



TECHNISCHE
UNIVERSITÄT
DRESDEN

Vorlesung
Automotive Software Engineering
Teil 5 E/E-Entwicklung (1-2)
Sommersemester 2015

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

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

CAN (Controller Area Network)

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

CAN-Standard (1)

- Seit circa zwei Jahrzehnten unterliegt das Automobil einer regelrechten Elektronifizierung: Immer mehr elektronische Systeme sorgen für ein zunehmendes Maß an Sicherheit und Komfort beim Autofahren. Eine Schlüsselposition nehmen dabei serielle Kommunikationssysteme ein. Denn viele wertvolle elektronische Systeme wären ohne Datenaustausch nicht realisierbar.
- Die im Kfz am häufigsten eingesetzte serielle Kommunikationstechnologie ist CAN (Controller Area Network). Sie ist prädestiniert für den Einsatz im Kfz, da sie auch unter schwierigsten Umgebungsbedingungen einen sicheren Datenaustausch gewährleistet.
- Seit 1993 ist CAN standardisiert und liegt als ISO-Norm (International Standardization Organization) vor: ISO 11898. Während sie sich anfangs lediglich aus drei Teilen zusammensetzte, umfasst sie heute fünf Teile. Der erste Teil beschreibt das ereignisorientierte Kommunikationsprotokoll. Eine zeitgesteuerte Erweiterung findet man im vierten Teil.

CAN-Standard (2)

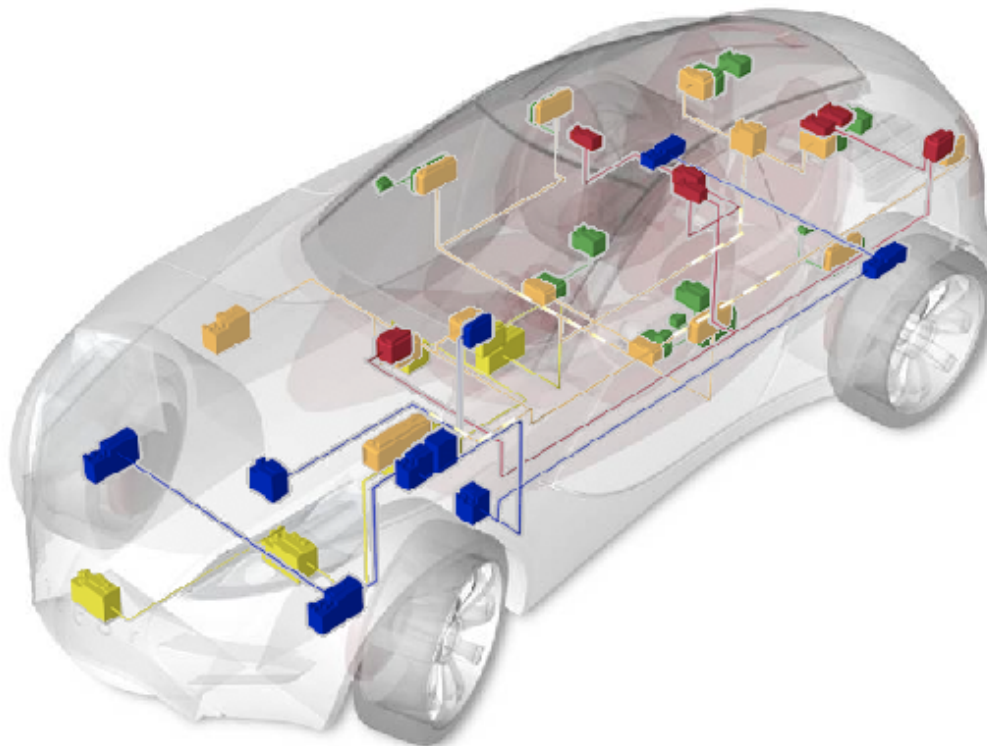
- Der zweite und dritte Teil umfasst Angaben zur Busankopplung und zur physikalischen Datenübertragung: Dabei unterscheidet man die High-Speed-Variante (Datenraten bis 1 MBit/s) von der Low-Speed-Variante (Datenraten bis 125 KBit/s). Der letzte Teil beschreibt das Verhalten eines CAN-Knotens im High-Speed-Netzwerk im „Low Power Mode“. Aus der Grafik CAN-Standard ist die Einordnung der für die Praxis derzeit bedeutsamen ersten drei Teile der ISO-Norm 11898 in das von der ISO veröffentlichte OSI-Modell (Open System Interconnection) ersichtlich.
- Speziell für den Einsatz im Automobil entwickelt, kommt CAN vor allem als High-Speed-CAN im Antrieb und Fahrwerk, sowie als Low-Speed-CAN im Komfort zum Einsatz.

CAN Lowspeed Bus (Innenraum/Karosserie-CAN)



Elektronikvernetzung im Kfz

✕ schließen



■ CAN H

■ **CAN L**

■ LIN

■ FlexRay

■ MOST

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

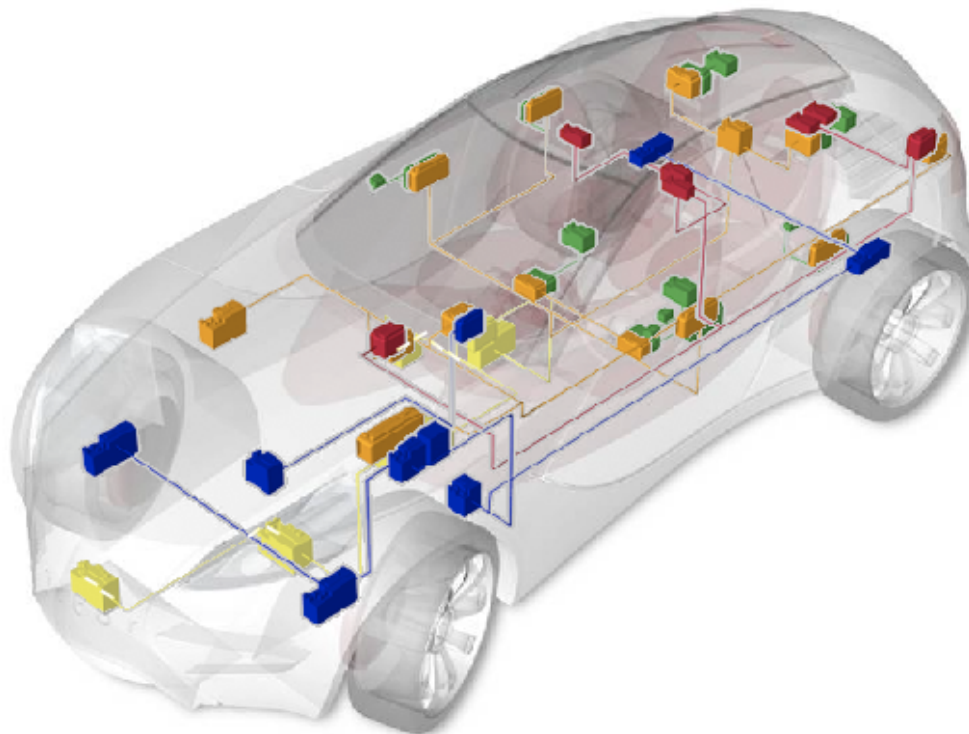
- Antriebsstrang (Powertrain)
- Fahrwerk (Chassis)
- **Karosserie (Body)**
- Multi-Media (Telematics)

CAN Highspeed Bus (Motor-CAN)



Elektronikvernetzung im Kfz

✕ schließen



- **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.

CAN (Controller Area Network)

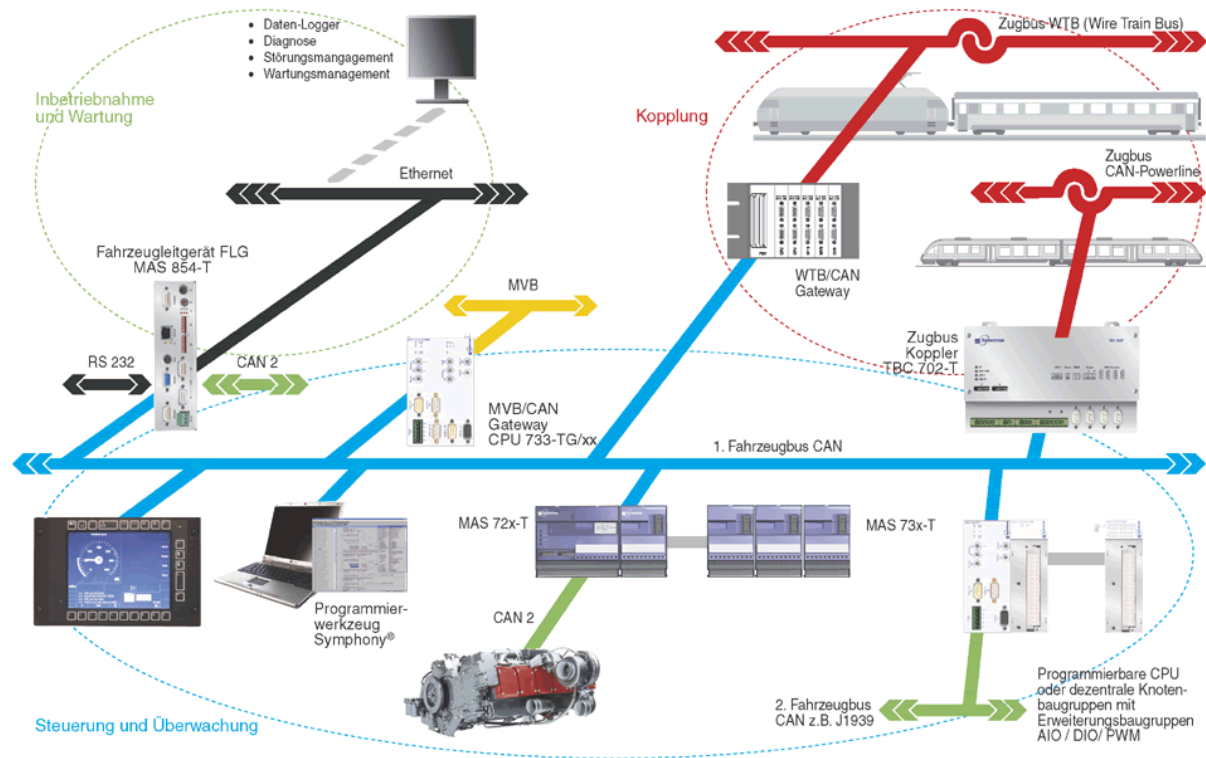


- Entwickelt von Bosch und Intel 1981
- ISO/OSI-Standard
- Eigenschaften:
 - Bandbreite bis 125 kbit/s (Lowspeed-CAN, Karosserie-CAN, Body-CAN)
 - Bandbreite bis 1000 kbit/s (Highspeed-CAN, Motor-CAN, Powertrain-CAN)
 - Asynchroner Betrieb ohne Busmaster mit Arbitrierung
 - Variante für Synchronbetrieb: 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/>

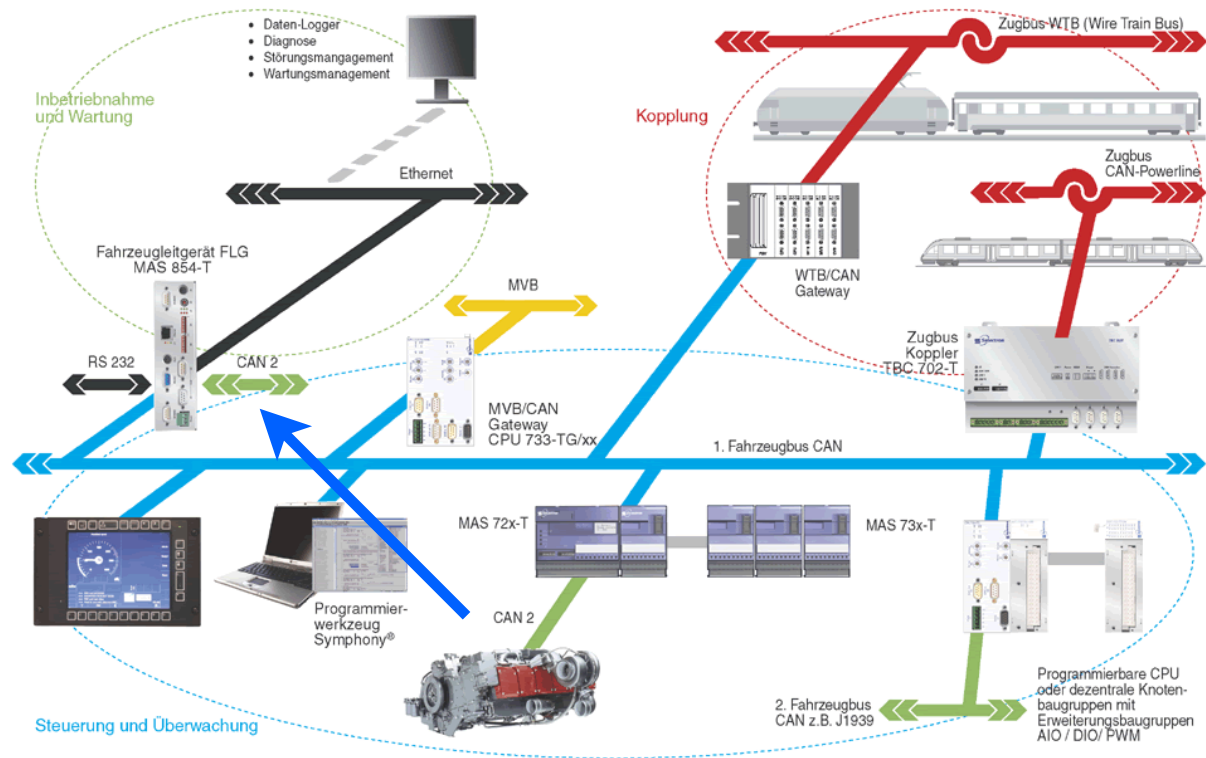
Applikationsbeispiel 3 - Schienenfahrzeuge

TCMS (Train Control and Monitoring System)



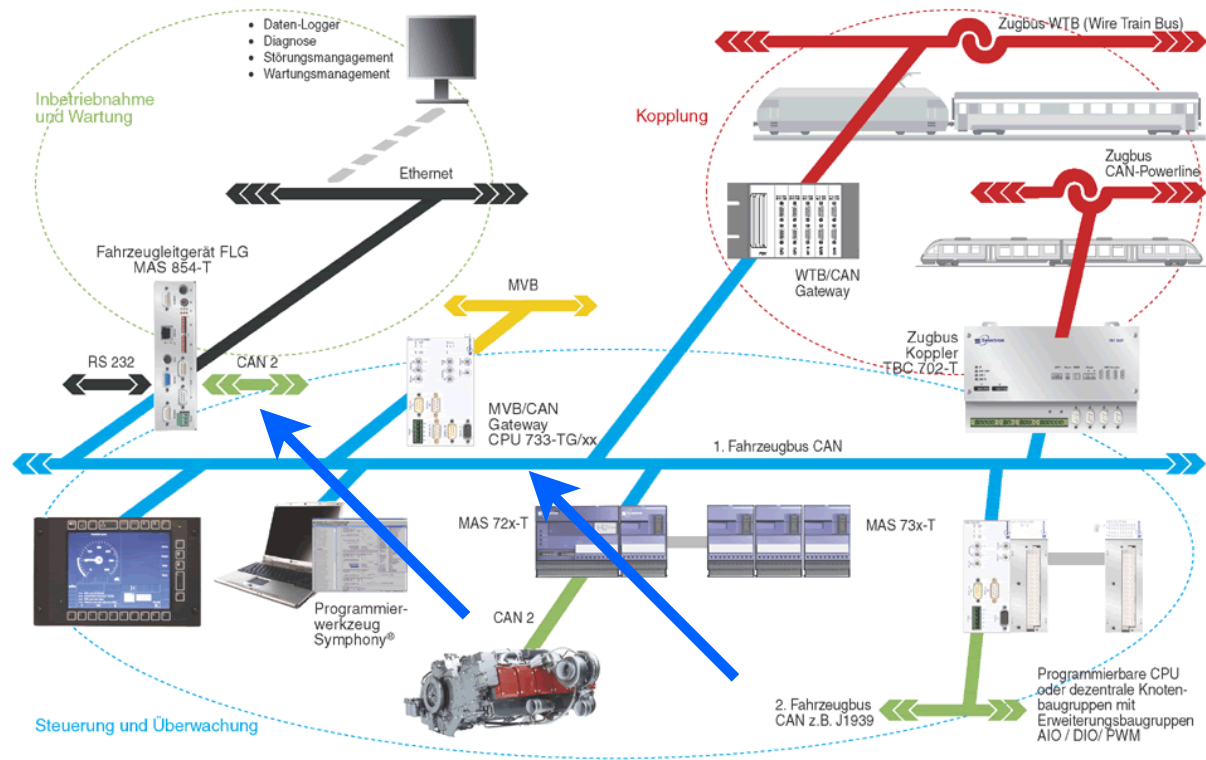
Applikationsbeispiel 3 - Schienenfahrzeuge

TCMS (Train Control and Monitoring System)



