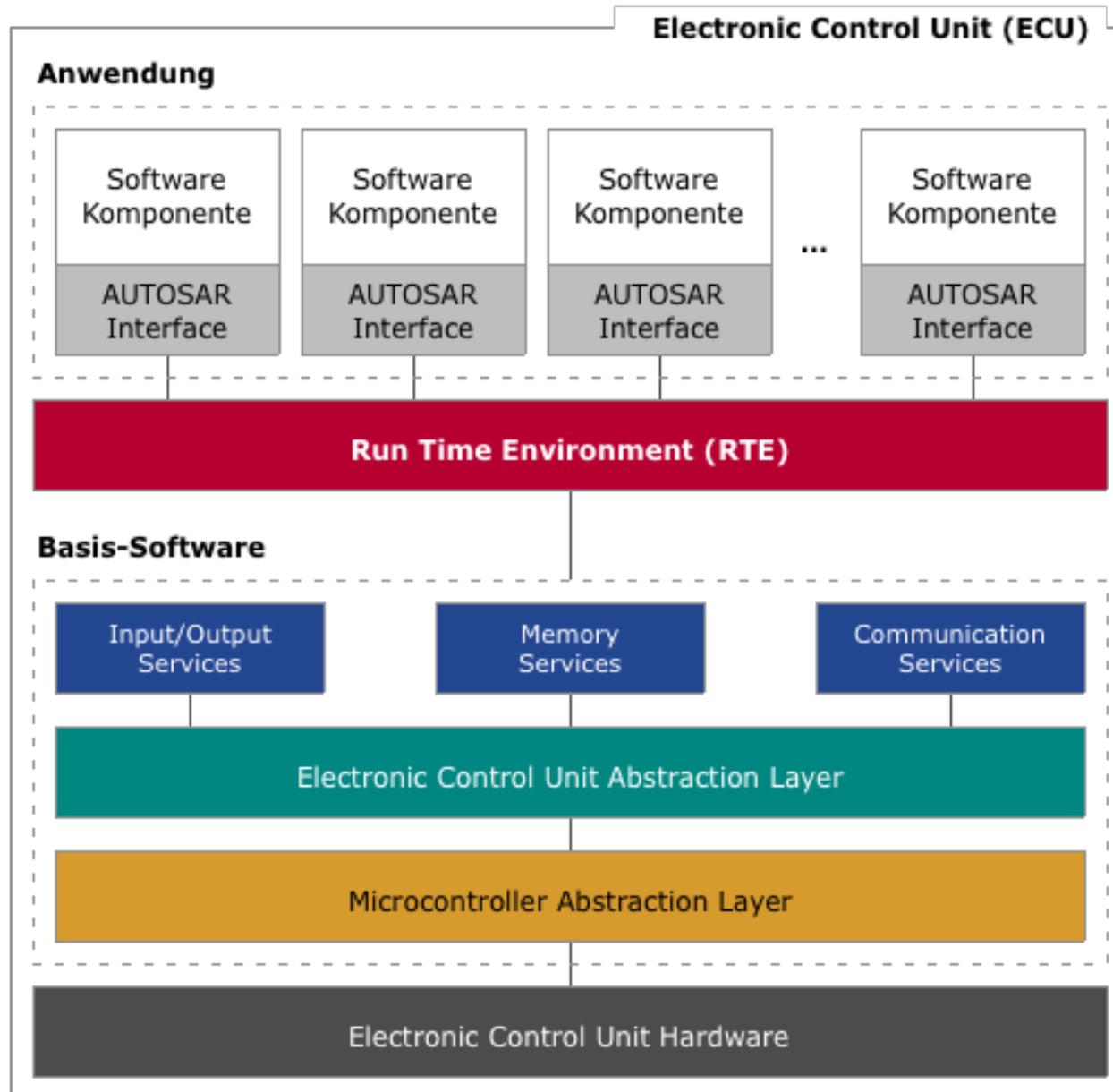
Schicht-1-Protokoll





Einführung

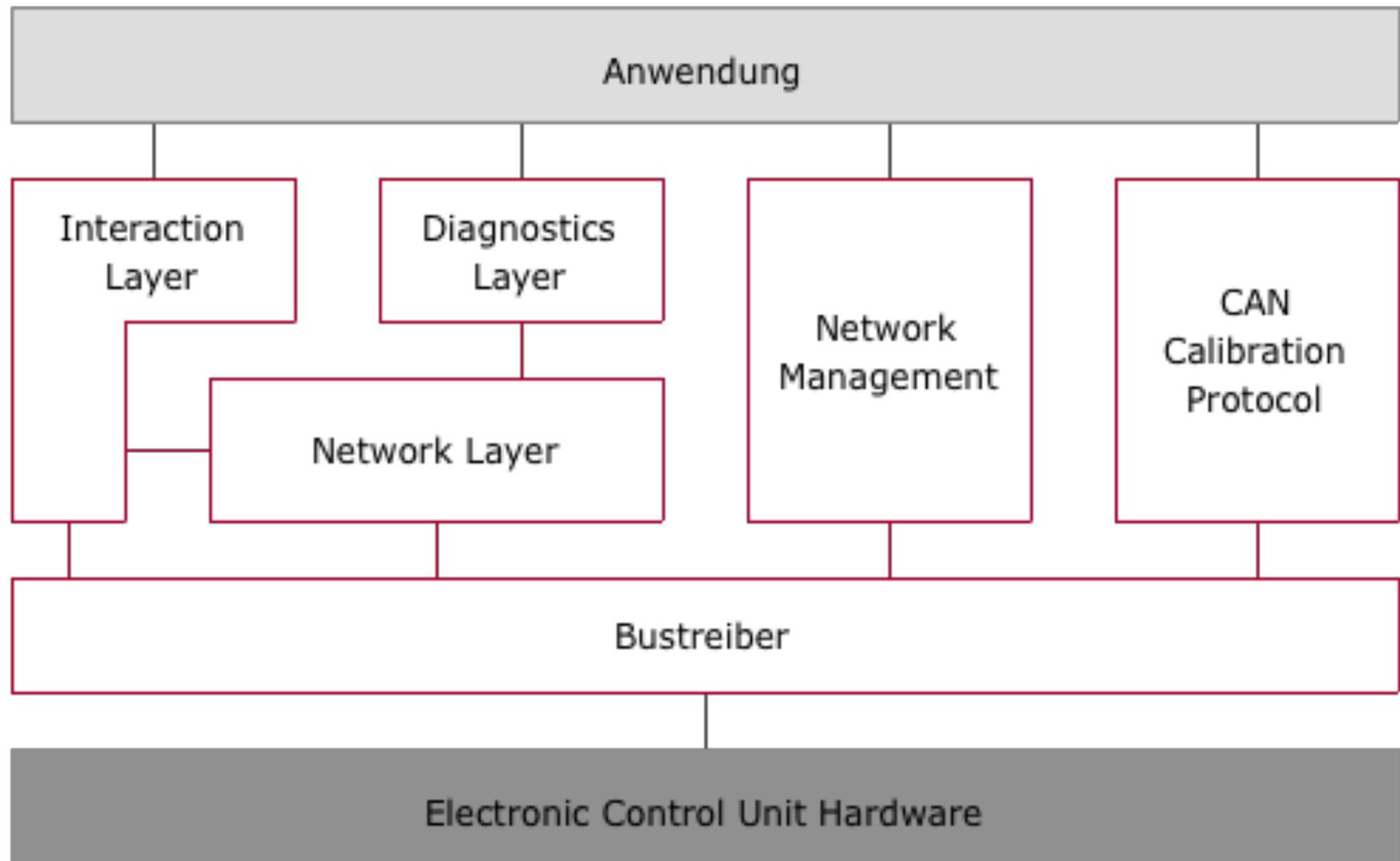
AUTOSAR-Architektur





Einführung

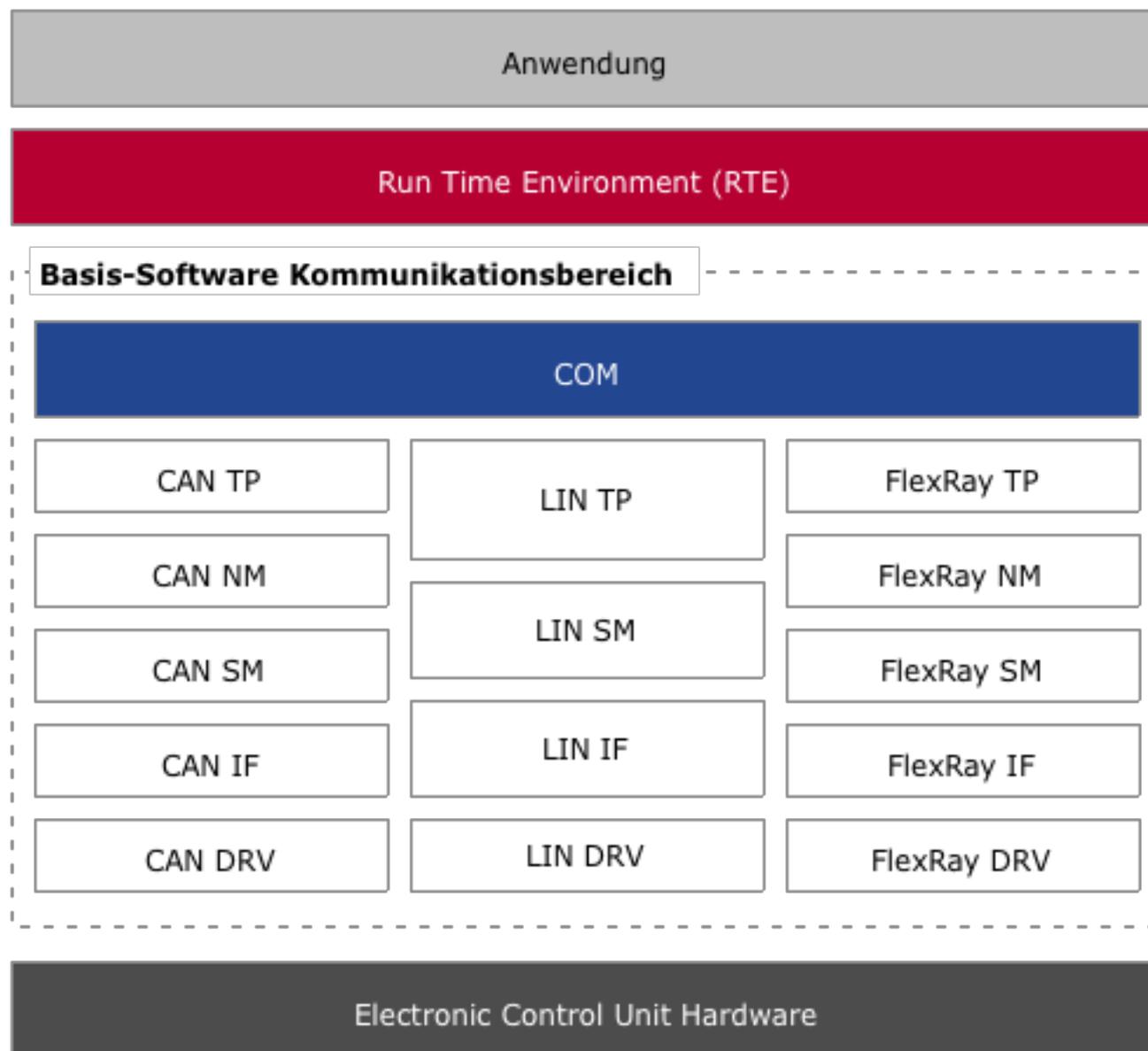
ISO/OSEK-Architektur





Einführung

AUTOSAR-Architektur



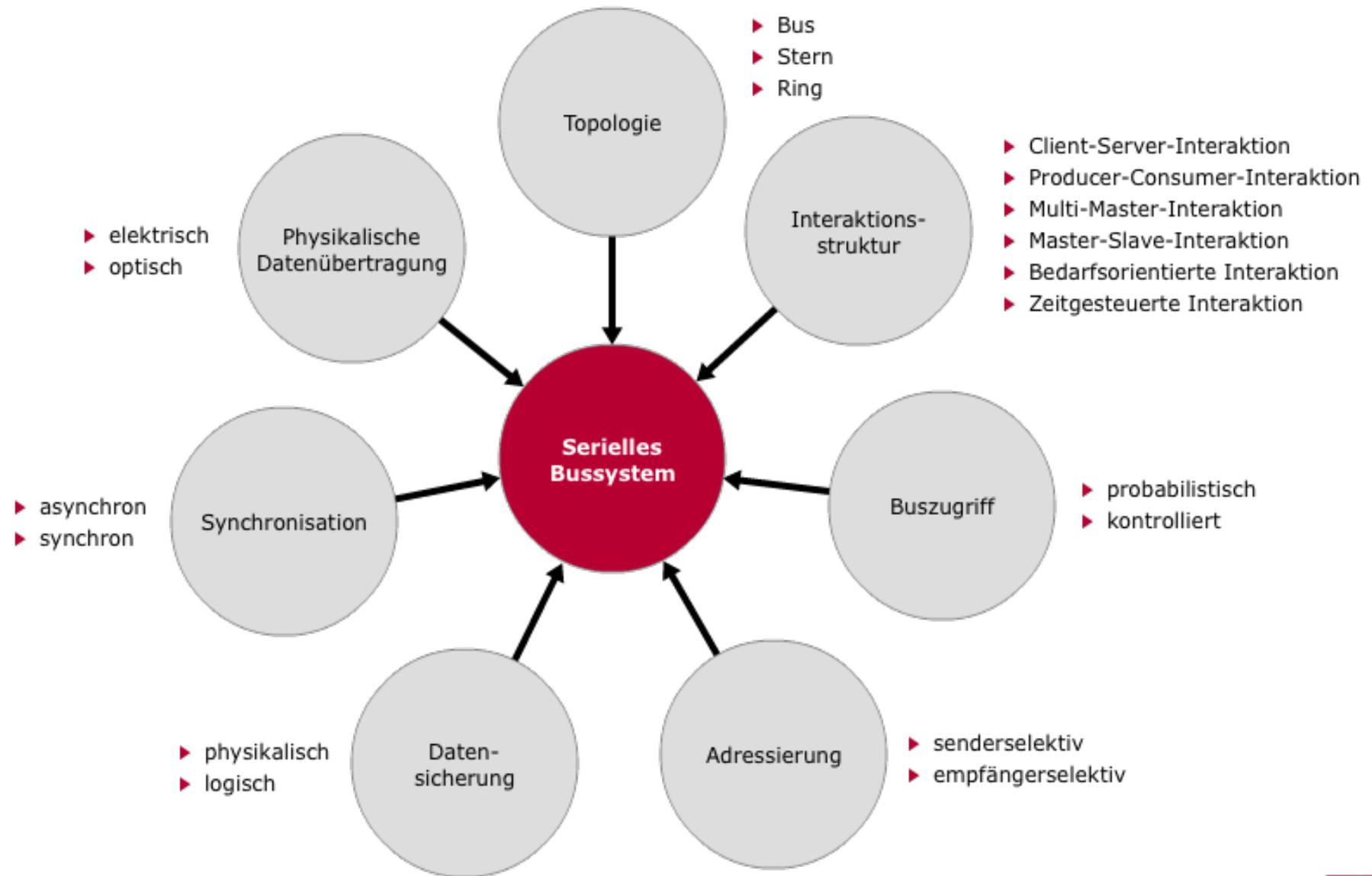
COM: Communication
TP: Transportprotokoll
NM: Netzmanagement
SM: Statusmanagement
IF: Interface
DRV: Driver

- 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

Komponenten serieller Bussysteme

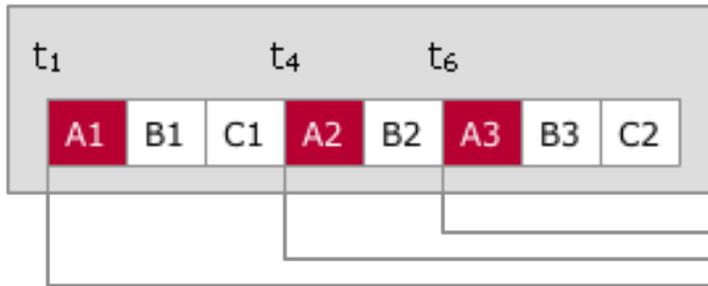




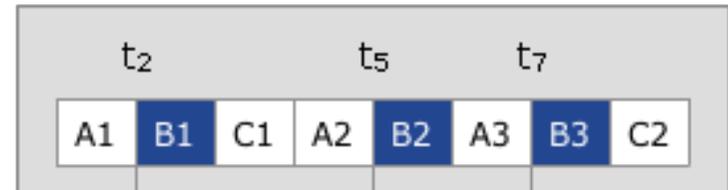
Architektur serieller Bussysteme

Systemintegration

Busknoten A

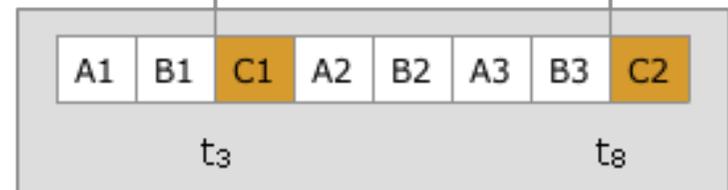
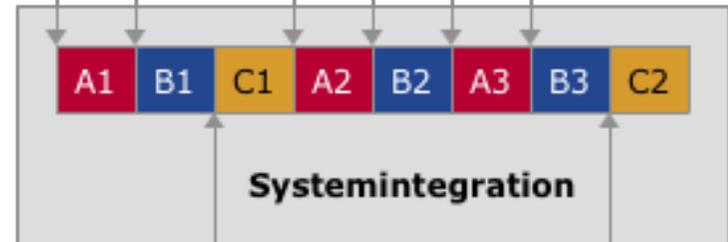


Busknoten B



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

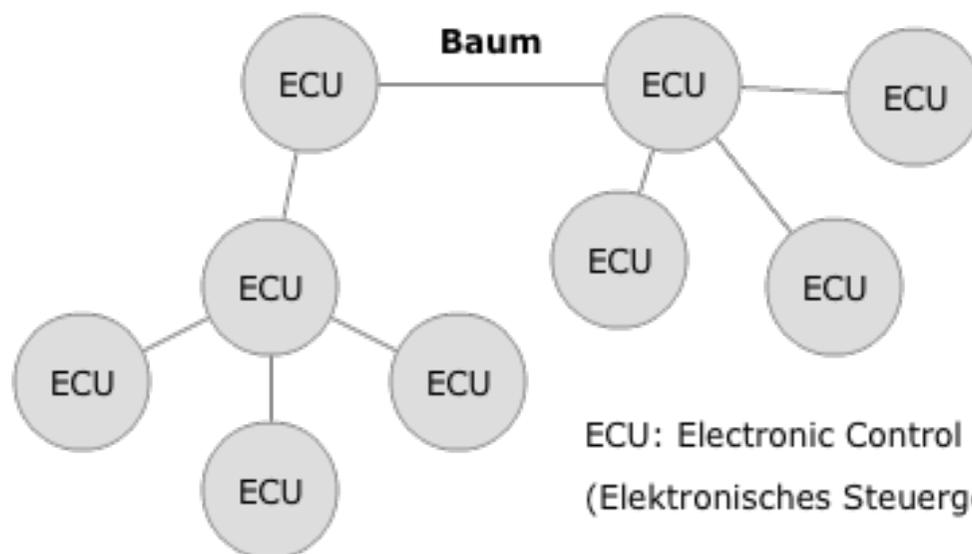
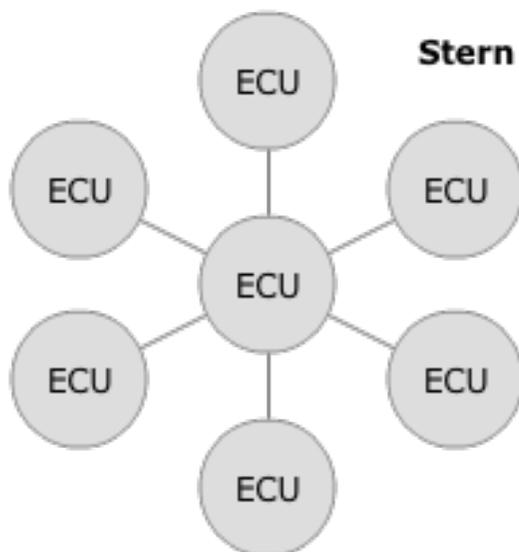
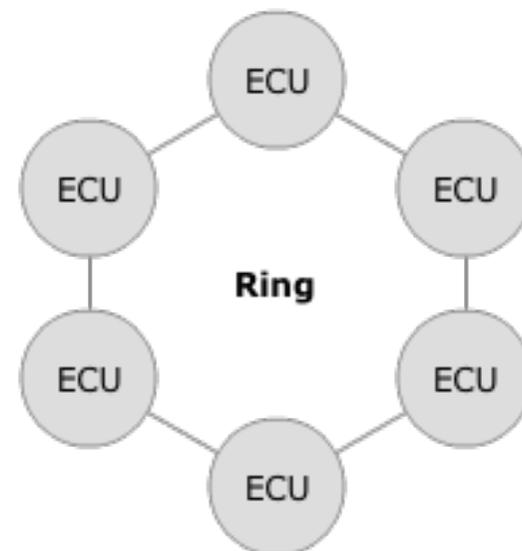
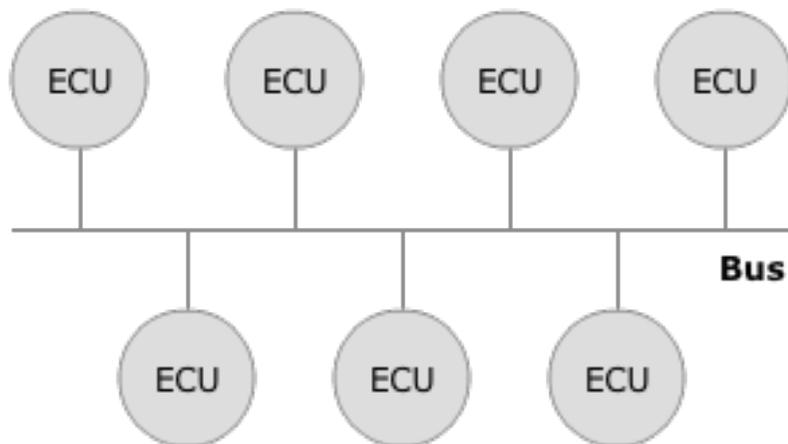


- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - **2.2 Topologie**
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3: FlexRay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



Architektur serieller Bussysteme

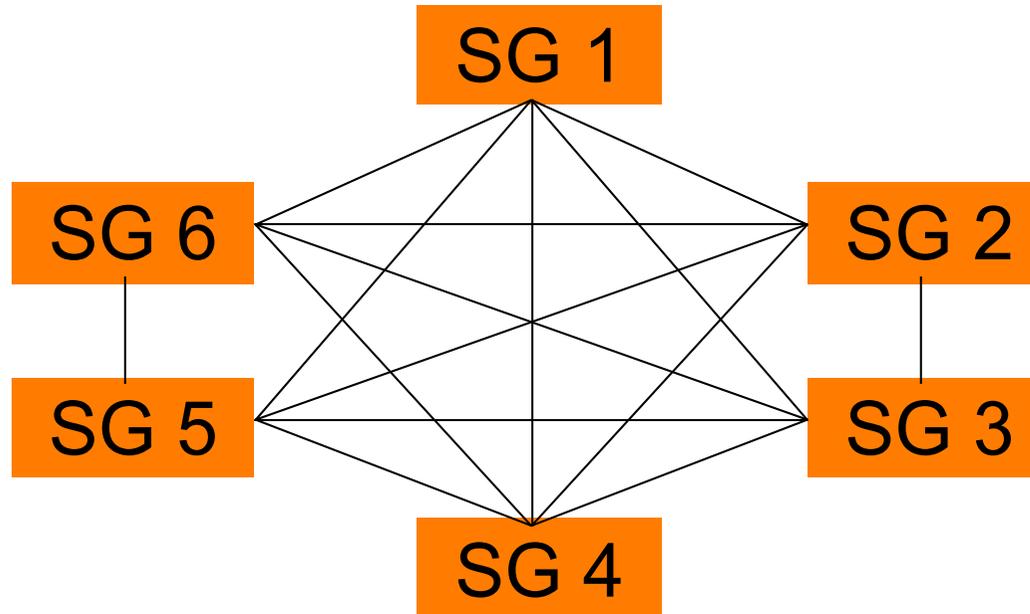
Topologien



ECU: Electronic Control Unit
(Elektronisches Steuergerät)

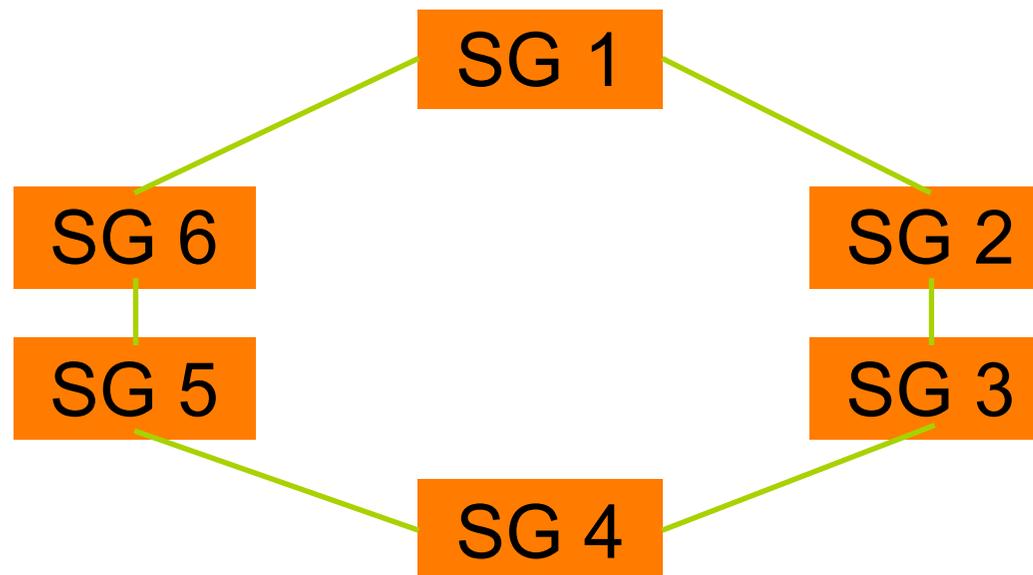
Topologien - Direkte Verbindung

- 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)



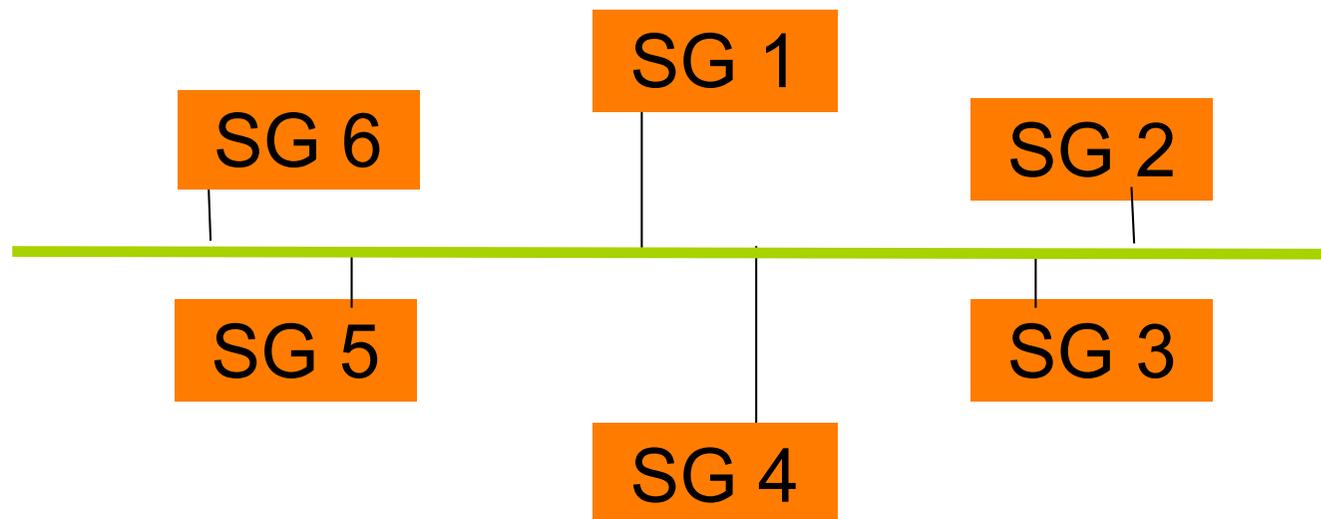
Topologien - Ring

- 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



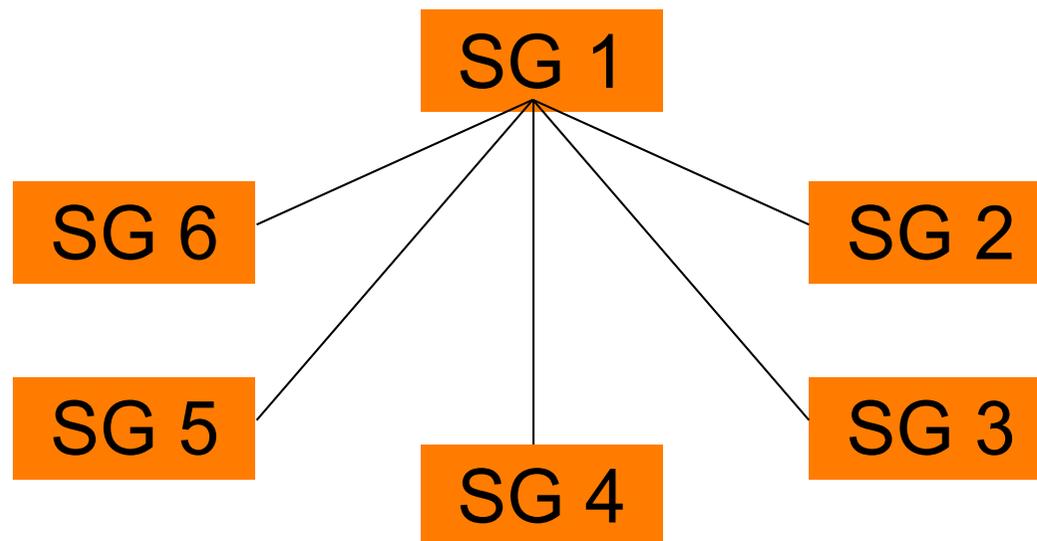
Topologien - Bus

- 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

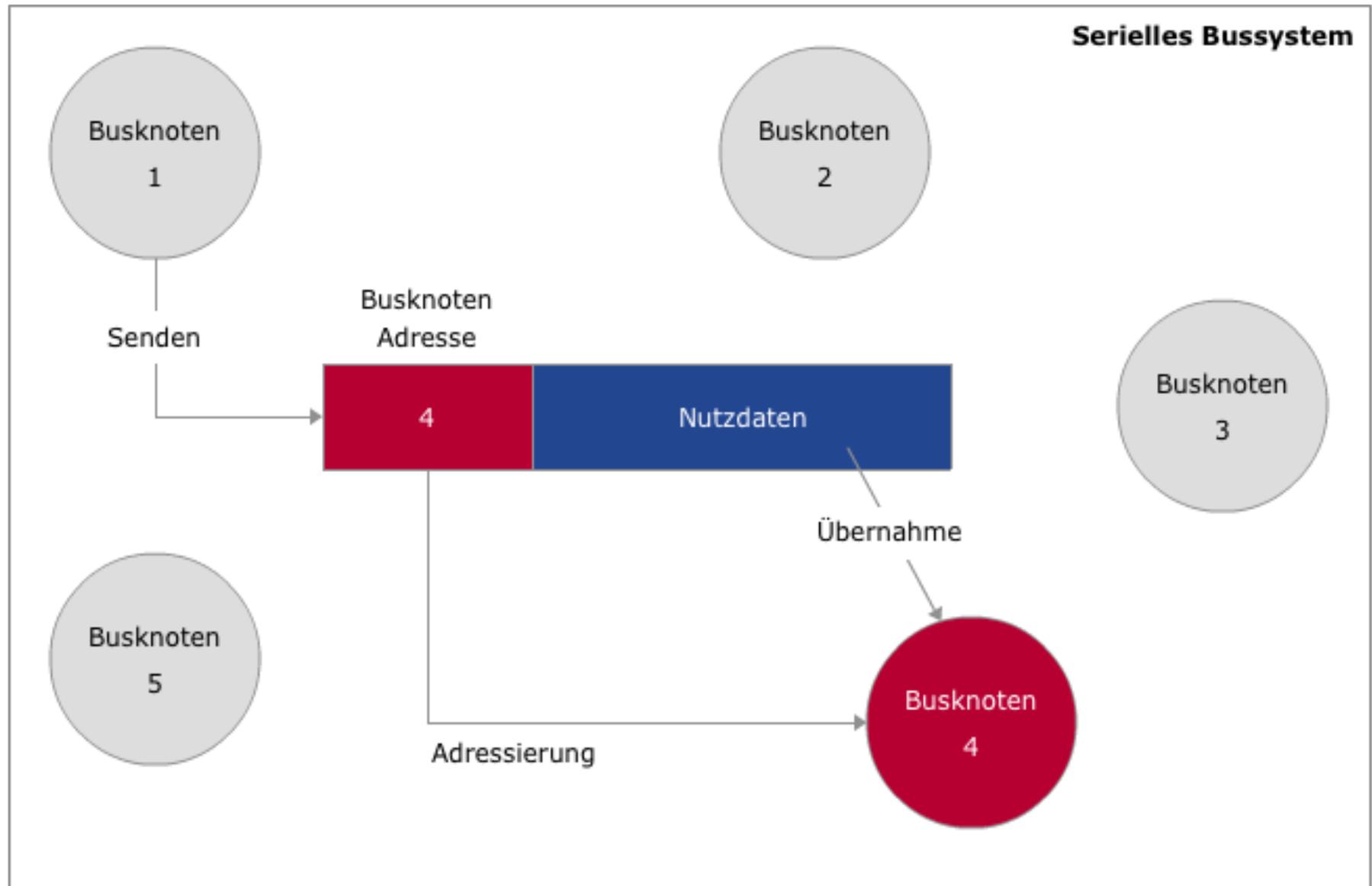


- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - **2.3 Adressierung und Framing**
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3: FlexRay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



