



Diagnosesysteme im Automobil

Anwendungen für Messen & Kalibrieren

Dr.-Ing. Jörg Supke

emotive GmbH, Stuttgart

Prof. Dr.-Ing. Werner Zimmermann

Hochschule Esslingen

e**motive**
Automotive Software Solutions

1

Einleitung ASAM MCD

2

ASAM MCD 1

3

D-PDU-API

4

ASAM MCD 2 FIBEX

5

ASAM MCD 2MC

Open System Interconnection (OSI) Schichtenmodell (ISO 1978)

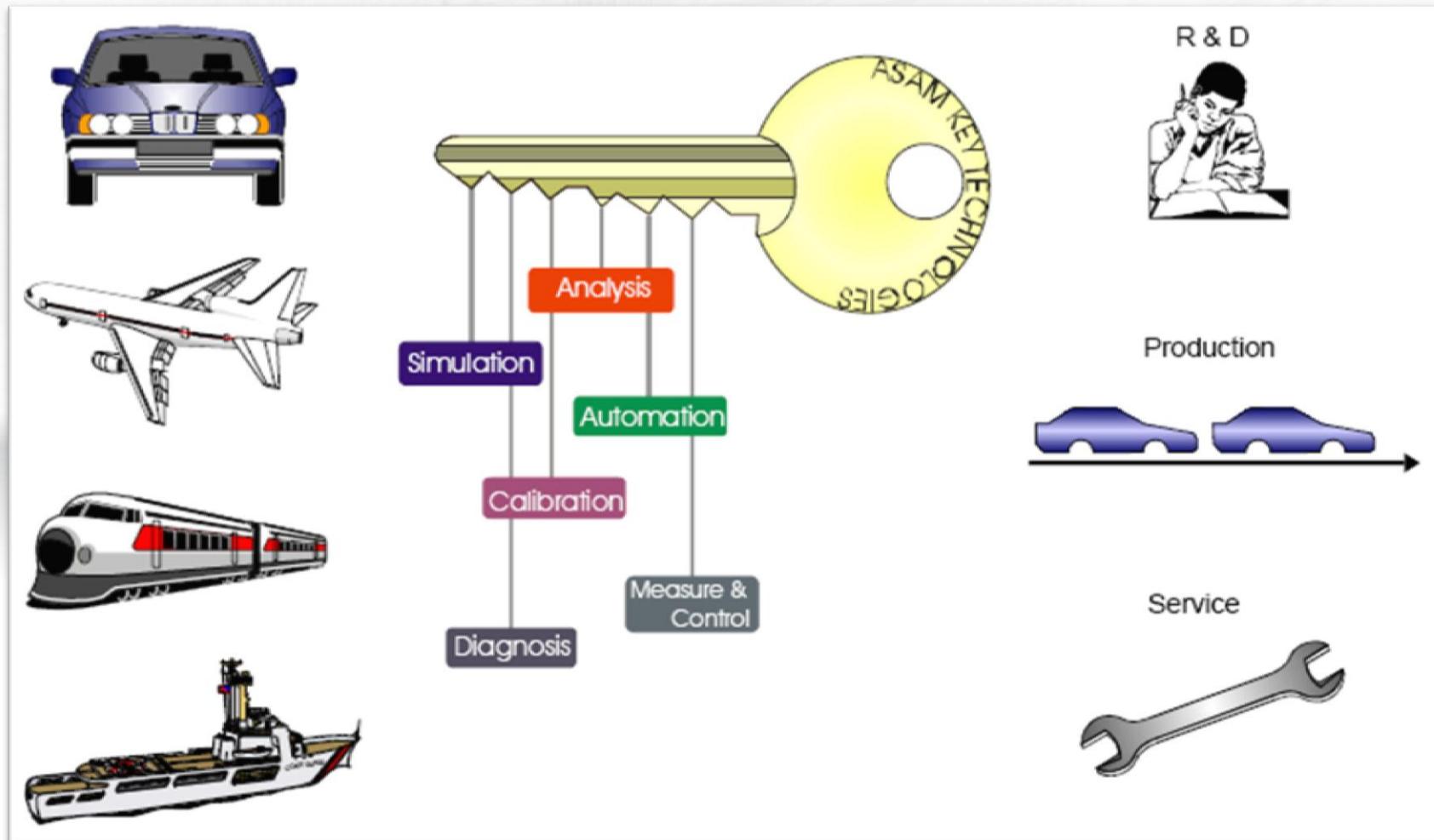


Eigentliche Anwendung (On-Board z.B. Motorsteuerung oder Off-Board z.B. Diagnosetester)