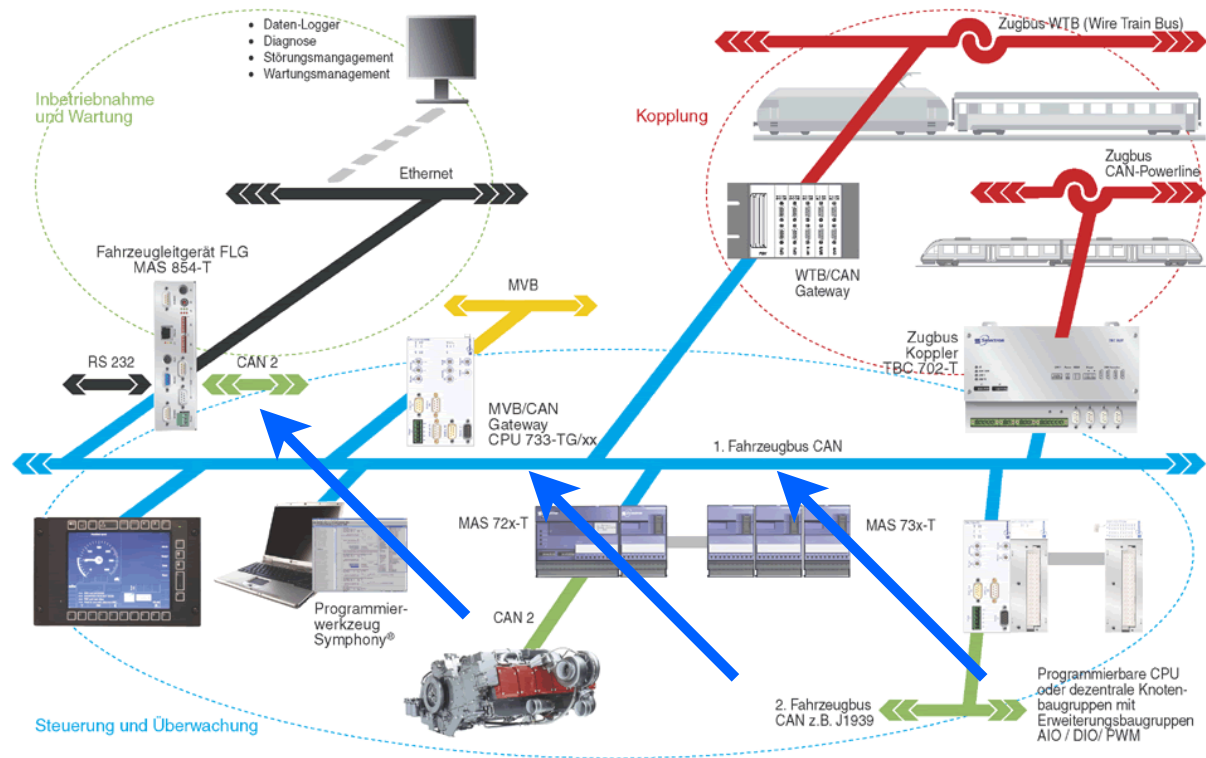
Applikationsbeispiel 3 - Schienenfahrzeuge

TCMS (Train Control and Monitoring System)



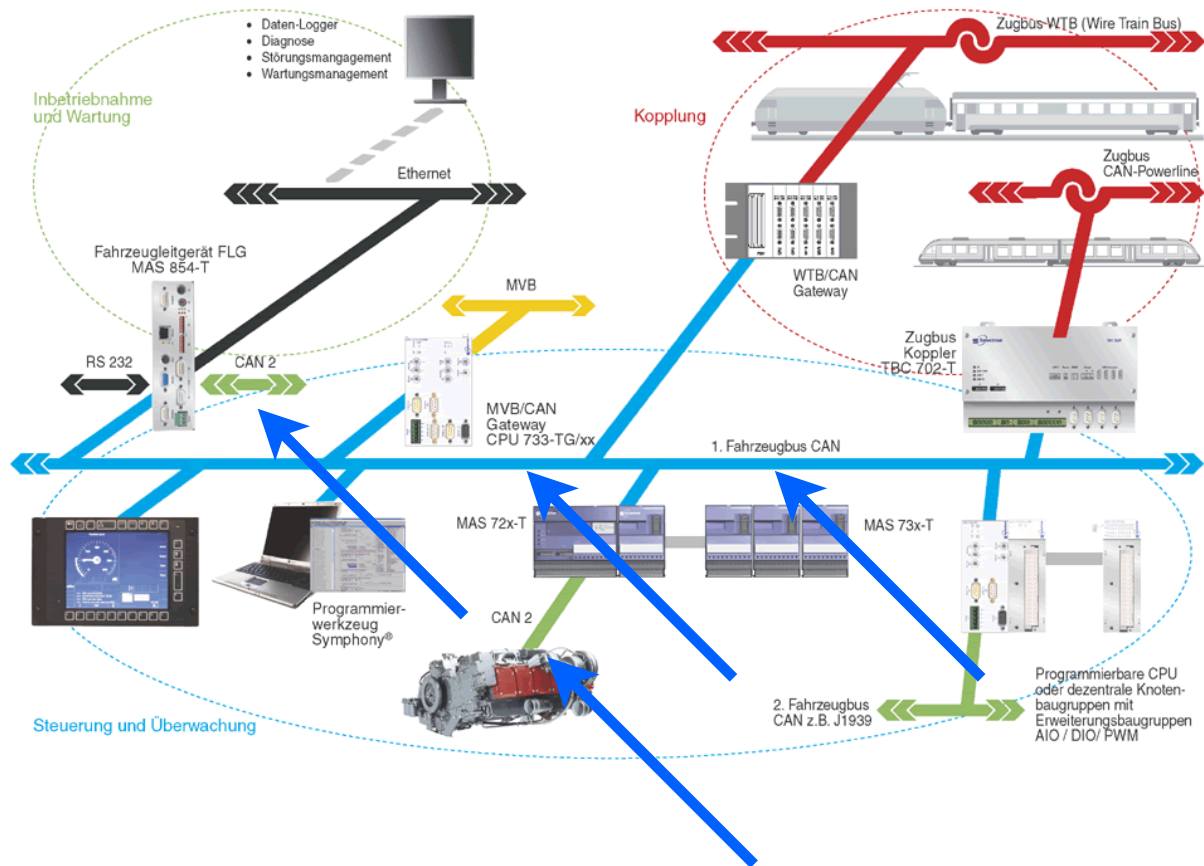
Applikationsbeispiel 3 - Schienenfahrzeuge

TCMS (Train Control and Monitoring System)



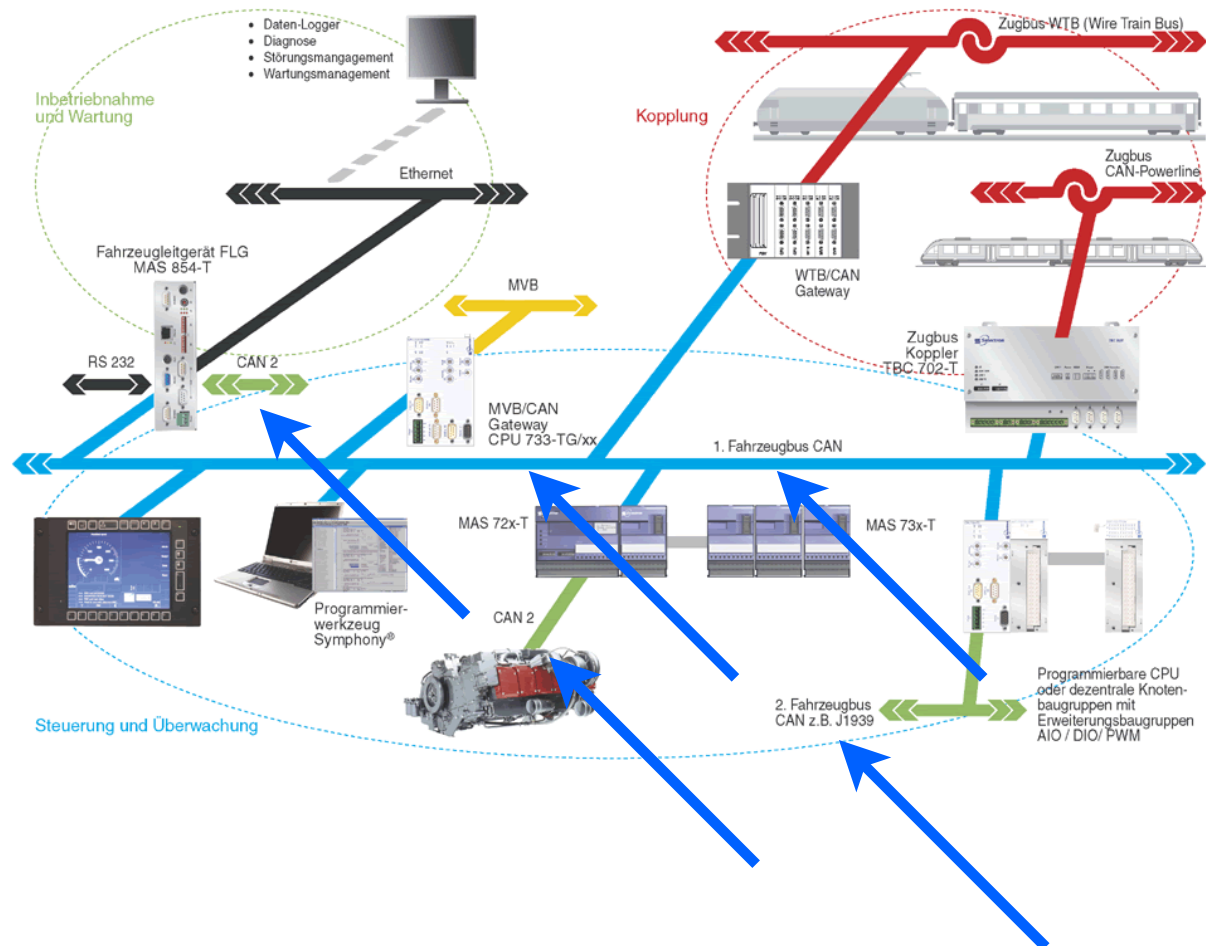
Applikationsbeispiel 3 - Schienenfahrzeuge

TCMS (Train Control and Monitoring System)



Applikationsbeispiel 3 - Schienenfahrzeuge

TCMS (Train Control and Monitoring System)

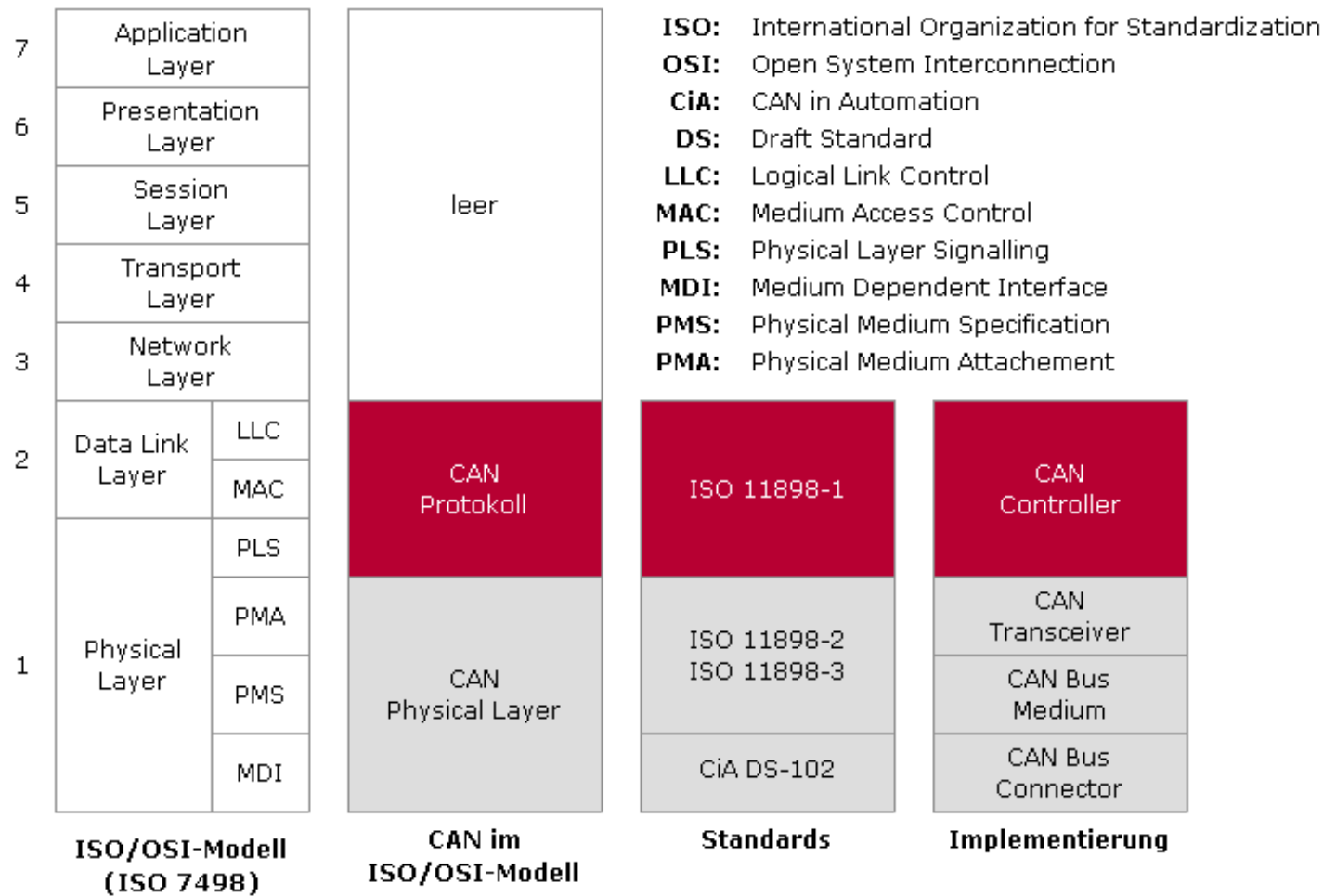


CAN-Standard

- 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 (1)

- Ein CAN-Netzwerk ist ein Systemverbund aus CAN-Knoten (elektronische Steuergeräte mit CAN-Schnittstelle), die über ihre jeweiligen CAN-Schnittstellen und einem alle CAN-Schnittstellen verbindendes Übertragungsmedium (CAN-Bus) untereinander Daten austauschen. Eine CAN-Schnittstelle setzt sich aus zwei Teilen zusammen: der Kommunikationssoftware und der Kommunikationshardware.
- Während die Kommunikationssoftware höhere Kommunikationsdienste umfasst, sind die grundlegenden Kommunikationsfunktionen in Hardware implementiert: Der CAN-Controller sorgt für die einheitliche Abwicklung des CAN-Kommunikationsprotokolls, der CAN-Transceiver sorgt für die Ankopplung des CAN-Controllers an den CAN-Bus.
- Als Übertragungsmedium kommt üblicherweise eine verdrehte Zweidrahtleitung (Twisted Pair) zum Einsatz. Die darüber stattfindende symmetrische Signalübertragung ist sehr unempfindlich gegenüber äußeren Störungen. Die Buspegel hängen von der eingesetzten Busankopplung ab. Man unterscheidet zwischen der CAN-High-Speed-Busankopplung (ISO 11898-2) und der CAN-Low-Speed-Busankopplung (ISO 11898-3).

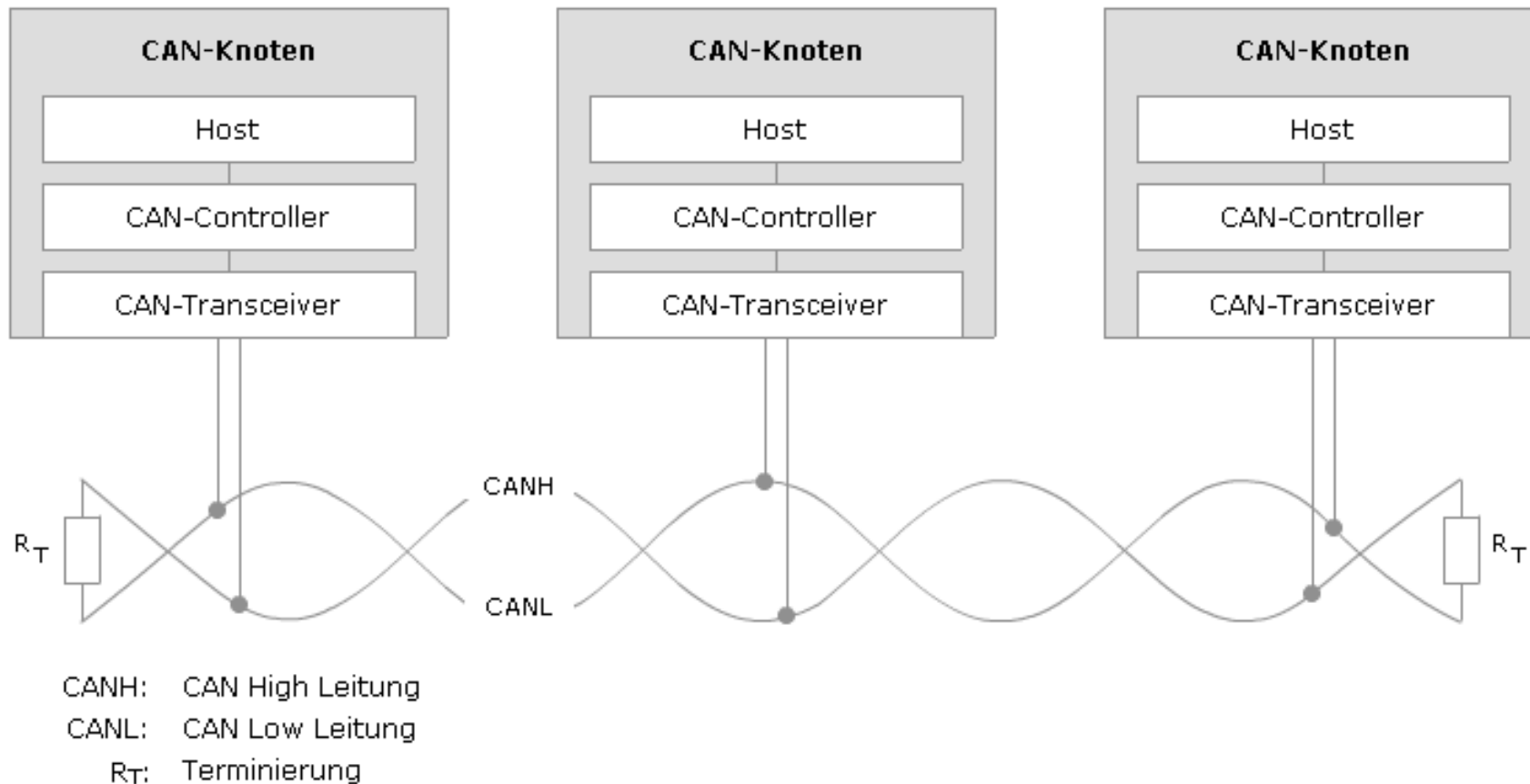
CAN-Netzwerk (2)

- 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



CAN-Datenübertragung (1)

- Die Datenübertragung im CAN-Netzwerk erfolgt mittels Nachrichtenrahmen, den so genannten „CAN Data Frames“. Nutzdaten bis zu acht Byte Länge können mit einem Data Frame im Data Field übertragen werden. Die genaue Anzahl der Nutzbytes geht aus dem DLC (Data Length Code) hervor. Jeder Data Frame steht jedem CAN-Knoten zur Übernahme zur Verfügung (Broadcasting).
- Gesichert werden die Nutzdaten mithilfe des CRC-Verfahrens (Cyclic Redundancy Check): Der Sender hängt den Nutzdaten eine Prüfsumme (CRC-Sequenz) an, die vom Empfänger unter Anwendung des CRC-Algorithmus ausgewertet wird. In Abhängigkeit des Auswertergebnisses quittiert der Empfänger im Anschluss an die CRC-Sequenz im ACK-Slot positiv oder negativ.
- Jeder Data Frame wird über einen Identifier gekennzeichnet. Dieser umfasst im Standard-Format 11 Bit, im Extended-Format 29 Bit. In Abhängigkeit des Formats stehen in einem CAN-Netzwerk somit entweder 211 oder 229 Identifier zur Kennzeichnung von Data Frames zur Verfügung. Die Grafik zeigt einen CAN Data Frame im Standard-Format.