Architektur serieller Bussysteme

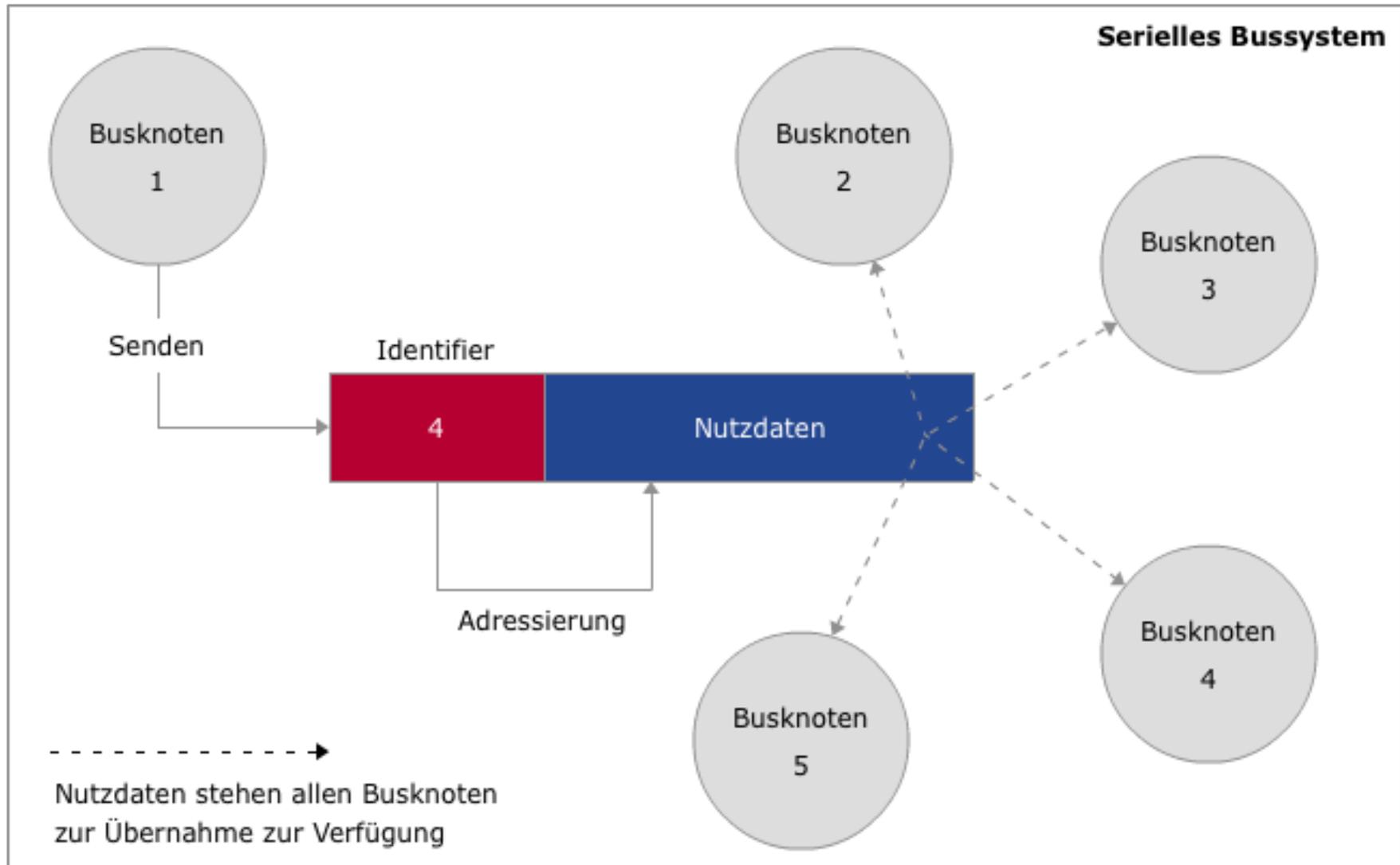
Senderselektive Adressierung





Architektur serieller Bussysteme

Empfängerselektive Adressierung



- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - **2.5 Datensicherung 06.05,**
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3: FlexRay
- 4 Lernzielkontrolle

■ 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:

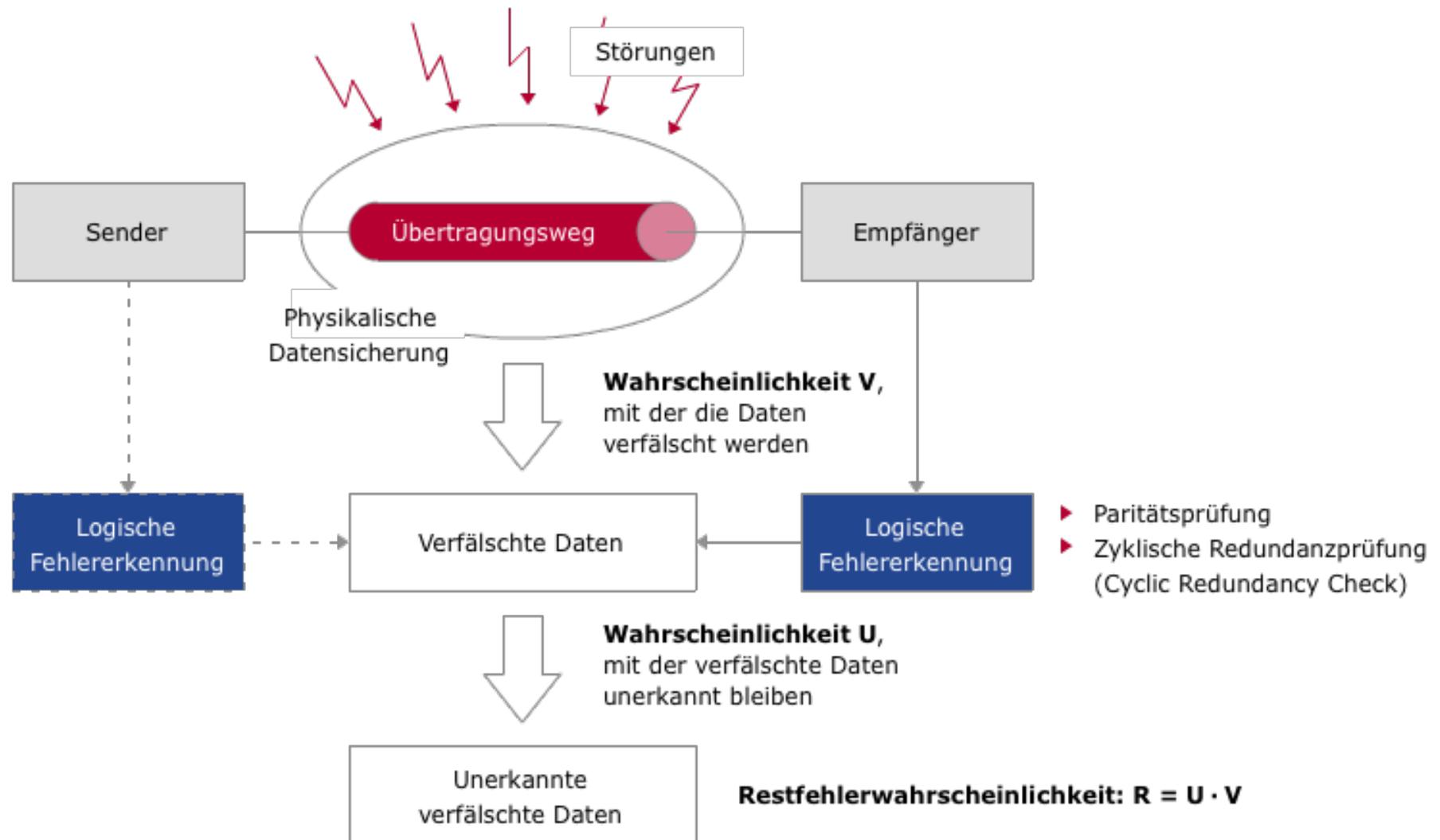
- Verwendet nichtlineare Eigenschaften bestimmter mathematischer Gruppen (Ideale) zur Optimierung des Verhältnisses von Hammingdistanz zu Bandbreitenbedarf
- Unempfindlich gegen die meisten statistischen und systematischen Fehler

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

Prinzip der Datensicherung





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

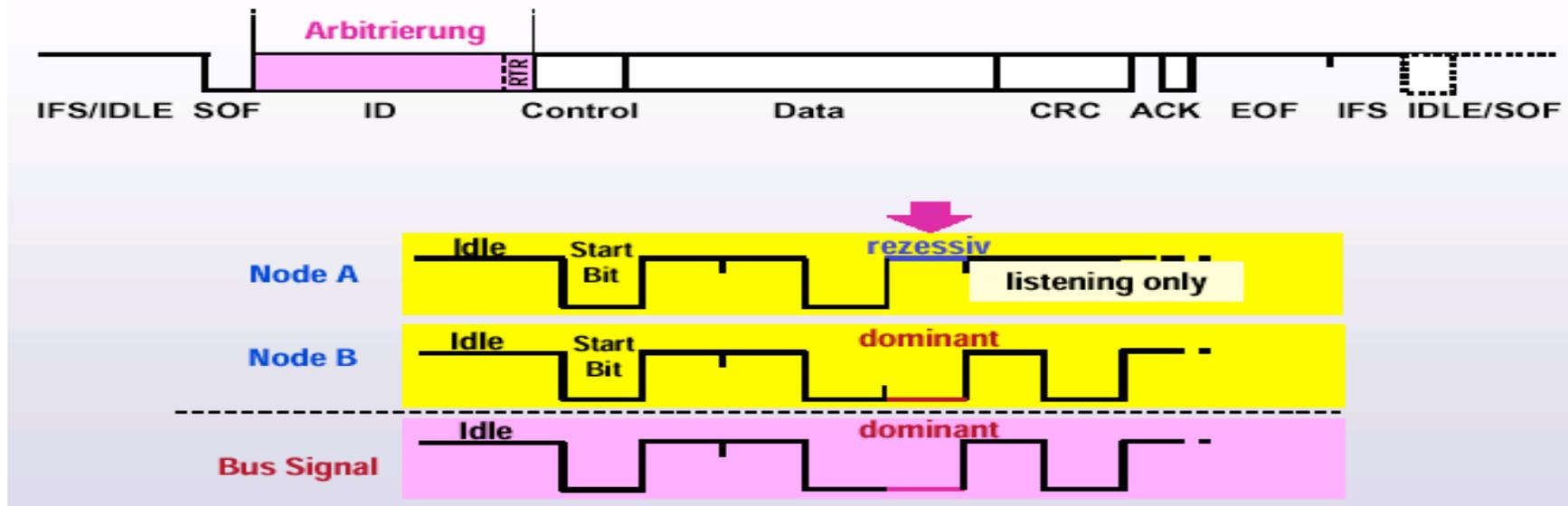
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Paritätsbit
(Even Parity)

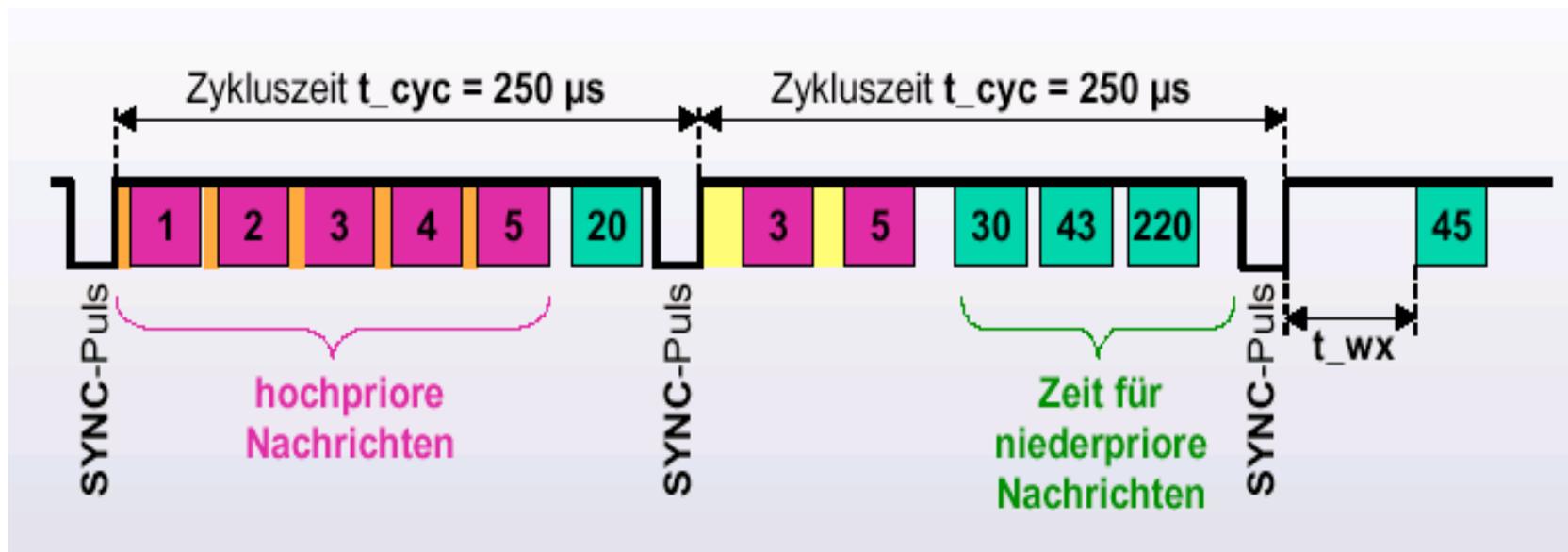


- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - **2.6 Synchronisation**
 - 2.7 Physikalische Übertragung
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3: FlexRay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

- 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



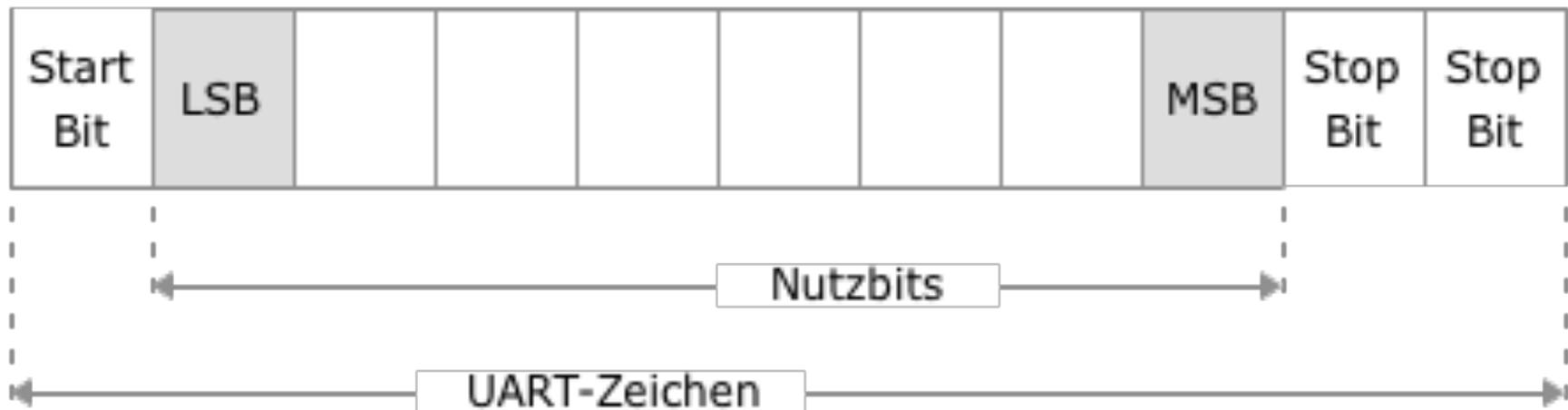
- Ein Busmaster synchronisiert alle Teilnehmer durch Senden eines Synchronisationssignals (SYNC-Puls)
- Innerhalb eines Zyklus gibt es für bestimmte (hochpriore) Nachrichten-IDs festgelegte Zeitfenster, in denen deren Übertragung garantiert wird.
- Freibleibende Zeit kann für asynchrone (niederpriore)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





Architektur serieller Bussysteme

UART-Format



UART: Universal Asynchronous Receiver Transmitter

LSB: Least Significant Bit

MSB: Most Significant Bit





Architektur serieller Bussysteme

Manchester Codierung

Logischer Bitpegel



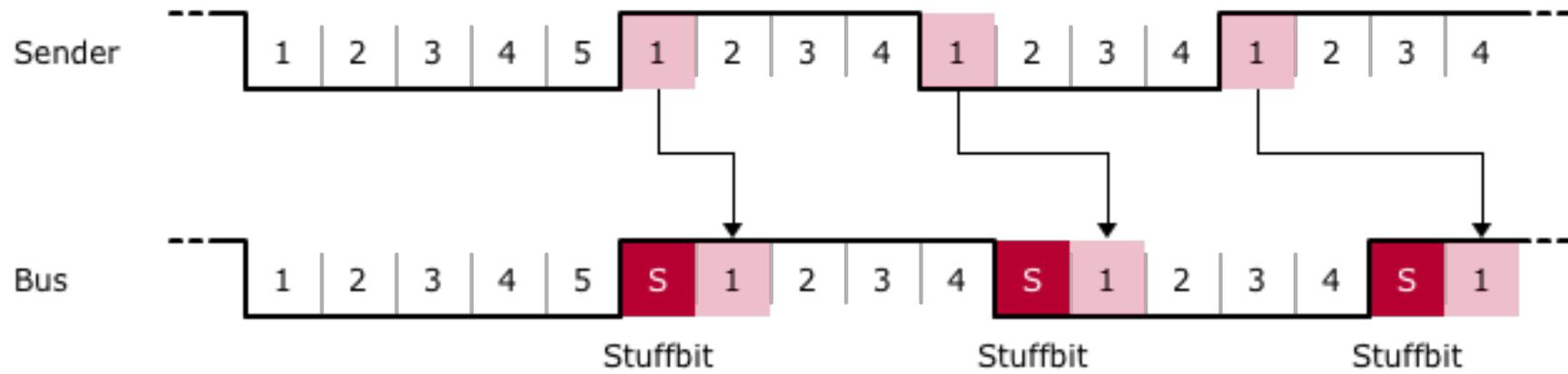
Manchester Codierung





Architektur serieller Bussysteme

Bitstuffing

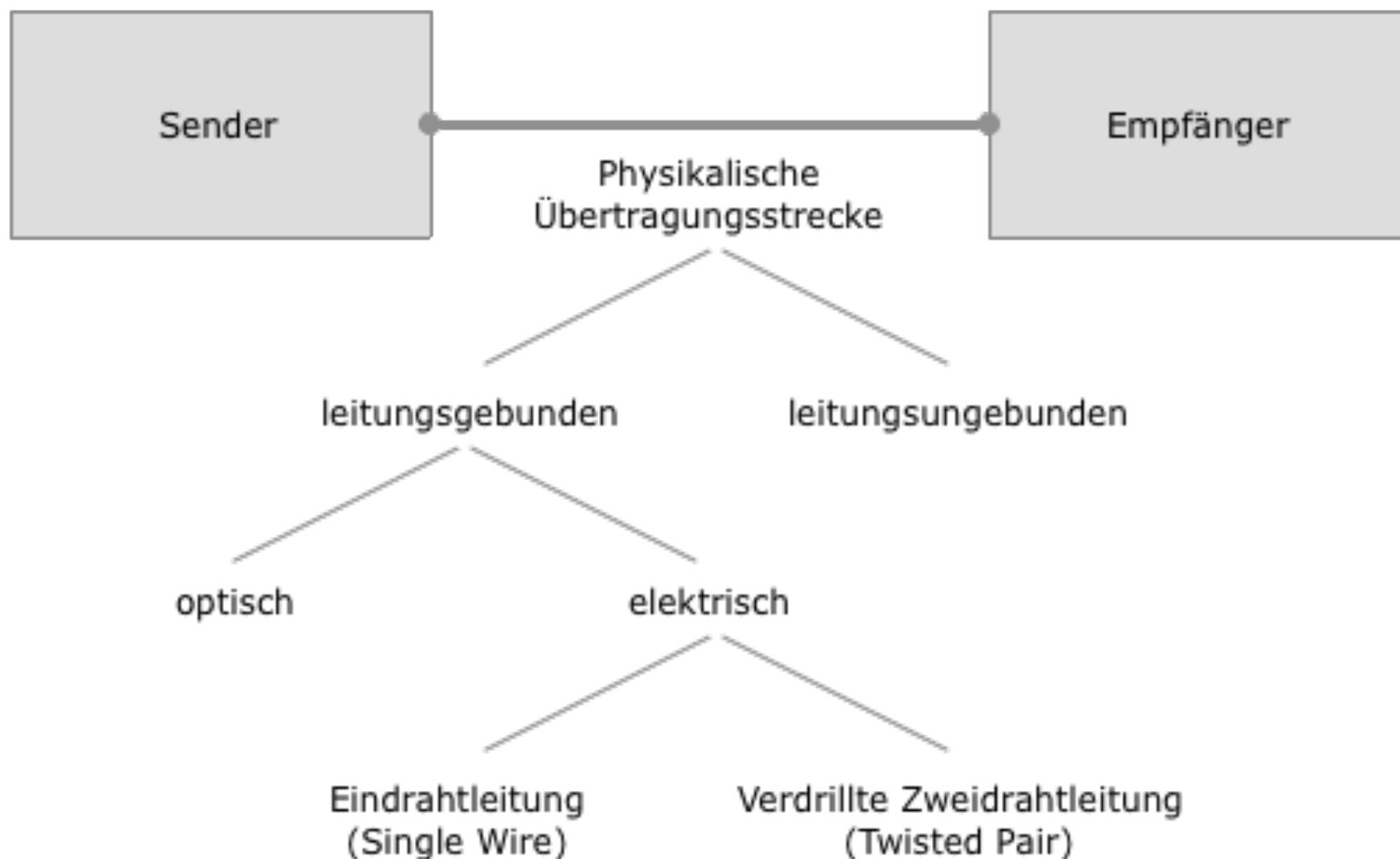


- 1 Einführung
- **2 Architektur serieller Bussysteme**
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - **2.7 Physikalische Übertragung**
- 3 Serielle Bussysteme im Kfz
 - 3.1 CAN
 - 3.2 LIN
 - 3.3: FlexRay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