Schicht	Bezeichnung	Anwendung im Fahrzeug	
7	Application Layer (Anwendung)	Anwendungsprogramm, fertige Dienste, z.B. Fehlerspeicher lesen	Diagnoseprotokolle
6*	Presentation Layer (Darstellung)	Unterschiedliche Darstellung der Daten	
5*	Session Layer (Sitzungssteuerung)	Steuert Verbindungsprozesse, z.B. Authentifizierung, Synchronisation	
4	Transport Layer (Transport)	Segmentierung der Botschaften	Transportprotokolle
3*	Network Layer (Vermittlung)	Routing, Adressierung, Teilnehmererkennung, - überwachung	
2	Data Link Layer (Sicherung)	Botschaftsaufbau, Buszugriff, Fehlererkennung, Flussregelung	Bussysteme
1	Physical Layer (Bitübertragung)	Signalpegel, Bitkodierung	

Busleitungen und Steckverbinder (Mechanik)

* Werden für Anwendungen im Fahrzeug z. Z. nicht verwendet; Aufgaben werden von den anderen Schichten übernommen.



Quelle: ASAM e.V., www.ASAM.net

Association for Standardization of Automation and Measuring Systems

- 1991 Gründung einer Initiative zur Standardisierung von Meß- und Automatisierungssystemen durch deutsche Fahrzeugherrsteller
- 1998 Gründung des ASAM e.V. mit damals 33 Mitgliedern → heute mehr als 120 weltweite Unternehmen (Fahrzeugherrsteller, Zulieferer und Toolhersteller)
- The **VISION** of ASAM is that standards enable products that can be freely interconnected with seamless data interchange.
- The **MISSION** of ASAM is to develop, maintain, and deploy platform independent extensible standards, and to enable products that use and are compliant with those standards.
- The **SCOPE** of ASAM includes measurement, automation, analysis, and simulation systems used within industry, and to support electronics software engineering methods.
- The **GOALS** of ASAM are to be a global organization, to cooperate with other standards organizations, to support collaboration between end users and suppliers for creating and using standards, and to enable software interfaces and data exchange for interconnecting computer and controller hardware and software products.

Mitglieder



Einleitung MCD 1 R-PRU-API FIBEX A2L

6

Supplier Members

ASOFT AND AMS ADDITIVE AFT AMS
AVL BOSCH OAF SAEI Compact Dynamics CRYSTAL Instruments
Cyber Metrics d2t DASAN DENSO DMDAK DSA
dSPACE ESG ETAS FEV HK
gedas HBM HGL Dynamics HORIBA
FICHAUTOMATION IAV imc InSitu influx TECHNOLOGY
JANUS USA KISTLER kratzer
KÖHLER LMS MK meaX MEIDEN
MESSRING MFP MTS MÜLLER-BBM NATIONAL INSTRUMENTS nCode
ONI SOKO Re-Sol JUMAR SIEMENS SIEMENS VDO Sigma Systems
RENIK SMART TGS TERCUNE Diagnostic Solutions The Motilink Toyota Macs T-Systems TTTech
vector VISUM xio works

OEM Members

AUDI BMW DAIMLERCHRYSLER GM MAINS PORSCHE
RENAULT VOLVO

Academic Members

PROPS FZI HTWK UNIKASSEL UNIVERSITÄT

Quelle: ASAM e.V., www.ASAM.net, Stand 2007

Was sind die ASAM Standards?

■ ASAM AE (Automotive Electronics)

Includes **MCD** (Measurement, Calibration and Diagnosis) and **MSR** (Manufacturer Supplier Relationship) activities which defines interfaces and data structures for the automotive electronics development and test processes.