CAN-Datenübertragung (2)

- Die Übertragung eines Data Frames beginnt mit einem Startzeichen (Start of Frame: SOF). Es dient den CAN-Knoten dazu, sich mit dem Sender zu synchronisieren. Während der Übertragung halten die CAN-Knoten den Takt, indem sie jeden Flankenwechsel auswerten und ggf. den Takt anpassen. Das Bitstuffing-Verfahren garantiert, dass nach spätestens fünf homogenen Bits ein Flankenwechsel auftritt. Ein Endezeichen (End Of Frame: EOF) markiert das Ende eines Data Frames.

Framing (1)

- Die Datenkommunikation in einem seriellen Bussystem erfolgt mittels sog. Frames. Bei einem Frame handelt es sich um einen Rahmen (Nachricht, Botschaft), der neben den eigentlichen Nutzdaten weitere Informationen enthält. Dabei handelt es sich im Wesentlichen um Informationen zur Herstellung einer eindeutigen Zuordnung zwischen Nutzdaten und Busknoten (Adressierung), um Informationen zur Synchronisierung der Kommunikationspartner und um Informationen zur Sicherung der Nutzdaten. Es folgt die grobe Beschreibung der bei CAN, LIN und FlexRay am häufigsten eingesetzten Frames.
- Der im CAN-Netzwerk zur Datenübertragung zum Einsatz kommende Data Frame beginnt mit einem Synchronisationsbit, dem sog. Startbit (Start Of Frame - SOF). Diesem folgt die Kennung (Identifizier) des Frames. Das folgende RTR-Bit (Remote Transmission Request) zeigt den Frametyp (Data oder Remote Frame) an. Über das sich anschließende IDE-Bit (Identifizier Extension) zeigt der Sender das Frame-Format (Standard- bzw. Extended-Format) an: Im Standard-Format umfasst der Identifizier 11 Bit, im Extended-Format 29 Bit.

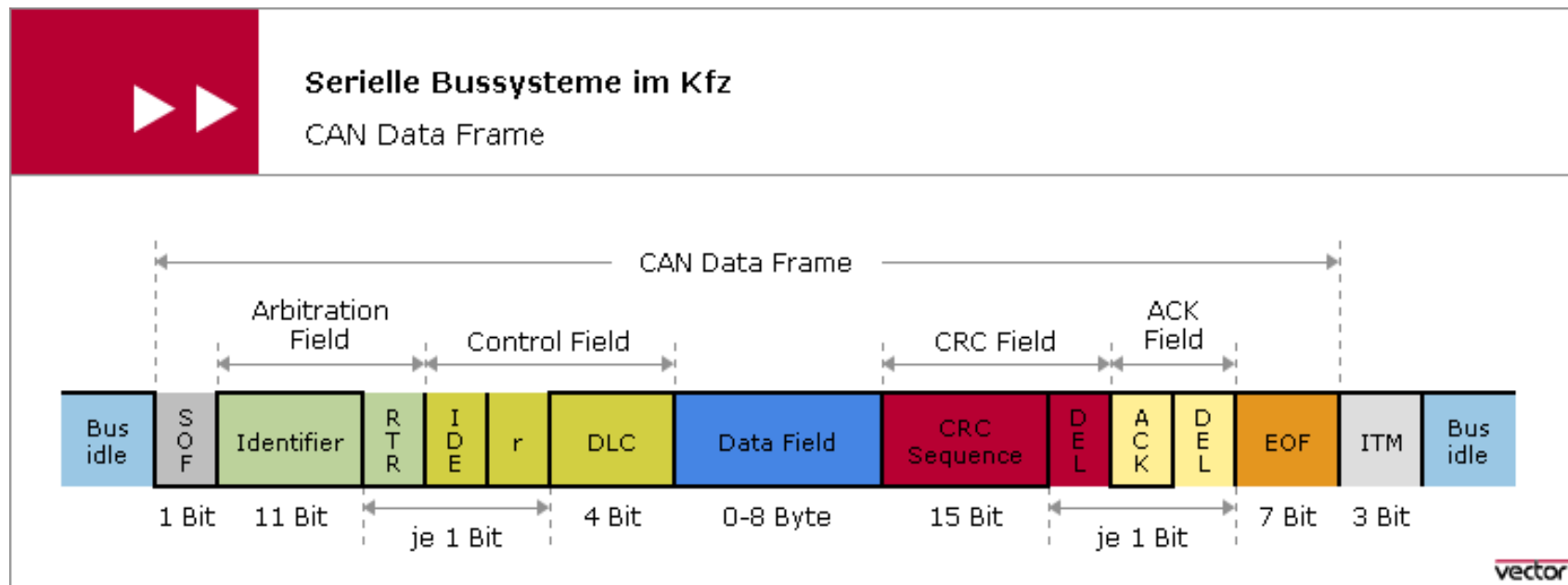
Framing (2)

- Dann folgt der DLC (Data Length Code), der die Anzahl der Nutzbytes anzeigt. Erst dann folgt das Data Field, welches maximal acht Nutzbytes umfasst. Daran schließt sich zur Sicherung sämtlicher Informationen und Nutzdaten die CRC-Sequenz an. Noch vor dem Endezeichen EOF (End Of Frame) befindet sich das ACK-Field (Acknowledgement).

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.

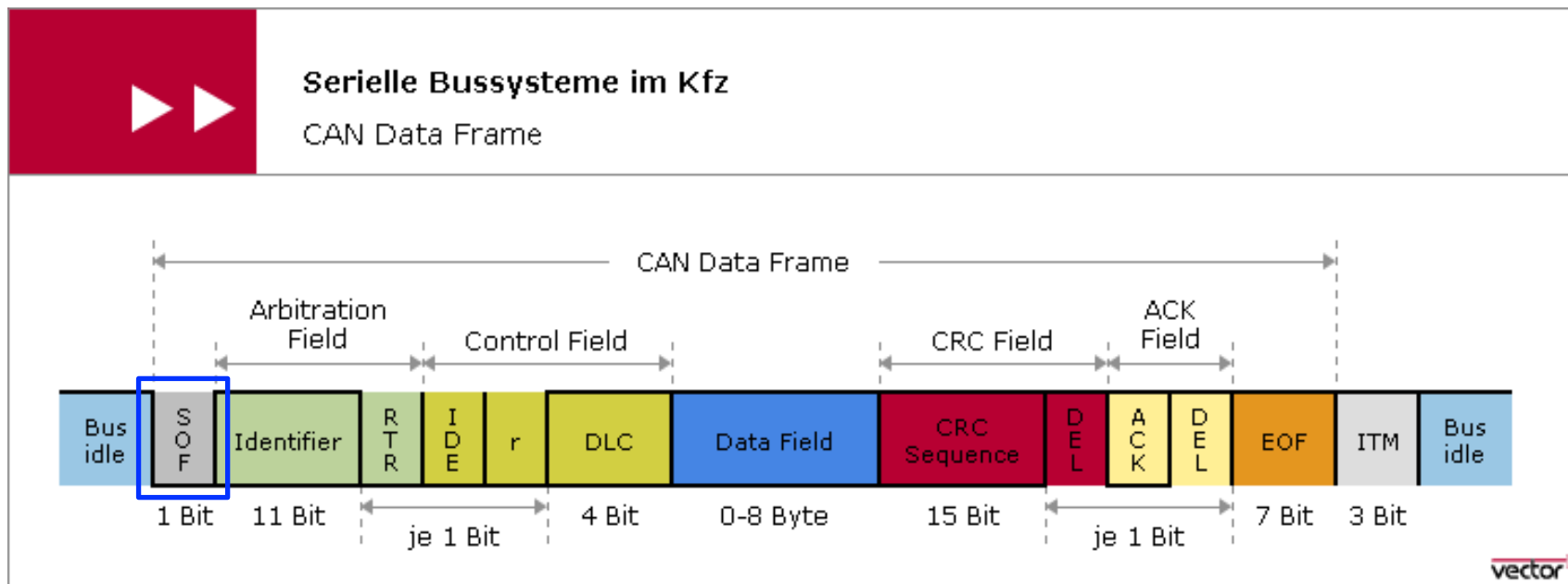
CAN-Datenübertragung (2)



CAN-Datenübertragung (3)

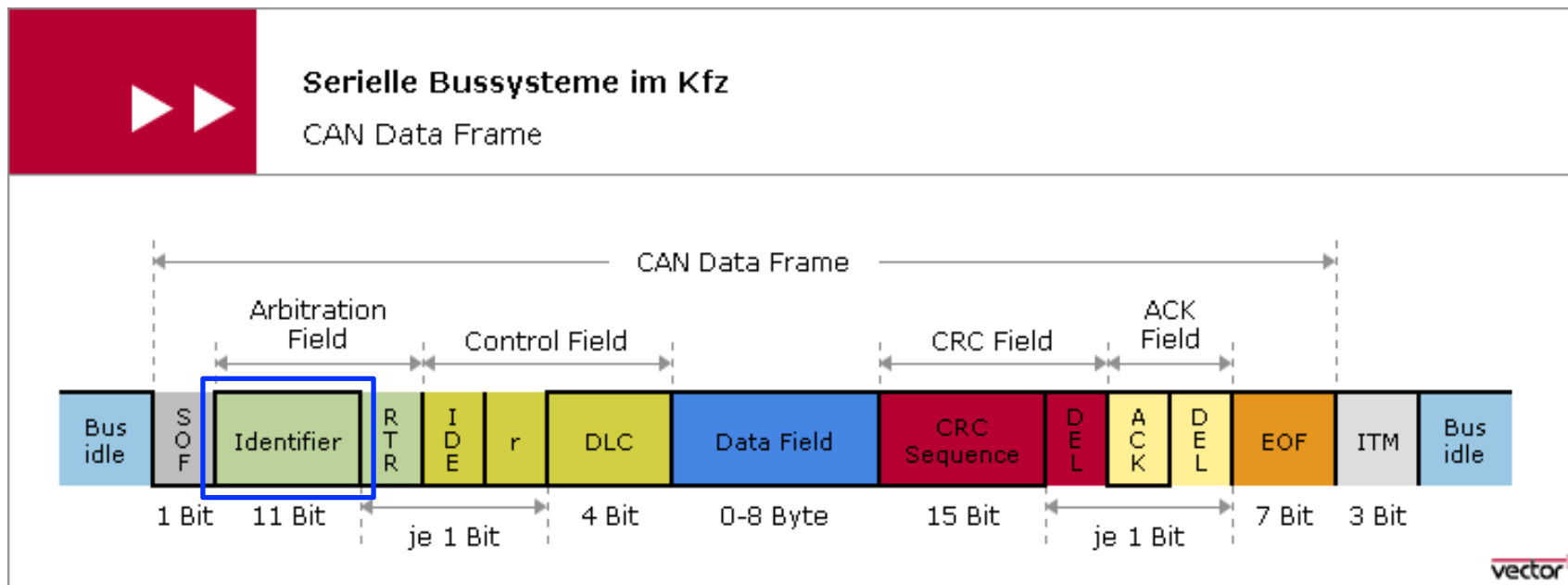
- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)
- Kennung (Identifizier) des Frames.
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).
- IDE-Bit (Identifizier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - 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)

CAN-Datenübertragung (3)



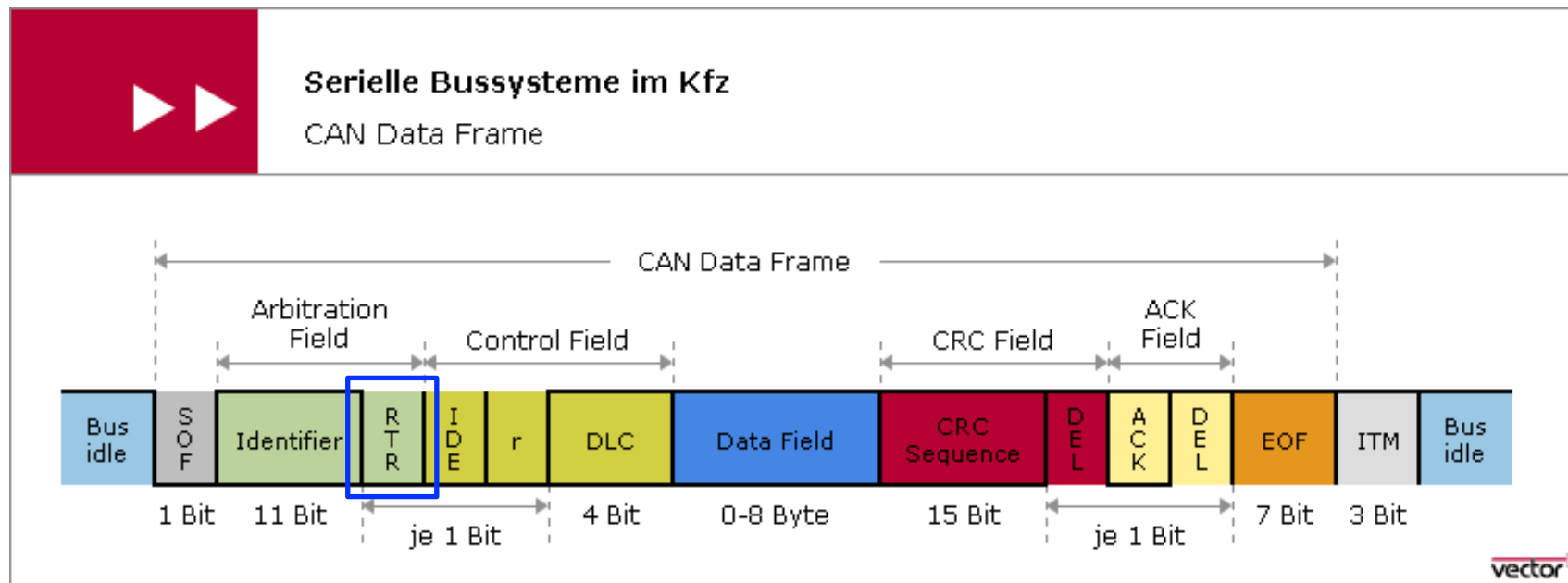
- Startzeichen (Start of Frame: SOF): Synchronisation des CAN-Knotens mit dem Sender (Synchronisationsbit)

CAN-Datenübertragung (3)



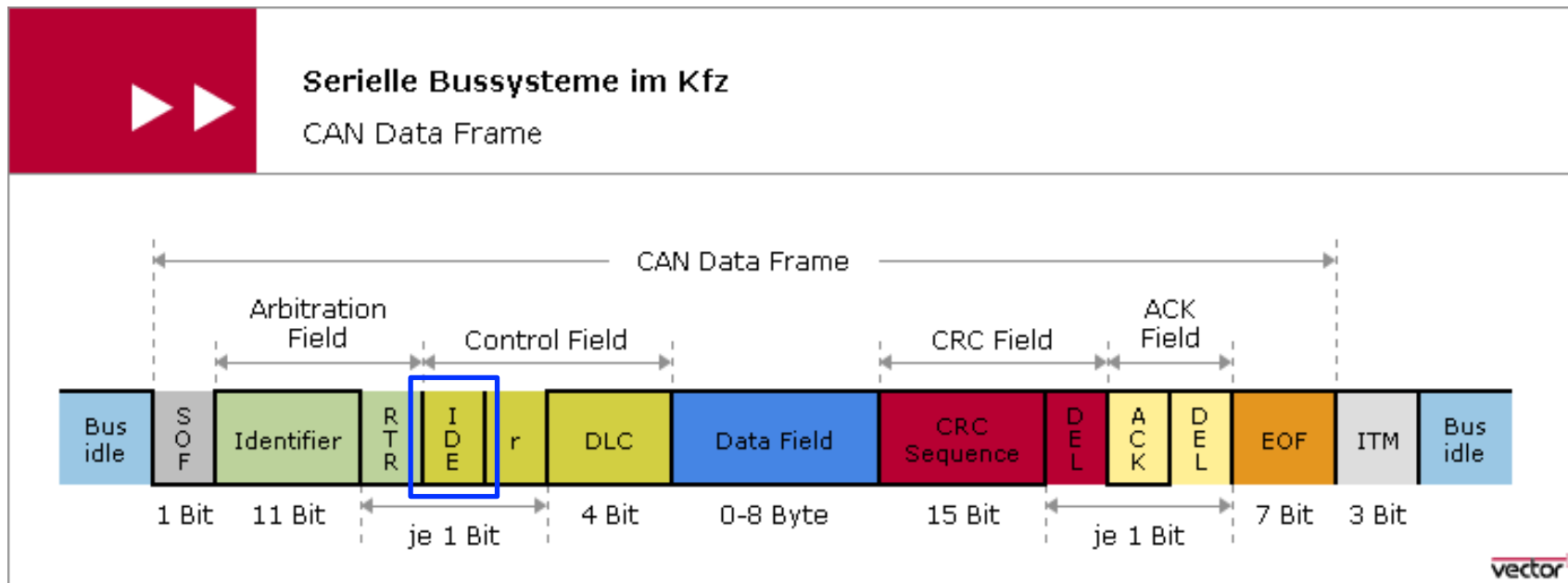
- Kennung (Identifizier) des Frames.