Architektur serieller Bussysteme

Physikalische Übertragungsstrecke

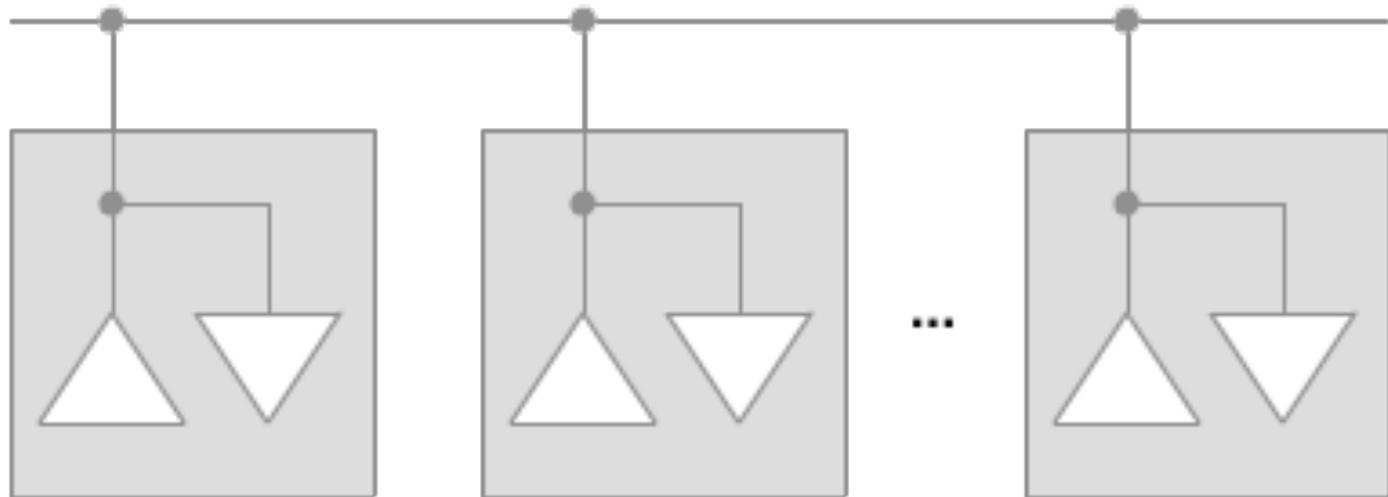




Architektur serieller Bussysteme

Asymmetrische Signalübertragung

Signalleitung



Sender/
Empfänger

Sender/
Empfänger

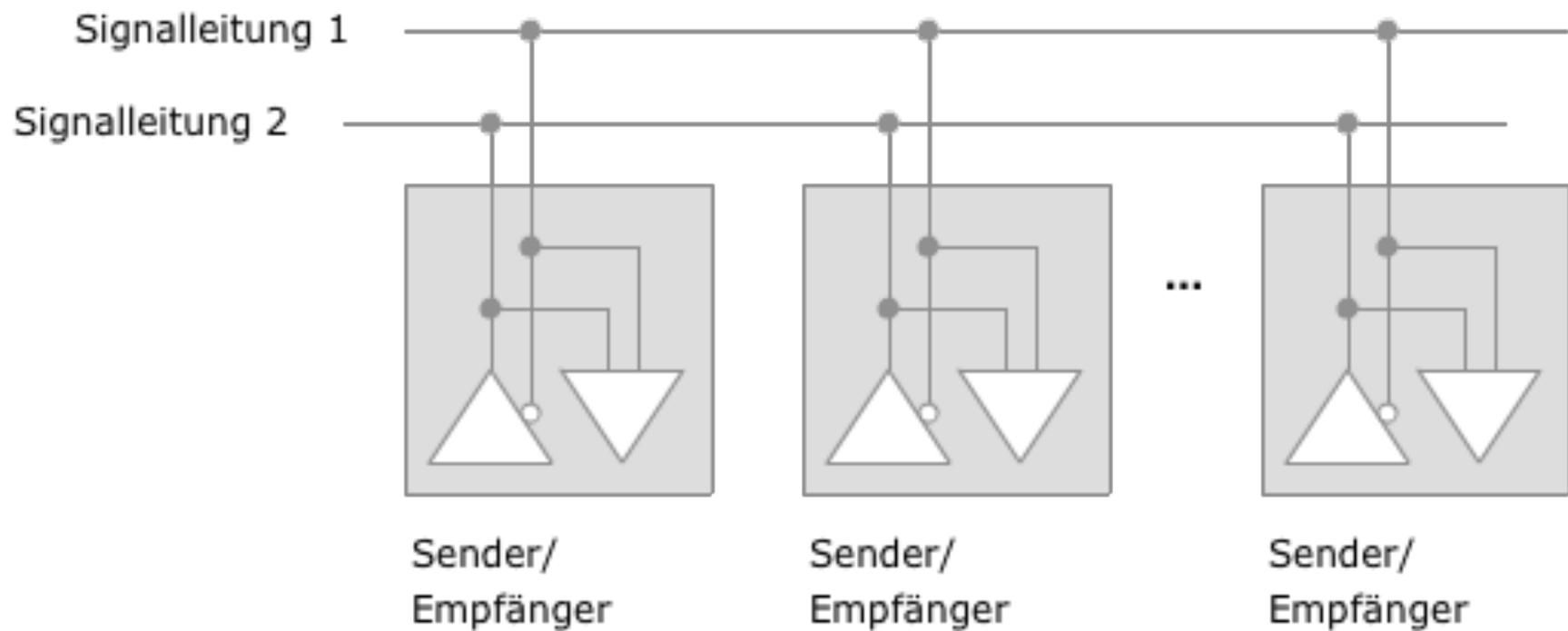
Sender/
Empfänger





Architektur serieller Bussysteme

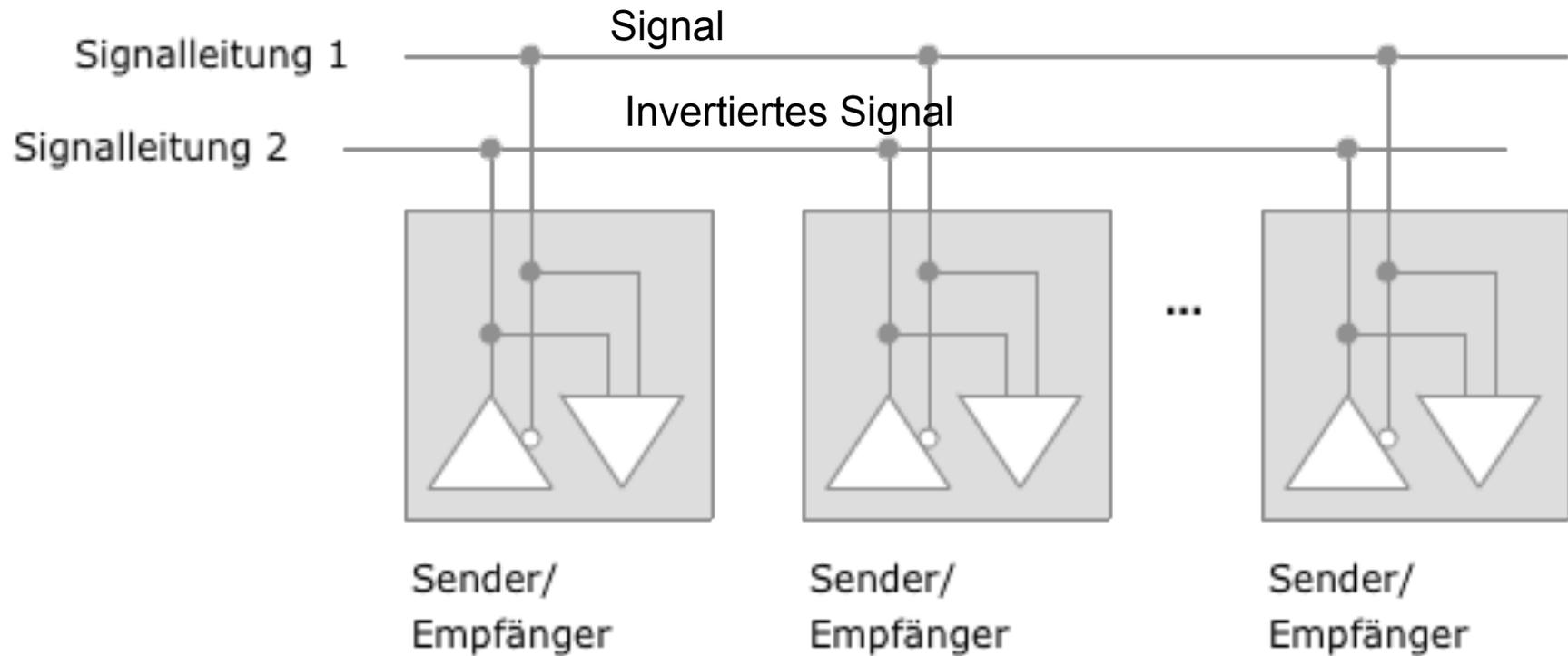
Symmetrische Signalübertragung





Architektur serieller Bussysteme

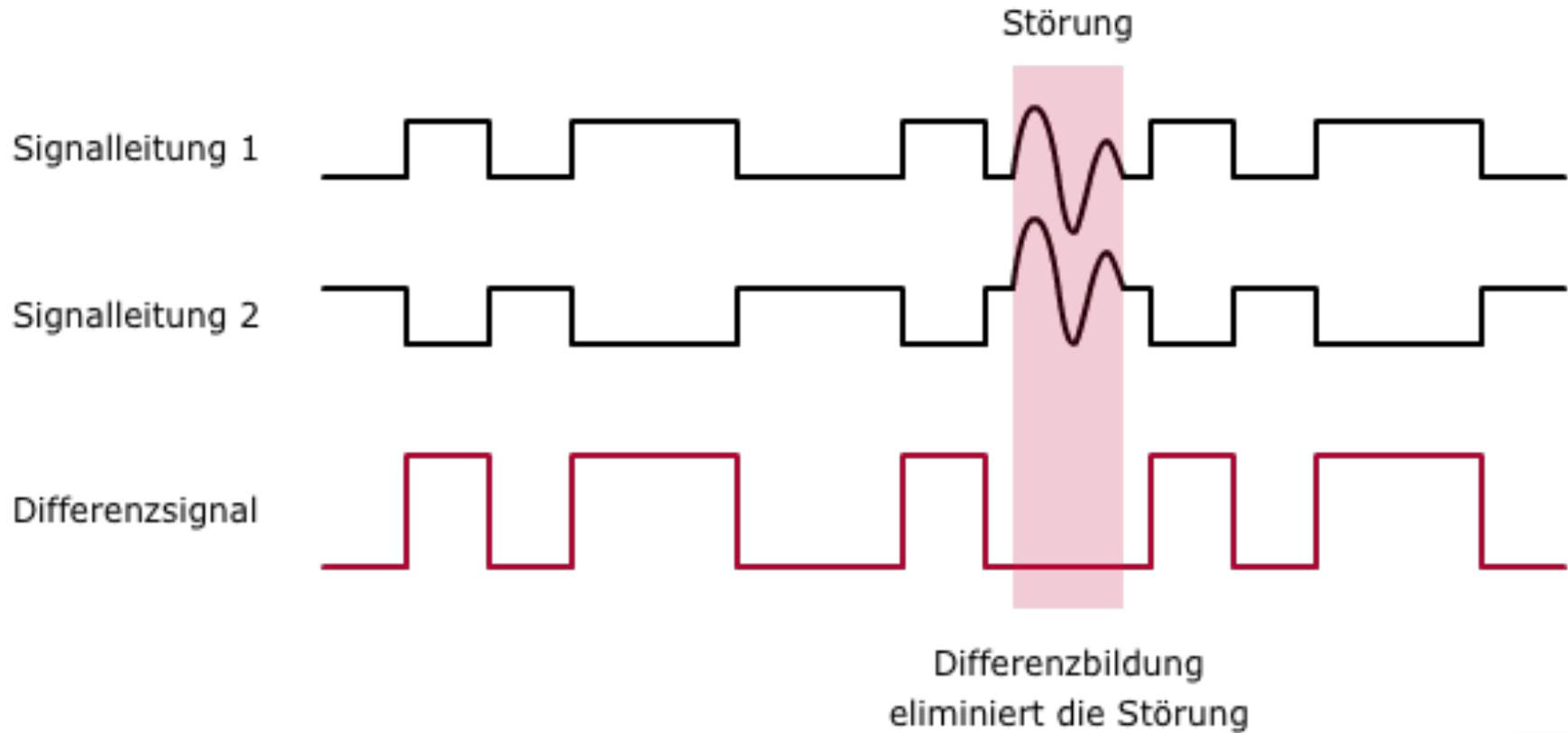
Symmetrische Signalübertragung





Architektur serieller Bussysteme

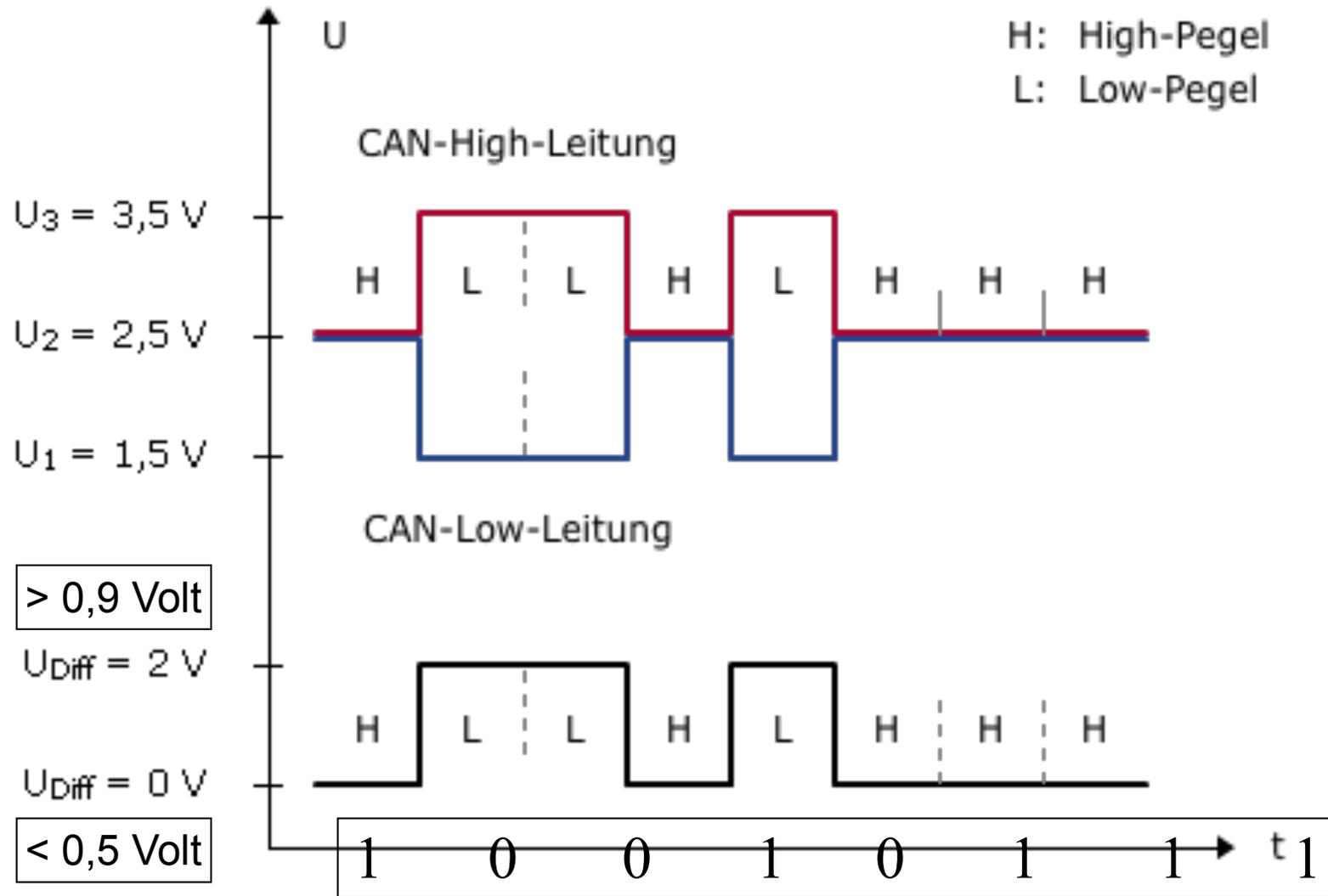
Differenzbildung





Architektur serieller Bussysteme

CAN-Signalspezifikation (High-Speed)





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

Klasse	Übertragungsraten	Anwendung	Vertreter
Klasse A	Geringe Datenraten (bis 10 kBit/s)	Vernetzung von Aktoren und Sensoren	LIN
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 (> 10 MBit/s)	Vernetzung von Steuergeräten im Telematik- und Multimediabereich	MOST

Quelle: BOSCH: Kraftfahrtechnisches Taschenbuch, Vieweg+Teubner, 27. Auflage, 2011.

Klasse	Übertragungsraten	Anwendung	Vertreter
Klasse A	Geringe Datenraten (bis 10 kBit/s)	Vernetzung von Aktoren und Sensoren	LIN
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 (> 10 MBit/s)	Vernetzung von Steuergeräten im Telematik- und Multimediabereich Verliert an Bedeutung	MOST
		in Einführung	Ethernet

Quelle: BOSCH: Kraftfahrtechnisches Taschenbuch, Vieweg+Teubner, 27. Auflage, 2011.

Gelbe Felder ergänzt

Vergleich: Bussysteme im Kfz und Funknetze



Übertragungsrate	Übertragungssicherheit	Übertragungstrecke	Bussysteme	Funknetze
Gering	Gering	Kurz		
Mittel - hoch	Hoch	Mittel		

Vergleich: Bussysteme im Kfz und Funknetze

Übertragungsrate	Übertragungssicherheit	Übertragungstrecke	Bussysteme	Funknetze
Gering	Gering	Kurz		
Mittel - hoch	Hoch	Mittel		

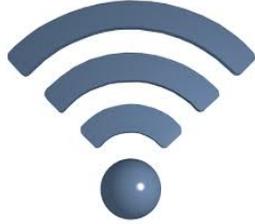
Vergleich: Bussysteme im Kfz und Funknetze

Übertragungsrate	Übertragungssicherheit	Übertragungstrecke	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

Übertragungsrate	Übertragungssicherheit	Übertragungsstrecke	Bussysteme	Funknetze
Gering	Gering	Kurz		
Mittel - hoch	Hoch	Mittel		

Bussysteme im KFZ - Überblick

Quelle: 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: > 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: > 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: > Datenübertragung bis zu 10Mbit/s > redundante Ausführung aller Netzteile	Kfz-Technik (z.B. sicherheitsrelevante Anwendungen, Brake-by-Wire)

Weitere Informationen: [Overview of Current Automotive Protocols \(Vector Informatik GmbH\)](#)

CAN (Controller Area Network)



- CAN-Standard
 - CAN-Netzwerk
 - CAN-Datenübertragung
 - CAN-Kommunikationsprinzip
 - CAN-Datensicherung
-
- Material: E-Learning Plattform Vector Informatik GmbH

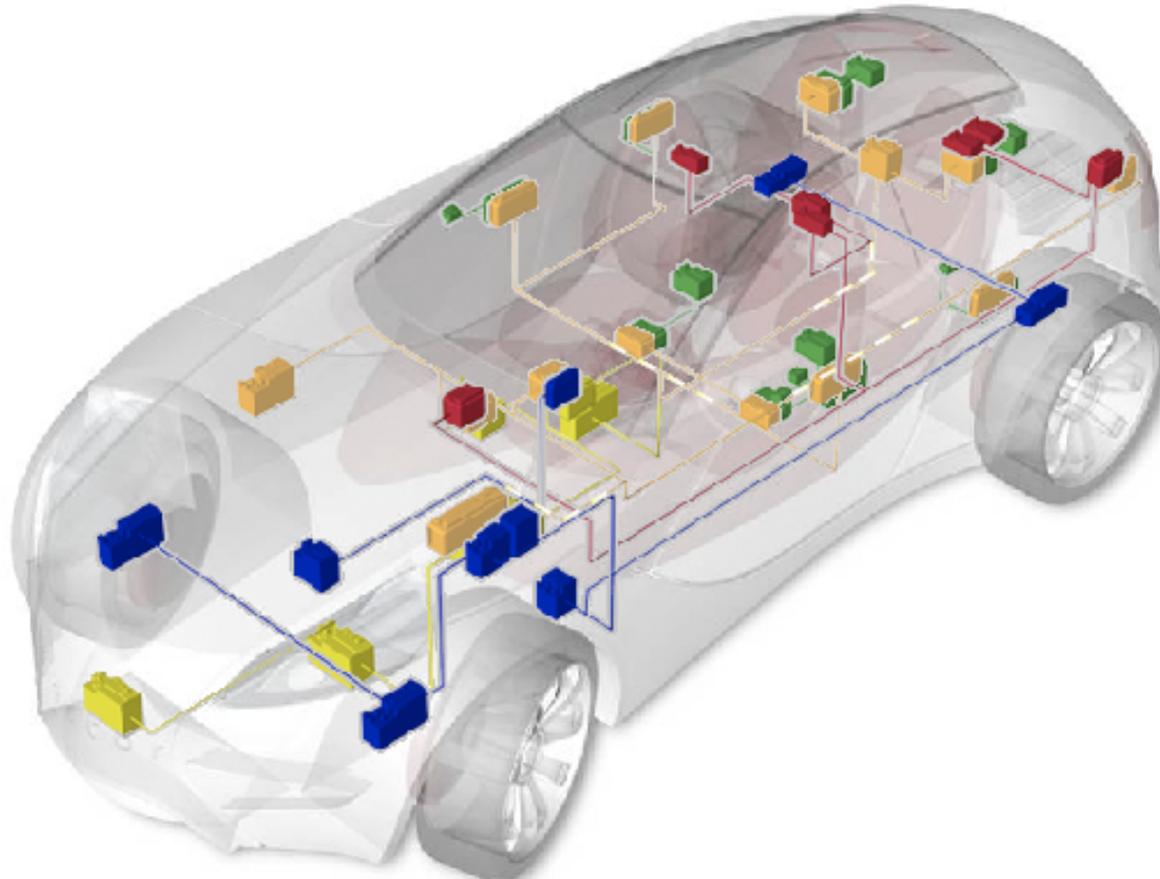
CAN Lowspeed Bus Innenraum- oder Karosserie-CAN



Elektronikvernetzung im Kfz

✕ schließen

- Antriebsstrang (Powertrain)
- Fahrwerk (Chassis)
- Karosserie (Body)
- Multi-Media (Telematics)



■ CAN H

■ **CAN L**

■ LIN

■ FlexRay

■ MOST

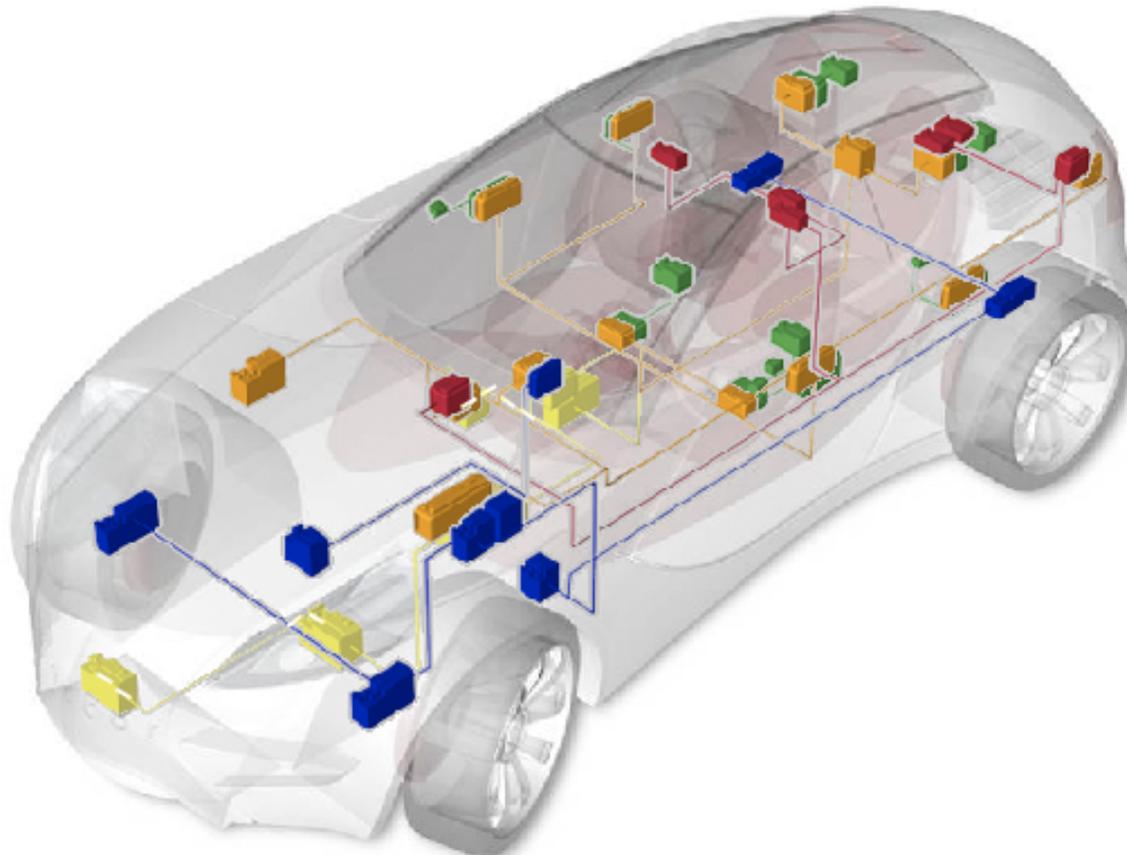
CAN Lowspeed Bus für die Vernetzung der Steuergeräte im Innenraum, Komfortfunktionen z.B. Sitzsteuergerät.



Elektronikvernetzung im Kfz



- Antriebsstrang (Powertrain)
- Fahrwerk (Chassis)
- Karosserie (Body)
- Multi-Media (Telematics)



■ **CAN H** ■ CAN L ■ LIN ■ FlexRay ■ MOST

CAN Highspeed Bus für die Vernetzung der Steuergeräte im Motorraum, z.B.ESP-Steuergerät.

Protokolle und Bussysteme: CAN (Controller Area Network)



- Entwickelt von Bosch und Intel 1981
- ISO/OSI-Standard
- Eigenschaften:
 - Bandbreite bis 125 kbit/s (Low-speed-CAN, Karosserie-CAN, Body-CAN)
 - Bandbreite bis 1000 kbit/s (High-speed-CAN, Motor-CAN, Powertrain-CAN)
 - Asynchroner Betrieb ohne Busmaster mit Arbitrierung
 - Variante für Synchronbetrieb: TTCAN (time triggered CAN)
 - Physical Layer: Bus, twisted pair
 - Fehlererkennung: CRC-16, d=6
- Große Zahl von Systemkomponenten verfügbar (Transceiver, Controller)
 - Deswegen auch Einsatz in anderen Bereichen, z. B. Luftfahrt, Eisenbahnen, Automatisierungstechnik, Rüstungsindustrie
- Geeignet für allgemeine, nichtkritische Anwendungen (Karosserie, Komfortsysteme)
- Nicht geeignet für Multimedia, x-by-Wire, Sicherheitssysteme

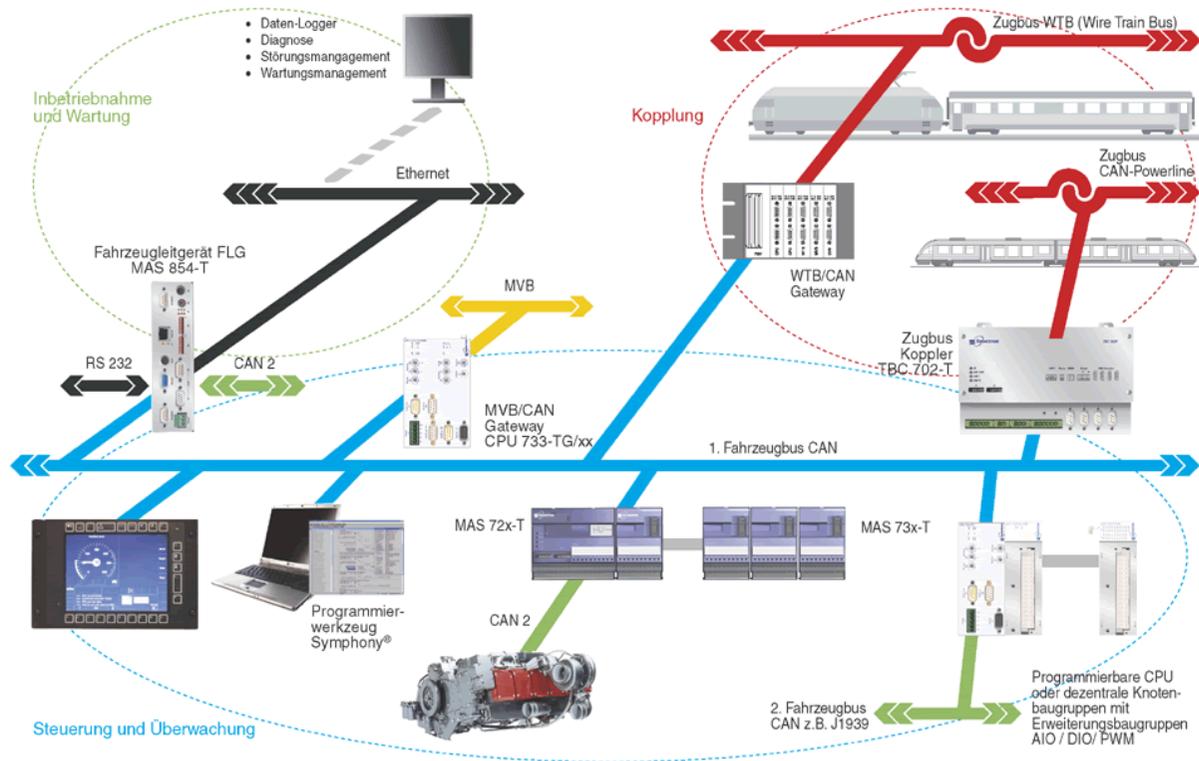
- <http://www.can-cia.org/>



Applikationsbeispiel 3 - Schienenfahrzeuge



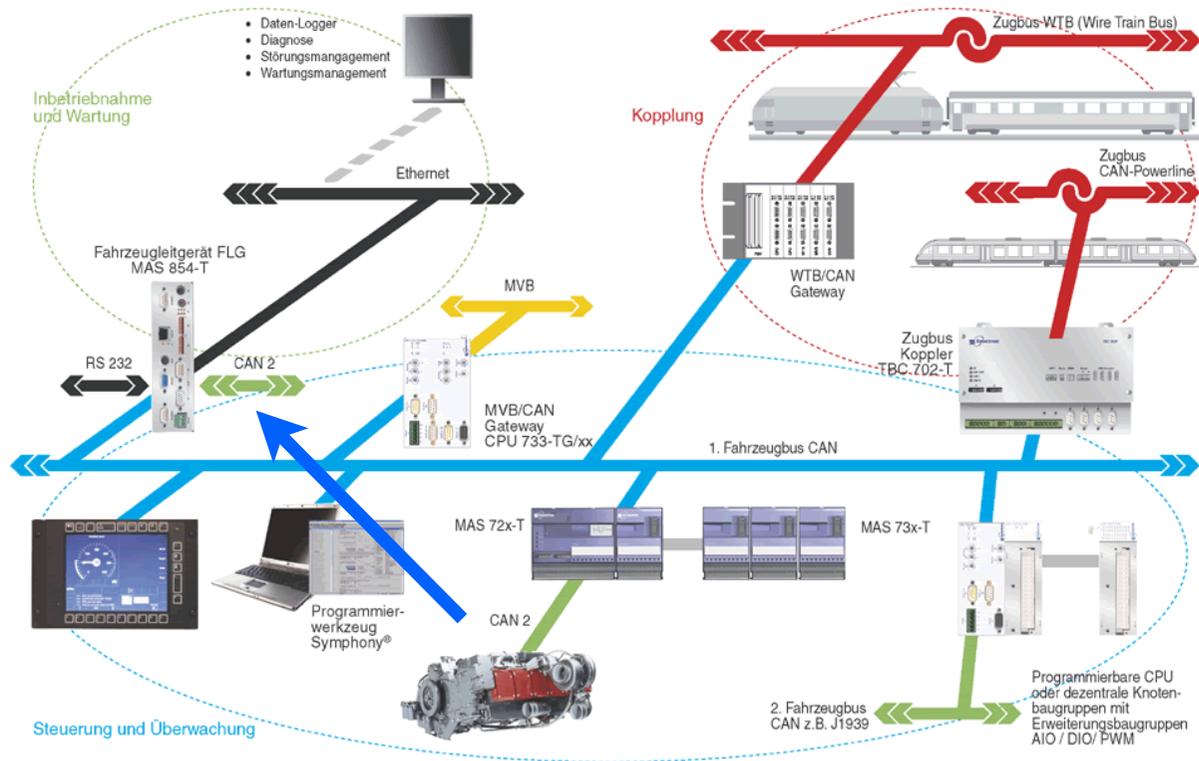
TCMS (Train Control and Monitoring System)



Applikationsbeispiel 3 - Schienenfahrzeuge



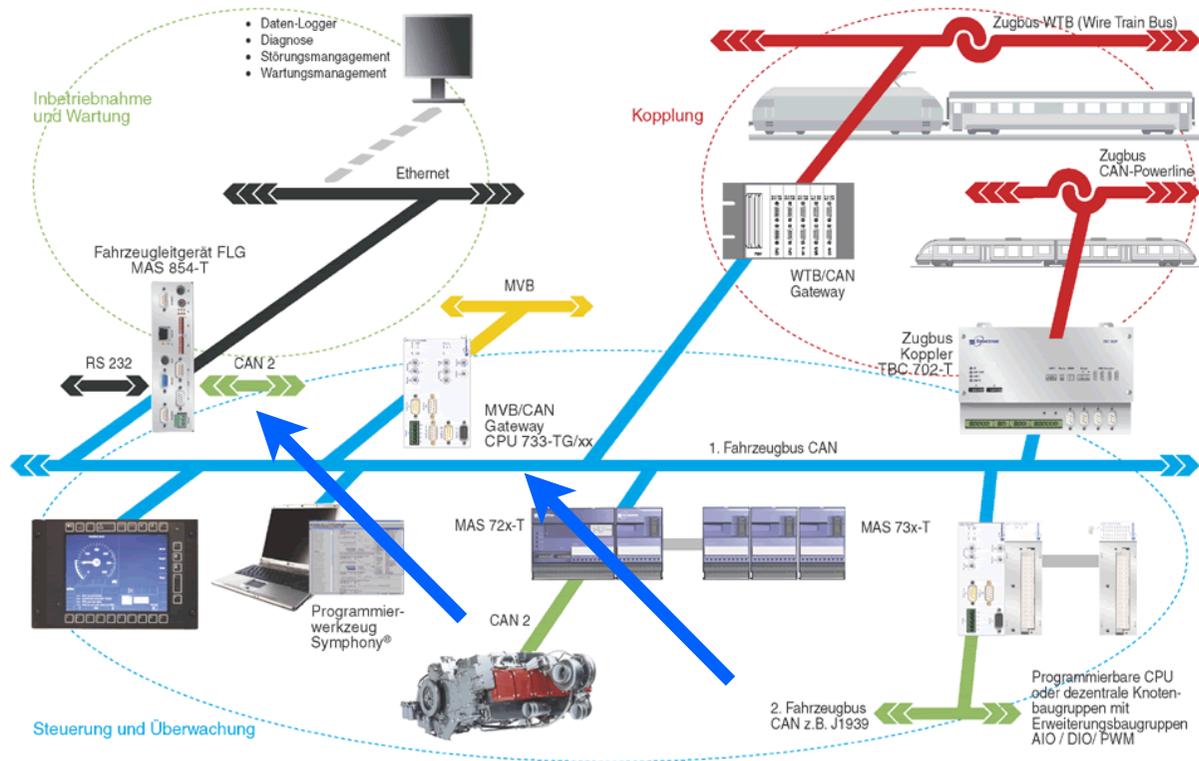
TCMS (Train Control and Monitoring System)