■ ASAM GDI (Generic Device Interface)

Defines interfaces to measuring devices and intelligent subsystems

■ ASAM ACI (Automatic Calibration Interface)

Defines the interface between optimization and automation components for microcontroller-based open-loop and closed-loop control systems

■ ASAM CEA (Components for Evaluation and Analysis)

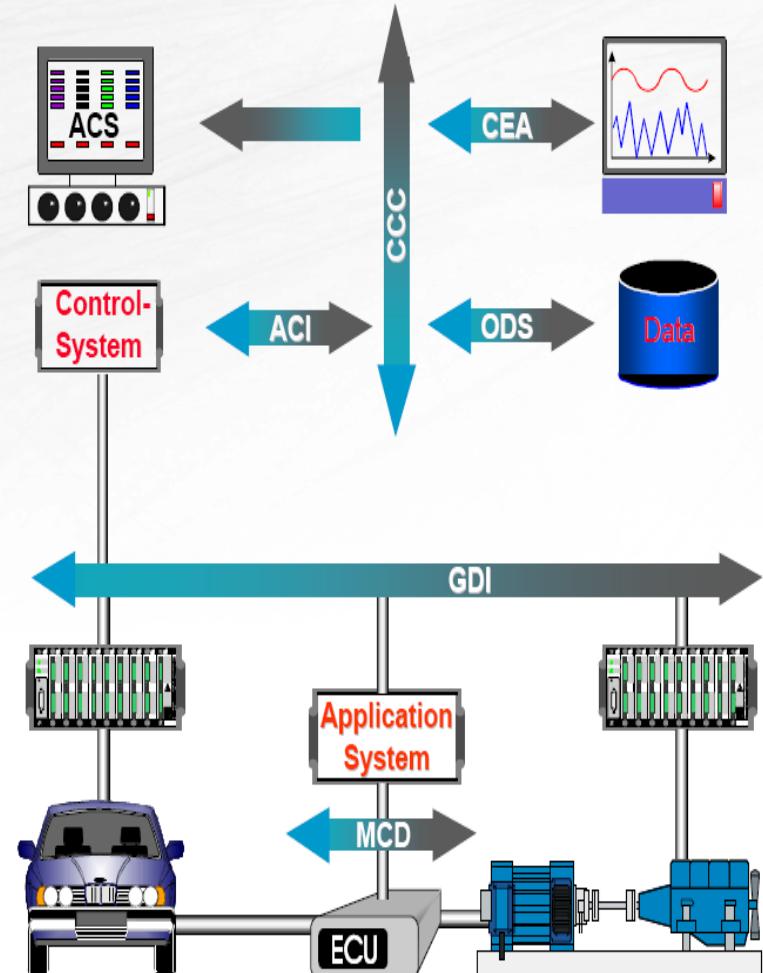
Defines a component interface and the necessary basic functionality for the creation of project-specific overall applications from modular measured data evaluation and analysis tools

■ ASAM ODS (Open Data Service)

Defines interfaces for storage, interpretation and exchange of data

■ ASAM CCC (Component Communication and Coordination)

Makes available methods facilitating comprehensive communication between software components