CAN-Datenübertragung (3)



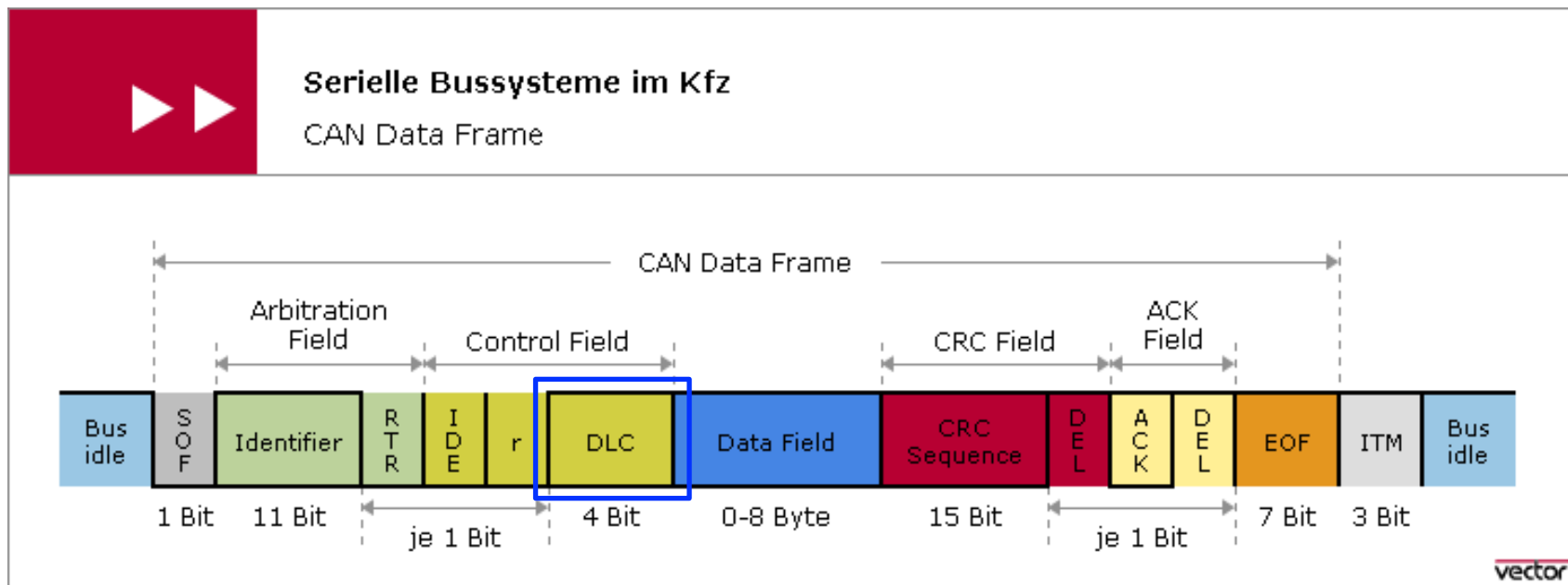
- RTR-Bit (Remote Transmission Request): Frametyp (Data oder Remote Frame).

CAN-Datenübertragung (3)



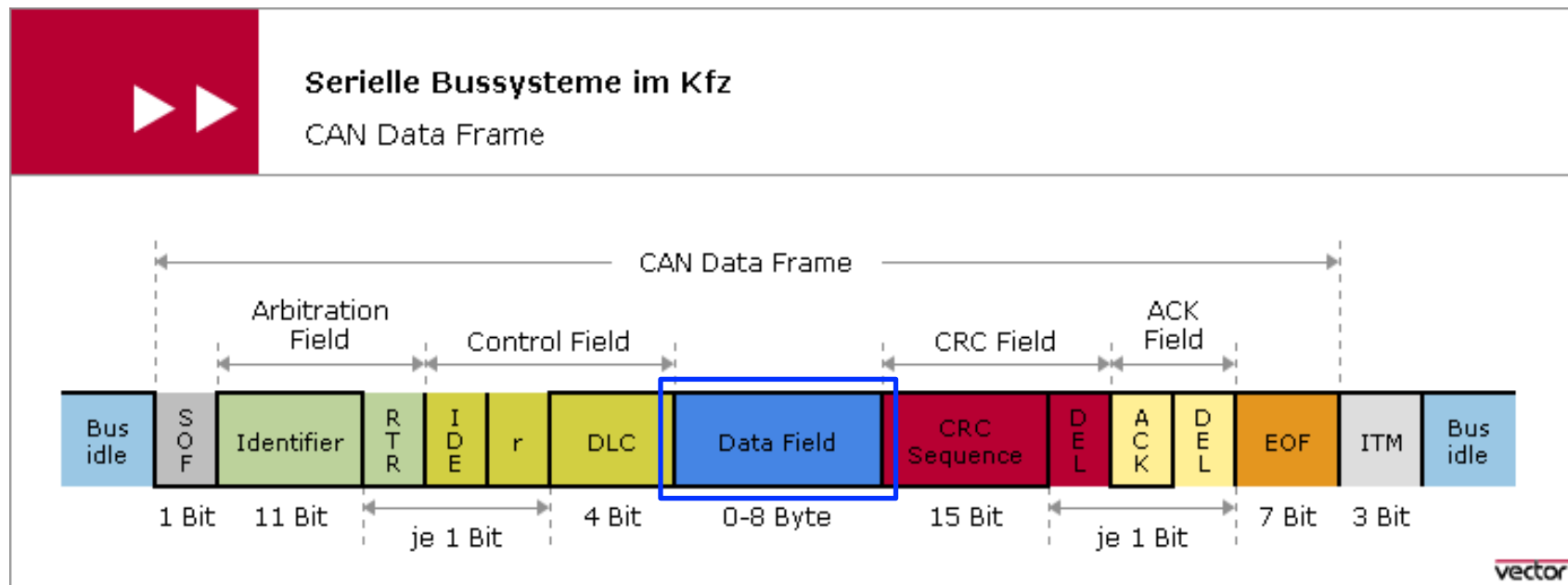
- IDE-Bit (Identifier Extension): Frame-Format (Standard- bzw. Extended-Format):
 - Im Standard-Format umfasst der Identifier 11 Bit, im Extended-Format 29 Bit.

CAN-Datenübertragung (3)



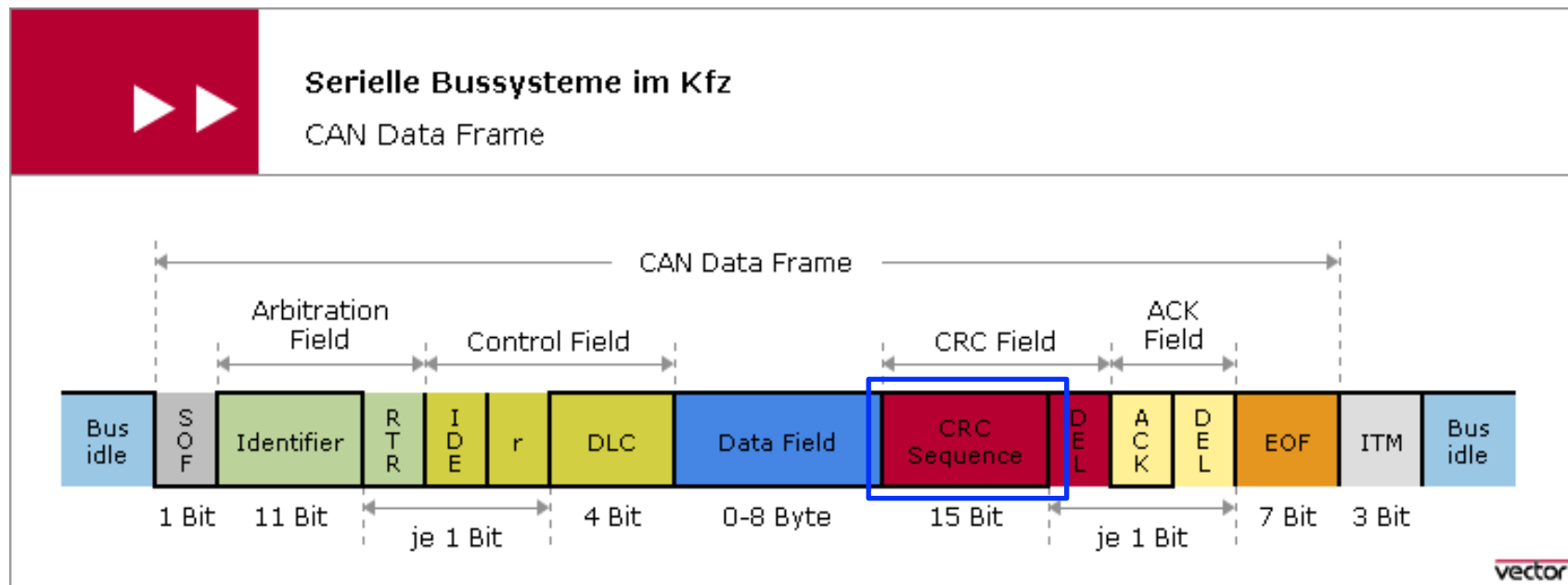
- **DLC (Data Length Code):** Anzahl der Nutzbytes

CAN-Datenübertragung (3)



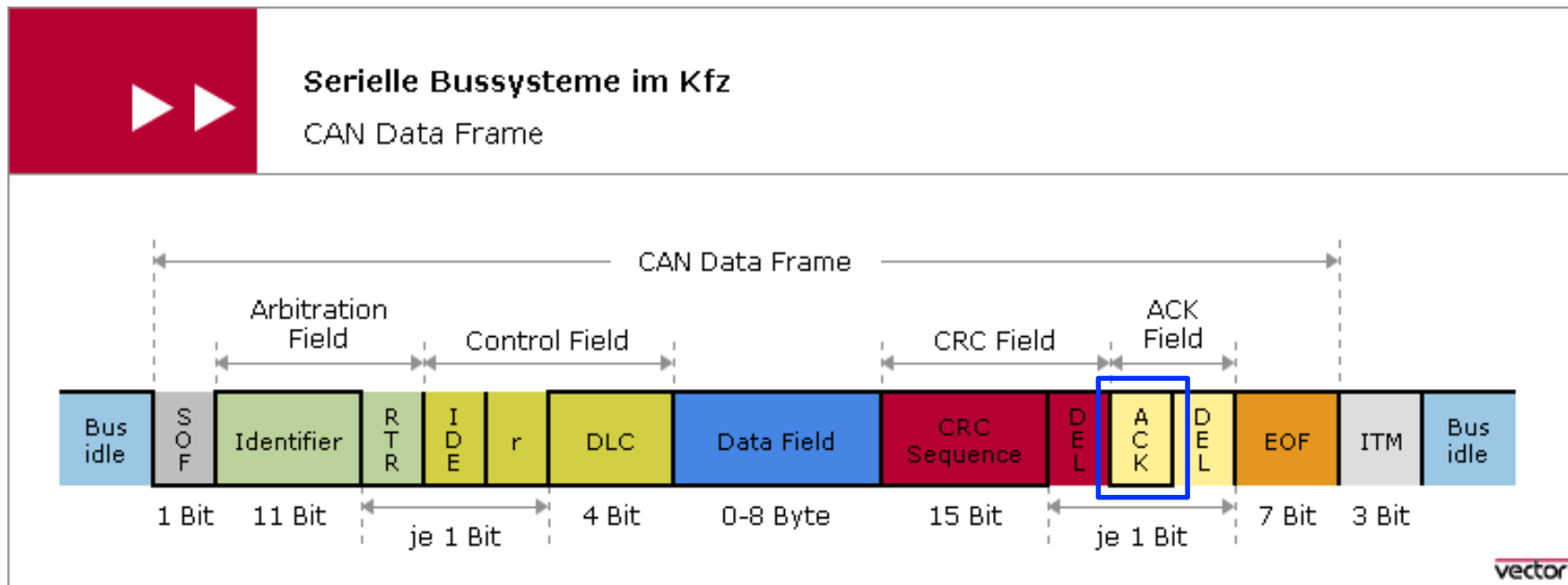
- Data Field: maximal acht Nutzbytes

CAN-Datenübertragung (3)



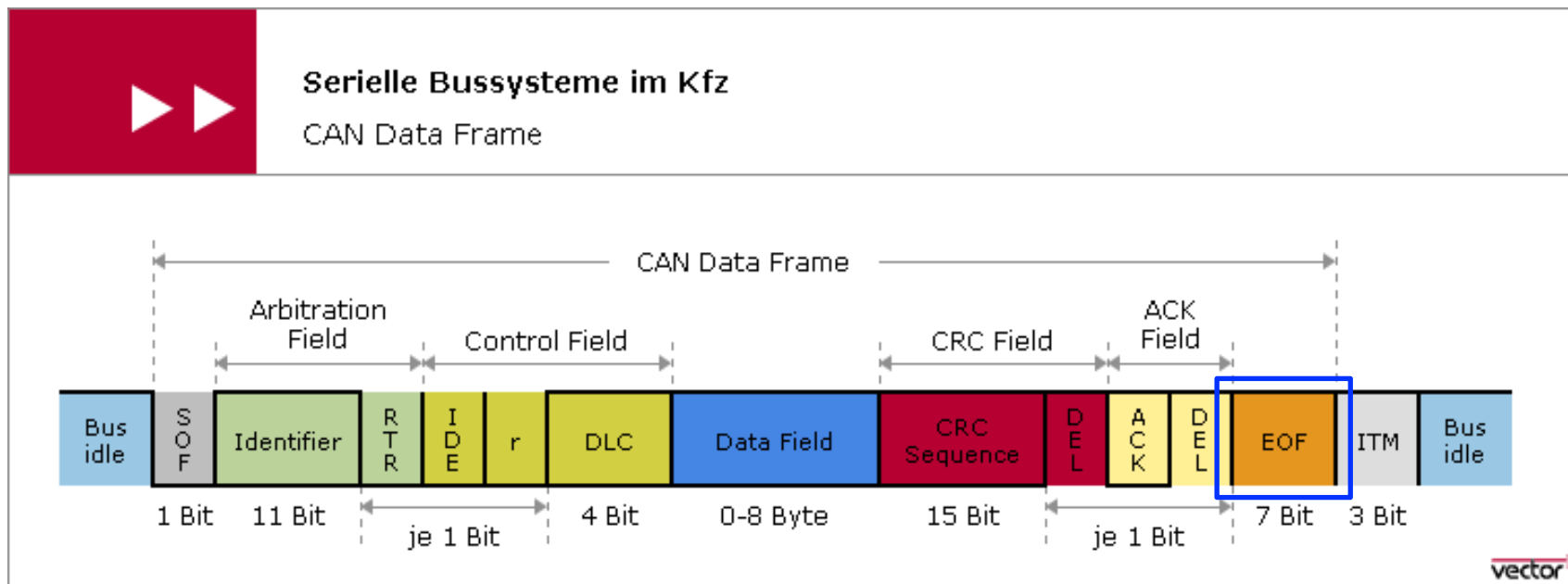
- CRC-Sequenz (Sicherung von Informationen und Nutzdaten).

CAN-Datenübertragung (3)



- ACK-Field (Acknowledgement).

CAN-Datenübertragung (3)



- Endezeichen (End Of Frame: EOF)

CAN-Kommunikationsprinzip (1)

- Die übertragenen Data Frames und deren Reihenfolge sind in einem CAN-Netzwerk nicht vom Fortschreiten der Zeit abhängig, sondern vom Auftreten spezieller Ereignisse. Jeder CAN-Knoten ist prinzipiell berechtigt, sofort nach Auftreten eines Ereignisses auf den CAN-Bus zuzugreifen. In Verbindung mit der vergleichsweise kurzen Nachrichtenlänge von maximal 130 Bit im Standard-Format und der hohen Datenübertragungsrate von bis zu 1 MBit/s ermöglicht das Verfahren schnelle Reaktionen auf asynchrone Vorgänge.
- Dies ist eine wichtige Voraussetzung für eine echtzeitfähige Datenübertragung im unteren Millisekundenbereich, die vor allem die Applikationen des Antriebs und des Fahrwerks verlangen. Da der CAN-Kommunikation kein Zeitplan zugrunde liegt, ergibt sich der Nachrichtenverkehr stets erst zur Laufzeit und birgt deshalb die Gefahr von Kollisionen. Diese steigt mit zunehmender Buslast und stellt die Echtzeitfähigkeit in Frage.
- Um trotz zufälligem Buszugriffs Echtzeitkommunikation zu garantieren, liegt dem Buszugriff das CSMA/CA-Verfahren (Carrier Sense Multiple Access/Collision Avoidance) zugrunde. Das CSMA/CA-Verfahren sorgt zum einen dafür, dass sendewillige CAN-Knoten erst dann auf den CAN-Bus zugreifen, wenn dieser frei ist. Zum anderen sorgt das CSMA/CA-Verfahren bei simultanen Buszugriffen dafür, dass sich immer jener CAN-Knoten mit dem höchst prioren Data Frame durchsetzt.

CAN-Kommunikationsprinzip (2)

- Mit zunehmender Buslast wachsen die Verzögerungen niederpriorer Data Frames an, was zur Beeinträchtigung der Echtzeitfähigkeit führen kann. Nutzen Sie die interaktive Grafik „Prinzip des Buszugriffs“ um sich den Buszugriff im CAN-Netzwerk zu veranschaulichen.

CAN-Kommunikationsprinzip (3)


- 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)
- Schnelle Reaktion auf asynchrone Vorgänge im unteren Millisekundenbereich
 - Echtzeitfähige Datenübertragung (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).