Applikationsbeispiel 3 - Schienenfahrzeuge



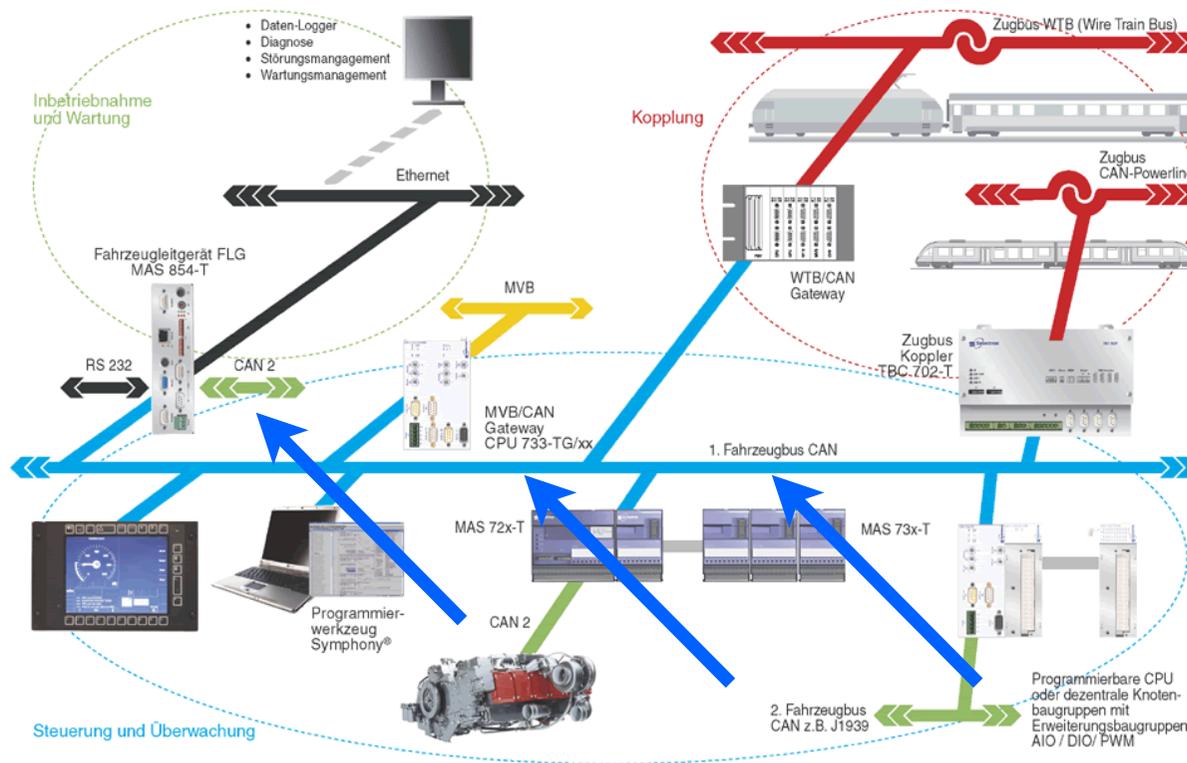
■ TCMS (Train Control and Monitoring System)



Applikationsbeispiel 3 - Schienenfahrzeuge



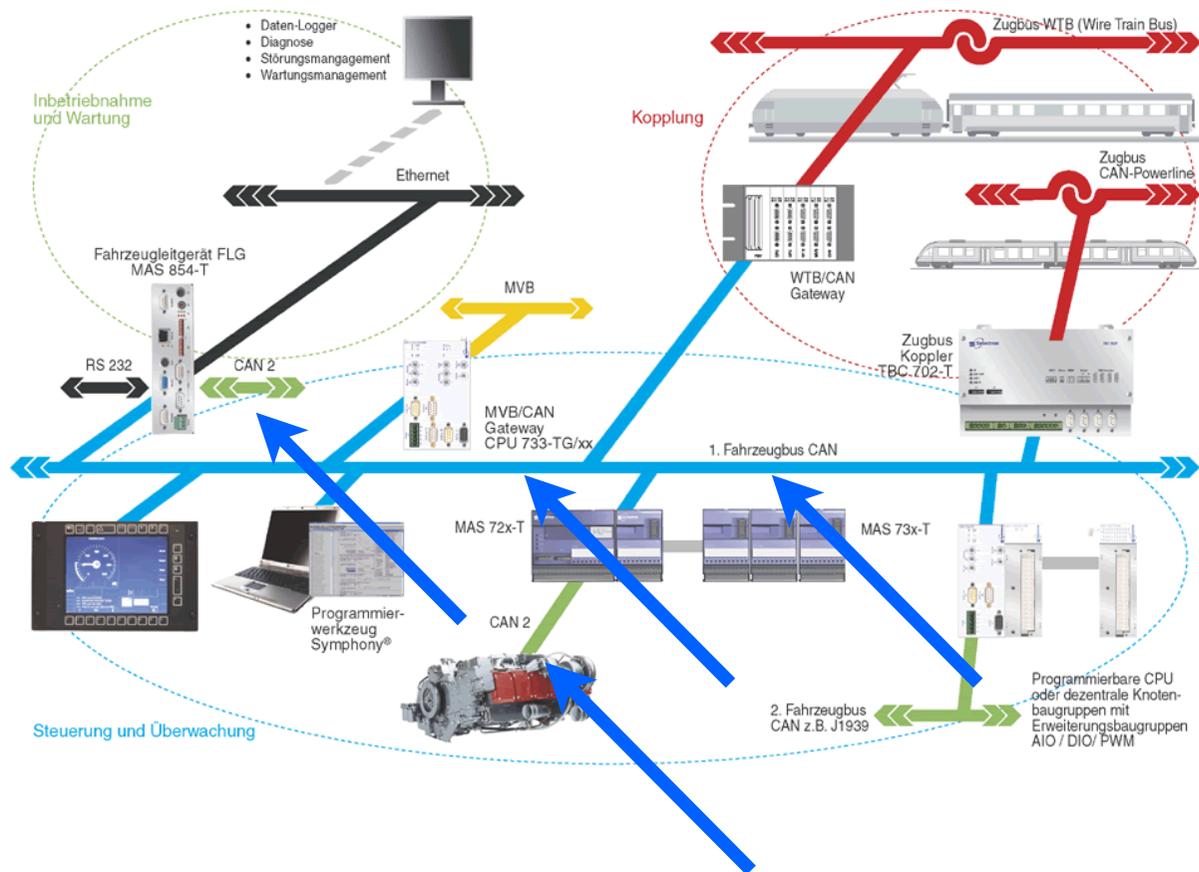
TCMS (Train Control and Monitoring System)



Applikationsbeispiel 3 - Schienenfahrzeuge



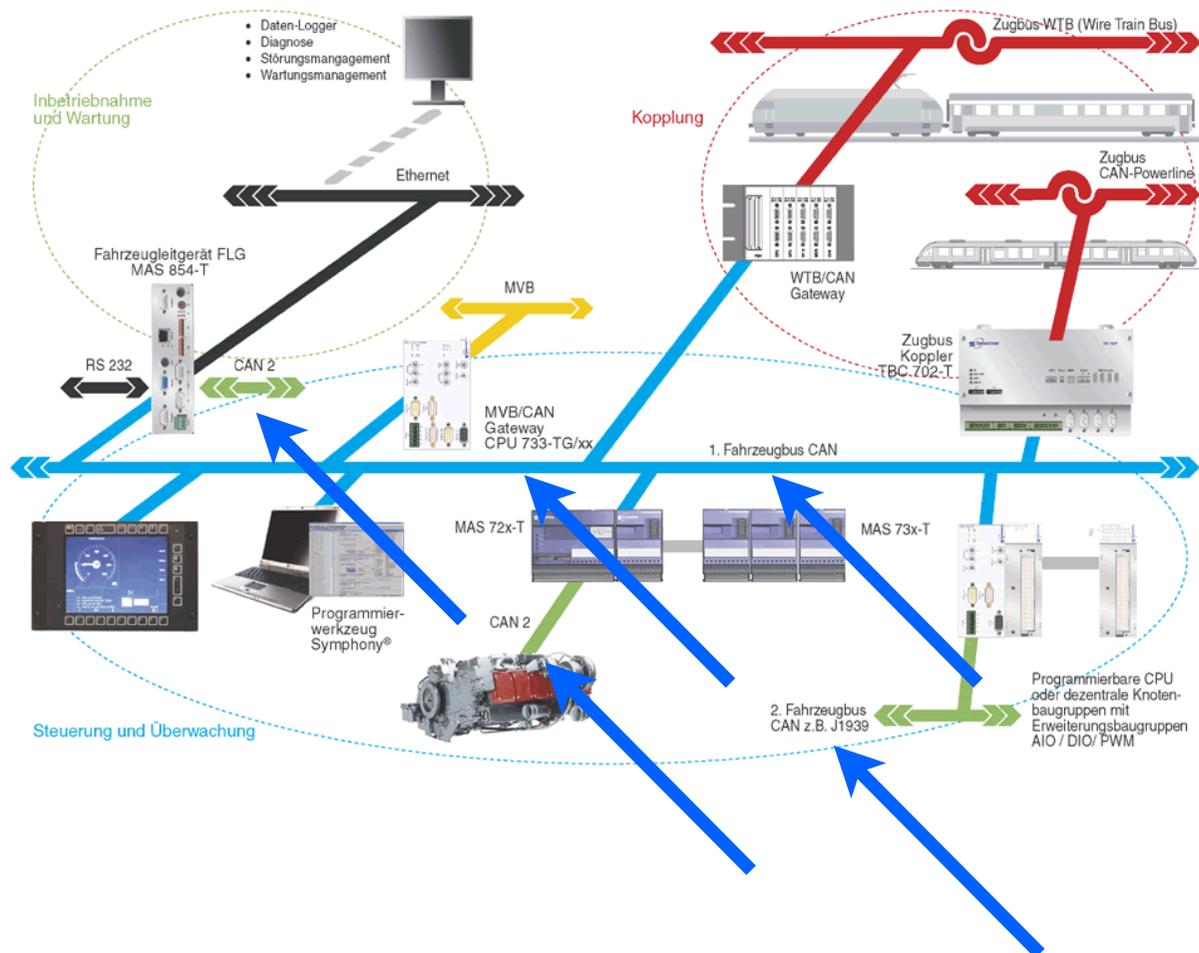
■ TCMS (Train Control and Monitoring System)



Applikationsbeispiel 3 - Schienenfahrzeuge



■ TCMS (Train Control and Monitoring System)

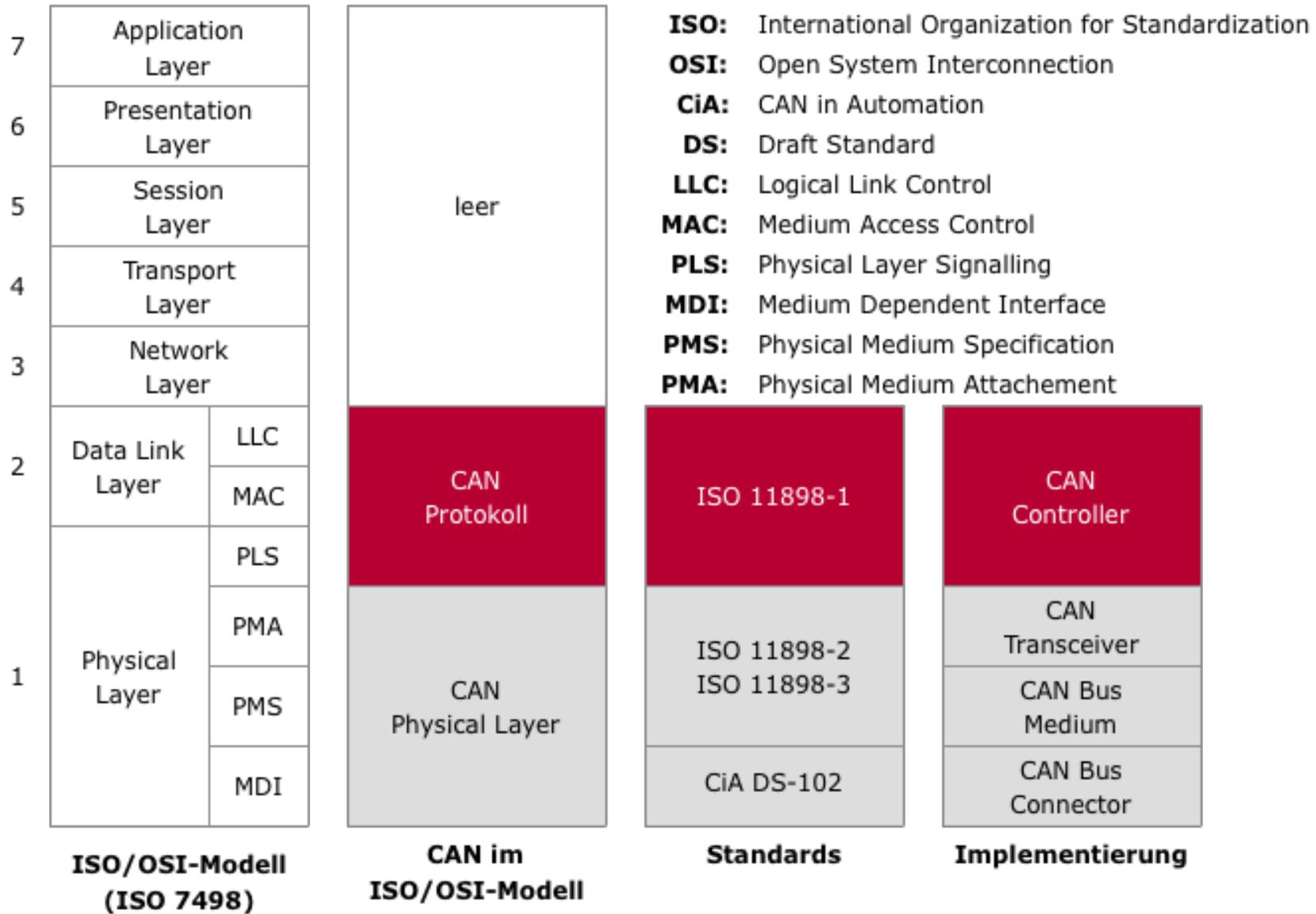


- Seit 1993 ISO 11898
- High-Speed-Variante (Datenraten bis 1 MBit/s) für Antrieb und Fahrwerk
- Low-Speed-Variante (Datenraten bis 125 KBit/s) für Innenraum / Komfort
- Ursprünglich drei, jetzt fünf Teile
- Teil 1: Beschreibung des ereignisorientierten Kommunikationsprotokolls.
- Teil 2: Busankopplung und Physikalische Datenübertragung High-Speed-Variante (Datenraten bis 1 MBit/s)
- Teil 3: Busankopplung und Physikalische Datenübertragung Low-Speed-Variante (Datenraten bis 125 KBit/s).
- Teil 4: Zeitgesteuerte Erweiterung TTCAN (time triggered CAN).
- Teil 5: Verhalten eines CAN-Knotens im High-Speed-Netzwerk im „Low Power Mode“.
- Einordnung der ersten drei Teile der ISO-Norm 11898 (CAN-Standard) in das ISO-OSI-Modell (Open System Interconnection) siehe nächste Folie



Serielle Bussysteme im Kfz

CAN-Standard

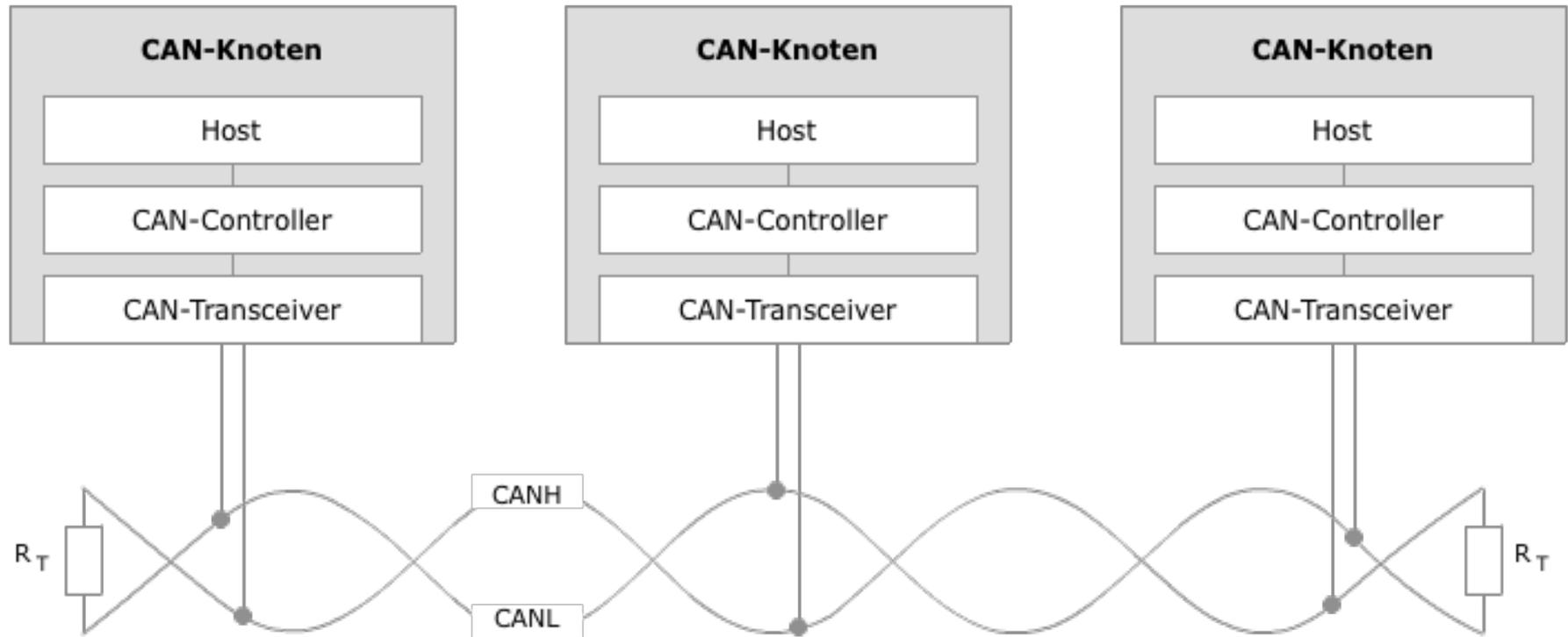


- CAN-Netzwerk:
Systemverbund aus CAN-Knoten (elektronische Steuergeräte mit CAN-Schnittstelle)
- Datenaustausch über die jeweiligen CAN-Schnittstellen, bestehend aus
 - Kommunikationssoftware
 - Kommunikationshardware.
- und einem alle CAN-Schnittstellen verbindendes Übertragungsmedium (CAN-Bus)
- CAN-Controller: Einheitliche Abwicklung des CAN-Kommunikationsprotokolls
- CAN-Transceiver: Ankopplung des CAN-Controllers an den CAN-Bus.
- Übertragungsmedium: Verdrillte Zweidrahtleitung (Twisted Pair)
- Symmetrische Signalübertragung, sehr unempfindlich gegenüber äußeren Störungen.
- Buspegel abhängig von Busankopplung
 - CAN-High-Speed-Busankopplung (ISO 11898-2)
 - CAN-Low-Speed-Busankopplung (ISO 11898-3).



Serielle Bussysteme im Kfz

CAN-Netzwerk



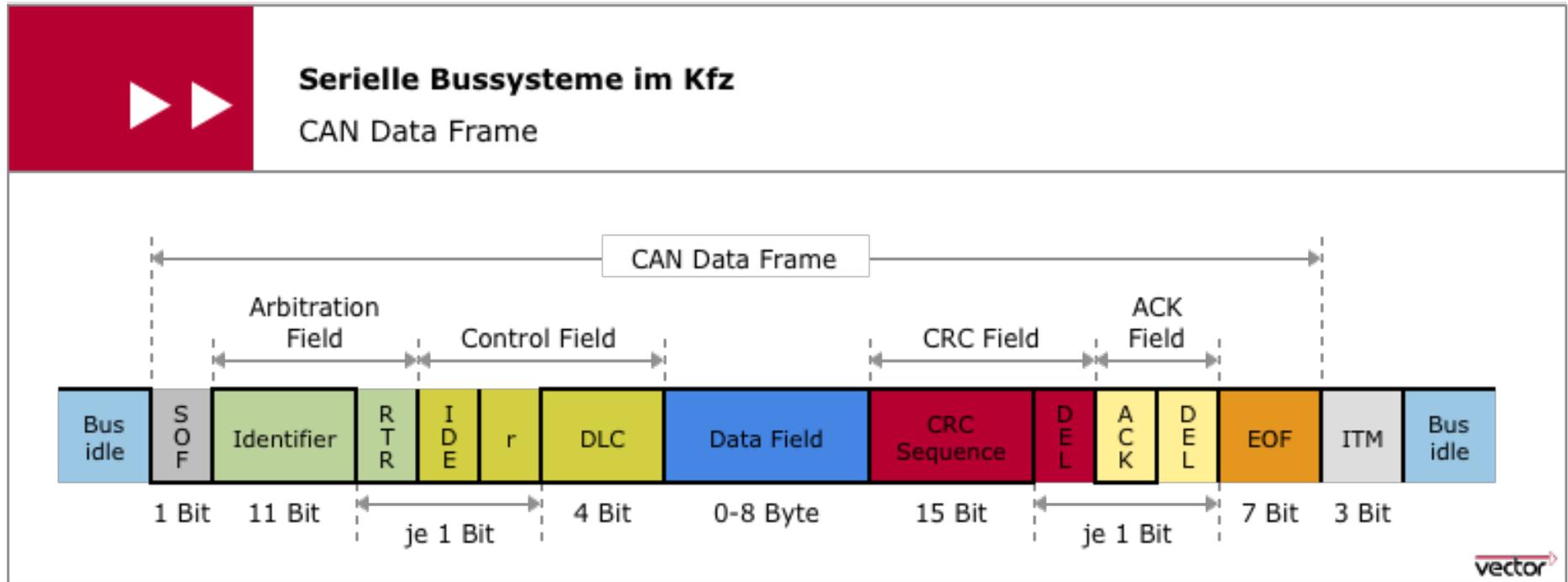
CANH: CAN High Leitung

CANL: CAN Low Leitung

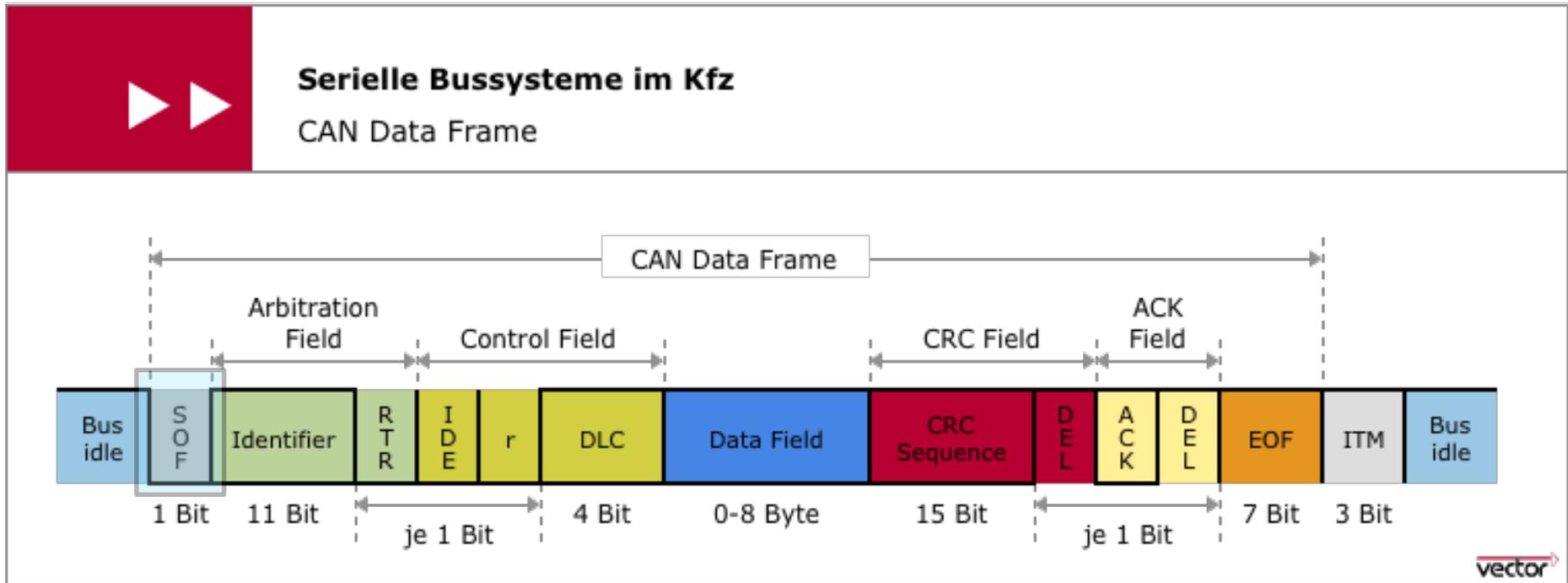
R_T : Terminierung

CAN-Datenübertragung (1)

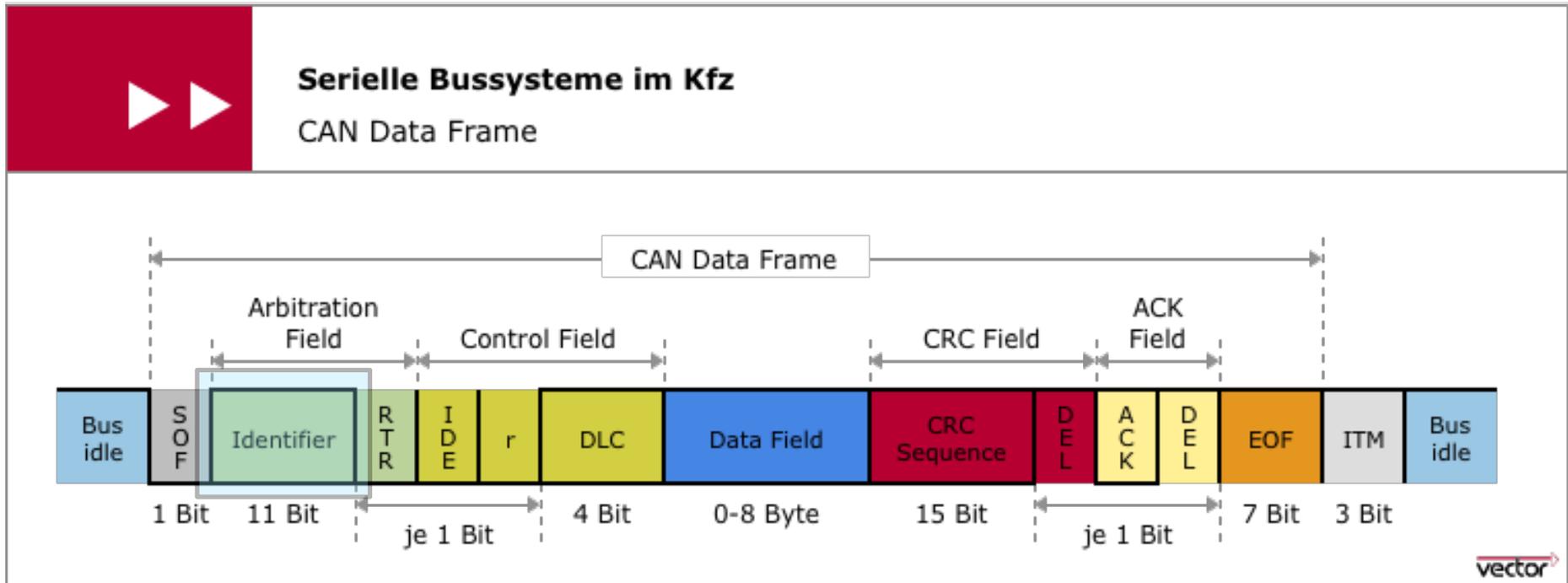
- Datenübertragung mittels Nachrichtenrahmen „CAN Data Frames“.
- Übertragung von Nutzdaten bis zu acht Byte Länge mit Data Frame im Data Field
- DLC (Data Length Code): Genaue Anzahl der Nutzbytes
- Jeder Data Frame steht jedem CAN-Knoten zur Verfügung (Broadcasting).
- Sicherung der Nutzdaten: CRC-Verfahrens (Cyclic Redundancy Check):
 - Der Sender hängt den Nutzdaten eine Prüfsumme (CRC-Sequenz)
 - Auswertung durch Empfänger (CRC-Algorithmus)
 - Je nach Ergebnis positive oder negative Quittung im ACK-Slot.
- Jeder Data Frame wird über einen Identifier gekennzeichnet.
 - Standard-Format 11 Bit
 - Extended-Format 29 Bit.