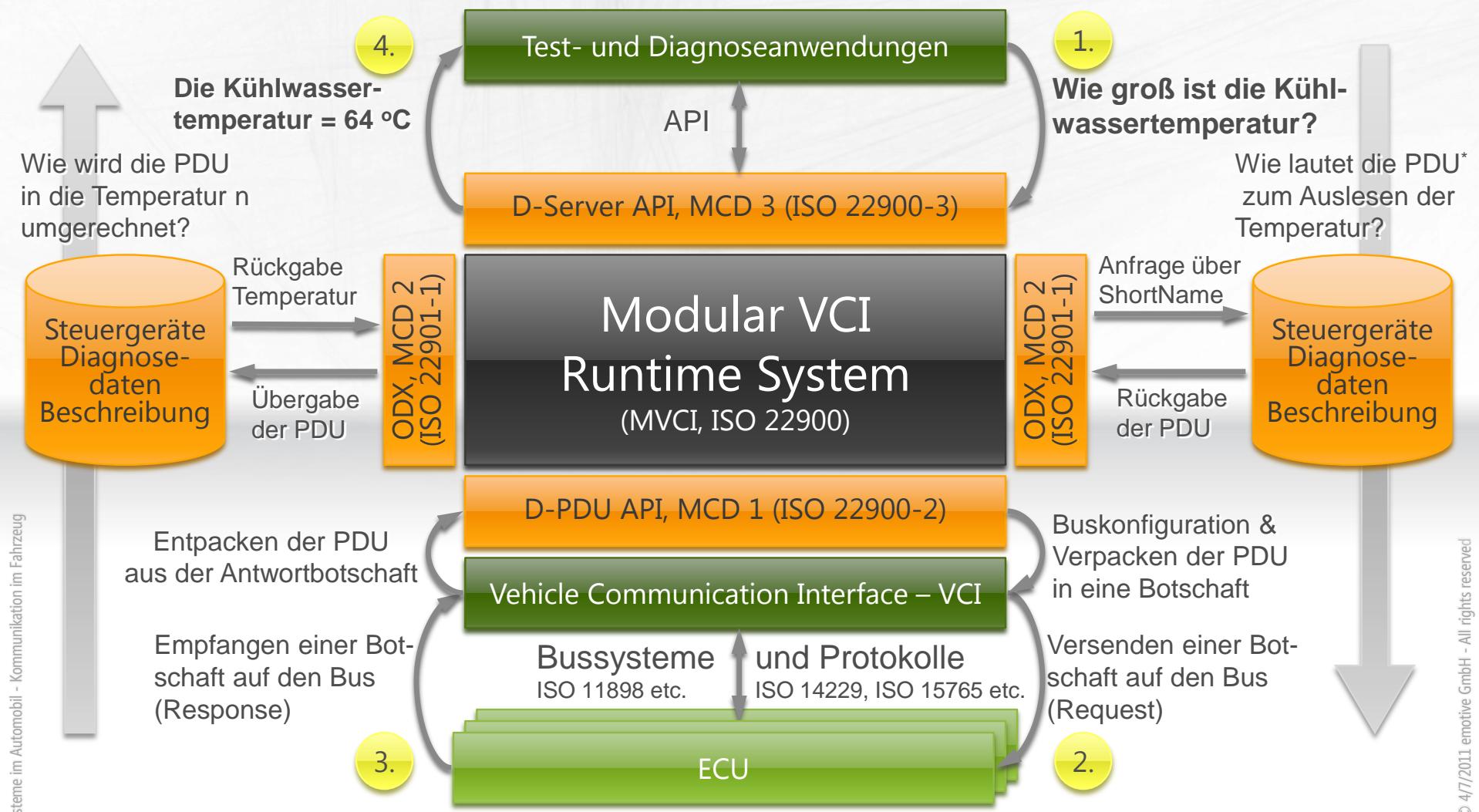
Quelle: ASAM e.V., www.ASAM.net

Überblick ASAM AE MCD D (MVCI)