CAN-Kommunikationsprinzip (4)

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

▶▶

CAN-Kommunikation
CAN-Kommunikationsprinzip



CAN-Knoten A

Host

CAN-Schnittstelle

Sende-puffer

Senden

Empfangs-puffer

Empfangen

Akzeptanz-prüfung

CAN-Bus

Empfangen

Akzeptanz-prüfung

Empfangs-puffer

Sende-puffer

Senden

CAN-Schnittstelle

Host

CAN-Knoten C

CAN-Knoten B

Host

CAN-Schnittstelle

Sende-puffer

Senden

ID=0x52

Akzeptanz-prüfung

Empfangen

CAN-Bus

Empfangen

Akzeptanz-prüfung

Empfangs-puffer

Sende-puffer

Senden

CAN-Schnittstelle


Host

CAN-Knoten D

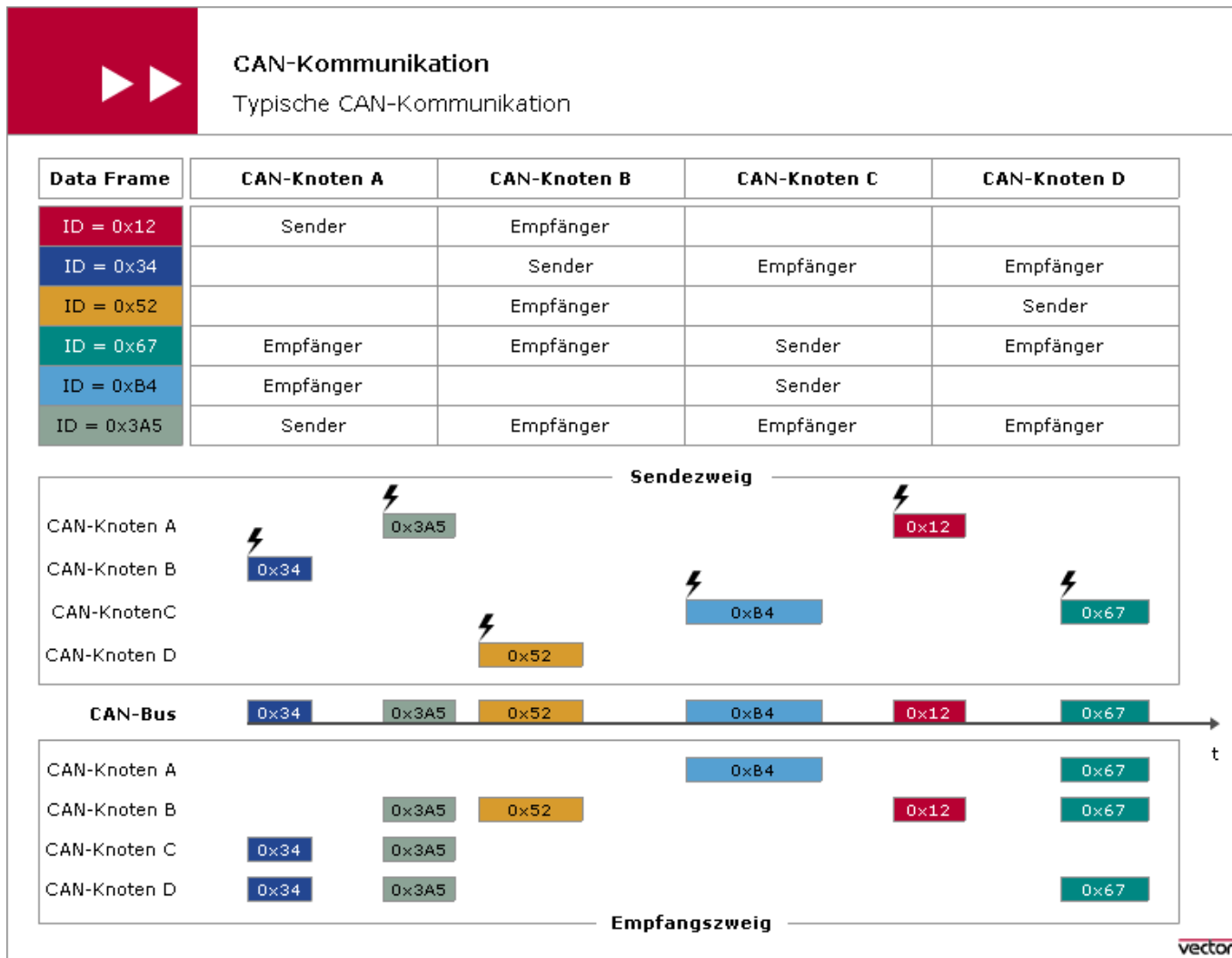
Kommunikationsmatrix

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

Der CAN-Controller des CAN-Knoten B informiert den Host über den Empfang von Nutzdaten (Interrupt) bzw. der Host bemerkt den Empfang von Nutzdaten (Polling). Der Host des CAN-Knoten B liest die Nutzdaten aus dem Empfangspuffer aus.

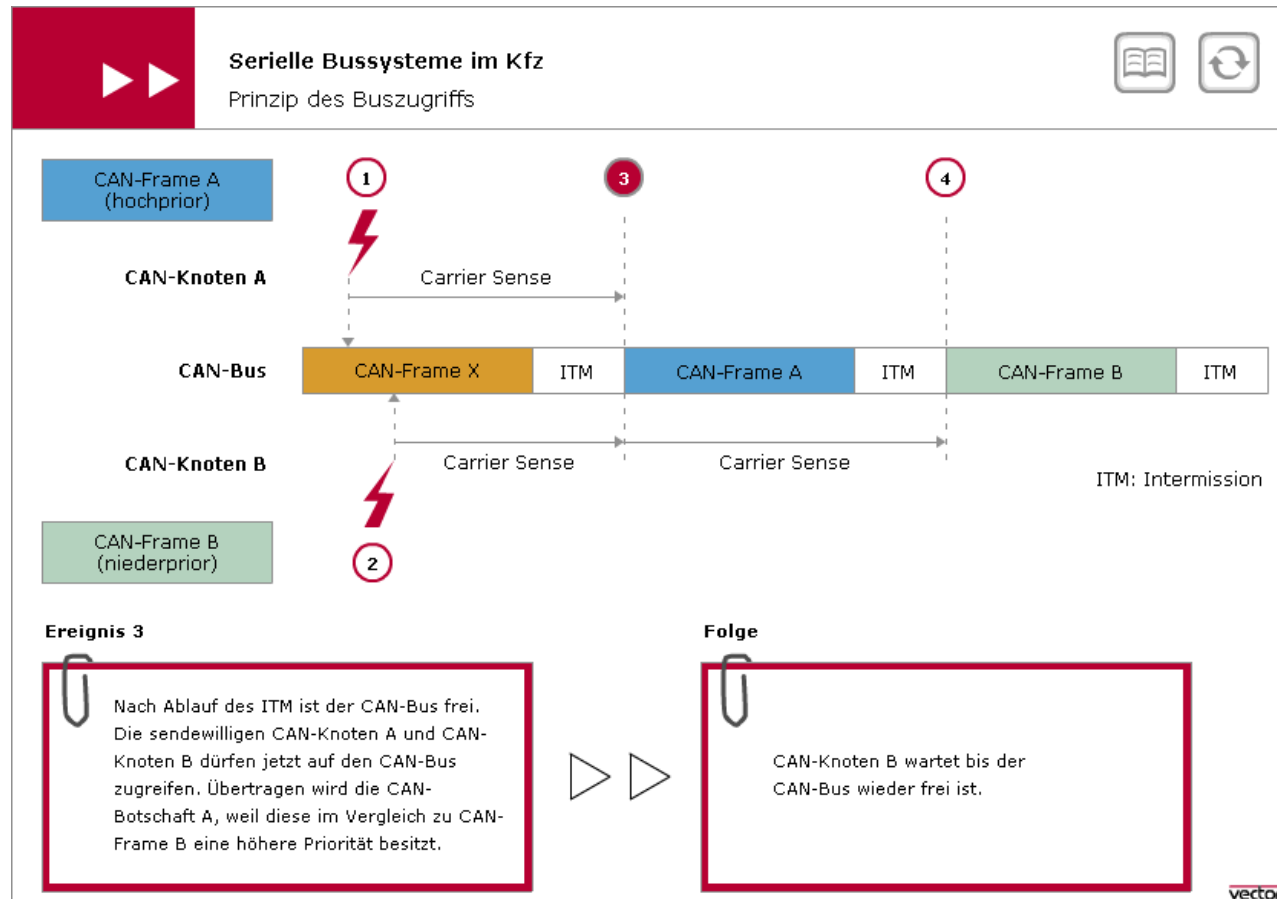


Typische CAN-Kommunikation



Prinzip des Buszugriffs

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



CAN-Datensicherung (1)

- 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)
 - Überprüfung der Einhaltung der Bitstuffingregel (Stuff Check)

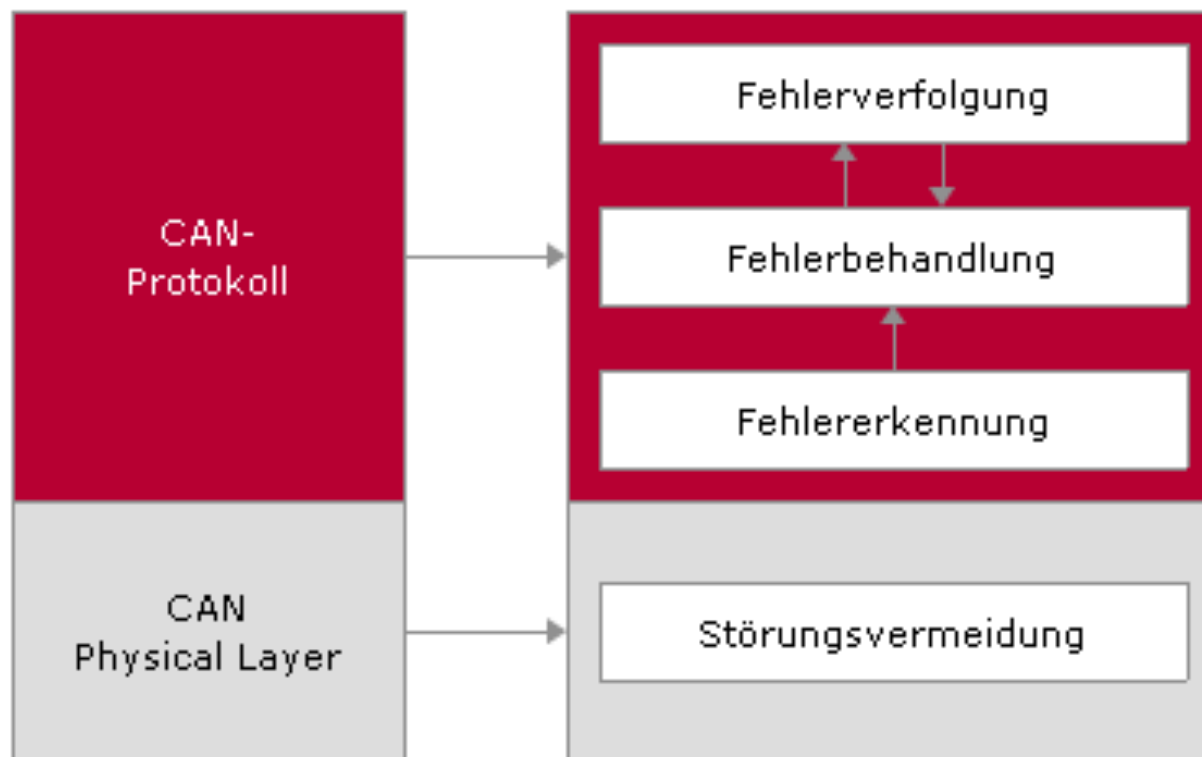
CAN-Datensicherung (2)

- 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.



Datensicherung im CAN-Netzwerk

Prinzip der Datensicherung

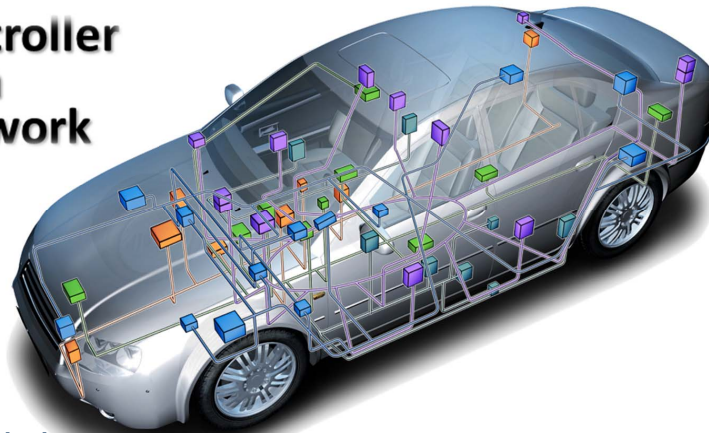


CAN FD (Flexible Data Rate)

- 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

Serielle Bussysteme

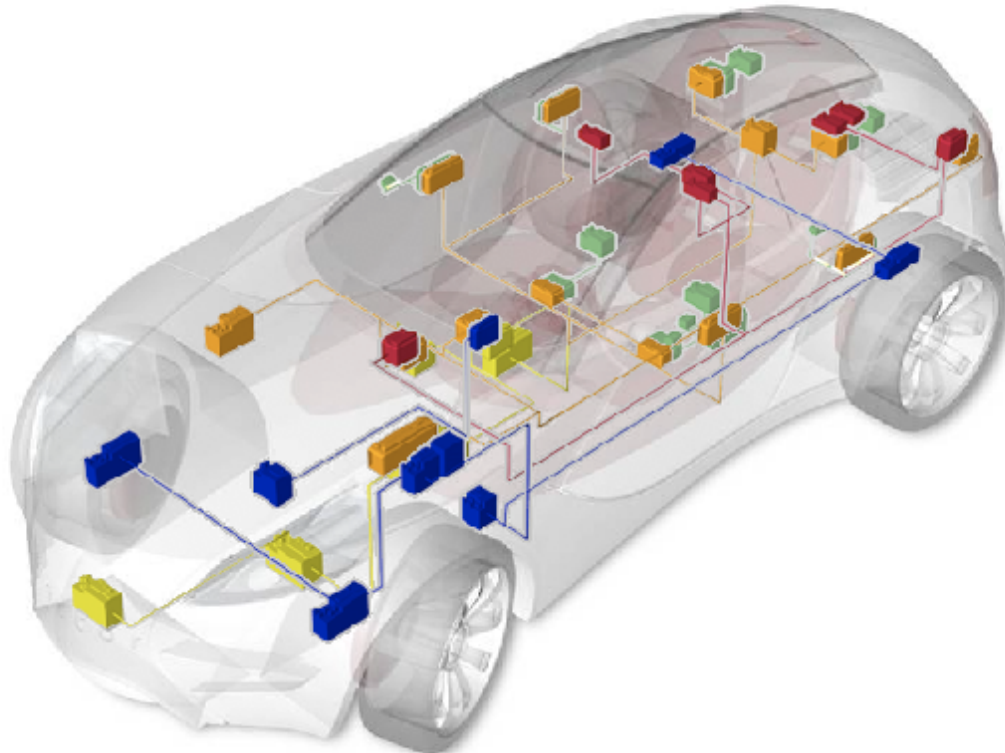
- 1 Einführung
- 2 Architektur serieller Bussysteme
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- **3 Serielle Bussysteme im Kfz**
 - 3.1 CAN
 - **3.2 LIN**
 - 3.3 FlexRay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

LIN Local Interconnect Network



Elektronikvernetzung im Kfz

✕ schließen



■ CAN H

■ CAN L

■ **LIN**

■ FlexRay

■ MOST

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

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

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/>



Serielle Bussysteme

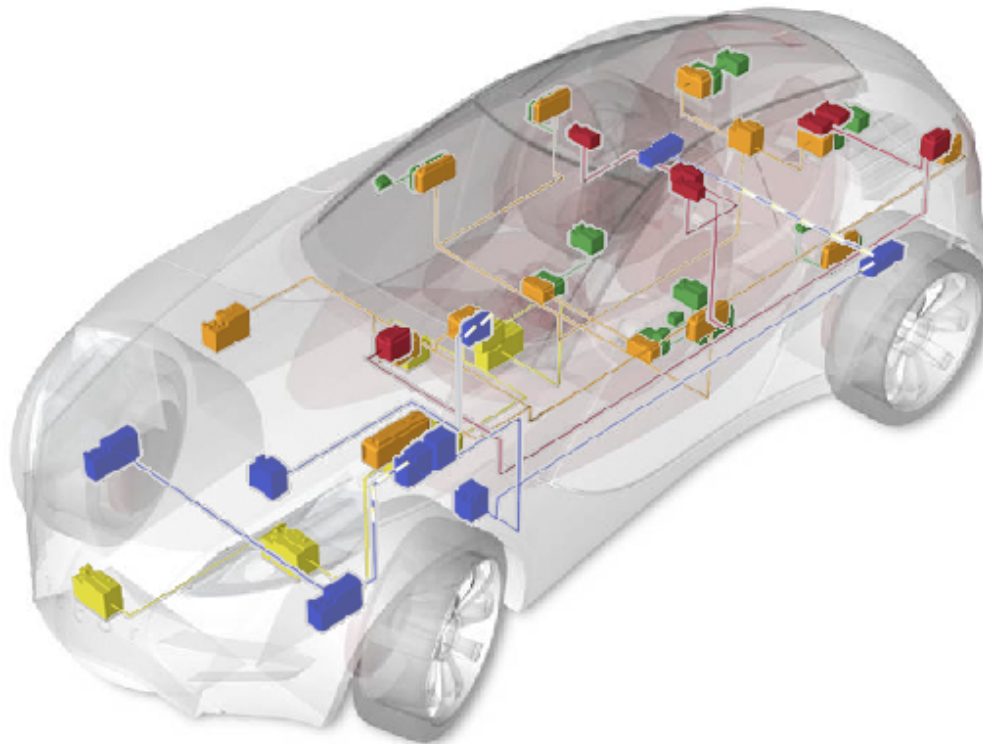
- 1 Einführung
- 2 Architektur serieller Bussysteme
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- **3 Serielle Bussysteme im Kfz**
 - 3.1 CAN
 - 3.2 LIN
 - **3.3 FlexRay**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

FlexRay



Elektronikvernetzung im Kfz

✕ schließen



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

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

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

Protokolle und Bussysteme:

- 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/>



Serielle Bussysteme

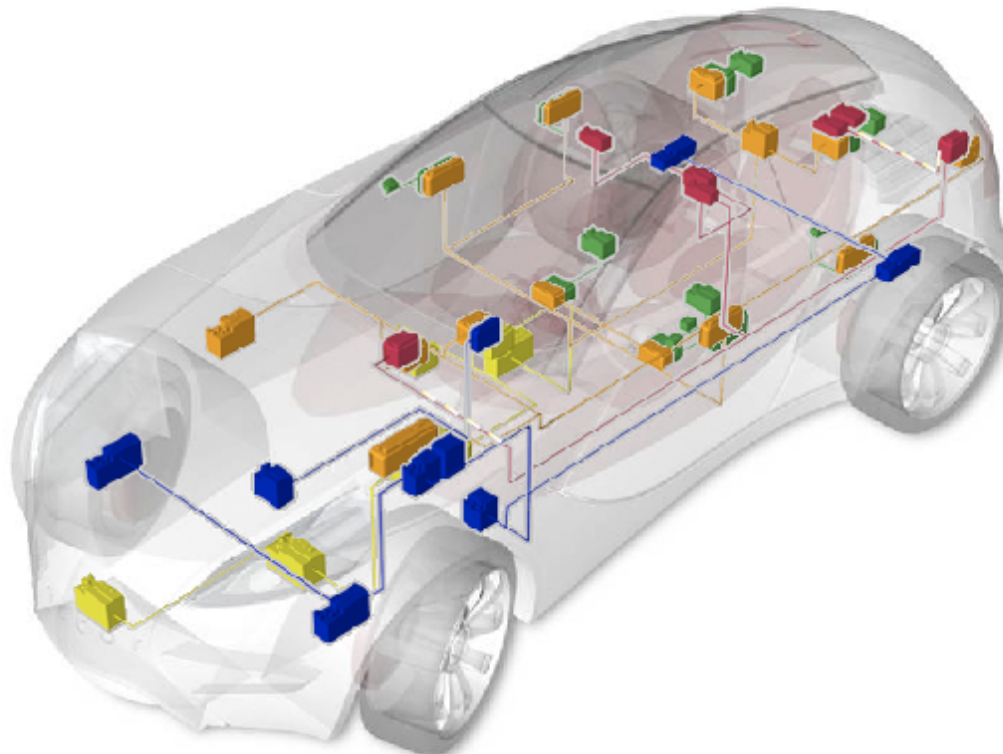
- 1 Einführung
- 2 Architektur serieller Bussysteme
 - 2.1 Interaktionsstruktur
 - 2.2 Topologie
 - 2.3 Adressierung und Framing
 - 2.4 Buszugriff
 - 2.5 Datensicherung
 - 2.6 Synchronisation
 - 2.7 Physikalische Übertragung
- **3 Serielle Bussysteme im Kfz**
 - 3.1 CAN
 - 3.2 LIN
 - 3.3 FlexRay
 - **3.4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

MOST



Elektronikvernetzung im Kfz

✕ schließen



■ CAN H

■ CAN L

■ LIN

■ FlexRay

■ **MOST**

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

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

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>

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

Ethernet

- 6. Vector Congress 2012
Beiträge: http://www.vector.com/vi_congress12_de.html
 - 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
- Vector Automotive Ethernet Symposium 2014
 - Beiträge:
http://vector.com/vi_aes14_de.html



Peter Schönenberg, BMW AG,
Germany.