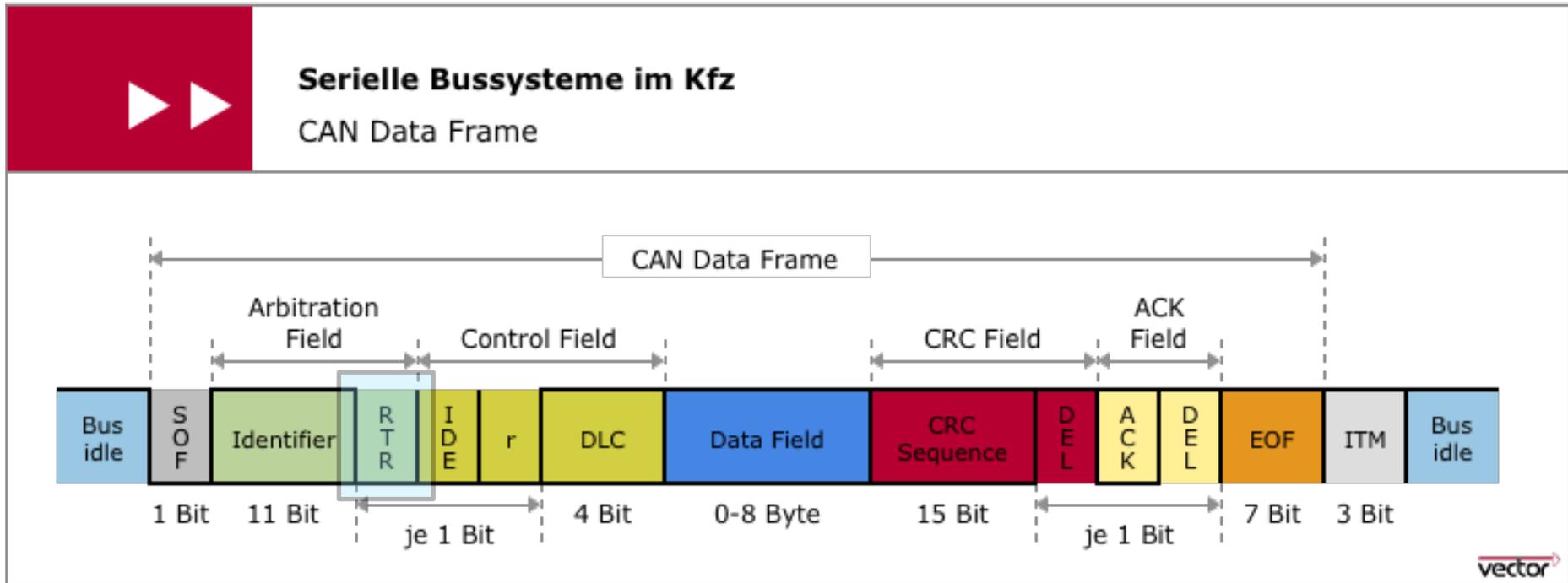
- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- Data Field: maximal acht Nutzbytes
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- ACK-Feld (Acknowledgement).
- Endezeichen (End Of Frame: EOF)



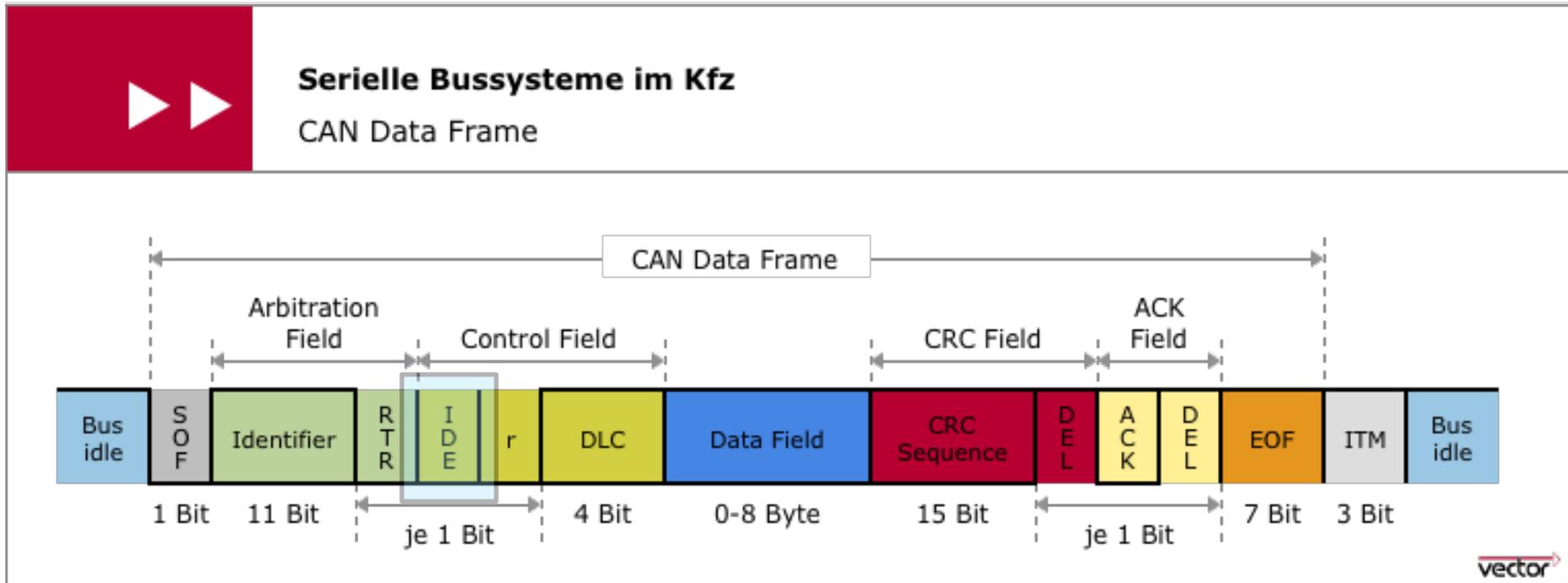
- **Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)**
- Kennung (Identifier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- Data Field: maximal acht Nutzbytes
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- ACK-Field (Acknowledgement).
- Endezeichen (End Of Frame: EOF)



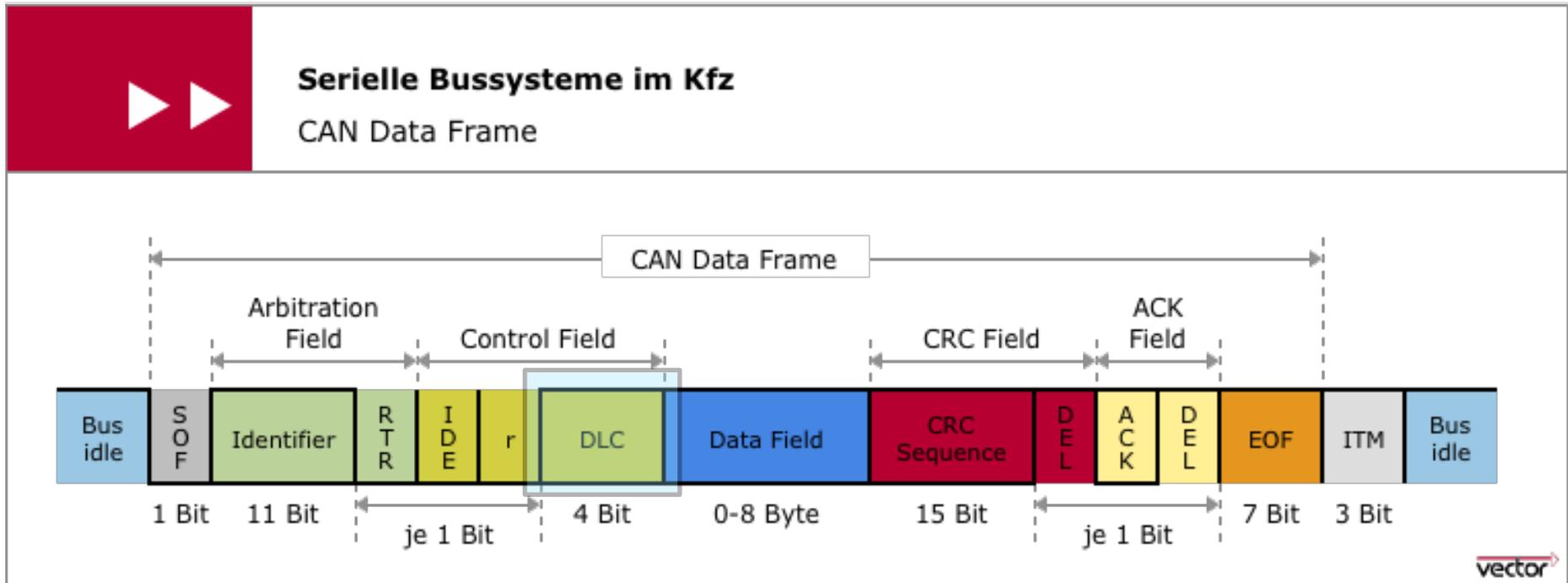
- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- **Kennung (Identifizier) des Frames.**
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- Data Field: maximal acht Nutzbytes
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- ACK-Field (Acknowledgement).
- Endezeichen (End Of Frame: EOF)



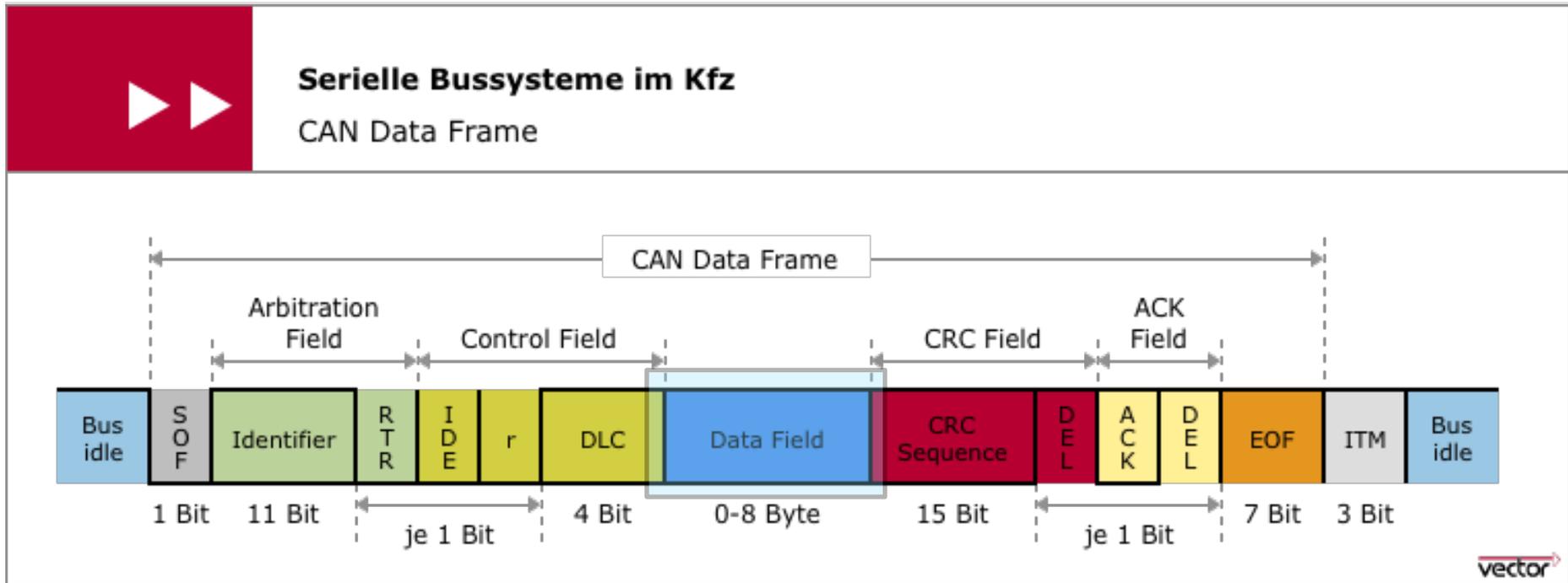
- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- **RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).**
- IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- Data Field: maximal acht Nutzbytes
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- ACK-Field (Acknowledgement).
- Endezeichen (End Of Frame: EOF)



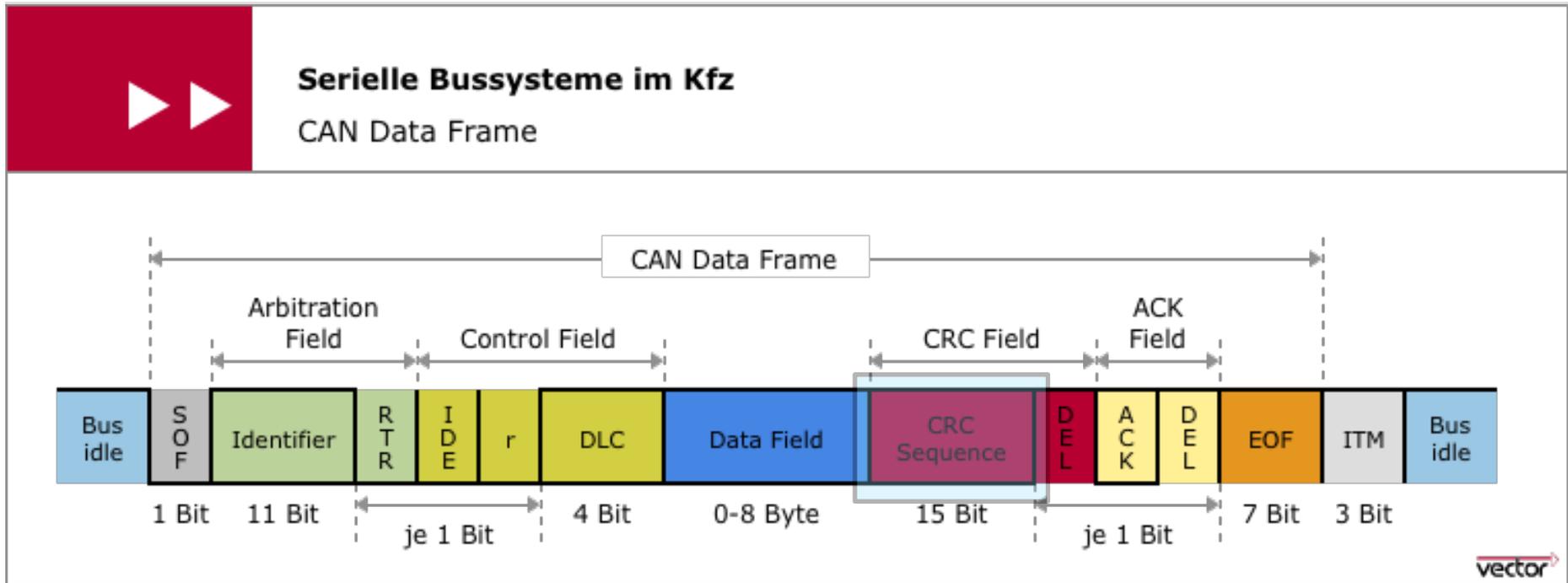
- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- **IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):**
 - Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- Data Field: maximal acht Nutzbytes
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- ACK-Feld (Acknowledgement).
- Endezeichen (End Of Frame: EOF)



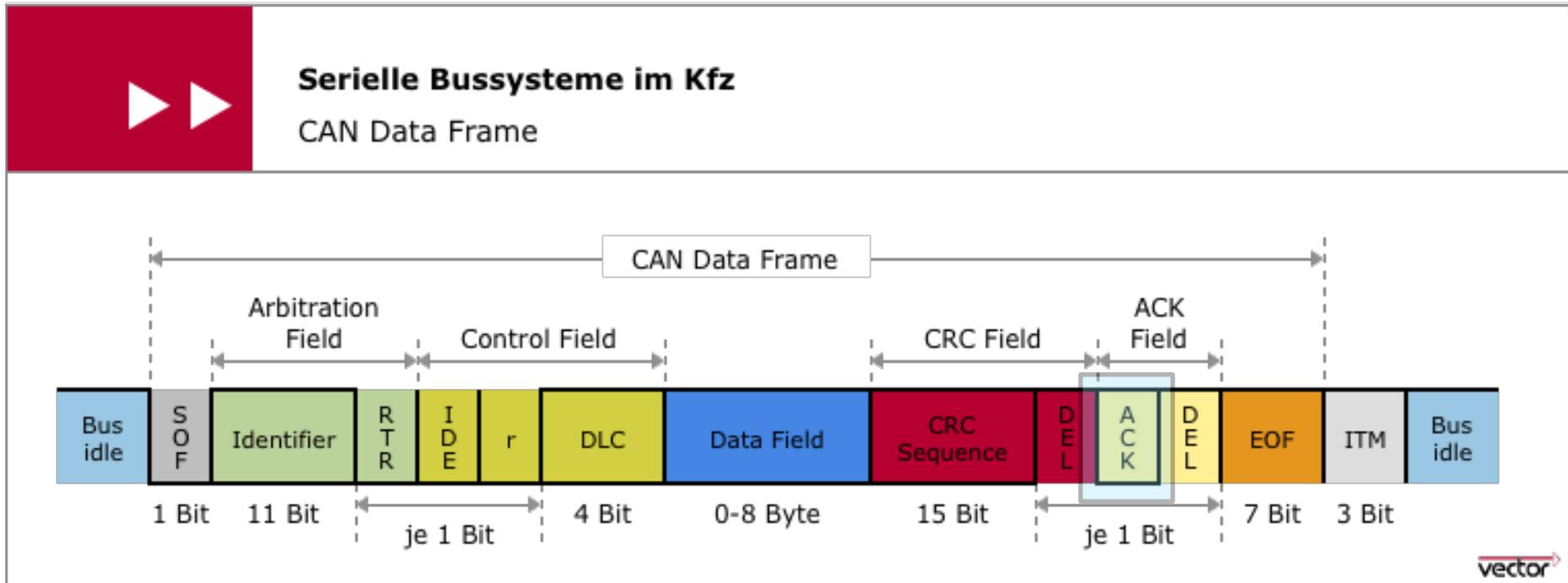
- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.
- **DLC (Data Length Code): Anzahl der Nutzbytes**
- Data Field: maximal acht Nutzbytes
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- ACK-Field (Acknowledgement).
- Endezeichen (End Of Frame: EOF)



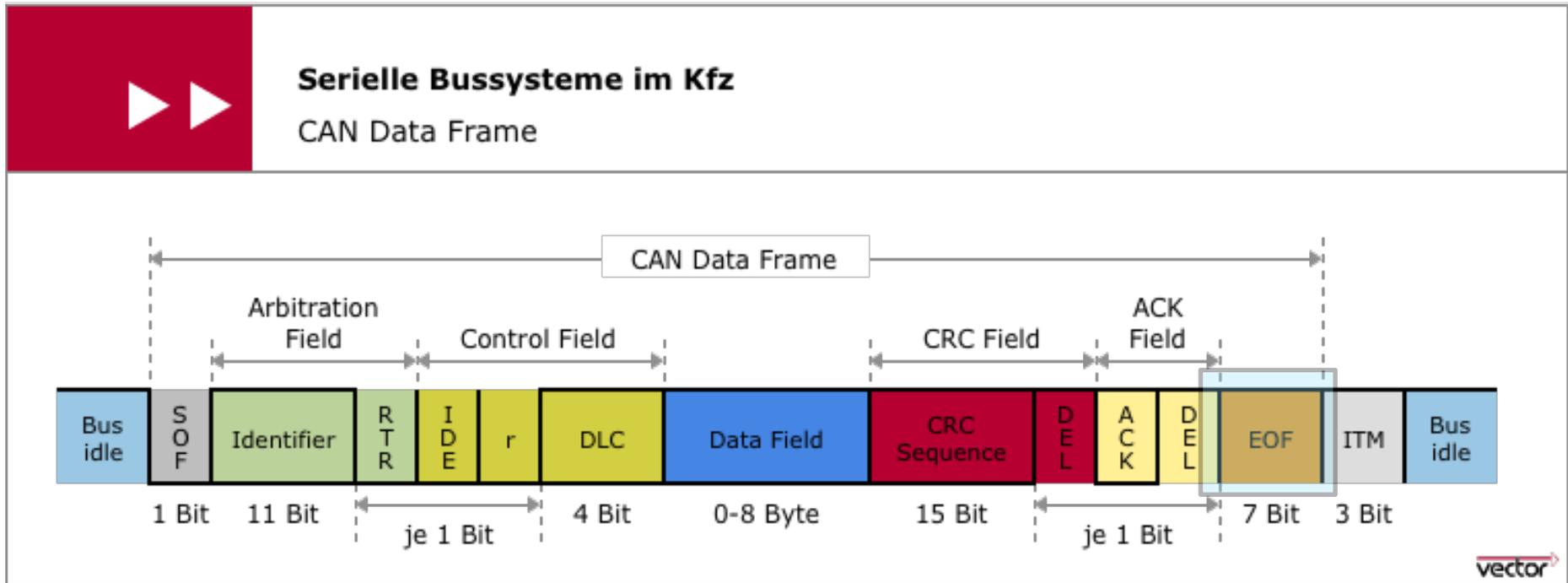
- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- **Data Field: maximal acht Nutzbytes**
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- ACK-Feld (Acknowledgement).
- Endezeichen (End Of Frame: EOF)



- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- Data Field: maximal acht Nutzbytes
- **CRC-Sequenz (Sicherung von Informationen und Nutzdaten).**
- ACK-Feld (Acknowledgement).
- Endezeichen (End Of Frame: EOF)



- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- Data Field: maximal acht Nutzbytes
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- **ACK-Field (Acknowledgement).**
- Endezeichen (End Of Frame: EOF)



- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.
- DLC (Data Length Code): Anzahl der Nutzbytes
- Data Field: maximal acht Nutzbytes
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).
- ACK-Feld (Acknowledgement).
- **Endezeichen (End Of Frame: EOF)**

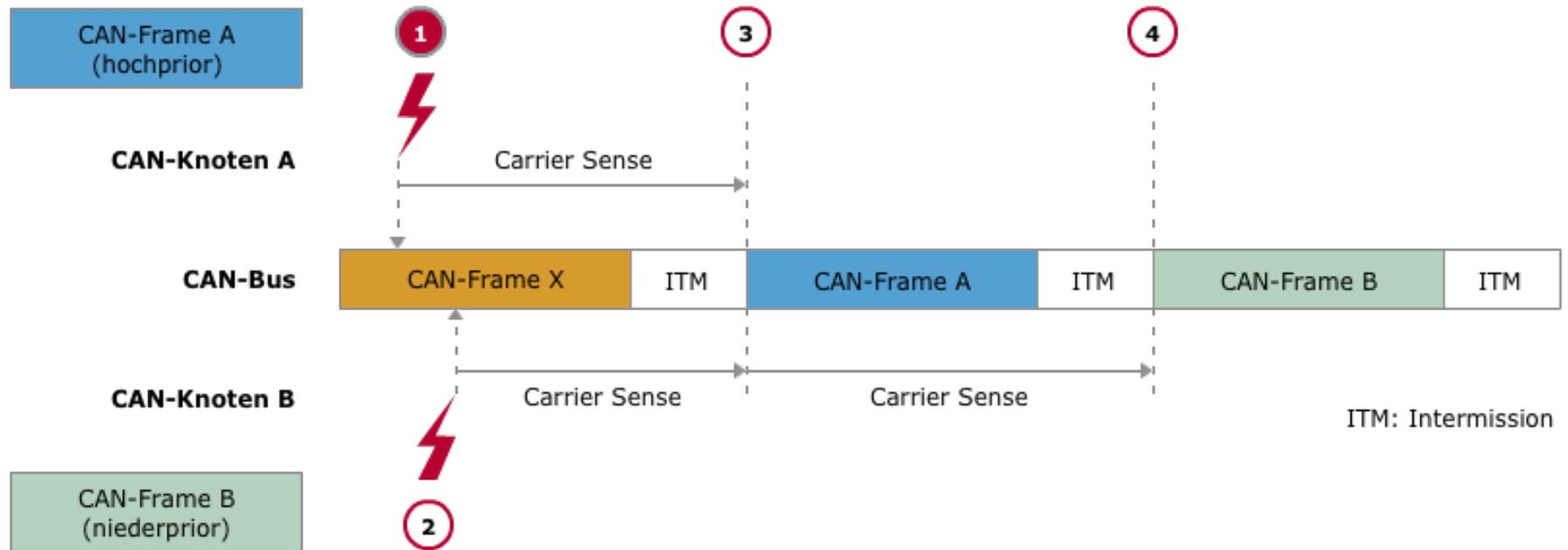
- Ereignisabhängige Kommunikation
- Jeder CAN-Knoten kann prinzipiell jederzeit auf den CAN-Bus zuzugreifen.
- Kurze Nachrichtenlänge (maximal 130 Bit im Standard-Format)
- Hohe Datenübertragungsrate (bis zu 1 MBit/s)
- Ergebnis: Schnelle Reaktion auf asynchrone Vorgänge
 - Echtzeitfähige Datenübertragung im unteren Millisekundenbereich (Applikationen des Antriebs und des Fahrwerks)
- Aber: Nachrichtenverkehr ergibt sich stets erst zur Laufzeit
 - Gefahr von Kollisionen.
 - Steigt mit zunehmender Buslast und stellt die Echtzeitfähigkeit in Frage.
- Buszugriff mit CSMA/CA-Verfahren (Carrier Sense Multiple Access/Collision Avoidance)
 - Ein sendewilliger CAN-Knoten kann erst dann auf den CAN-Bus zugreifen, wenn dieser frei ist.
 - Bei simultanen Buszugriffen setzt sich immer der CAN-Knoten mit dem höchst prioren Data Frame durch.
 - Mit zunehmender Buslast wachsen die Verzögerungen niederpriorer Data Frames an (Beeinträchtigung der Echtzeitfähigkeit).

- https://vector.com/index.php?wbt_ls_kapitel_id=450808&root=376493&seite=vl_can_introduction_de



Serielle Bussysteme im Kfz

Prinzip des Buszugriffs



Ereignis 1

CAN-Knoten A möchte auf den CAN-Bus zugreifen, um seine hochpriorie CAN-Botschaft zu übertragen. Er kann aber nicht, weil der CAN-Bus besetzt ist.

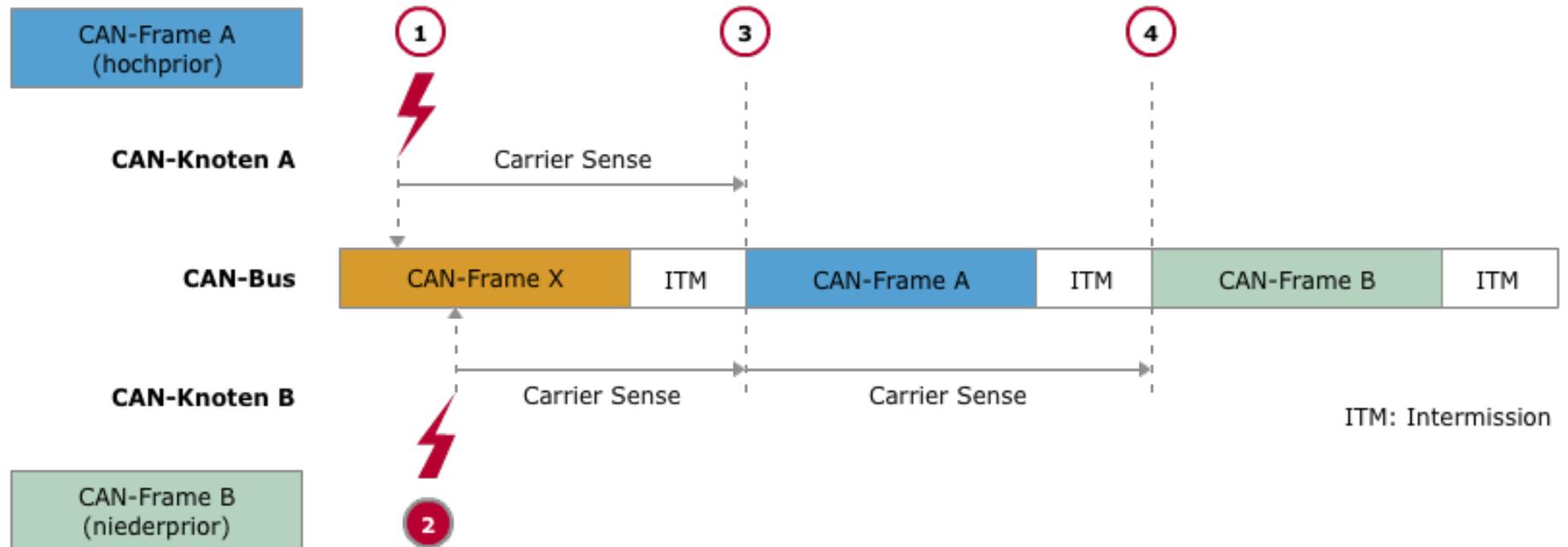
Folge

CAN-Knoten A darf erst dann auf den CAN-Bus zugreifen, wenn dieser frei ist. Dies tritt nach Ablauf des ITM ein. Dazu beobachtet CAN-Knoten A den CAN-Bus (Carrier Sense).



Serielle Bussysteme im Kfz

Prinzip des Buszugriffs



Ereignis 2

CAN-Knoten B möchte auf den CAN-Bus zugreifen, um seine niederpriorie CAN-Botschaft zu übertragen. Er kann aber nicht, weil der CAN-Bus besetzt ist.



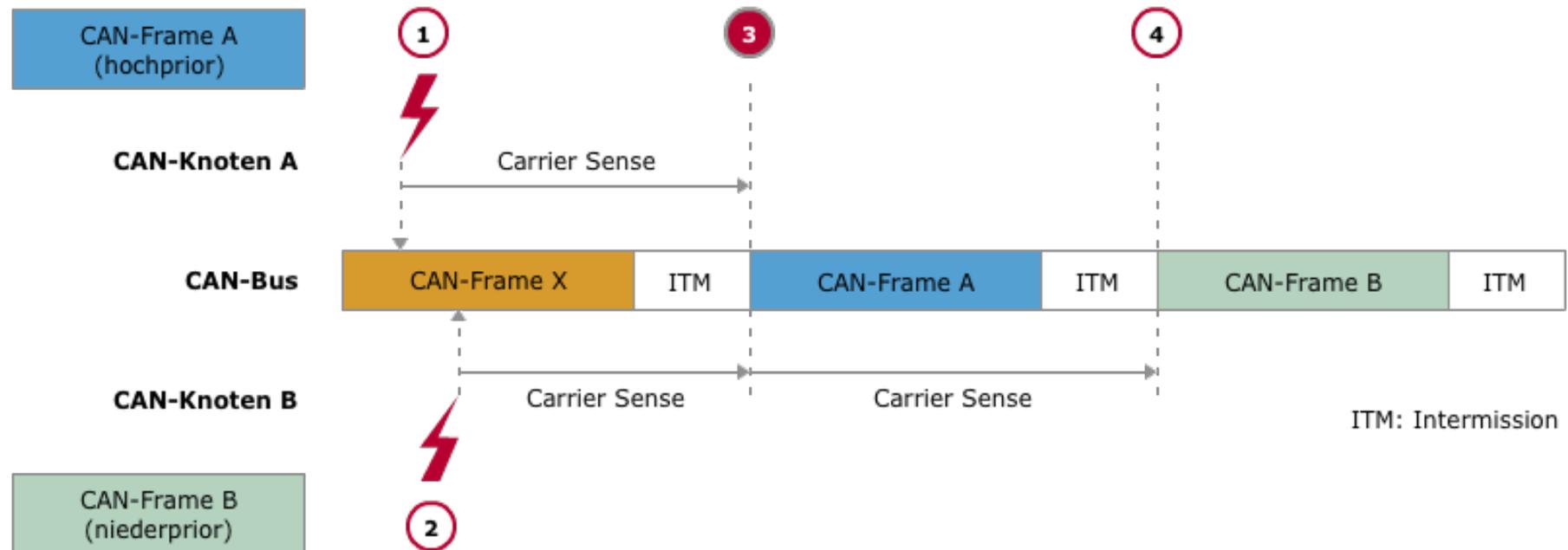
Folge

CAN-Knoten B darf erst dann auf den CAN-Bus zugreifen, wenn dieser frei ist. Dies tritt nach Ablauf des ITM ein. Dazu beobachtet CAN-Knoten B den CAN-Bus (Carrier Sense).



Serielle Bussysteme im Kfz

Prinzip des Buszugriffs



Ereignis 3

Nach Ablauf des ITM ist der CAN-Bus frei. Die sendewilligen CAN-Knoten A und CAN-Knoten B dürfen jetzt auf den CAN-Bus zugreifen. Übertragen wird die CAN-Botschaft A, weil diese im Vergleich zu CAN-Frame B eine höhere Priorität besitzt.



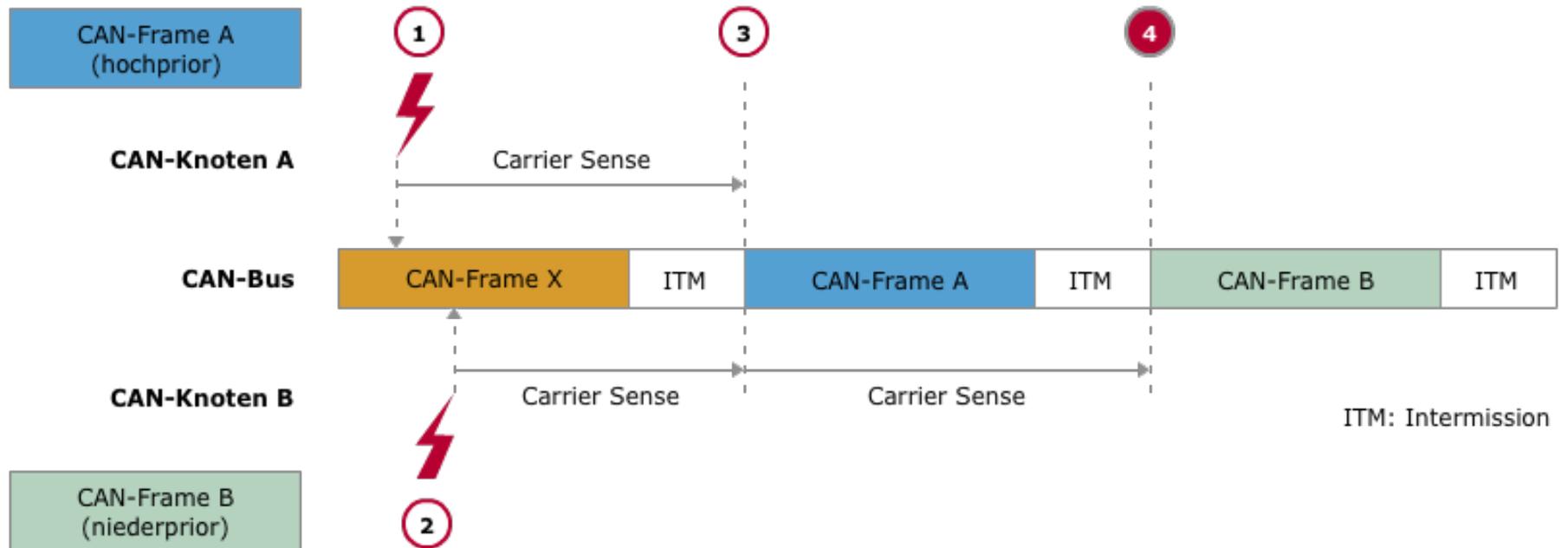
Folge

CAN-Knoten B wartet bis der CAN-Bus wieder frei ist.