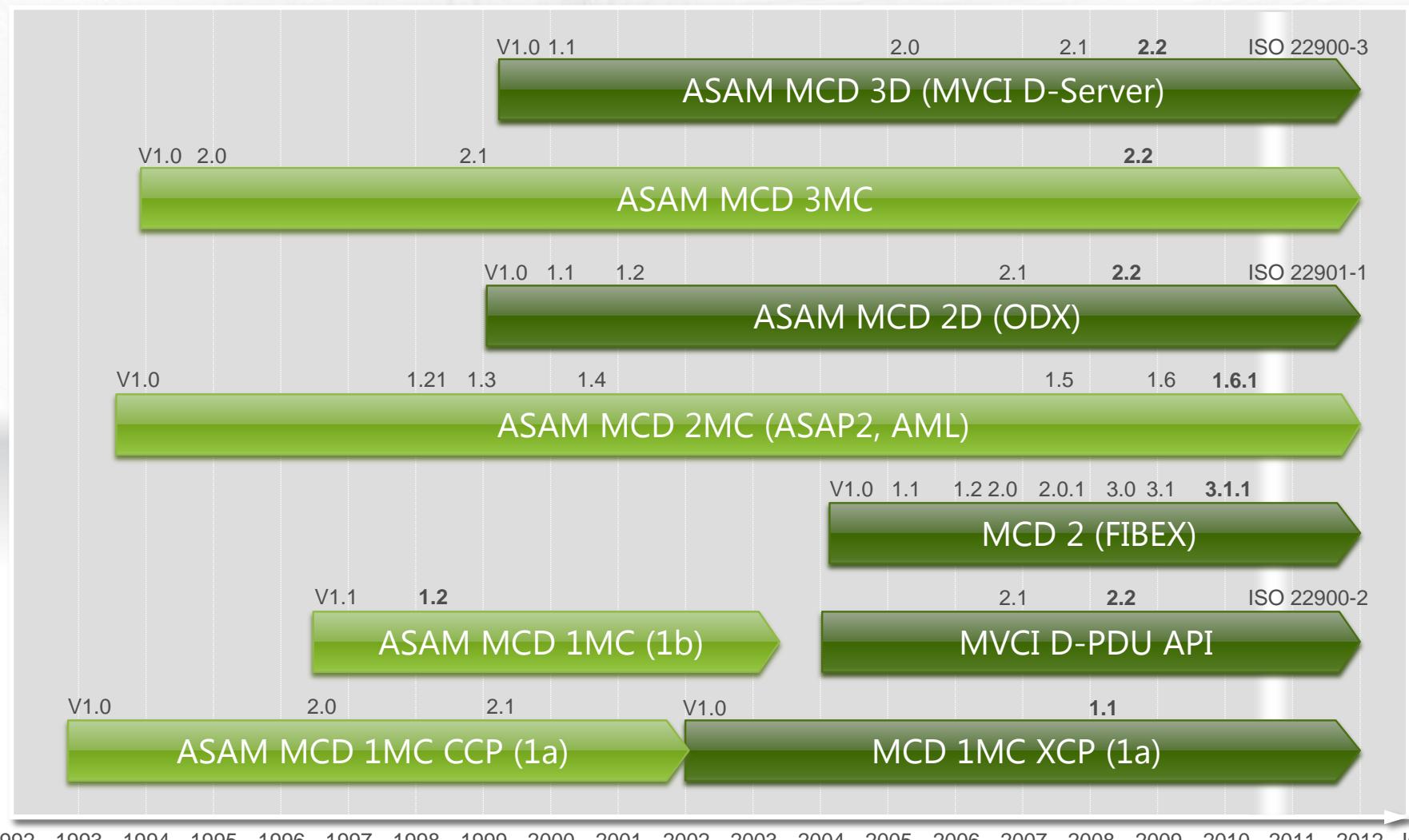


Einleitung MCD 1 R-PDU-API FIBEX A2L

8



Timeline



1

Einleitung ASAM MCD

2

ASAM MCD 1

3

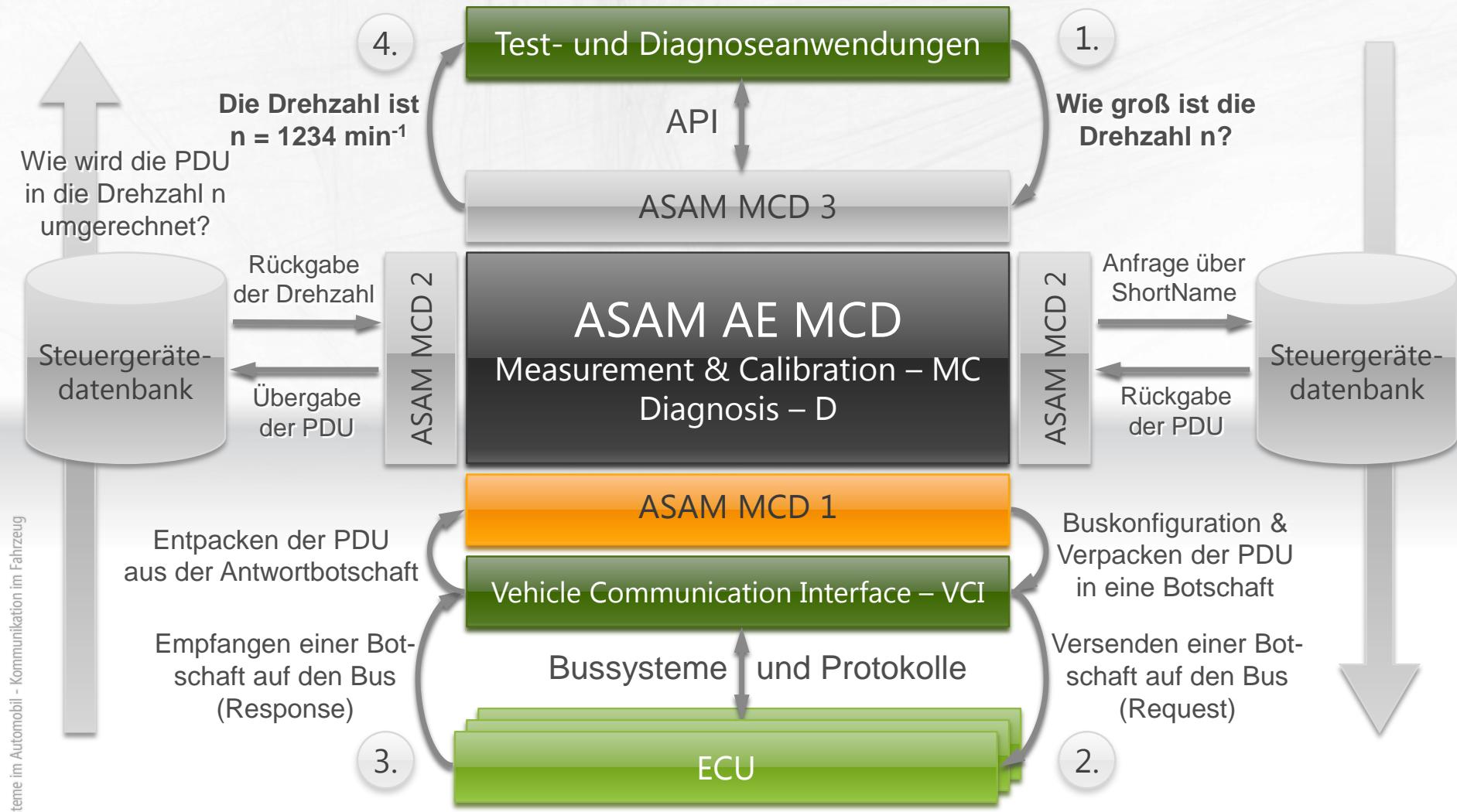
D-PDU-API

4

ASAM MCD 2 FIBEX

5

ASAM MCD 2MC



*PDU = Protocoll Data Unit

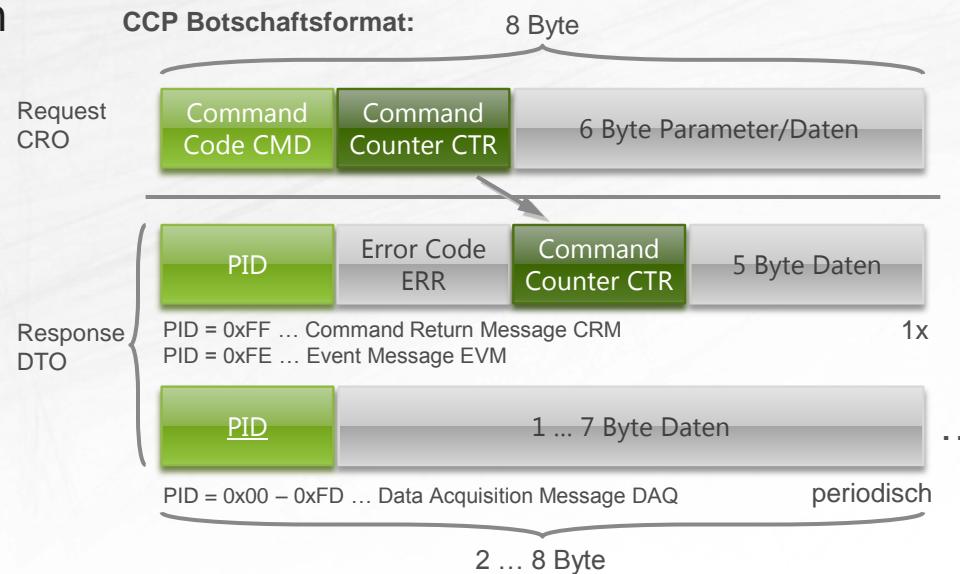
■ Kommunikation erfolgt ausschließlich über 2 CAN-Botschaften:

1. Command Receive Object (CRO)

- Request – überträgt Befehle von der Anwendung an das Steuergerät (Master → ECU)
- CMD Funktionsgruppen:
 - Verbindungsauftbau und Steuerung (z.B. CONNECT, GET_S_STATUS etc.)
 - Zugriff auf den Steuergerätespeicher (DNLOAD, UPLOAD, MOVE etc.)
 - Starten von Diagnosediensten (DIAG_SERVICE, ACTION_SERVICE etc.)
 - Flash-Programmierung PGM (CLEAR_MEMORY, PROGRAM etc.)
 - Kalibrieren CAL (SET_CAL_PAGE etc.)
 - Meßdatenerfassung DAQ (GET_DAQ_SIZE, START_STOP etc.)