INTRODUCTION OF ETHERNET.

6TH VECTOR CONGRESS
NOVEMBER 29, 2012, STUTTGART

BMW
GROUP



Continental 

IP/Ethernet

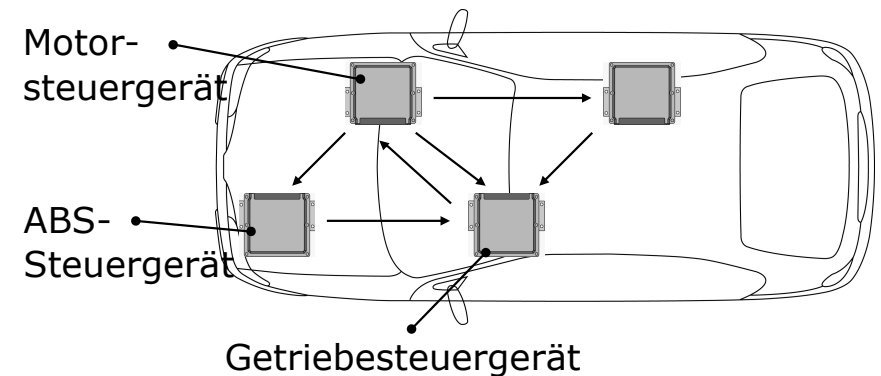
Next steps of pre-development

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**

Kommunikationsmatrix

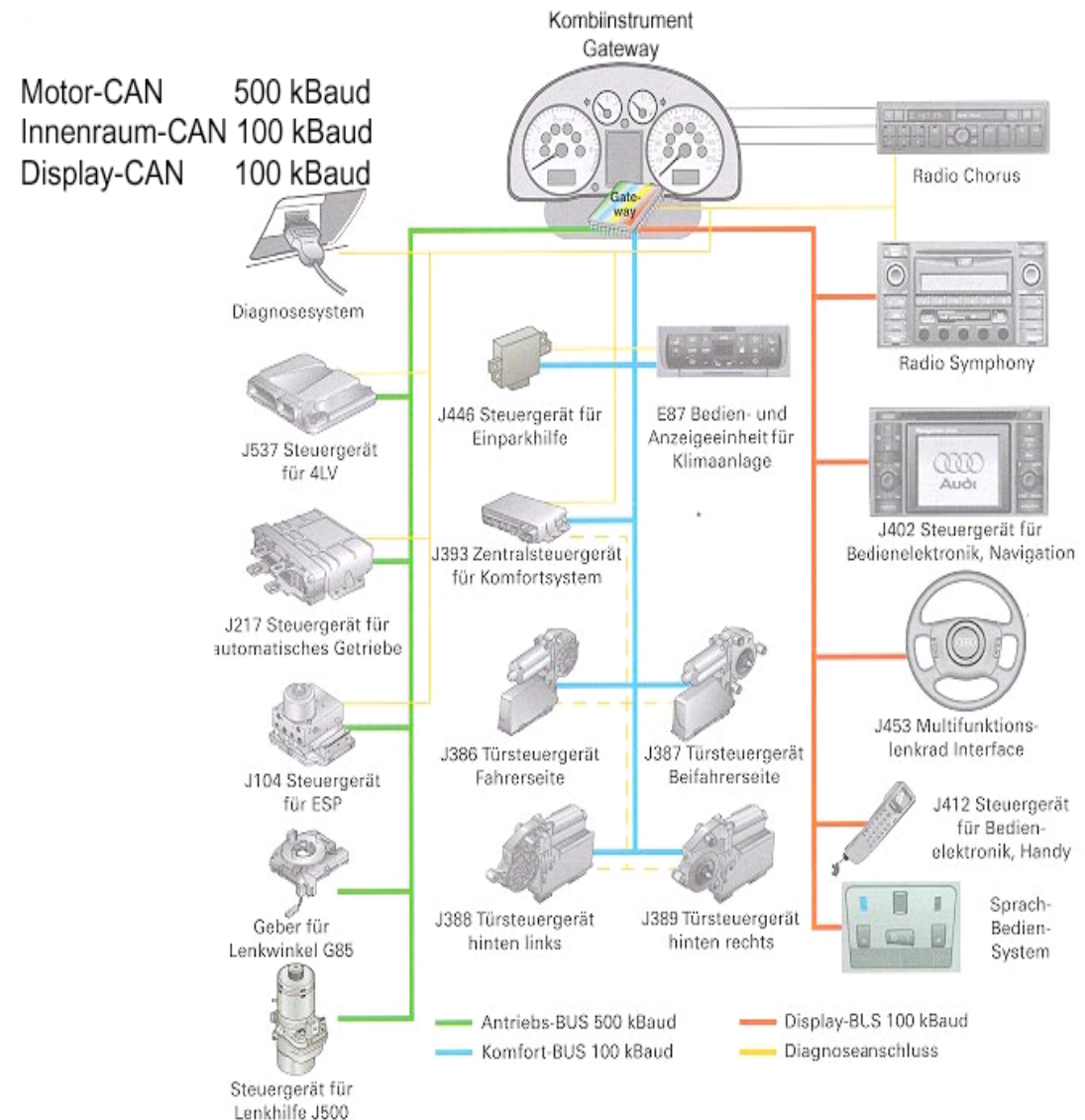
Netzknoten	Nachricht	Signal	ABS	MS	GS
ABS-Steuergerät (ABS)	ABS_1	Raddrehzahl vorne links	S		E
		Raddrehzahl vorne rechts	S		E
	ABS_2	Raddrehzahl hinten links	S		E
		Raddrehzahl hinten rechts	S		E
Motorsteuergerät (MS)	MS_1	Fahrpedalwert		S	E
	MS_2	Motordrehzahl Motortemperatur	E	S	E
Getriebesteuergerät (GS)	GS_1	Motorsollmoment		E	S



Quelle: Schäufele, Zurawka: Automotive SWE

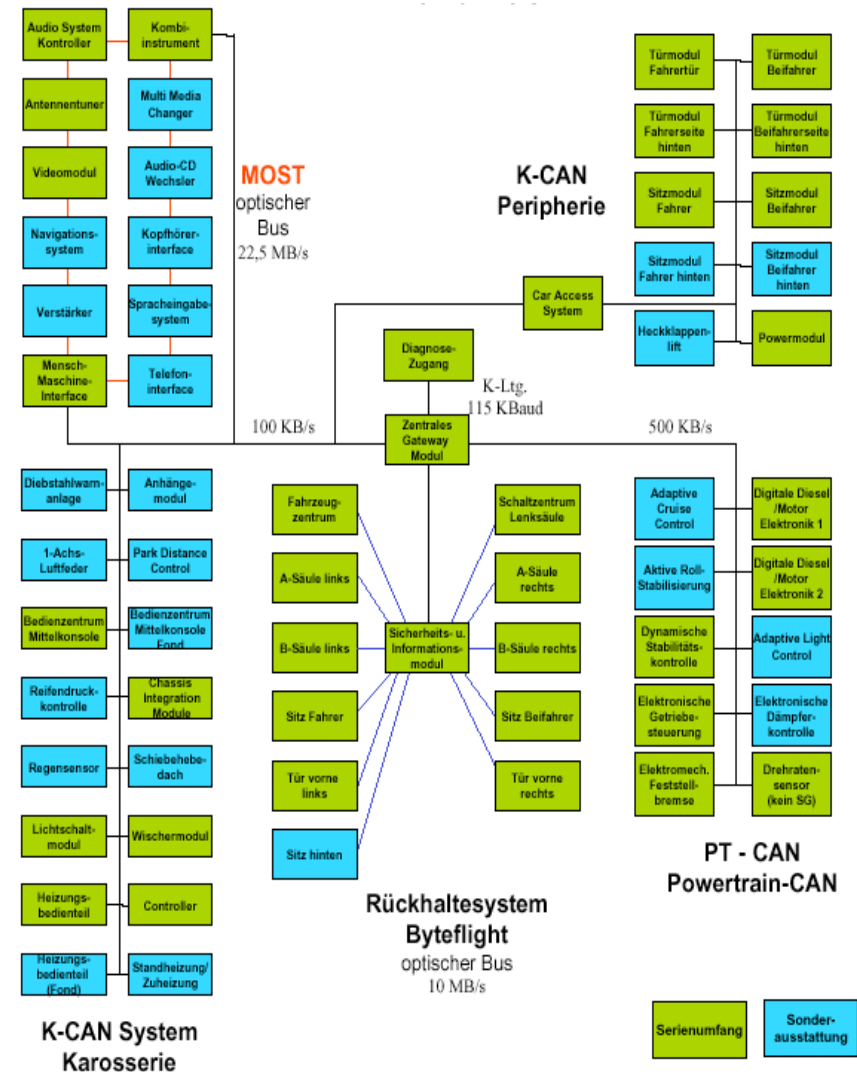
Beispiele

- 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



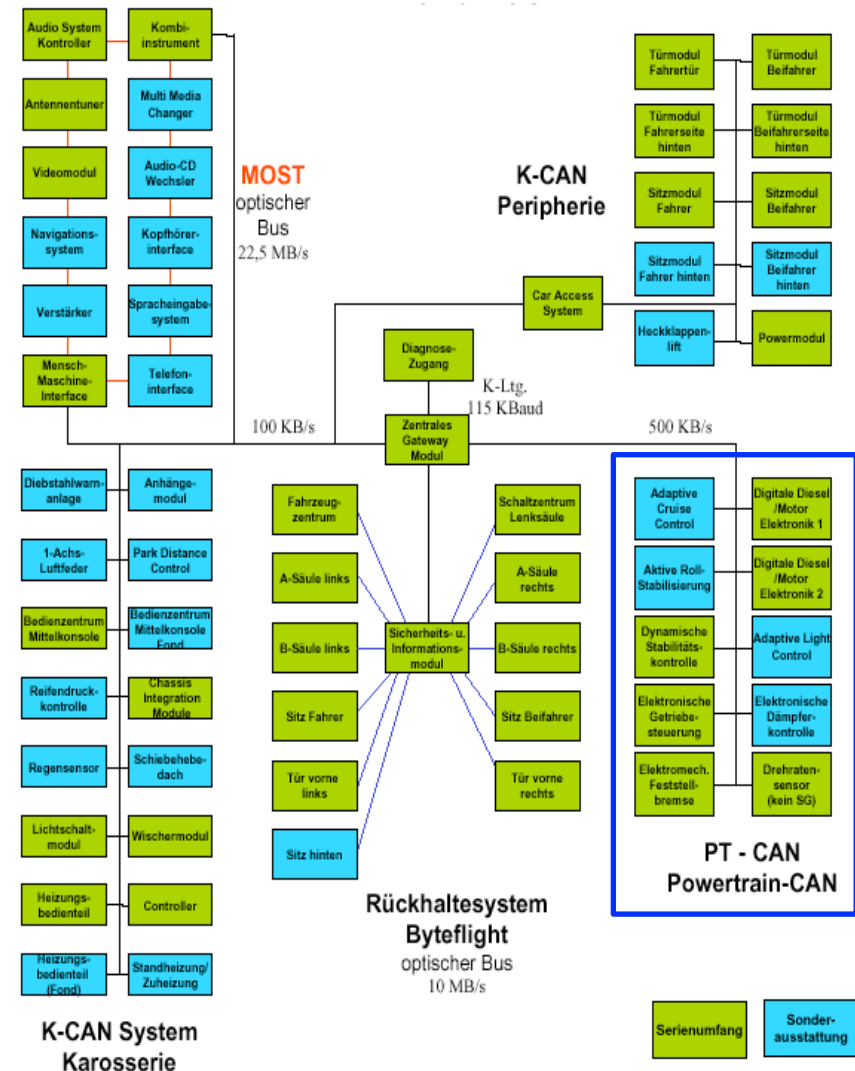
Beispiele

- 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)



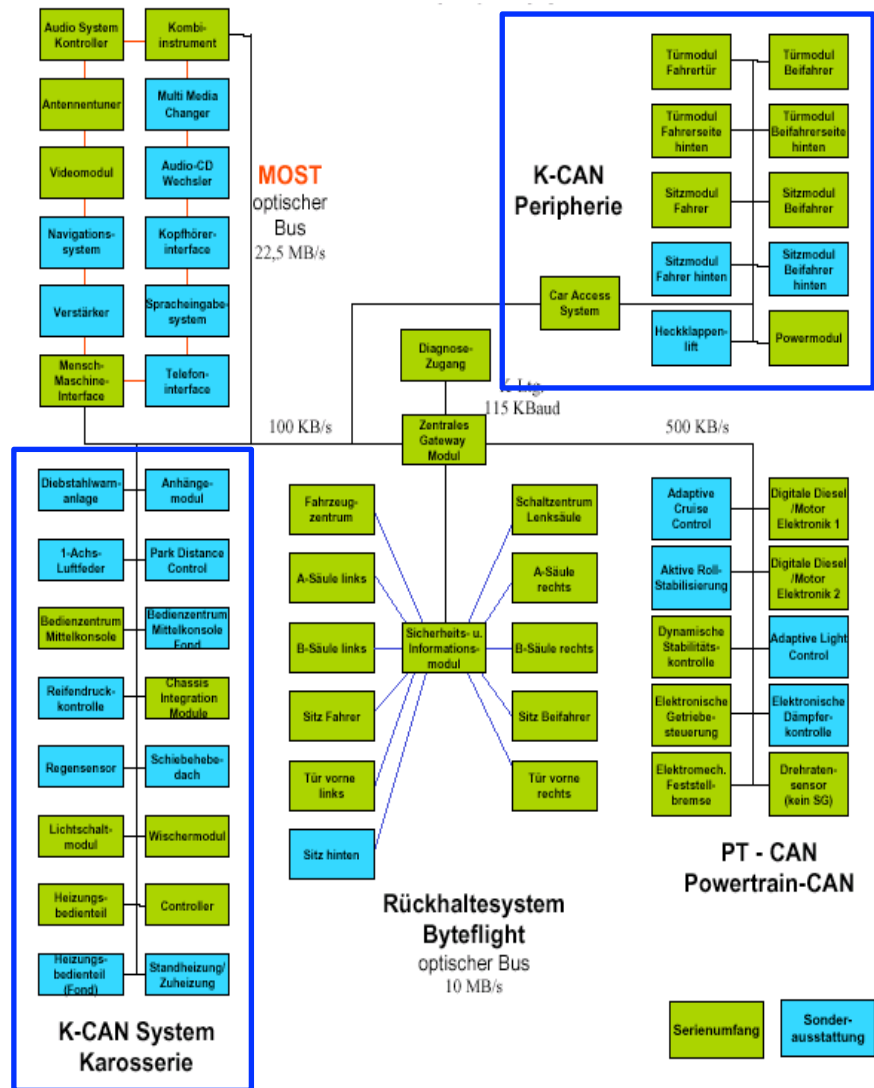
Beispiele

- 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)



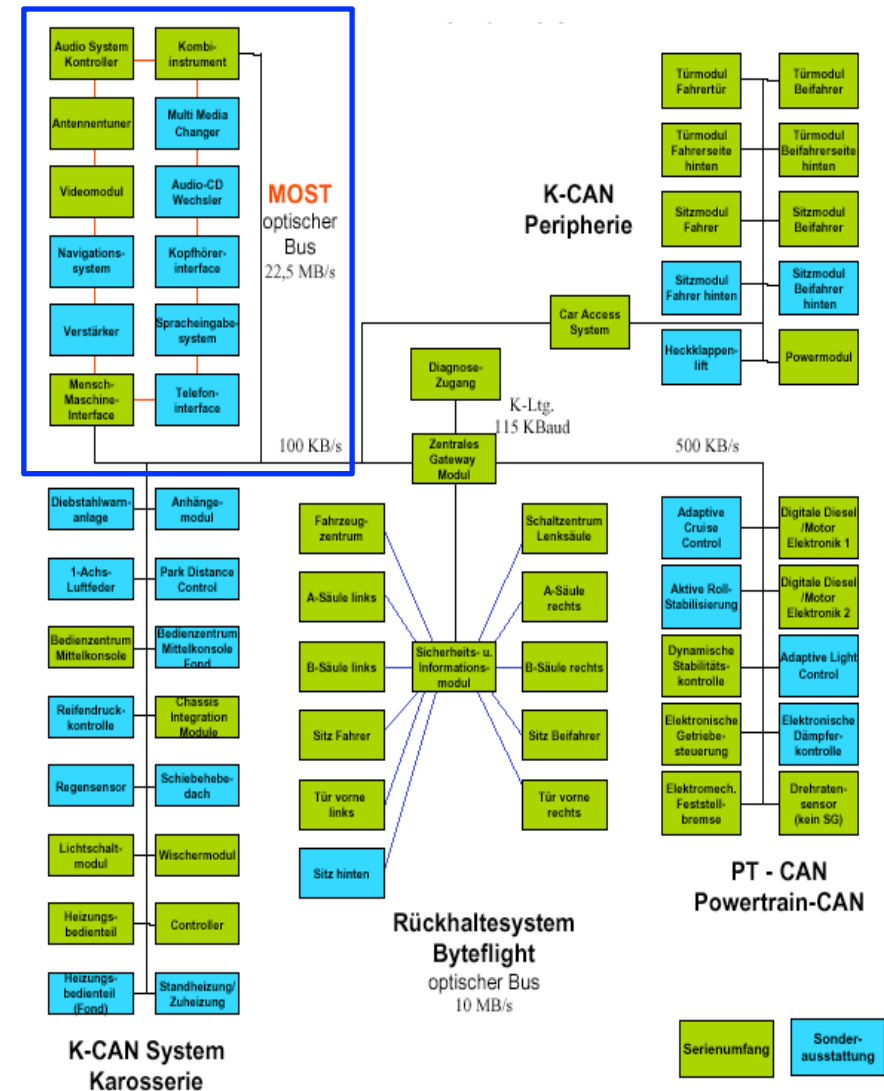
Beispiele

- 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)