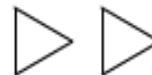
Serielle Bussysteme im Kfz

Prinzip des Buszugriffs



Ereignis 4

Nach Ablauf des ITM ist der CAN-Bus frei.



Folge

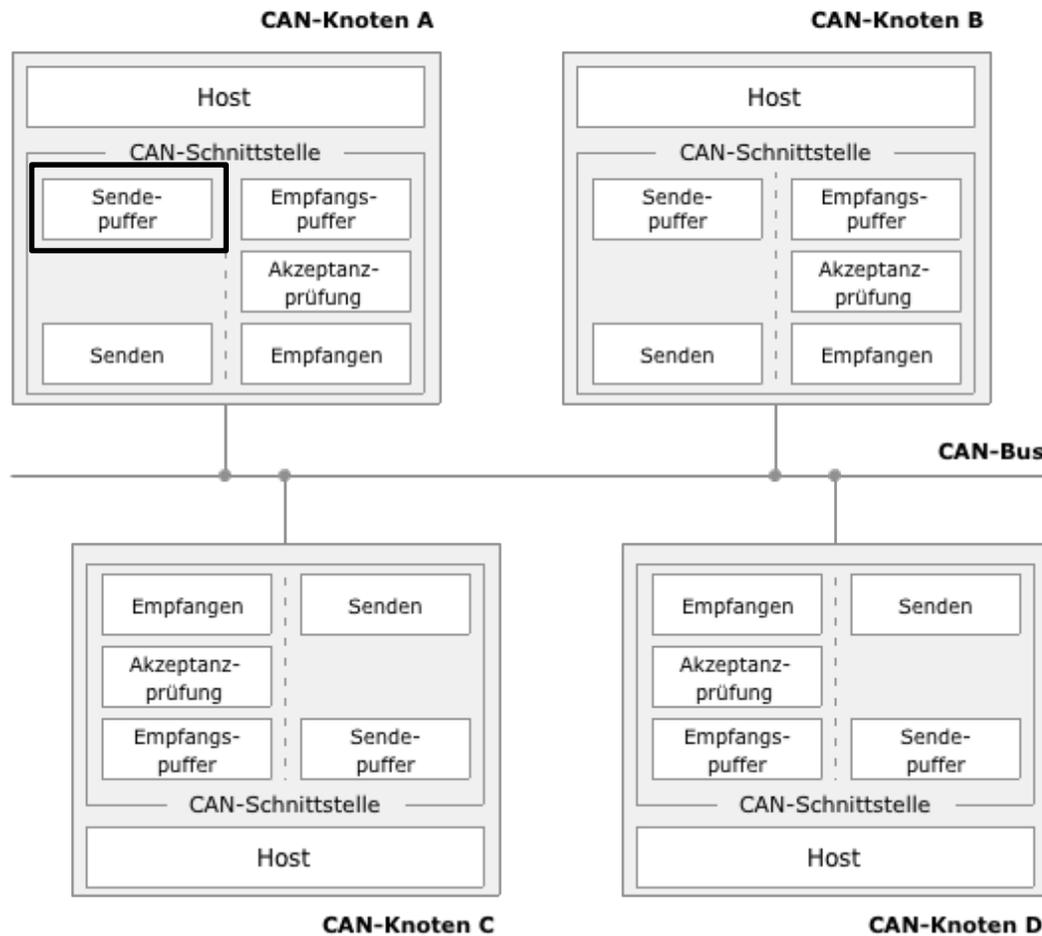
CAN-Knoten B greift auf den CAN-Bus zu und überträgt seine CAN-Botschaft.

- https://vector.com/index.php?&wbt_ls_seite_id=522809&root=376493&seite=vl_can_introduction_de



CAN-Kommunikation

CAN-Kommunikationsprinzip



Kommunikationsmatrix

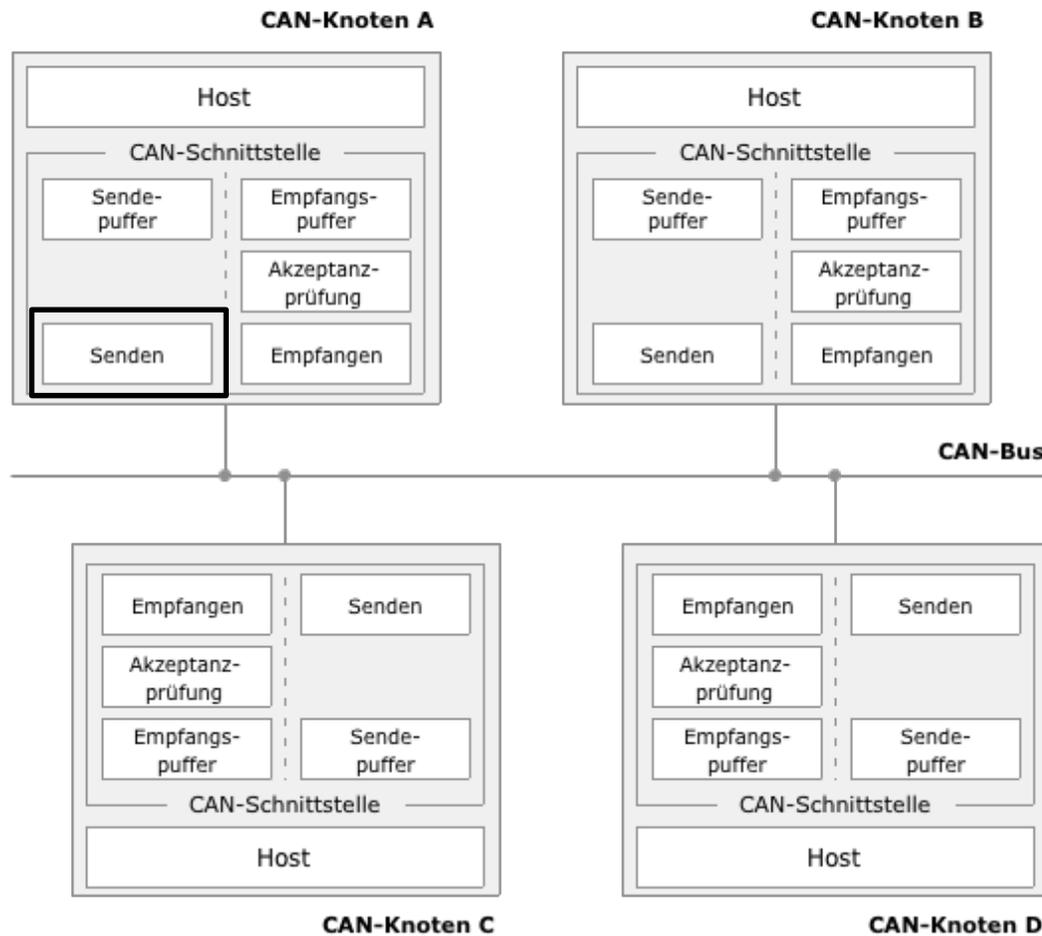
Data Frame	CAN-Knoten A	CAN-Knoten B	CAN-Knoten C	CAN-Knoten D
ID=0x12	Sender	Empfänger		
ID=0x34		Sender	Empfänger	Empfänger
ID=0x52		Empfänger		Sender
ID=0x67	Empfänger	Empfänger	Sender	Empfänger
ID=0xB4	Empfänger		Sender	
ID=0x3A5	Sender	Empfänger	Empfänger	Empfänger





CAN-Kommunikation

CAN-Kommunikationsprinzip



Kommunikationsmatrix

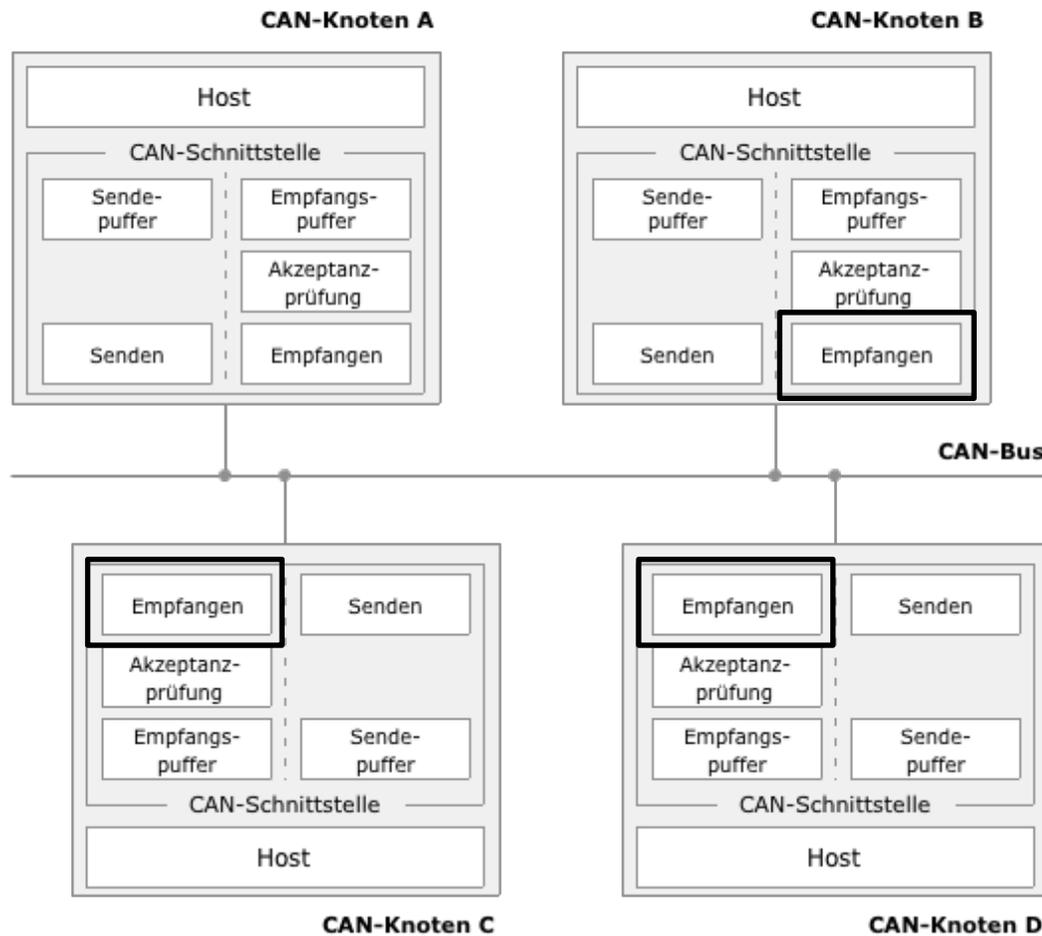
Data Frame	CAN-Knoten A	CAN-Knoten B	CAN-Knoten C	CAN-Knoten D
ID=0x12	Sender	Empfänger		
ID=0x34		Sender	Empfänger	Empfänger
ID=0x52		Empfänger		Sender
ID=0x67	Empfänger	Empfänger	Sender	Empfänger
ID=0xB4	Empfänger		Sender	
ID=0x3A5	Sender	Empfänger	Empfänger	Empfänger





CAN-Kommunikation

CAN-Kommunikationsprinzip



Kommunikationsmatrix

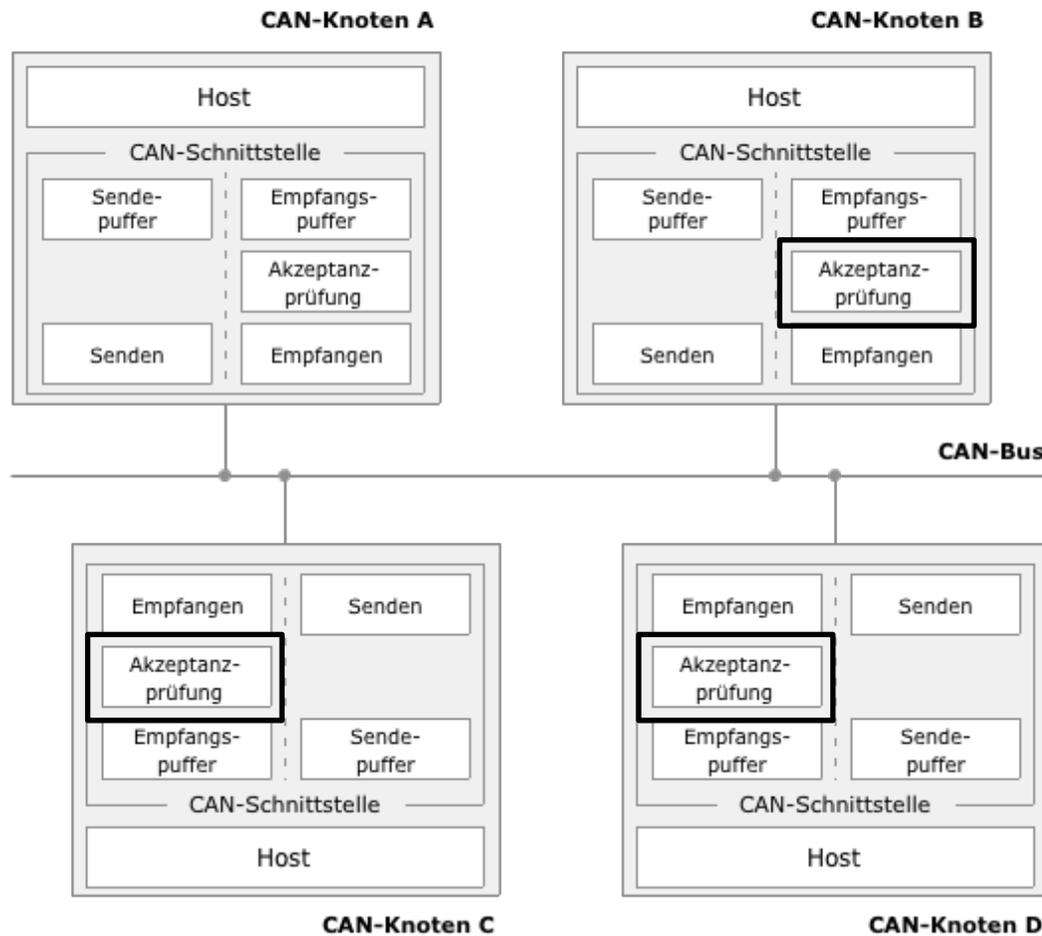
Data Frame	CAN-Knoten A	CAN-Knoten B	CAN-Knoten C	CAN-Knoten D
ID=0x12	Sender	Empfänger		
ID=0x34		Sender	Empfänger	Empfänger
ID=0x52		Empfänger		Sender
ID=0x67	Empfänger	Empfänger	Sender	Empfänger
ID=0xB4	Empfänger		Sender	
ID=0x3A5	Sender	Empfänger	Empfänger	Empfänger





CAN-Kommunikation

CAN-Kommunikationsprinzip



Kommunikationsmatrix

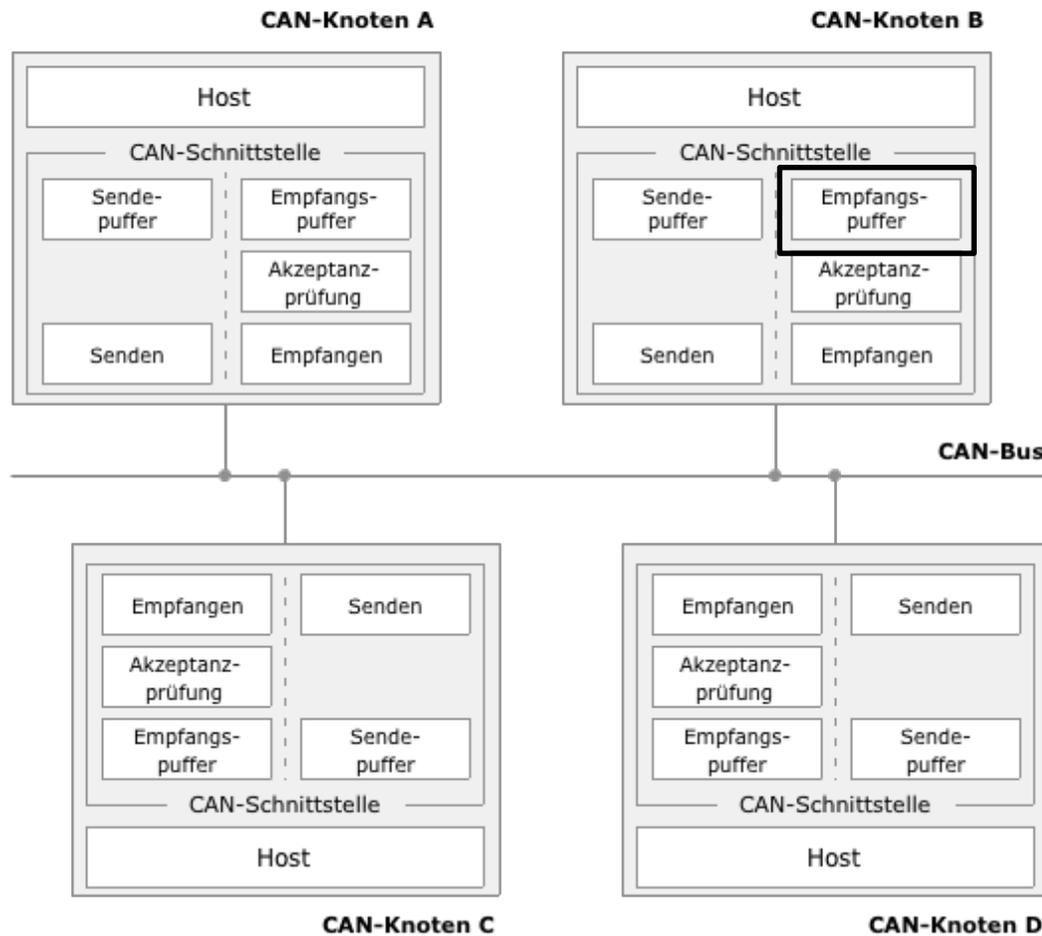
Data Frame	CAN-Knoten A	CAN-Knoten B	CAN-Knoten C	CAN-Knoten D
ID=0x12	Sender	Empfänger		
ID=0x34		Sender	Empfänger	Empfänger
ID=0x52		Empfänger		Sender
ID=0x67	Empfänger	Empfänger	Sender	Empfänger
ID=0xB4	Empfänger		Sender	
ID=0x3A5	Sender	Empfänger	Empfänger	Empfänger





CAN-Kommunikation

CAN-Kommunikationsprinzip



Kommunikationsmatrix

Data Frame	CAN-Knoten A	CAN-Knoten B	CAN-Knoten C	CAN-Knoten D
ID=0x12	Sender	Empfänger		
ID=0x34		Sender	Empfänger	Empfänger
ID=0x52		Empfänger		Sender
ID=0x67	Empfänger	Empfänger	Sender	Empfänger
ID=0xB4	Empfänger		Sender	
ID=0x3A5	Sender	Empfänger	Empfänger	Empfänger



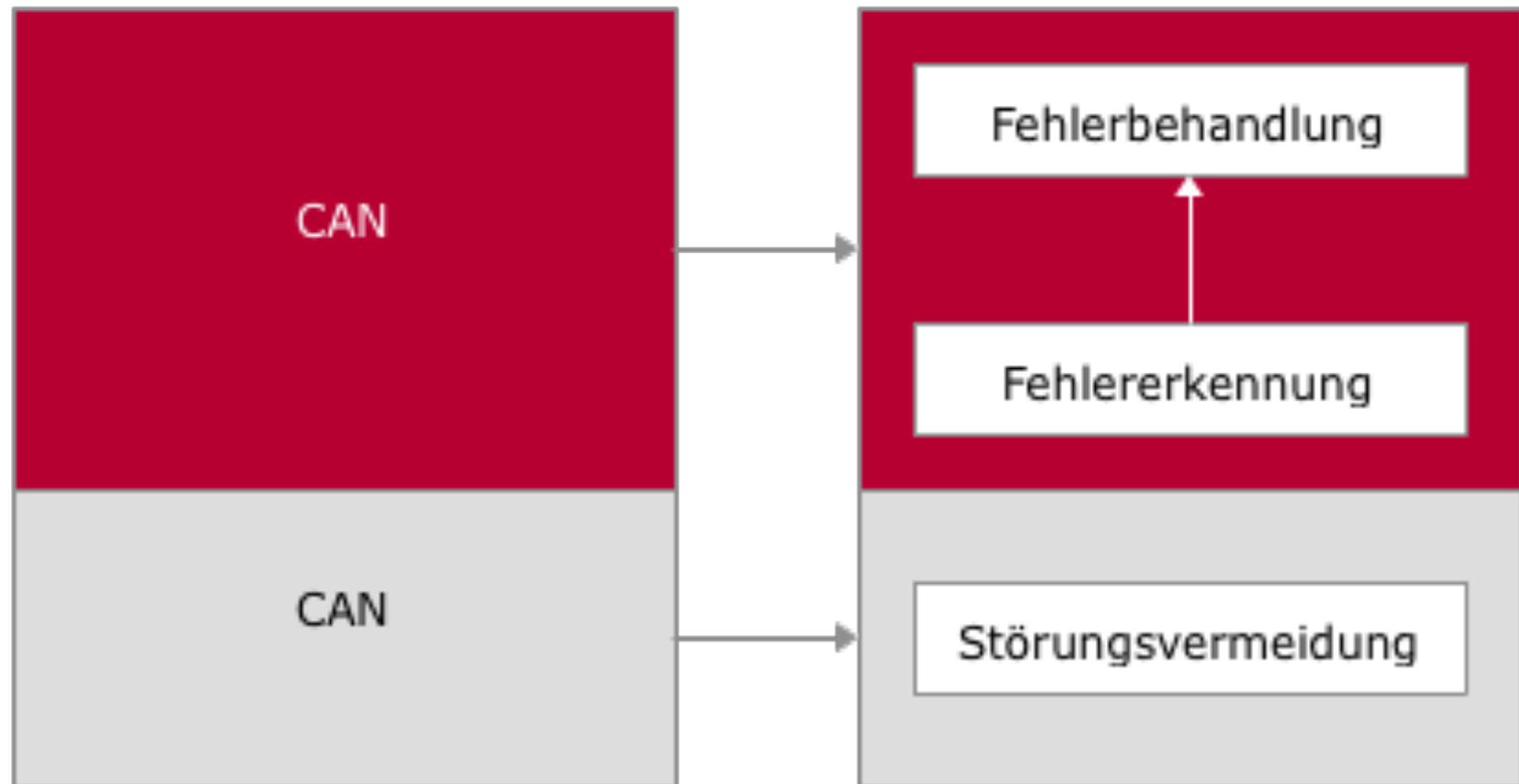
- Teilweise sehr zeit- und sicherheitskritischen Anwendungen
- Hohe Anforderungen an die Datenintegrität
 - Physikalische Datensicherungsmaßnahmen: Störungsvermeidung
 - Logische Datensicherungsmaßnahmen: Fehlererkennung und Fehlerbehandlung
- Physikalische Datensicherung
 - Symmetrische Signalübertragung über eine verdrehte Zweidrahtleitung.
- Logische Datensicherung: Fünf Fehlererkennungsmechanismen
 - Sender:
 - Vergleichen des gesendeten Bitpegels mit dem tatsächlichen Buspegel (Bitmonitoring)
 - Auswertung der Quittungen der CAN-Knoten (ACK-Check)
 - Empfänger:
 - Überprüfung der ankommenden CRC-Sequenz auf Korrektheit (Cyclic Redundancy Check) für jeden Data Frame, positive oder negative Quittung an Sender
 - Überprüfung des vorgeschriebenen Format (Form Check)
 - Empfänger: Überprüfung der Einhaltung der Bitstuffingregel (Stuff Check)

- Sobald ein CAN-Knoten einen Übertragungsfehler entdeckt, bricht er die Datenübertragung ab und überträgt sofort ein Fehlersignal (Error Flag).
- Dieses setzt sich aus sechs homogenen Bits zusammen. Dadurch stellen alle CAN-Knoten einen Bitstuffingfehler fest.
- Weil dann alle CAN-Knoten die Datenübertragung mittels Error Flag abbrechen, ist die Datenkonsistenz sichergestellt.
- Sobald der CAN-Bus wieder frei ist, wiederholt der Sender den abgebrochenen Data Frame.



Serielle Bussysteme im Kfz

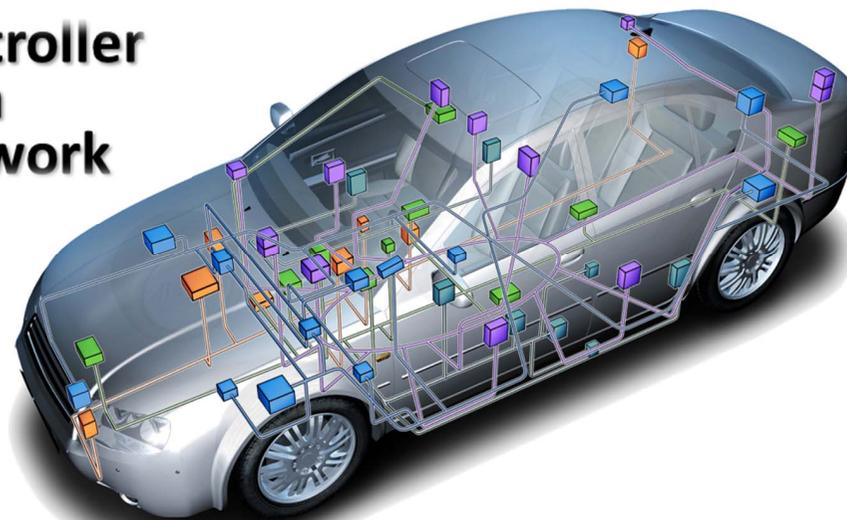
Datensicherung im CAN-Netzwerk



- CAN FD (PDF, 1.13 MB)
Thomas Lindenkreuz - Robert Bosch GmbH
- 6. Vector Congress 2012
Beiträge: http://www.vector.com/vi_congress12_de.html

CAN FD – CAN with Flexible Data Rate

**Controller
Area
Network**



Thomas Lindenkreuz
Robert Bosch GmbH
Automotive Electronics

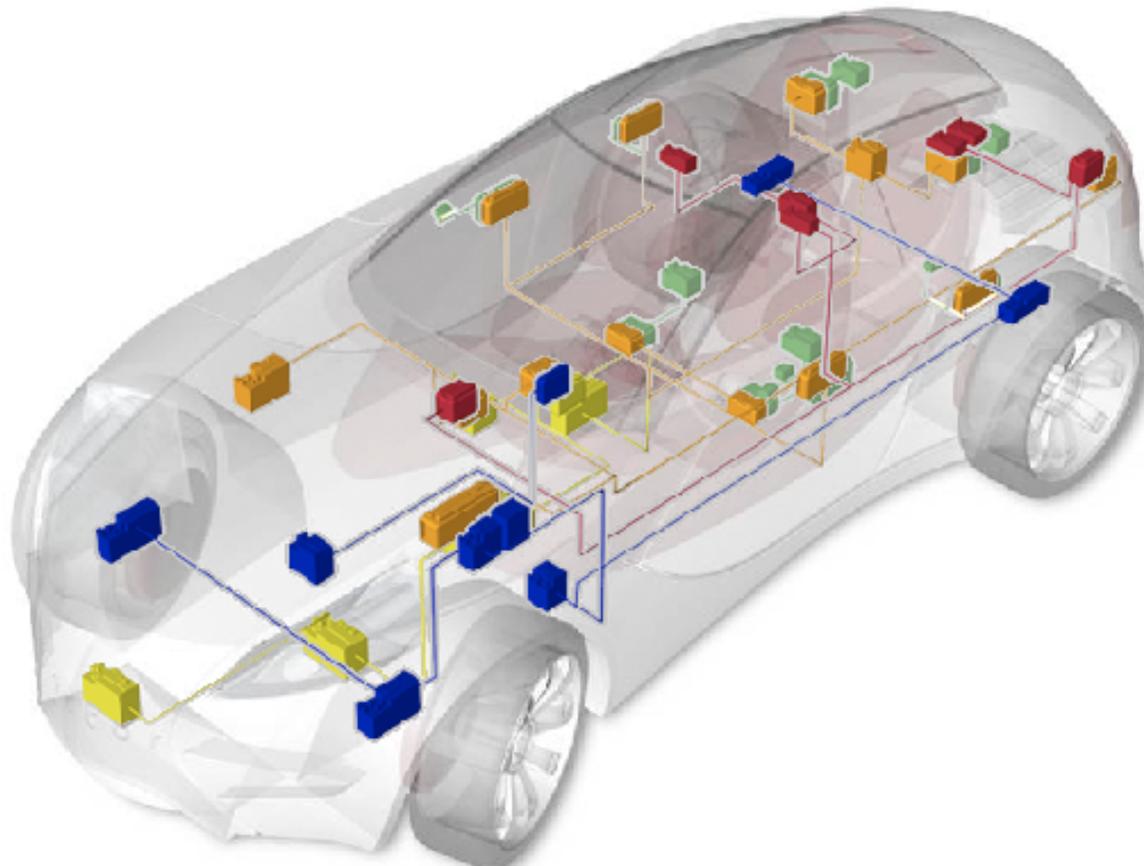
Vector Kongress 2012



Elektronikvernetzung im Kfz

✕ schließen

- Antriebsstrang (Powertrain)
- Fahrwerk (Chassis)
- Karosserie (Body)
- Multi-Media (Telematics)



■ CAN H

■ CAN L

■ LIN

■ FlexRay

■ MOST

Local Interconnect Network für verteilte, räumlich abgegrenzte elektronische Systeme, z.B. Vernetzung innerhalb eines Sitzes.