2. Data Transmission Object (DTO)

- Response – enthält die Antwort vom Steuergerät (ECU → Master)
- 3 verschiedene Antworten:
 - CRM = Command Return Message
 - Folgt direkt auf den Befehl
 - 8 Byte lang mit 5 Byte Antwortdaten



- ERR != 0 → Fehler aufgetreten
- EVM = Event Message
 - Wird vom SG bei Fehlern asynchron versendet
 - 8 Byte lang mit 5 Byte Antwortdaten
 - Synchronisation über CTR
- DAQ = Data Acquisition Message
 - Wird durch Request nur „angestoßen“
 - SG sendet zyklisch Daten (Periodisch oder nach Ereignis im SG)
 - PID = Nummer in der ODT (Object Descriptor Table)
 - Max. 8 Byte lang mit 1-7 Datenbytes

MCD 1a – Extended Calibration Protocol XCP

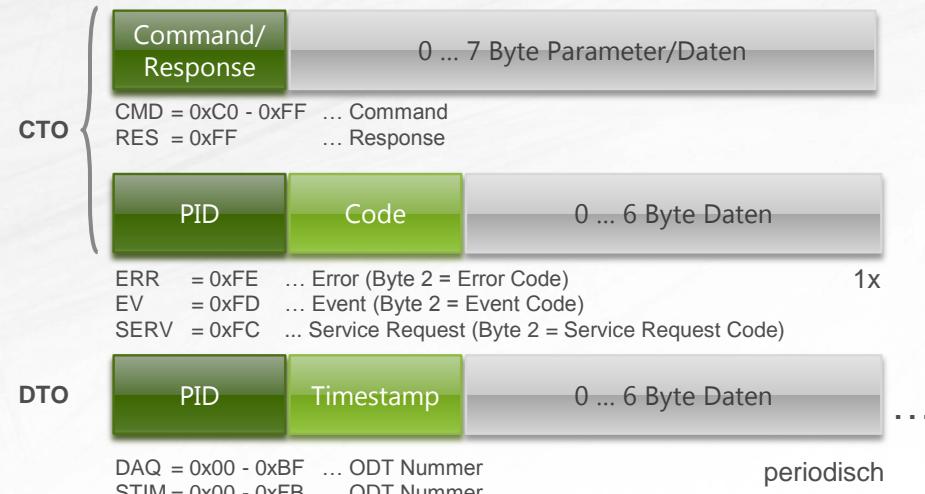


- Weiterentwicklung von CCP
- Unterstützte Bussysteme: CAN, Ethernet, FlexRay, SPI und USB
- Das selbe verbindungsorientierte Request-Response Kommunikation wie CCP
- Verwendet bei CAN für jedes SG eigenes Paar von CAN-Identifiern
- Nahezu die selben Dienste wie CCP
- Befehlskodes jedoch zu CCP inkompatibel
- Botschaften nahezu unabhängig vom Bussystem
- Kein Botschaftszähler im Botschaftsformat
- Keine Funktionsgruppe Diagnose → Verwendung von UDS oder ISOTP

XCP Botschaftsgruppen:



XCP Botschaftsformat:



- Neuer STIM-Modus für das DTO (Stimulus Data Packet)
 - Gegenstück zu DAQ für die Simulation von SG-Funktionen in der Entwicklung
- Neuer optionaler Blockmodus nach ISO 15765-2 (ISOTP)
 - Slave Block Mode: SG kann mit mehreren Antworten reagieren (Upload)
 - Master Block Mode: Applikation sendet mehrere Befehlsbotschaften an das SG (Download)

- Standardisierte API für den Zugriff des Applikationssystems auf CCP und XCP
- Parametrisierung über die MCD 2MC Datenbank (ASAP2 = AML = A2L)