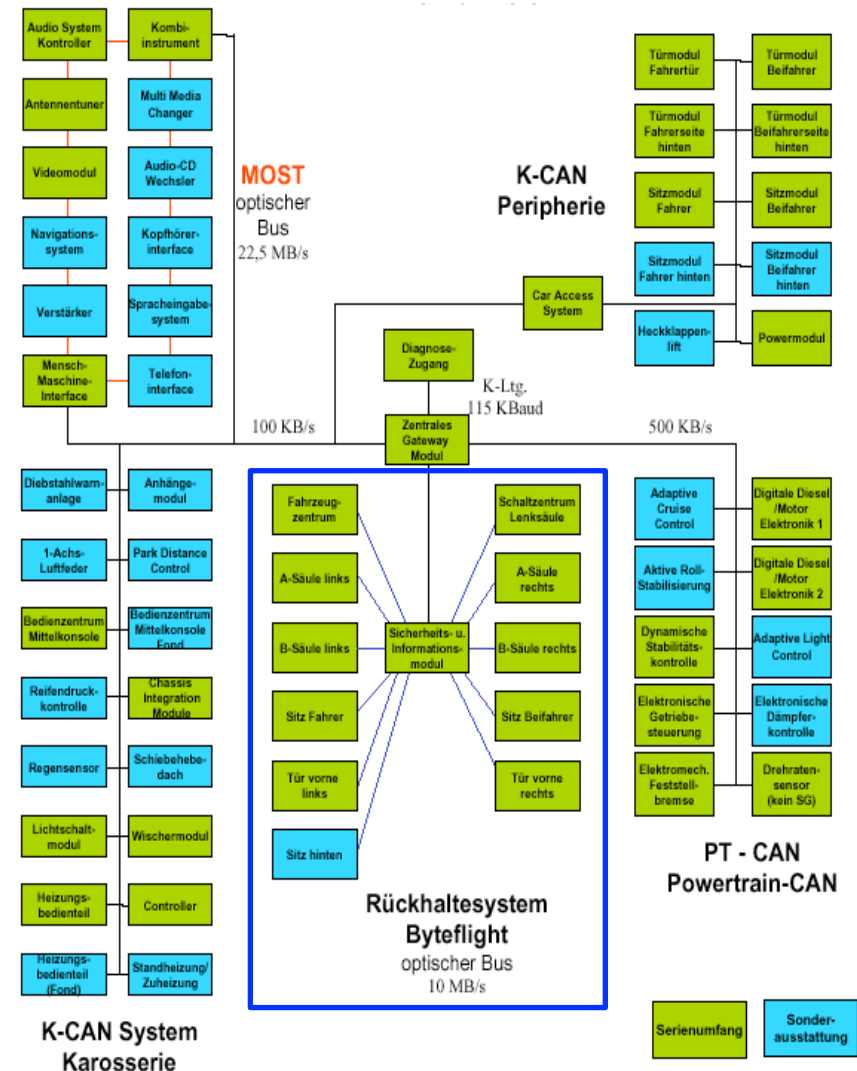
Beispiele

- 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)



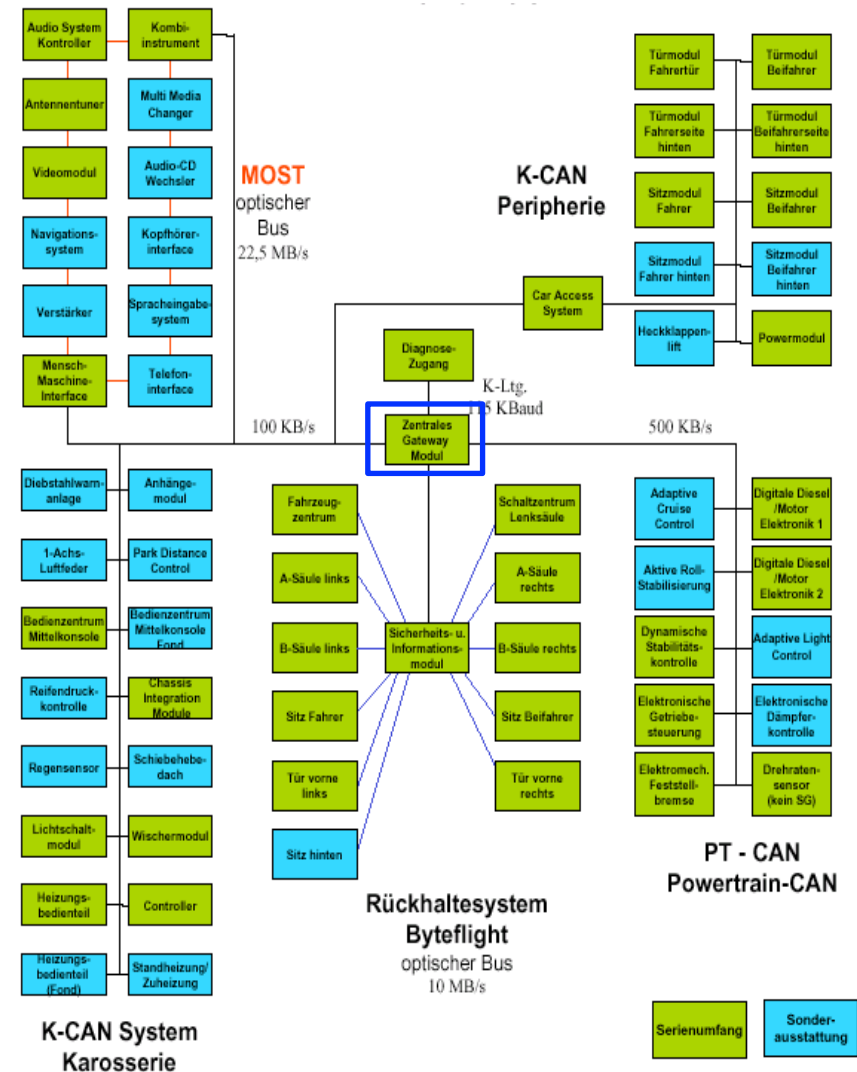
Beispiele

- 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)



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)

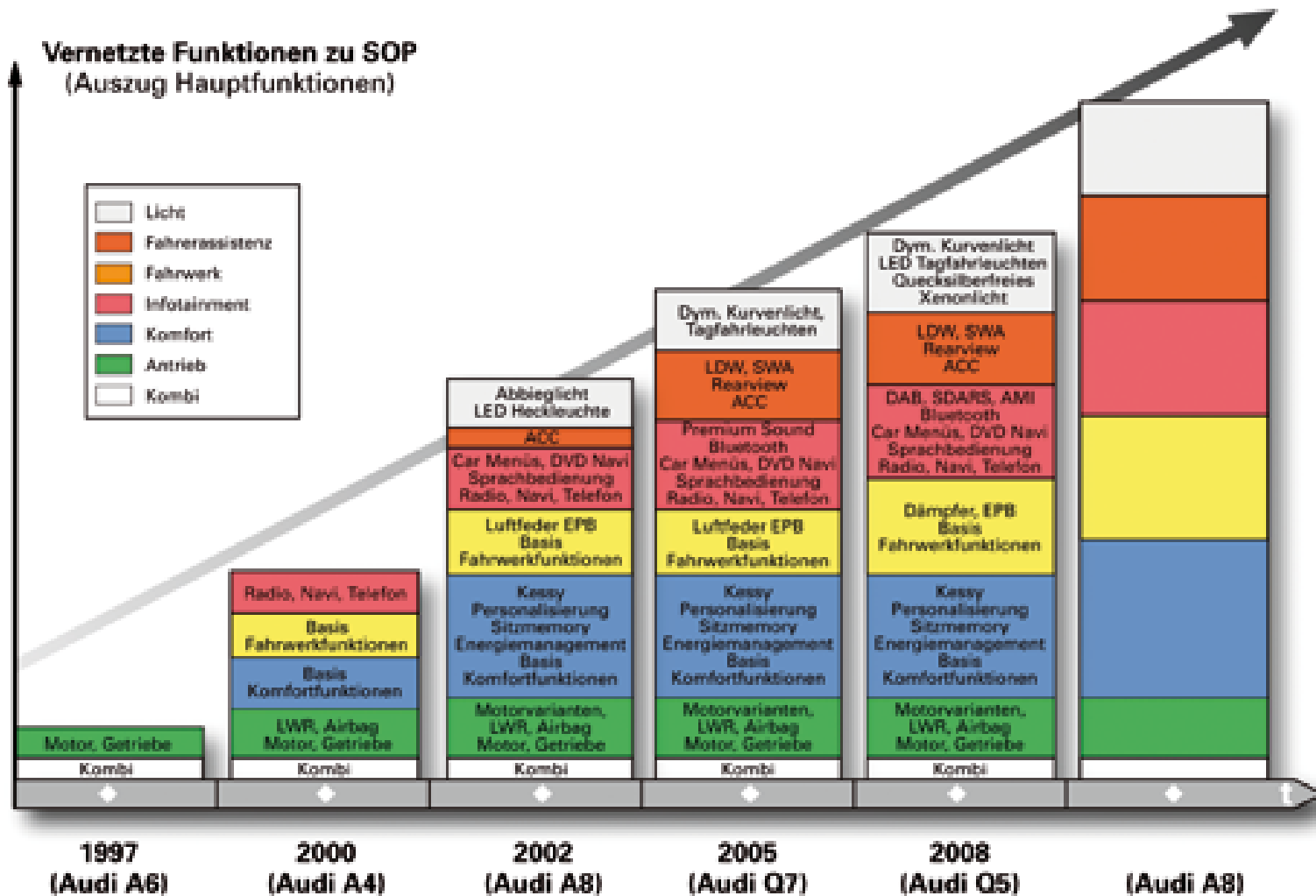


Beispiele



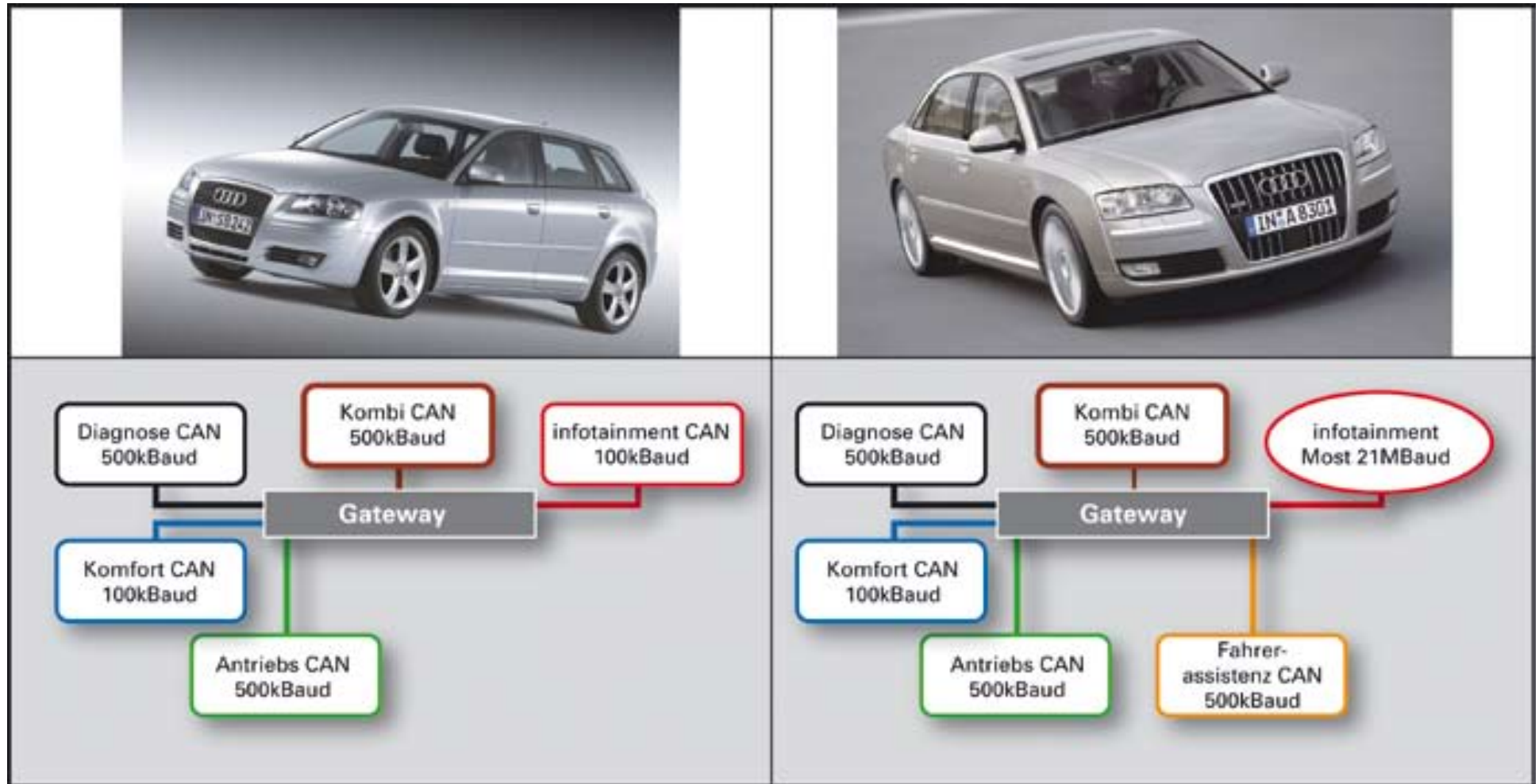
- S. Esch, B. Lang: Elektronik- und Vernetzungsarchitektur mit gesteigerter Leistungsfähigkeit, Sonderausgabe von ATZ und MTZ, Juni 2008

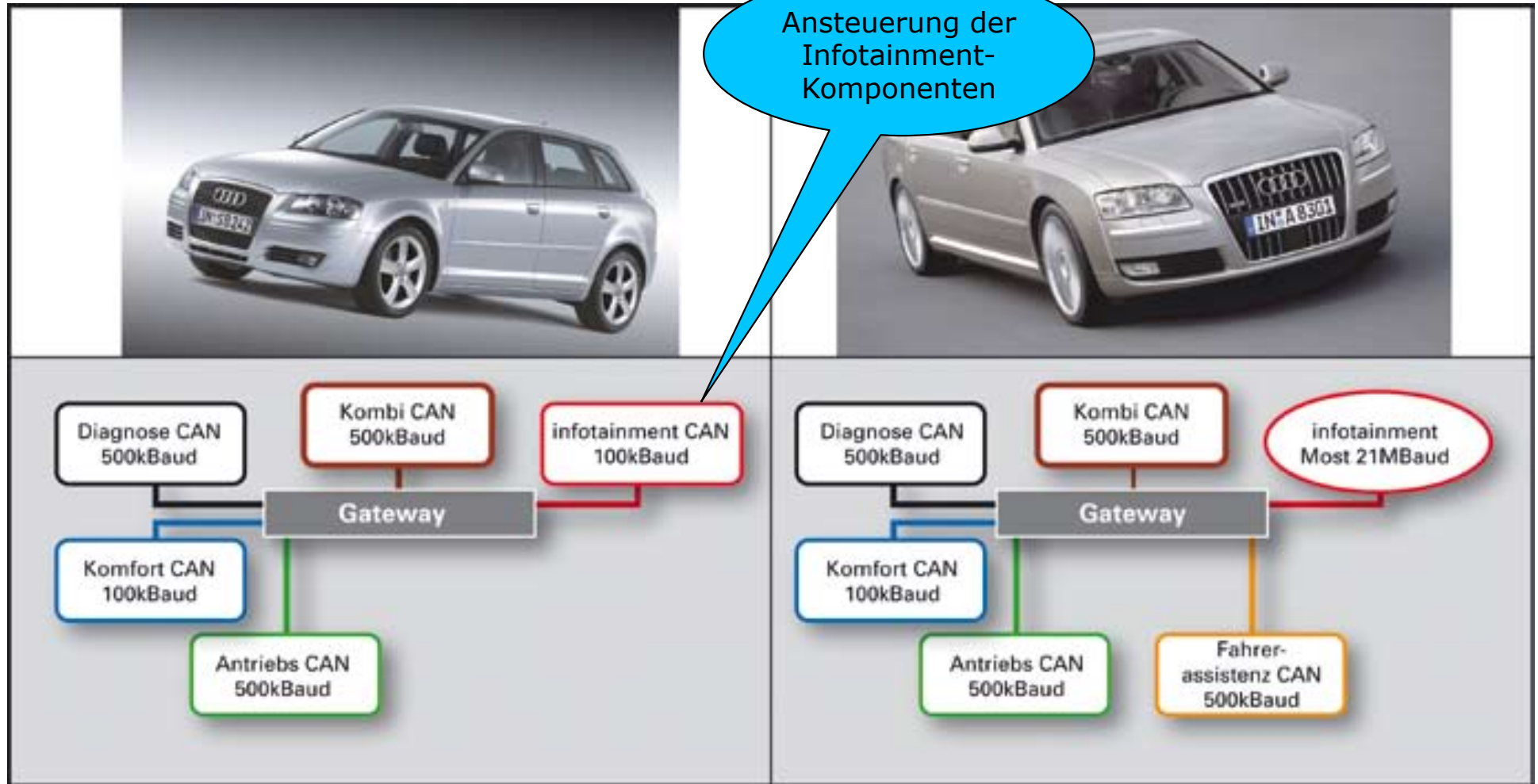
Zunahme der vernetzten Funktionen: Beispiel Audi