Protokolle und Bussysteme: LIN (Local Interconnect Bus)



- Entwickelt von internationalem OEM-Konsortium
- Nichtkommerziell
- Eigenschaften:
 - Bandbreite 19.2 kbit/s
 - Synchroner Master/Slave-Betrieb, optimierbar für P2P
 - Physical Layer: Bus, twisted pair
 - Fehlererkennung: inverted mod256-Checksum
- Mit gängigen Line Drivers realisierbar: kostengünstig
- Flexible, kostengünstige und hinreichend sichere Lösung für alle wenig datenintensiven Kommunikationsdienste und Point2Point-Verbindungen
 - Intelligente Sensoren
 - Backupleitungen
 - Diagnoseschnittstellen
- <http://www.lin-subbus.org/>

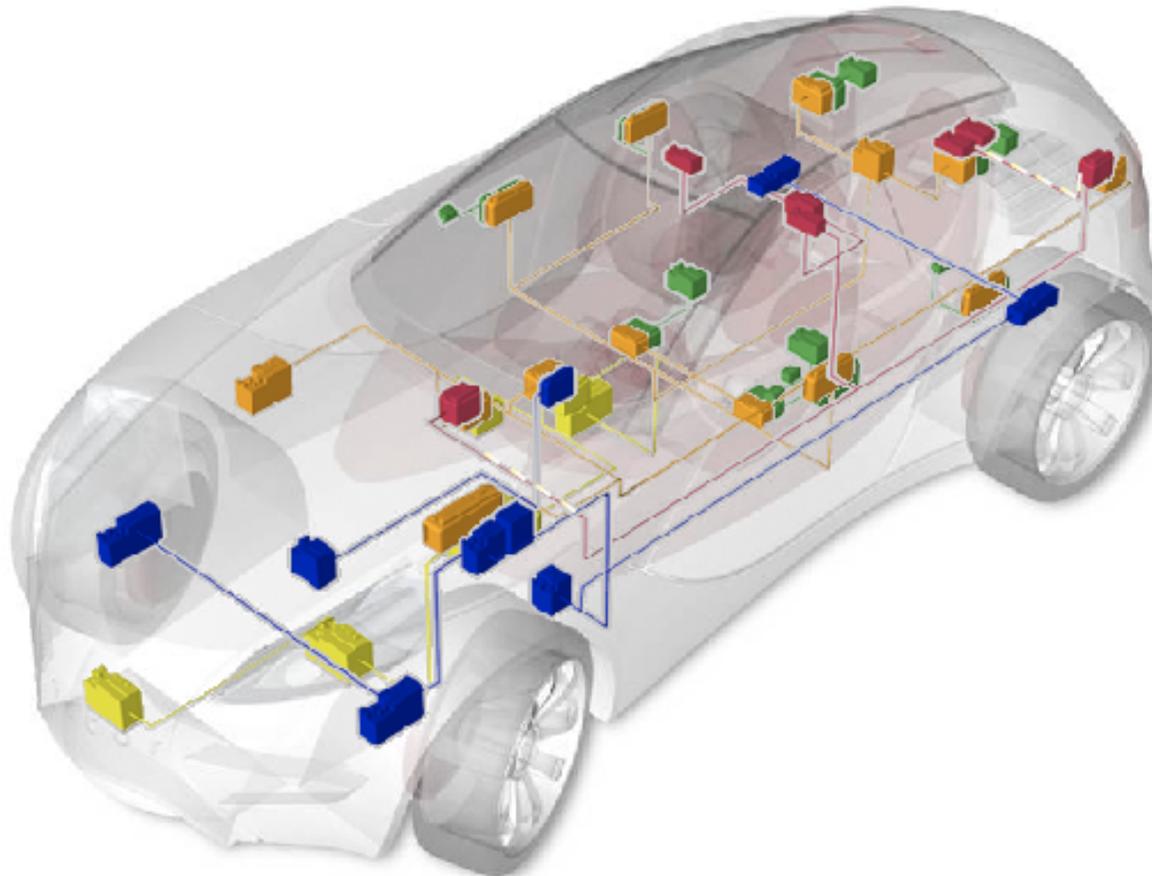




Elektronikvernetzung im Kfz

✕ schließen

- Antriebsstrang (Powertrain)
- Fahrwerk (Chassis)
- Karosserie (Body)
- **Multi-Media (Telematics)**



■ CAN H

■ CAN L

■ LIN

■ FlexRay

■ **MOST**

Optischer Multimedia-Bus zur Vernetzung von Infotainment-Komponenten im Kfz.

Protokolle und Bussysteme: MOST (Media Orientated System Transport)



- Entwickelt von BMW, DaimlerChrysler, Harman, OASIS seit 1998
- ISO/OSI-Standard, frei
- Eigenschaften:
 - Bandbreite 24.8 Mbit/s
 - Asynchroner und synchroner Betrieb mit Busmaster
 - Physical Layer: Stern, Kette, Ring (in praxi: Ring), Lichtwellenleiter
 - Fehlererkennung: CRC-16, d=6
- Geeignet für datenintensive Infotainment-Dienste
- Nicht geeignet für x-by-Wire, Sicherheitssysteme

- <http://www.mostcooperation.com/home/index.html>

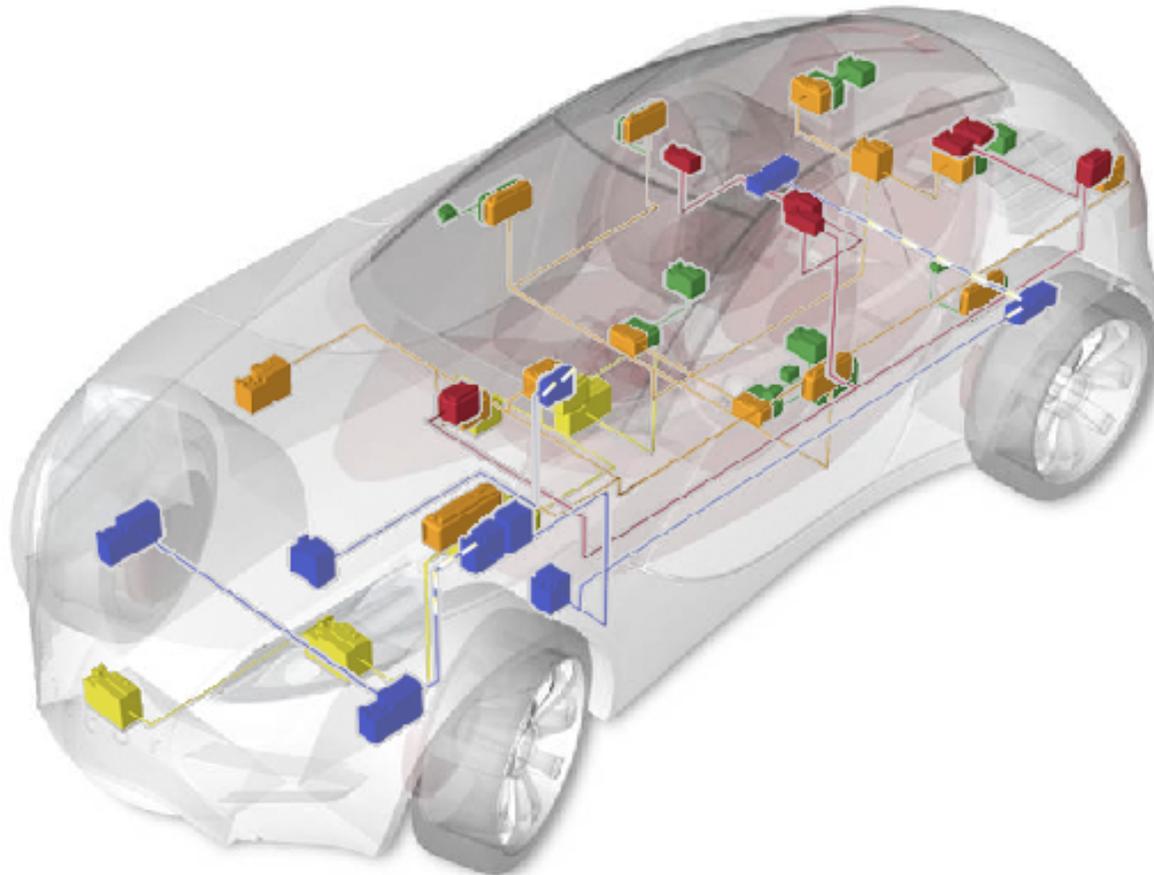




Elektronikvernetzung im Kfz

☒ schließen

- Antriebsstrang (Powertrain)
- Fahrwerk (Chassis)
- Karosserie (Body)
- Multi-Media (Telematics)



■ CAN H ■ CAN L ■ LIN ■ FlexRay ■ MOST

Bussystem für zeitkritische Systeme sowie für die Übertragung großer Datenmengen, z.B. BreakbyWire, SteerbyWire, XbyWire.

Protokolle und Bussysteme: FlexRay



- Entwickelt von BMW, DaimlerChrysler & div. HL-Herstellern seit 1999
- Nichtkommerziell, frei
- Eigenschaften:
 - Bandbreite bis 10 Mbit/s
 - Asynchroner und synchroner Betrieb mit Busmaster
 - Physical Layer: beliebige Topologie, beliebiges Medium
 - Redundante Systemauslegung möglich
 - Fehlererkennung: CRC-16, d=6
- Geeignet für sicherheitskritische Anwendungen (x-by-Wire)
- Für Multimedia u.U. Bandbreite zu niedrig

- <http://www.flexray.com/>



- Einführung Ethernet (PDF, 1.07 MB)
Peter Schönenberg - BMW Group
- Automotive IP/Ethernet: Nächste Schritte in der Vorentwicklung (PDF, 2.20 MB)
Jürgen Röder - Continental Automotive GmbH
- 6. Vector Congress 2012
Beiträge: http://www.vector.com/vi_congress12_de.html



Peter Schönenberg, BMW AG,
Germany.

INTRODUCTION OF ETHERNET.

6TH VECTOR CONGRESS
NOVEMBER 29, 2012, STUTT GART

BMW
GROUP



Continental 

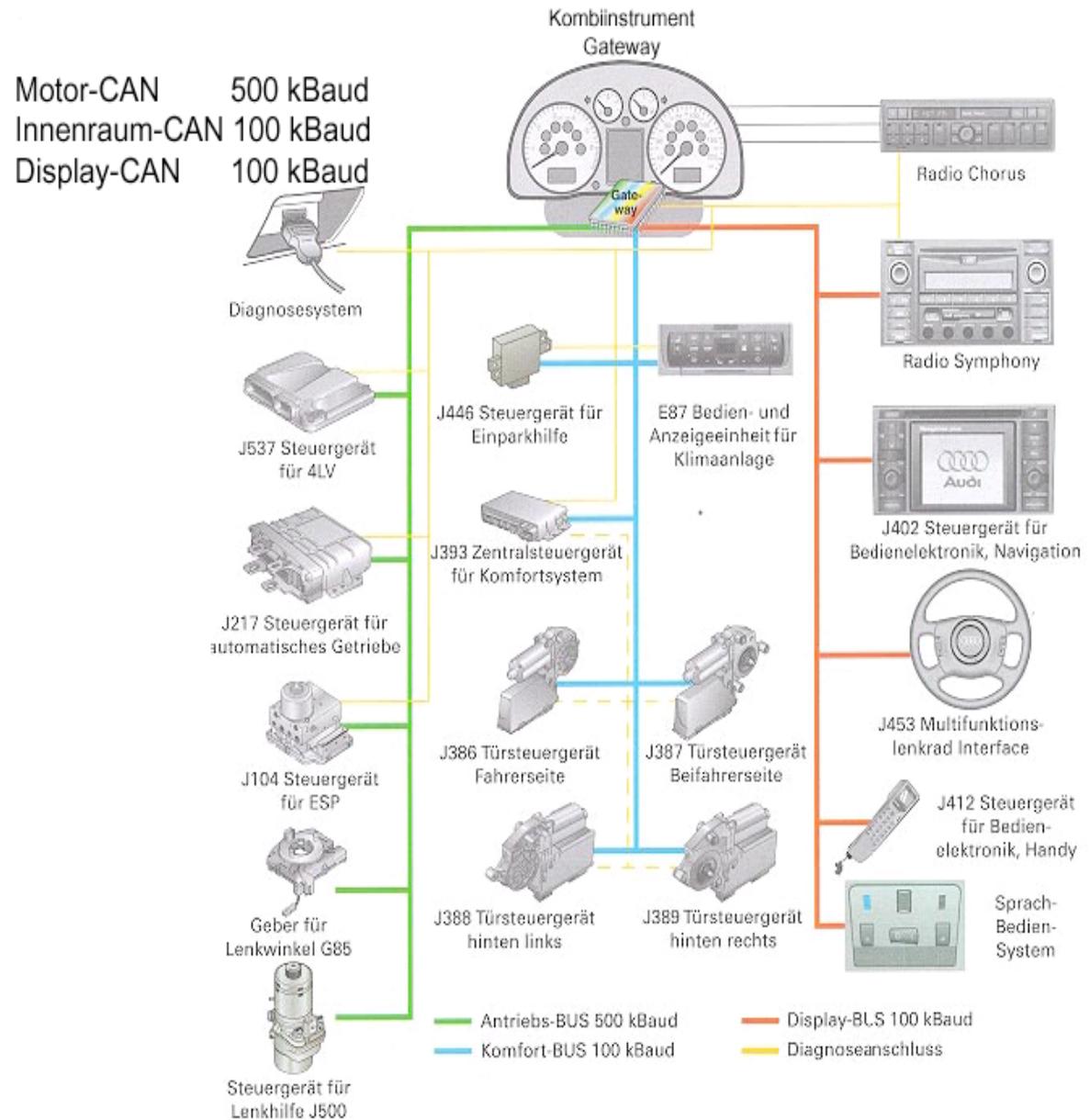
IP/Ethernet

Next steps of pre-development

Protokolle und Bussysteme: Beispiele

Audi A2, Modelljahr 2002

- Powertrain: Highspeed-CAN
- Innenraum/Karosserie: Lowspeed-CAN
- Display/MMI: Lowspeed-CAN
- Unabhängiger Diagnosebus
- Gateway: Kombi
 - Motor-CAN
 - Innenraum-CAN
 - Display-CAN

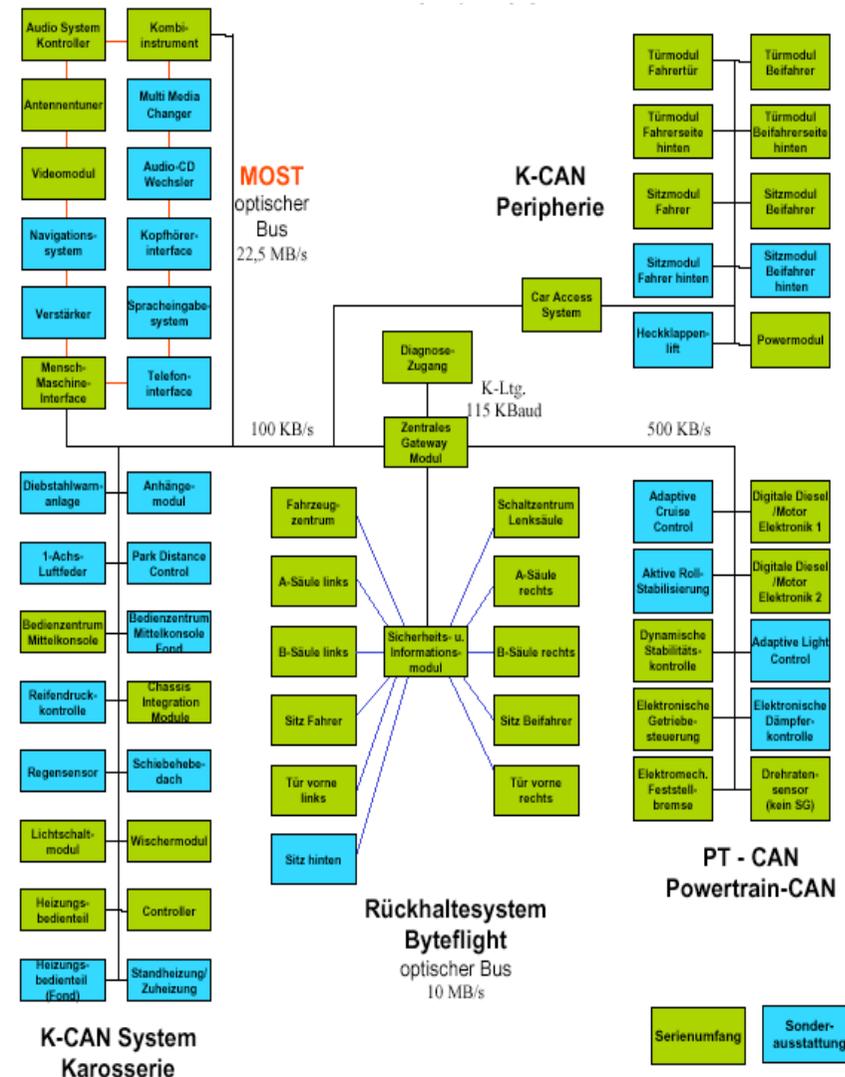


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

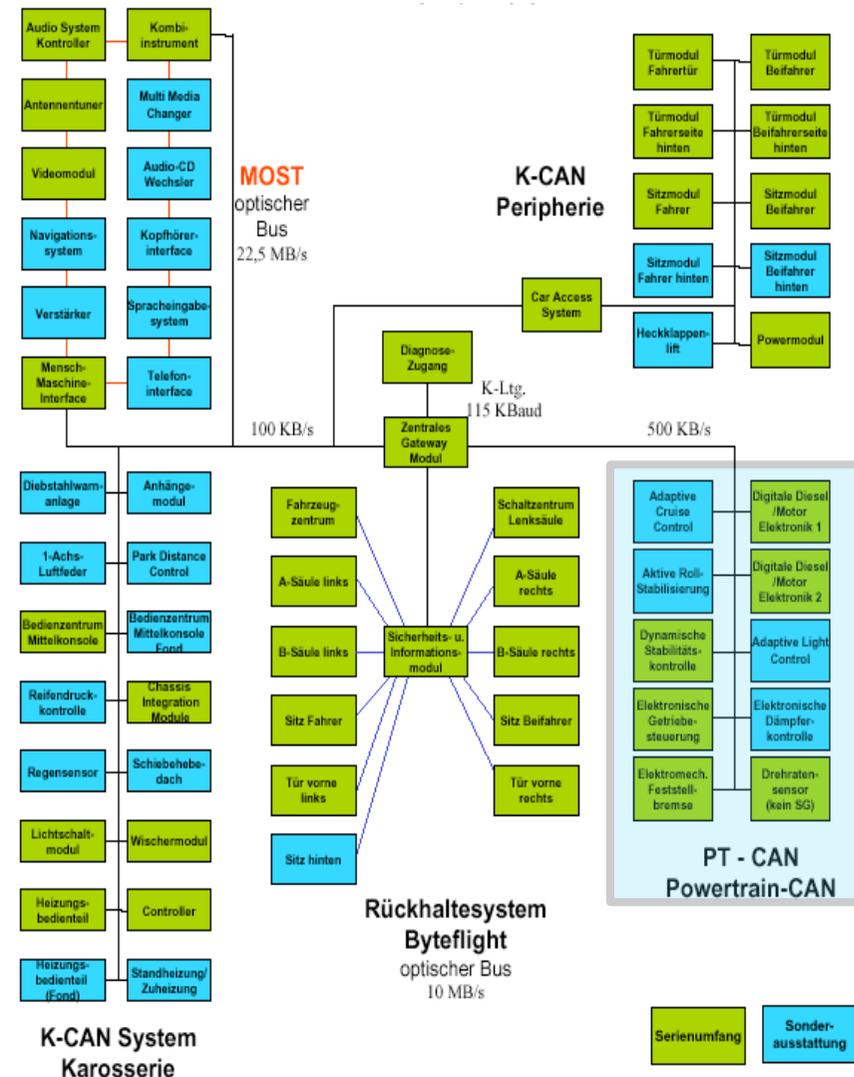


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- **Powertrain: Highspeed-CAN**
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

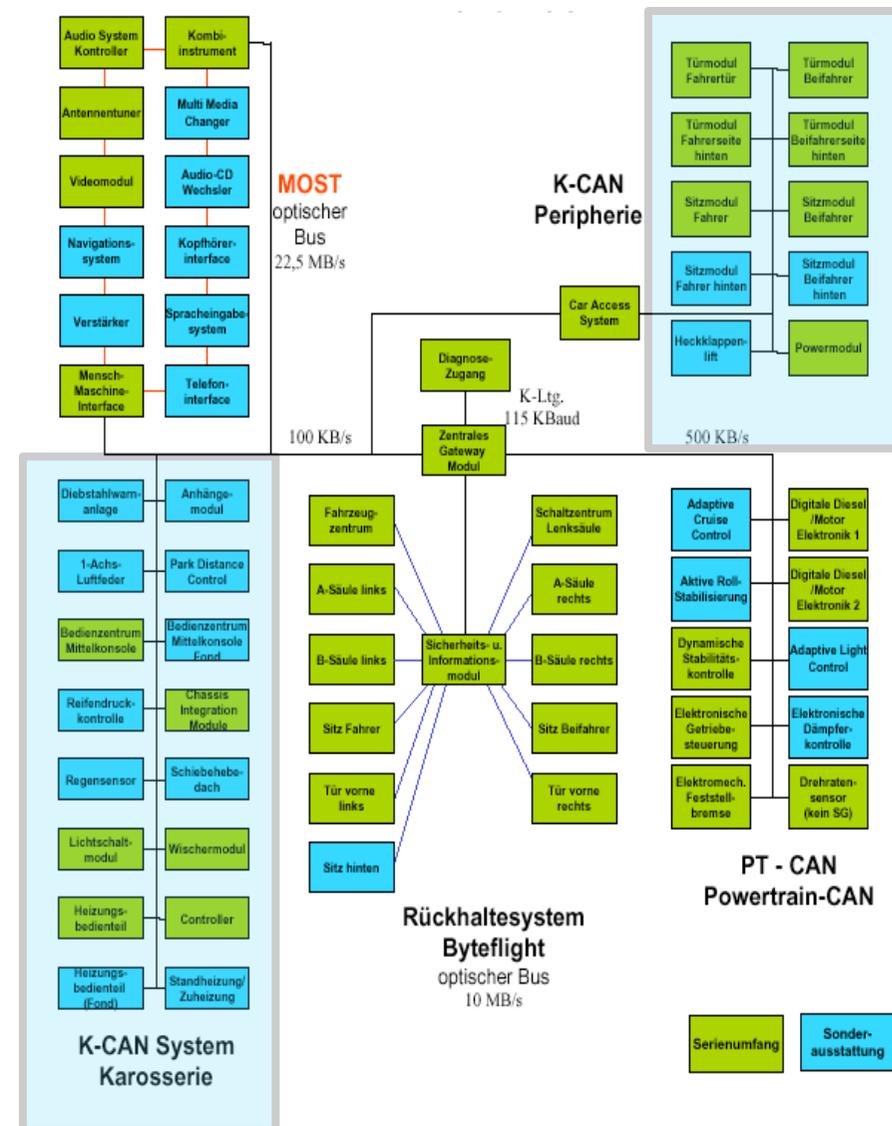


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- **Karosserie und Peripherie: Lowspeed-CAN**
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

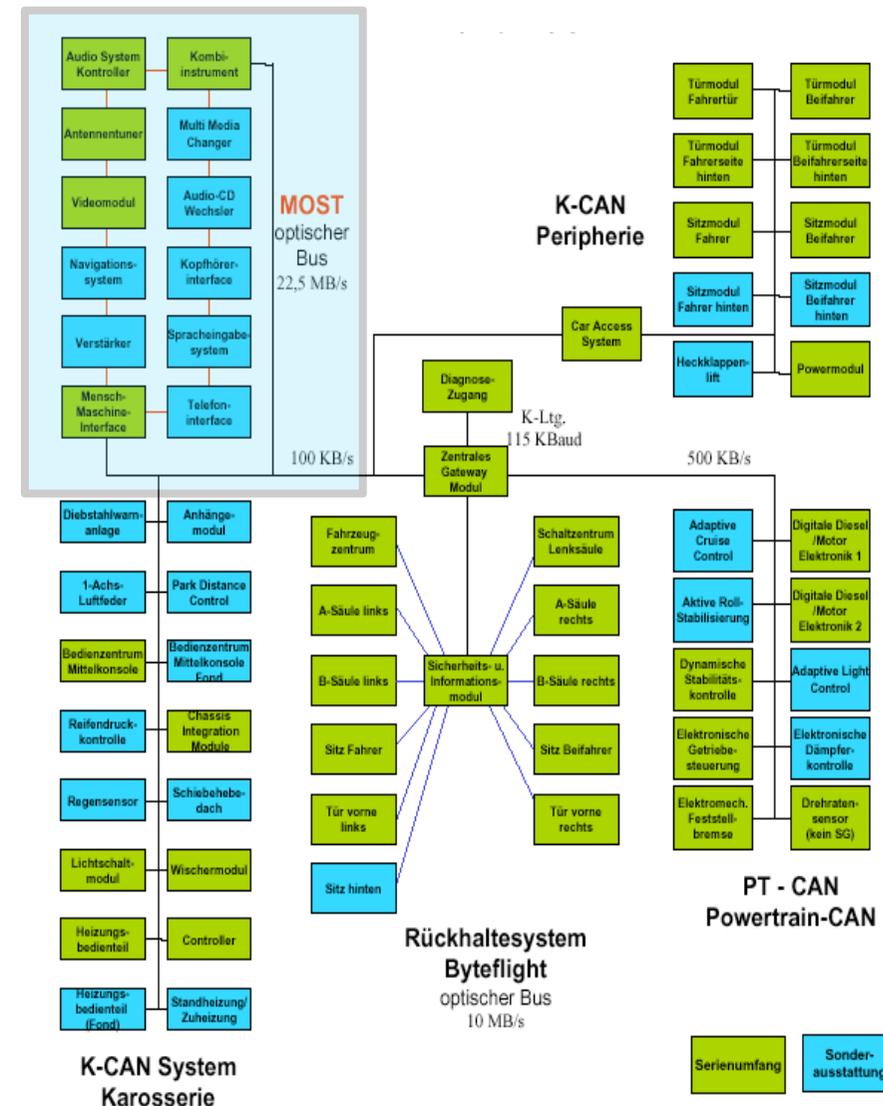


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

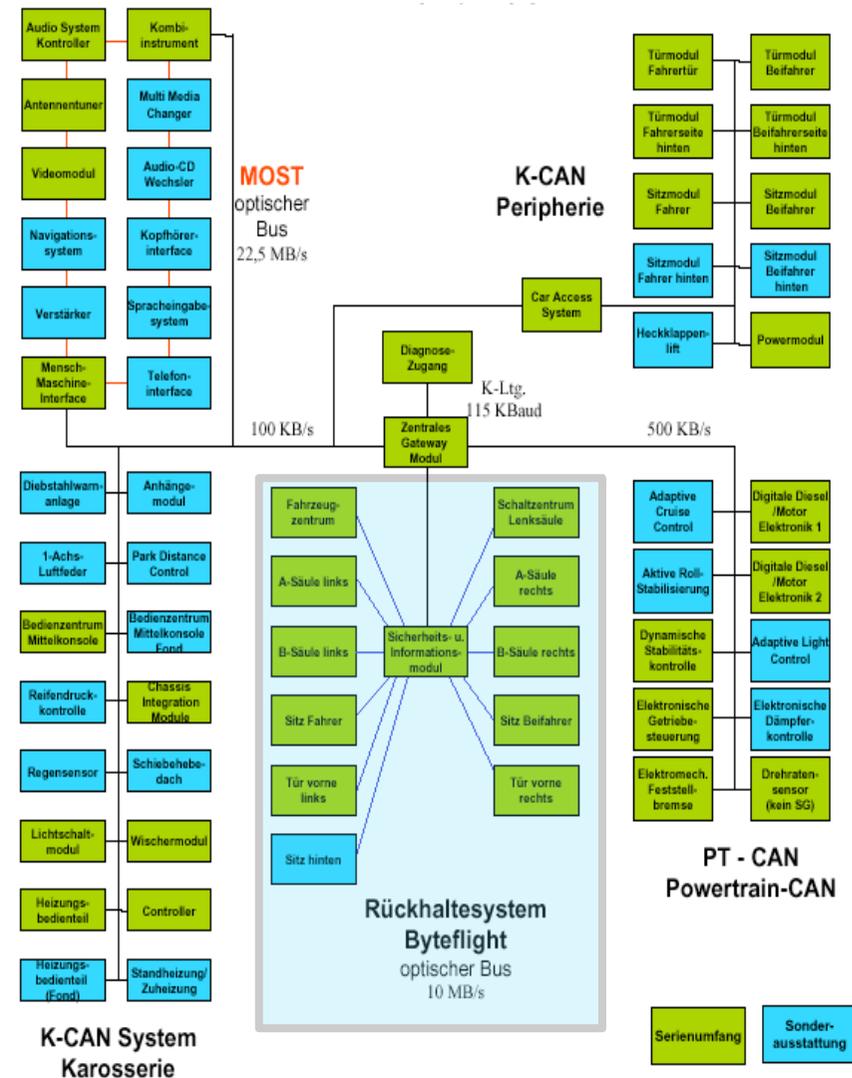


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- **Passive Sicherheit: byteflight**
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