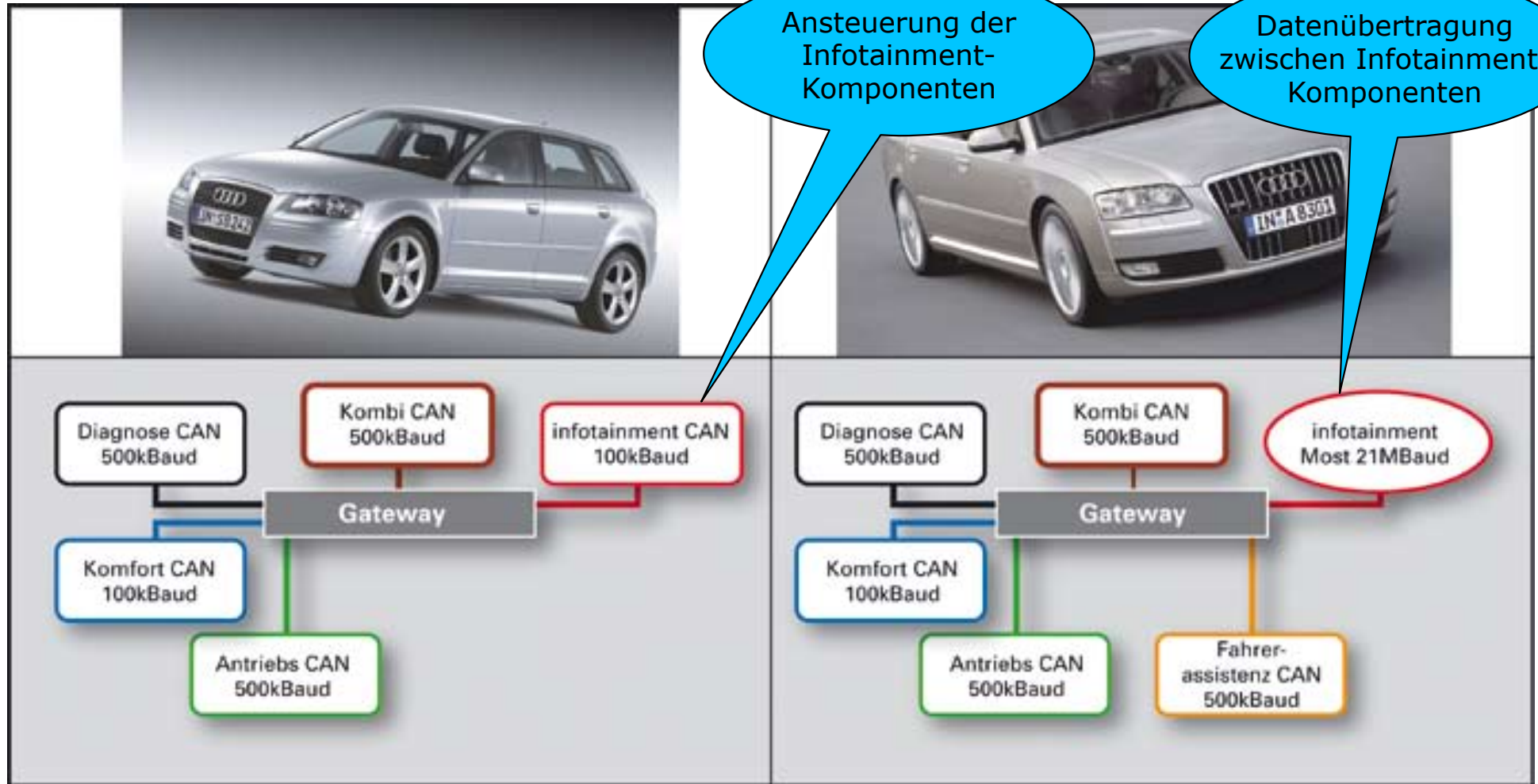


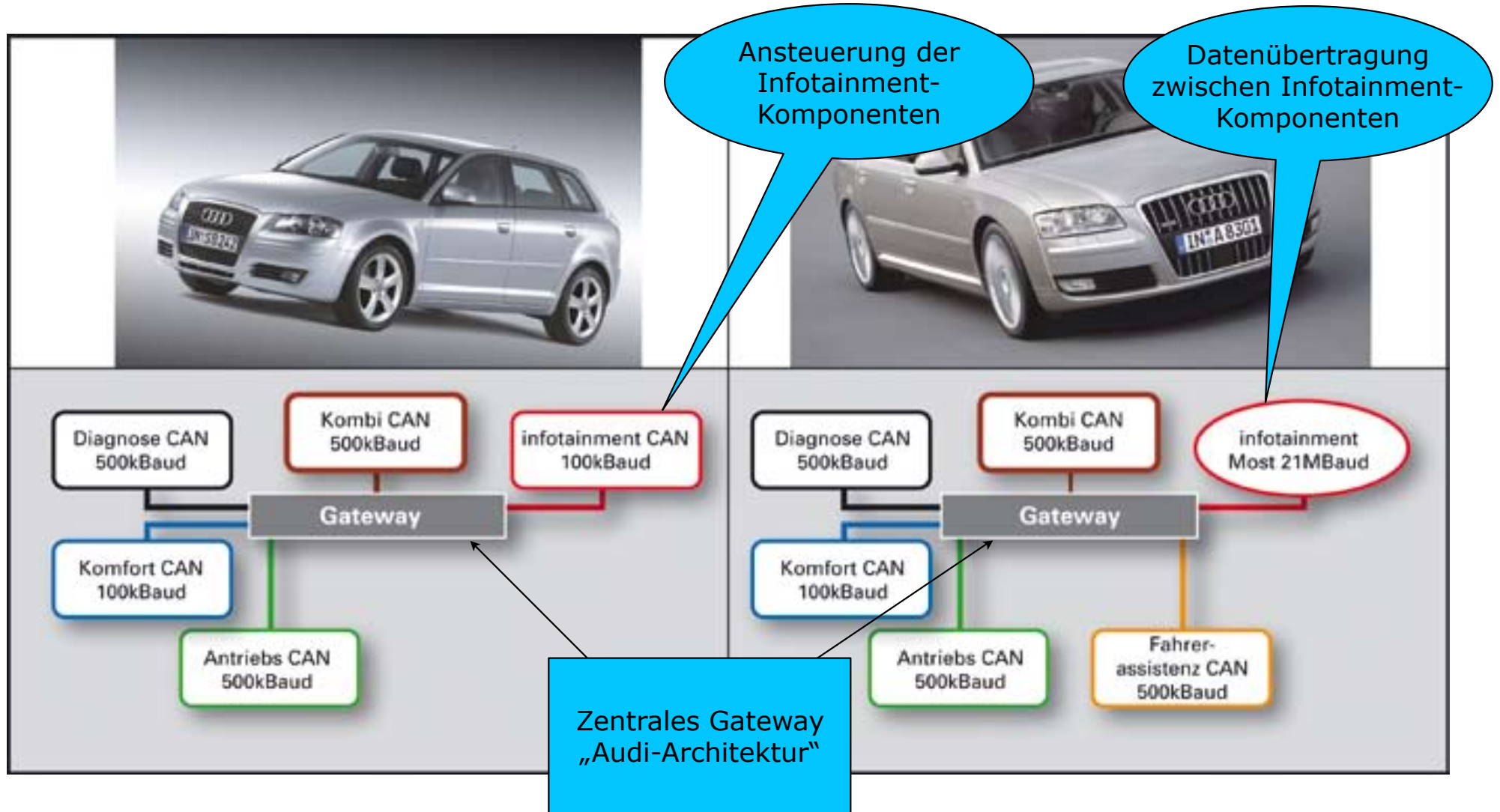
Audi Q5

- 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.



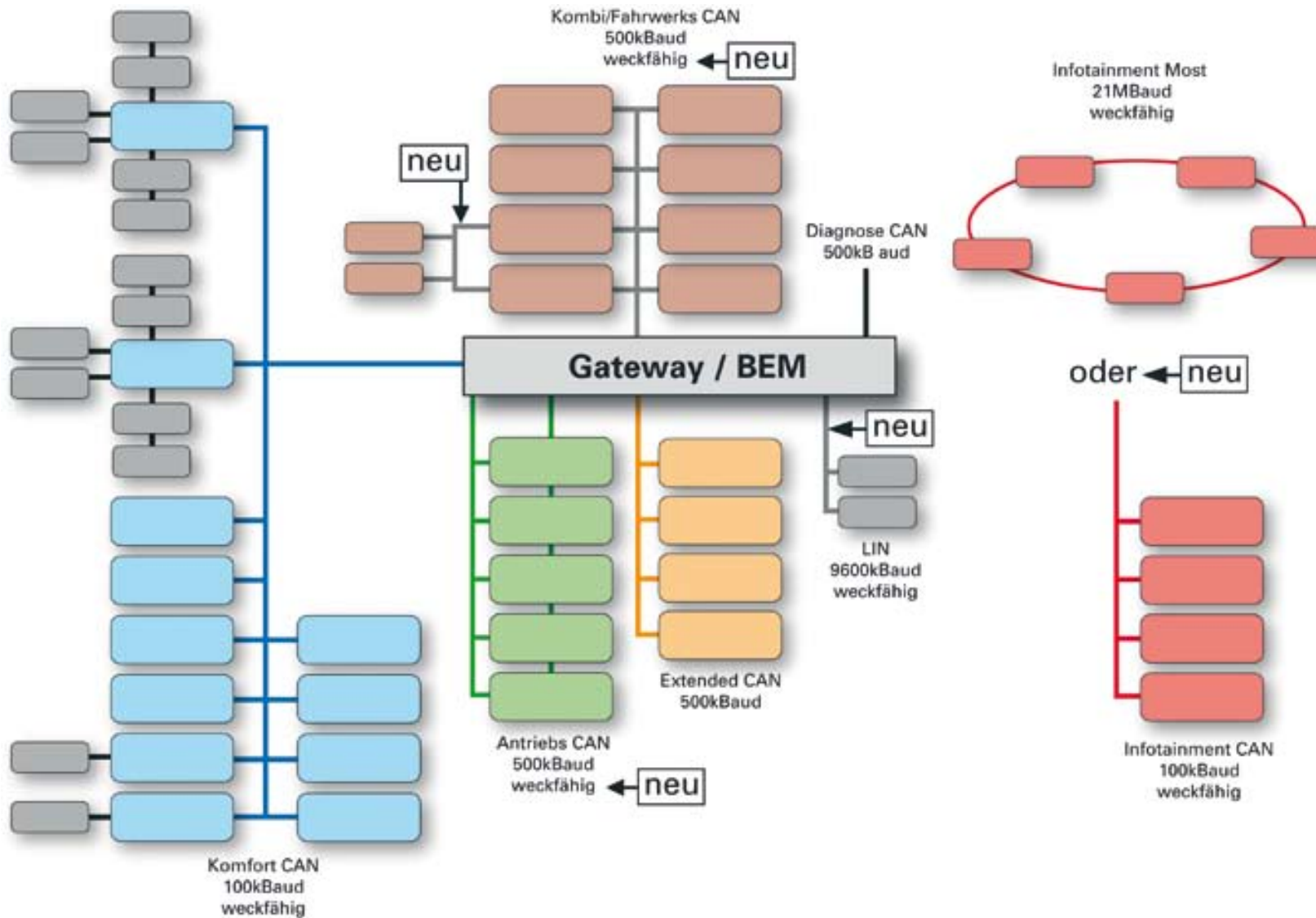




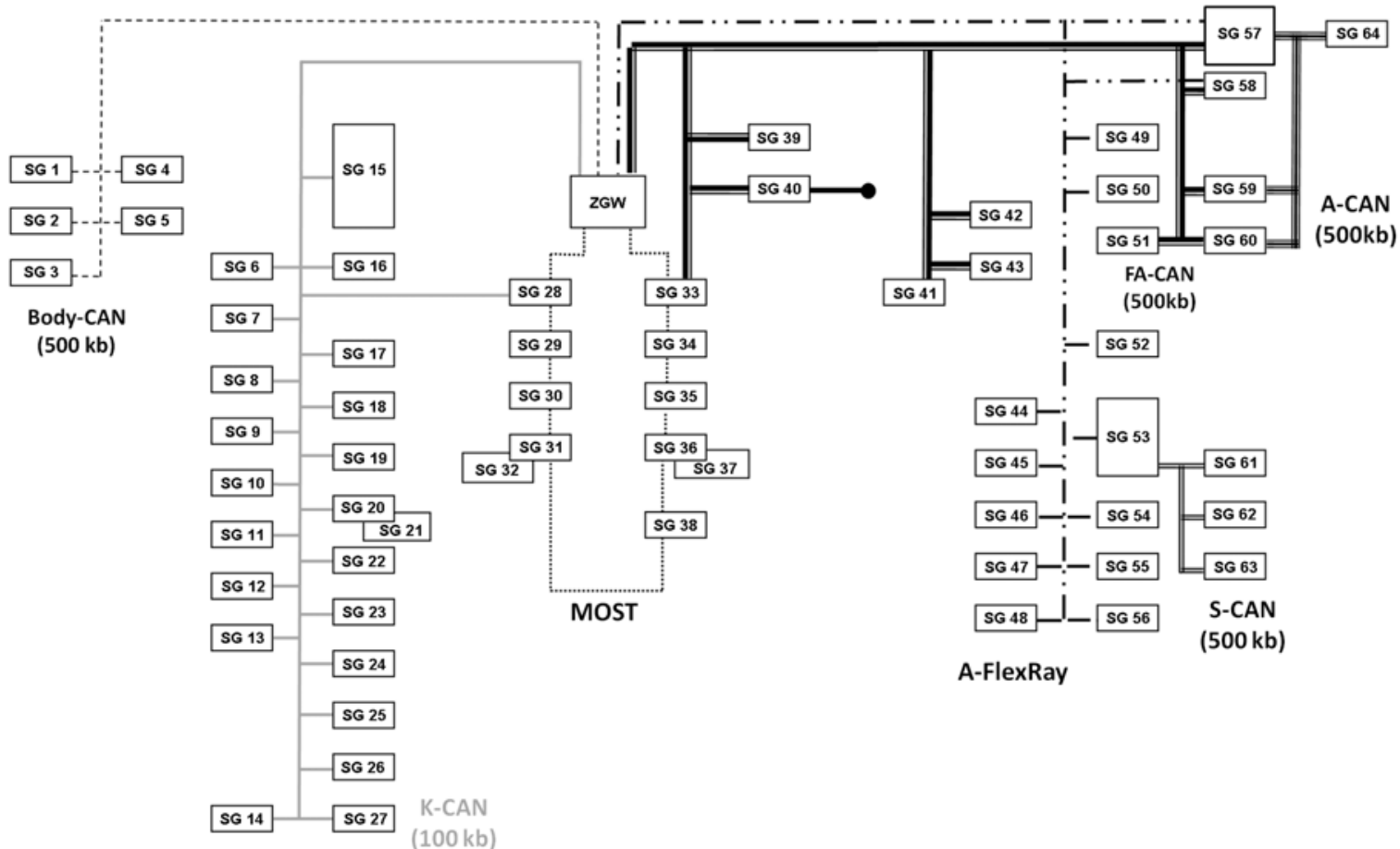


Audi Q5

- 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 hochskalierbaren Architektur.

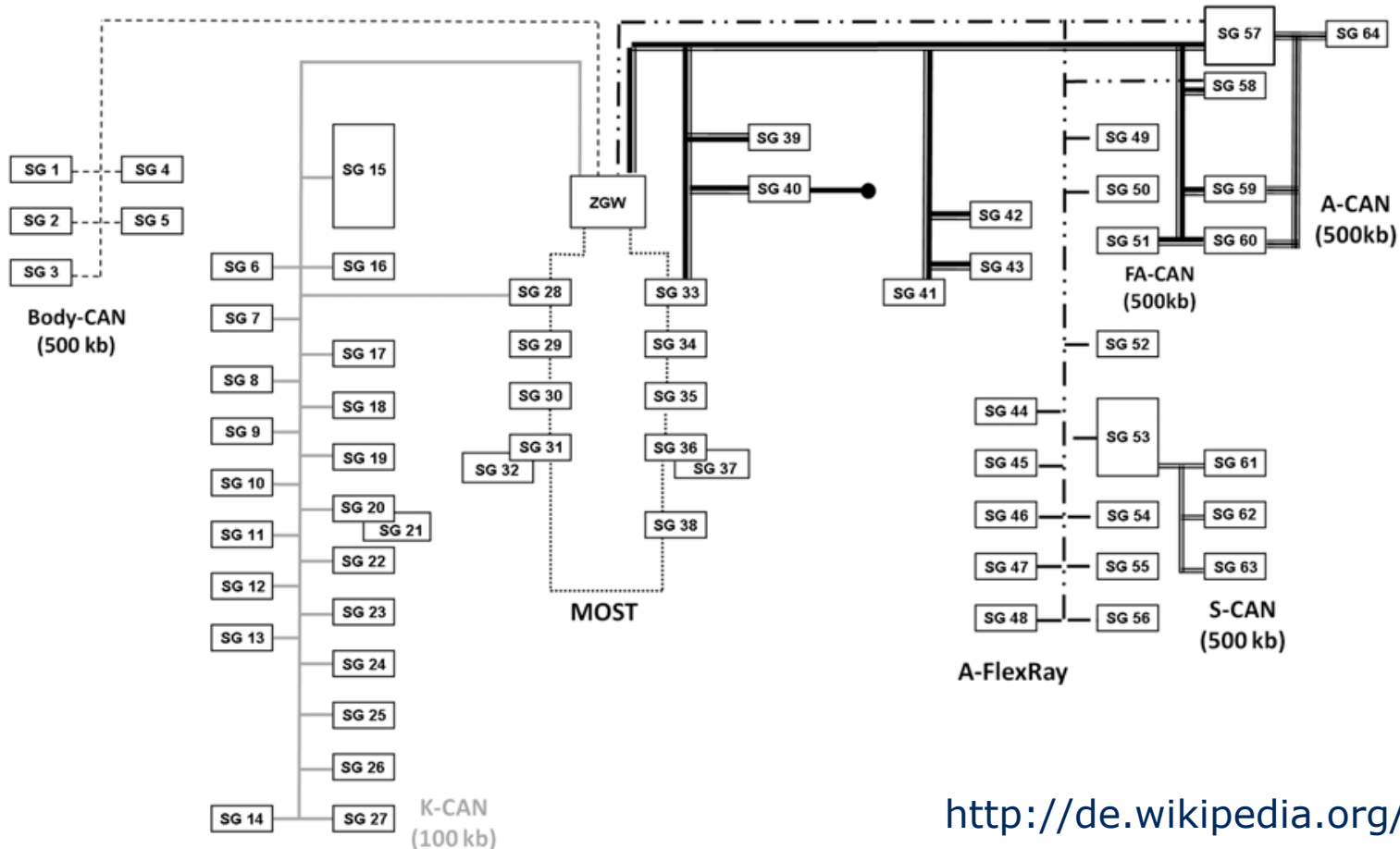


Datenkommunikationsbordnetz des BMW F01/F02



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

Datenkommunikationsbordnetz des BMW F01/F02



http://de.wikipedia.org/wiki/BMW_F01

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