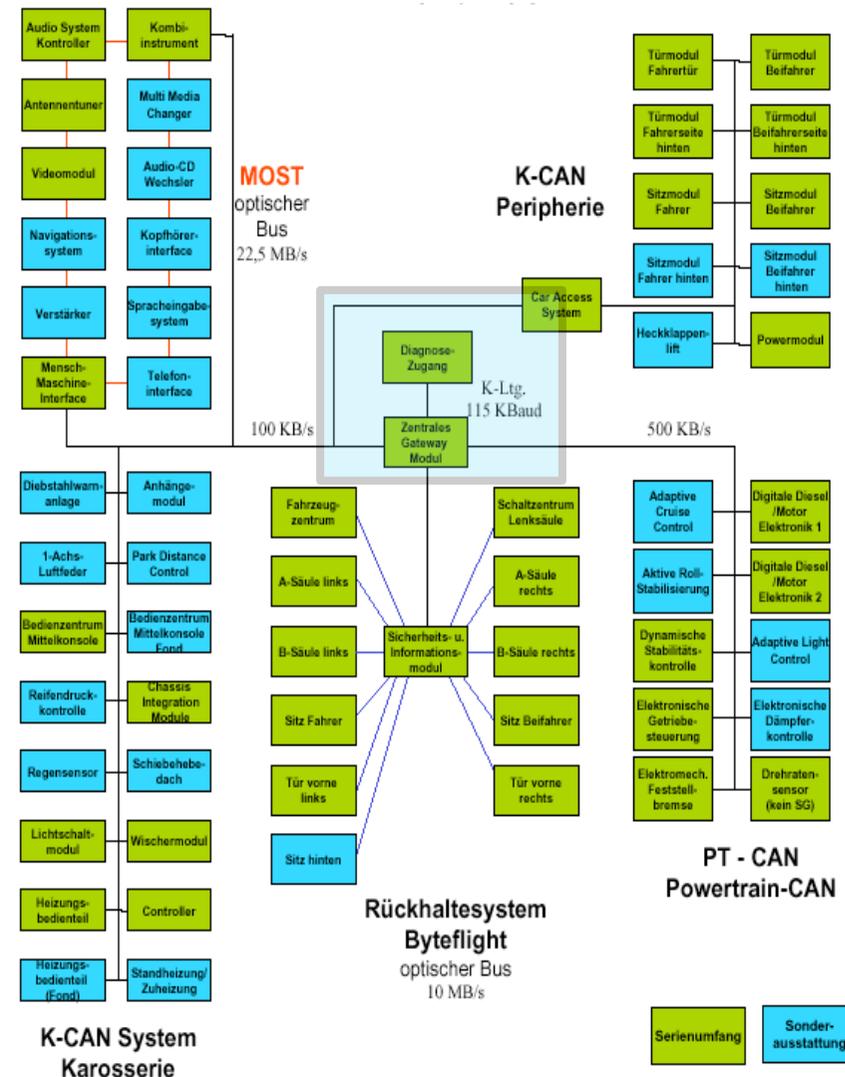


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

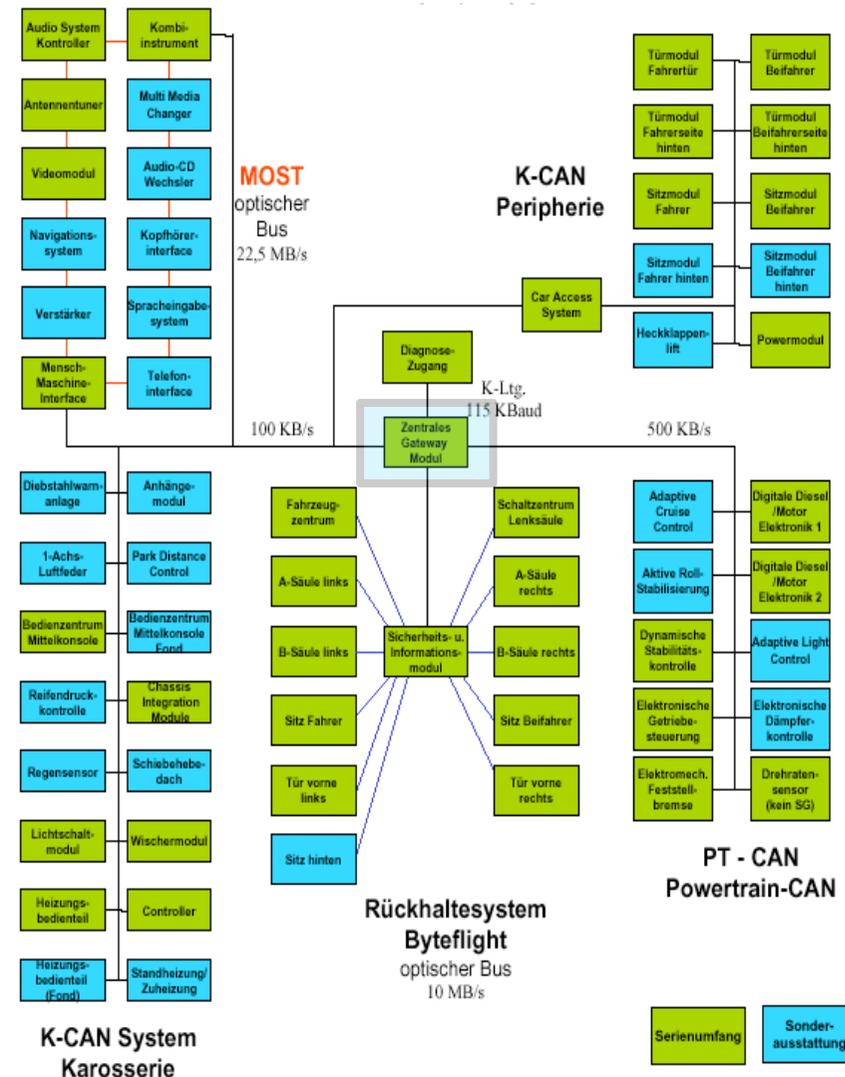


Protokolle und Bussysteme: Beispiele



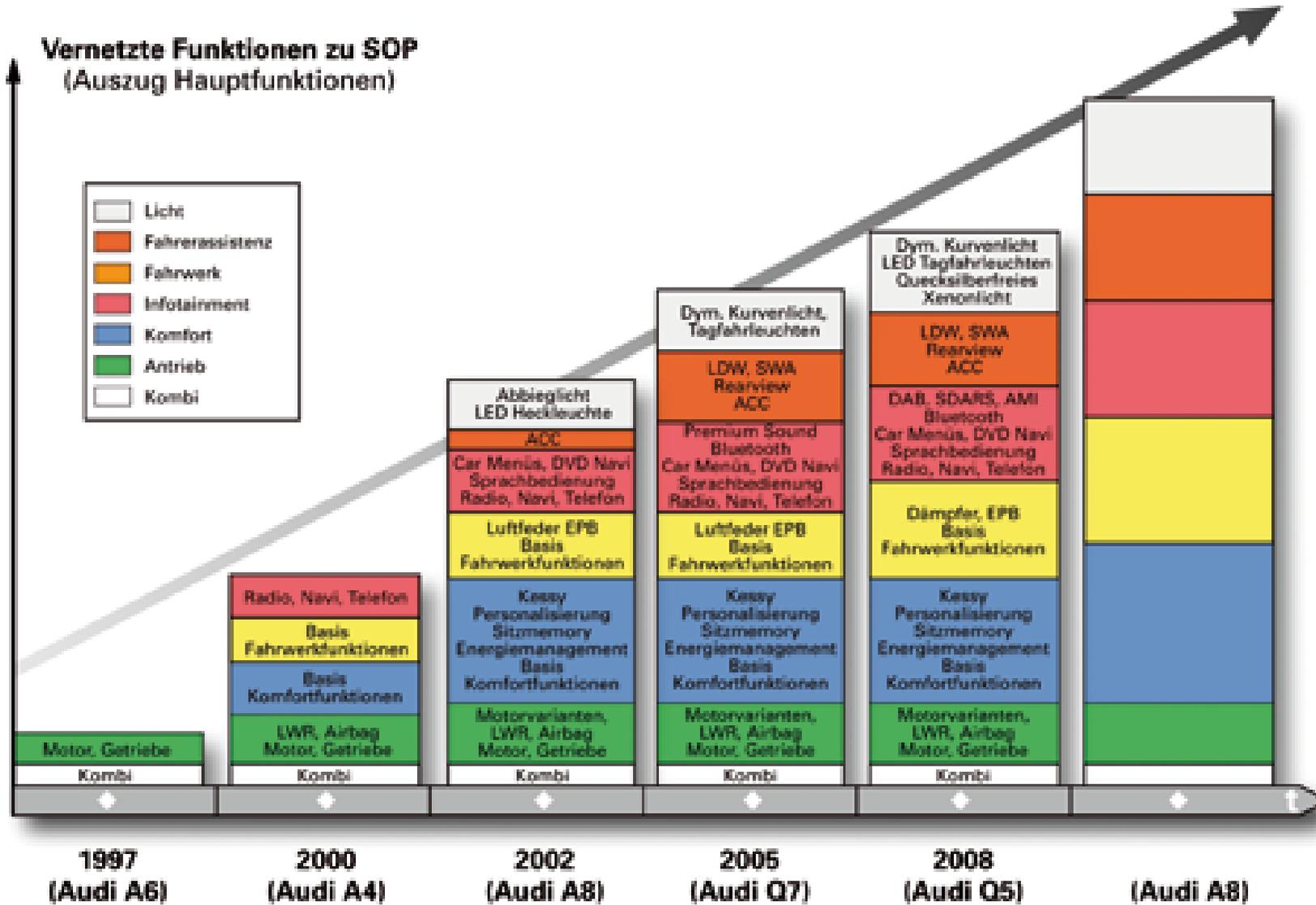
BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

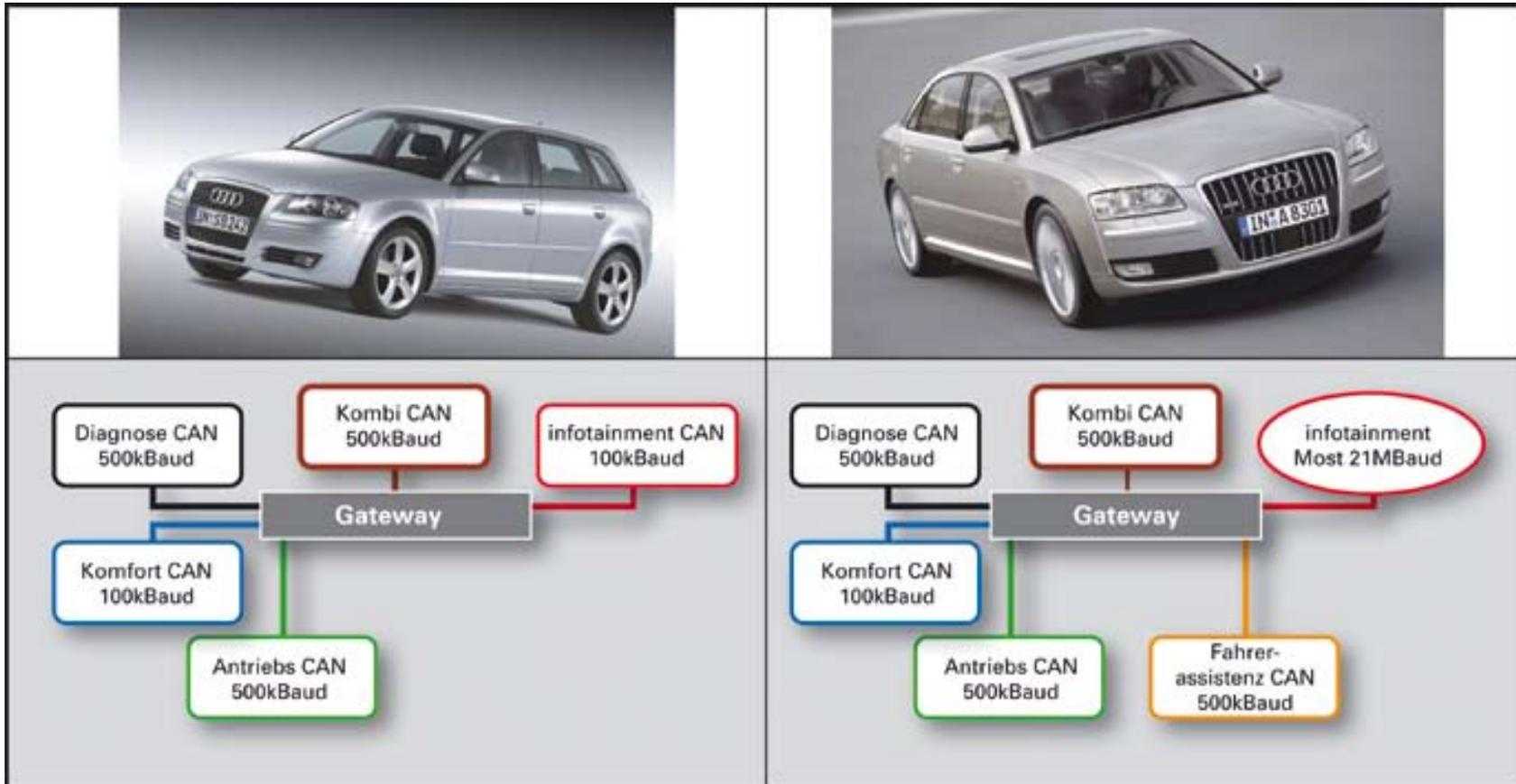




Zunahme der vernetzten Funktionen: Beispiel Audi

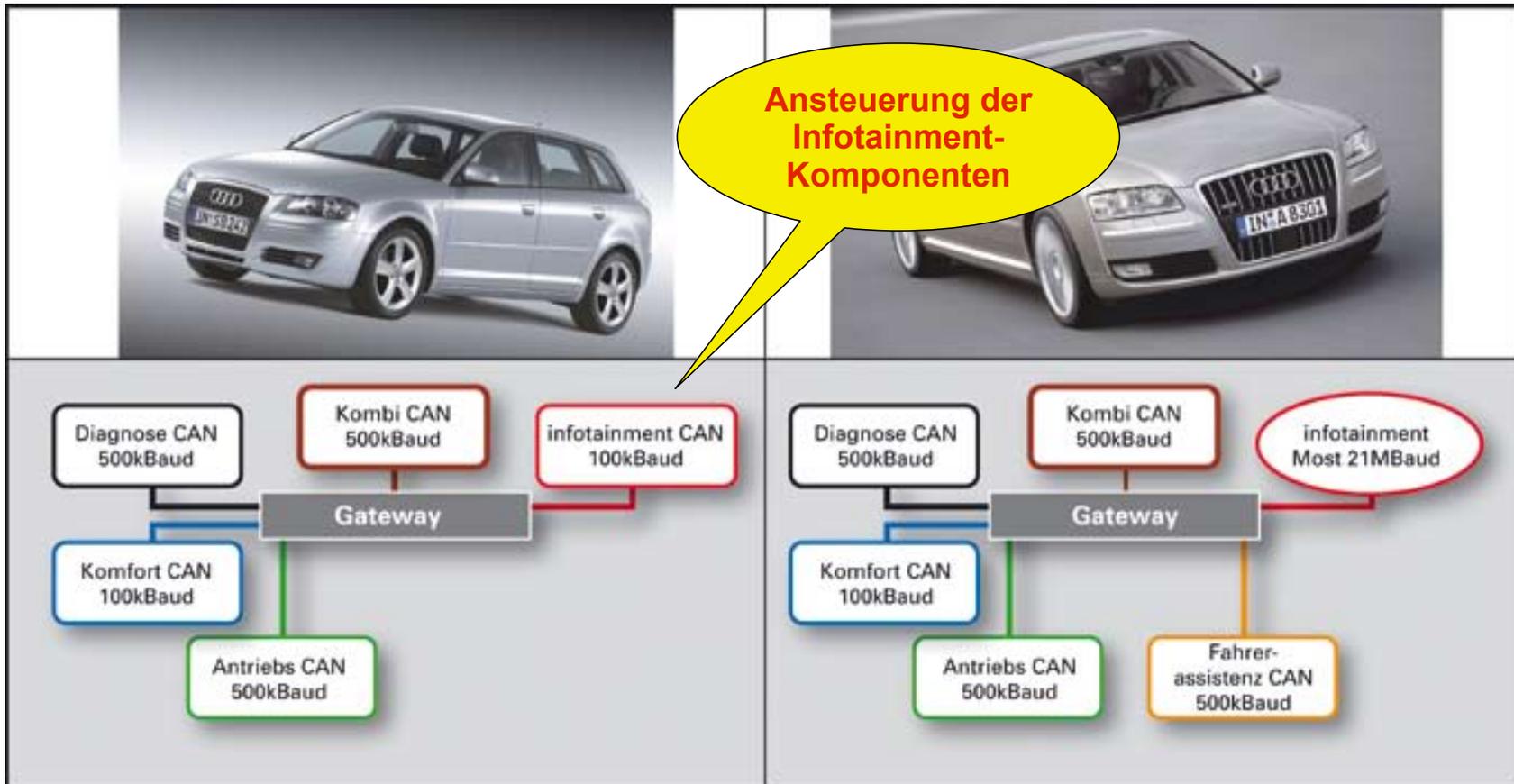


Im Infotainment-Bereich vereint der neue Audi Q5 Elemente des Audi A3 (CAN) und des A8 (MOST)



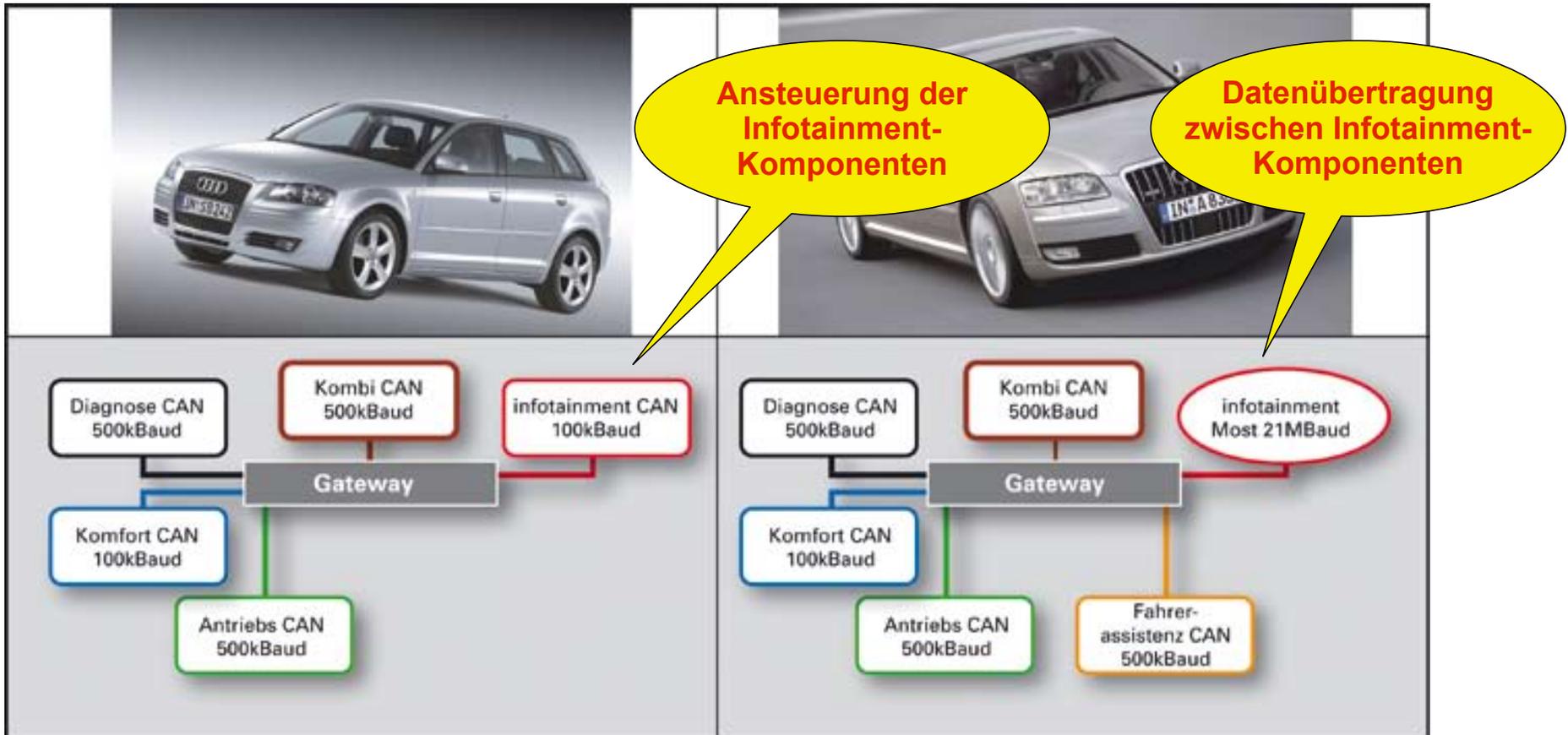
Im Audi A3 kommen ausschließlich Infotainmentsysteme auf Basis der CAN-Bustechnologie zum Einsatz, im Audi A8 kommunizieren die Infotainmentsysteme auf Basis der MOST-Technologie. Im Audi Q5 sollten beide Systemvarianten, CAN und MOST, ermöglicht werden.

Im Infotainment-Bereich vereint der neue Audi Q5 Elemente des Audi A3 (CAN) und des A8 (MOST)



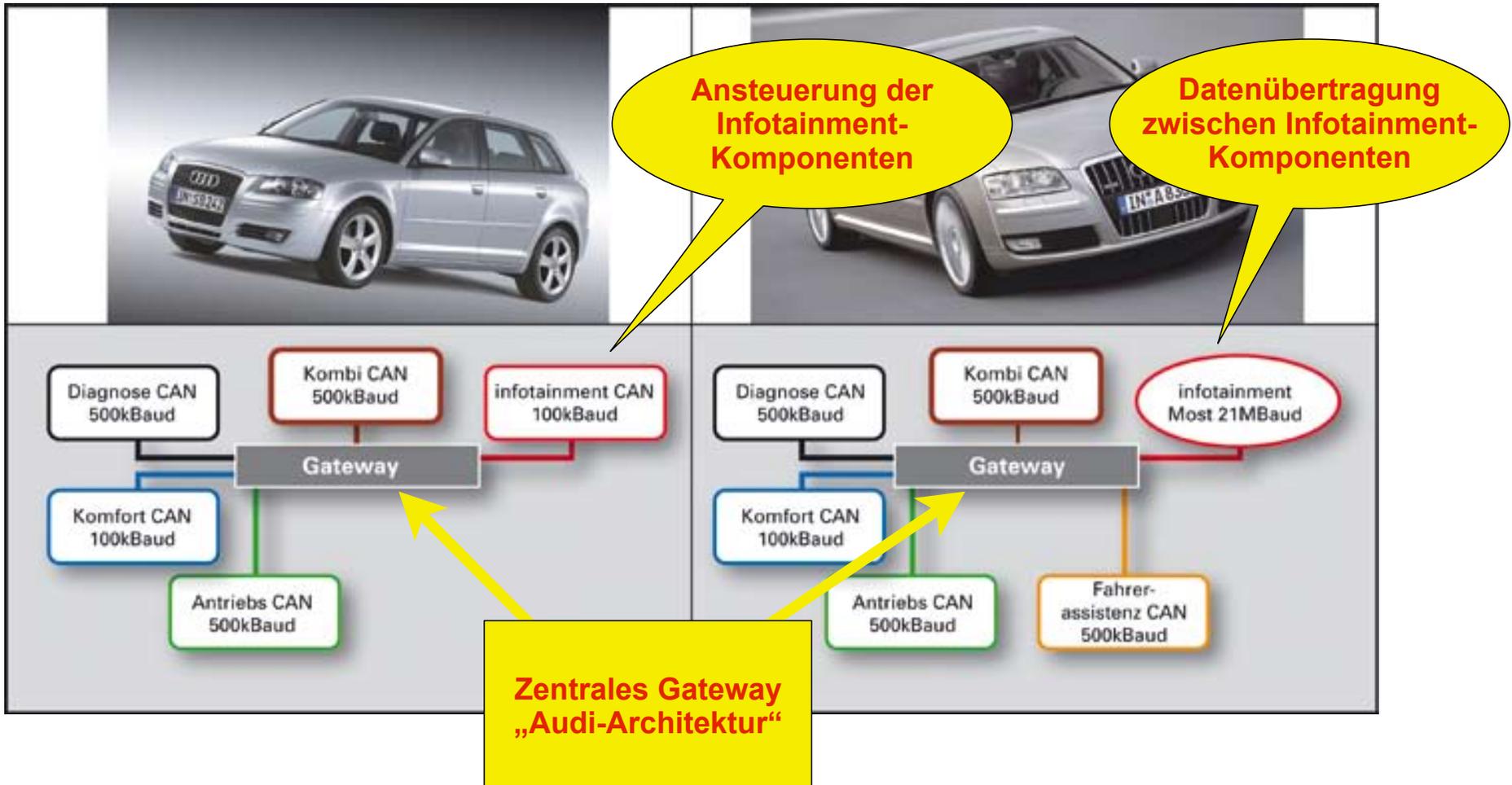
Im Audi A3 kommen ausschließlich Infotainmentsysteme auf Basis der CAN-Bustechnologie zum Einsatz, im Audi A8 kommunizieren die Infotainmentsysteme auf Basis der MOST-Technologie. Im Audi Q5 sollten beide Systemvarianten, CAN und MOST, ermöglicht werden.

Im Infotainment-Bereich vereint der neue Audi Q5 Elemente des Audi A3 (CAN) und des A8 (MOST)



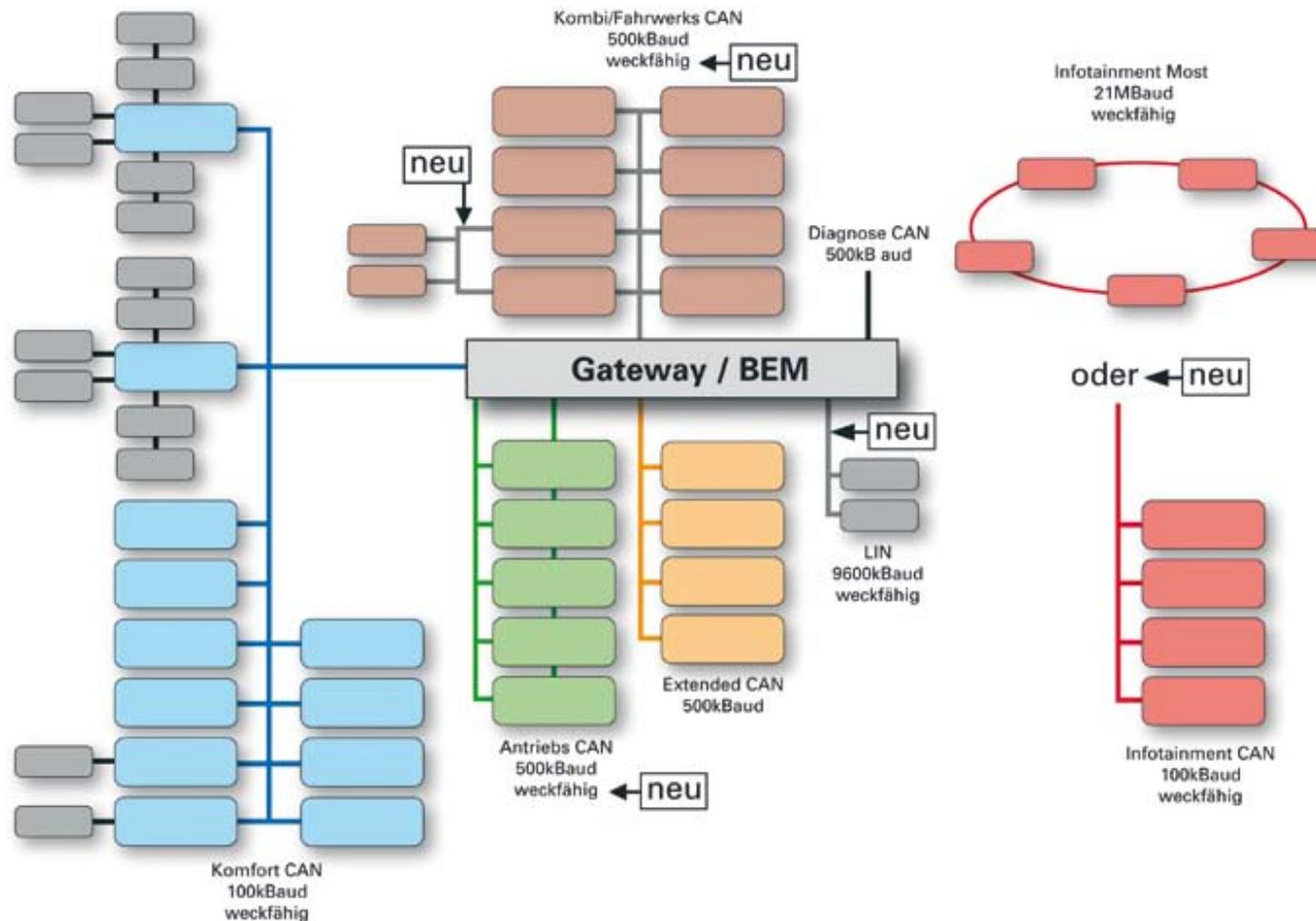
Im Audi A3 kommen ausschließlich Infotainmentsysteme auf Basis der CAN-Bustechnologie zum Einsatz, im Audi A8 kommunizieren die Infotainmentsysteme auf Basis der MOST-Technologie. Im Audi Q5 sollten beide Systemvarianten, CAN und MOST, ermöglicht werden.

Im Infotainment-Bereich vereint der neue Audi Q5 Elemente des Audi A3 (CAN) und des A8 (MOST)



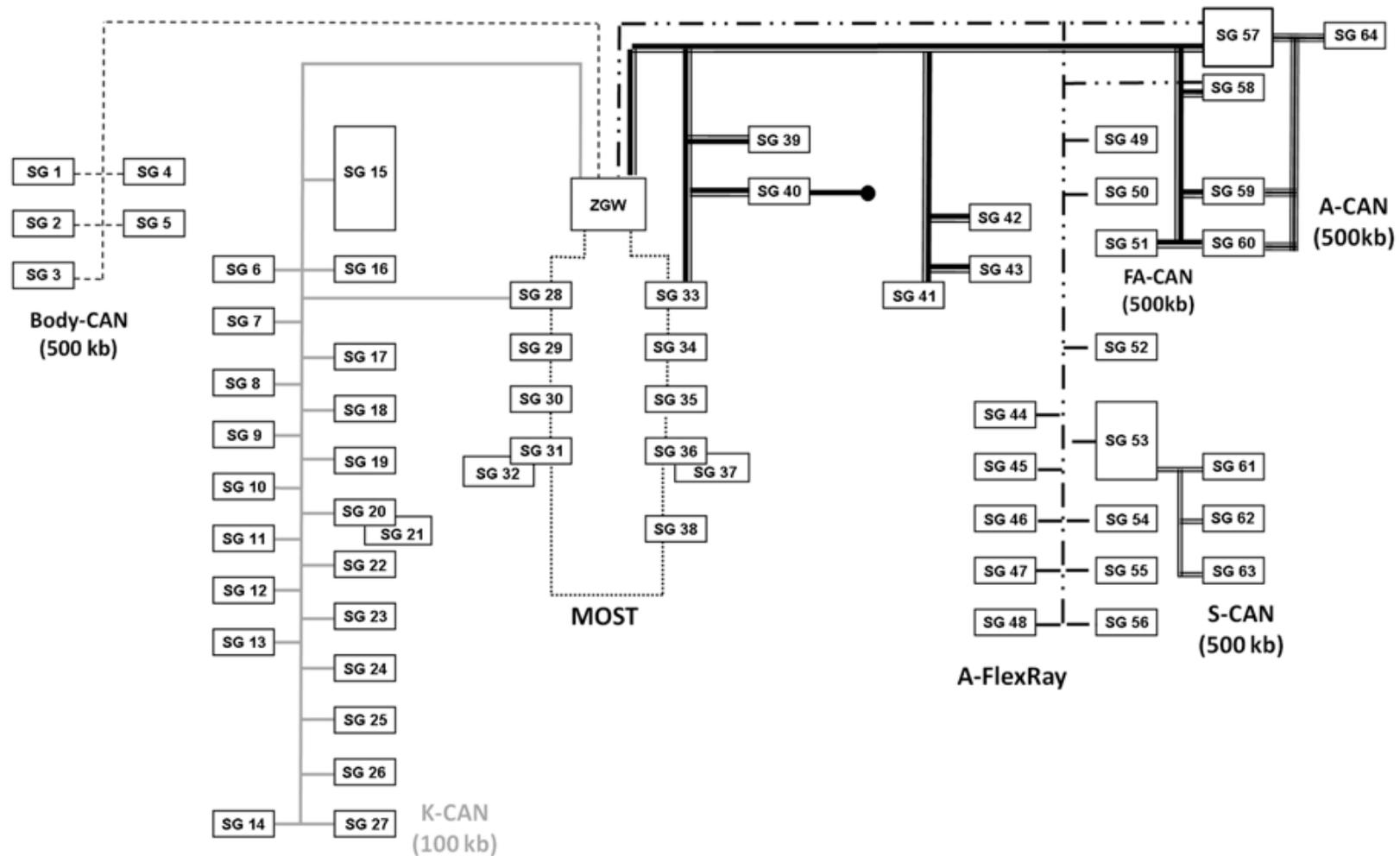
Im Audi A3 kommen ausschließlich Infotainmentsysteme auf Basis der CAN-Bustechnologie zum Einsatz, im Audi A8 kommunizieren die Infotainmentsysteme auf Basis der MOST-Technologie. Im Audi Q5 sollten beide Systemvarianten, CAN und MOST, ermöglicht werden.

Der Audi Q5 zeigt eine Elektronik-Gesamtarchitektur wie im Oberklasse-Segment



Des Weiteren musste das im Audi A8 bereits vorhandene Bussystem für die Fahrerassistenzsysteme eingesetzt werden. Zusätzlich wurde ein Konzept zur kostengünstigen Darstellung eines Basisumfangs analog Audi A3 notwendig. Die Abdeckung dieser Variantenvielfalt erforderte die Entwicklung einer

Datenkommunikationsbordnetz des BMW F01/F02



Quelle: F. Burgdorf: Eine kunden- und lebenszyklusorientierte Produktfamilienabsicherung für die Automobilindustrie, Dissertation, Karlsruher Institut für Technologie Fakultät für Elektrotechnik und Informationstechnik, 2010.