1

Einleitung ASAM MCD

2

ASAM MCD 1

3

D-PDU-API

4

ASAM MCD 2 FIBEX

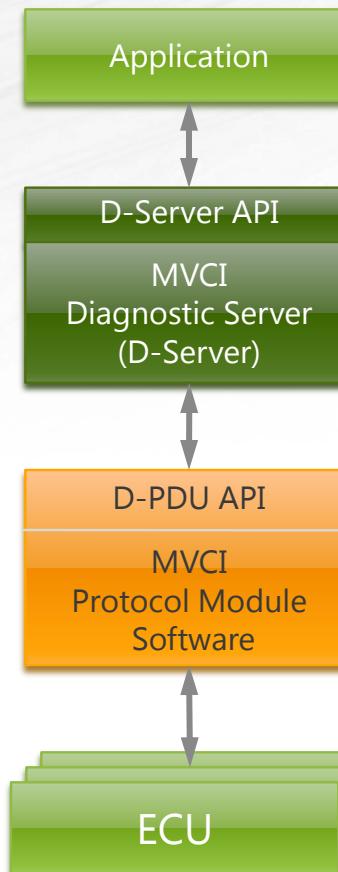
5

ASAM MCD 2MC

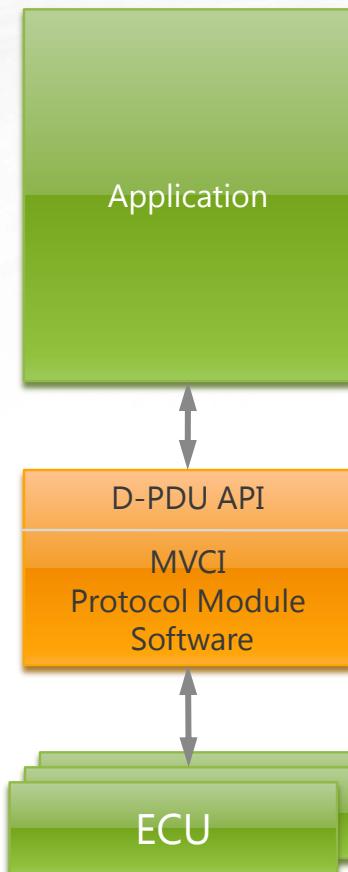
- Standardisierte Low-Level API für den Hardwarezugriff (VCI) nach ISO/DIS 22900-2
- Hardware Abstraction Layer (HAL)
- Für Tester- und Monitoring-Tools innerhalb der Fahrzeugdiagnose
- Flexible ANSI-C-basierende Schnittstelle mit mächtigem Parameter-Handling
 - RDF (**R**oot **D**escription **F**ile) – Registrierung aller auf dem Rechner installierten PDU-APIs
 - MDF (**M**odule **D**escription **F**ile) – Busse, Protokolle, Parameter, ShortName ↔ PIDs
 - CDF (**C**able **D**escription **F**ile) – Beschreibung der Verbindung OBD-Stecker ↔ Interface
- Plattformunabhängig
- Write once, run “everywhere”
- Unabhängig vom Hardwarelieferanten
- Reduzierung der Entwicklungskosten

Anwendungsszenarien:

A



B



Funktionen (Auswahl)	D-PDU API
PDUConstruct	Initialisierung der D-PDU API
PDURegisterEventCallback	(Optional) Registrierung der Callback-Funktionen. Über die Callback Funktion werden Ereignisse der PDU-API an die Applikation übergeben. Somit braucht die Applikation die PDU-API nicht zu pollen.
PDUCreateComLogicalLink	Erzeugen eines ComLogicalLink (abhängig von Protokoll, Pins, Bus)
PDUGetComParam	Lesen eines Kommunikationsparameters für den LogicalLink (CLL)
PDUSetComParam	Schreiben eines Kommunikationsparameters für den CLL
PDUSetUniqueRespIdTable	Einstellen der Unique Response Identifier Table für alle ECU Responses eines CCLs
PDUConnect	Verbindung des CLLs auf den Bus
PDUDisconnect	Trennen des CLLs vom Bus
PDUDestroyComLogicalLink	Zerstören des CLLs
PDURegisterEventCallback(NULL)	(Optional) Entfernen der Registrierung der Callback-Funktionen. Es werden keine Ereignisse mehr an die Anwendung weitergegeben.
PDUDestruct	Deinitialisieren der D-PDU API und freigeben aller internen Ressourcen

MVCI D-PDU API – Beispiel vereinfacht



```
// initialize API - AppTag will be returned in CallbackFunction -> App. could decide between different APIs
PDUConstruct(NULL, AppTag);

// Request available VCIs to get hMod
PDUGetModuleIds(&pModuleList);

// Request resources of detected VCI to get ResourceId
PDUGetResourceIds(hMod, resourceData, &pResourceIdList);

// Create ComLogicalLink (CLL) with specific resourceId, CllTag will be retuned in CallbackFunction
PDUCreateComLogicalLink(hMod, PDU_ID_UNDEF, resourceId, CllTag, &hCLL, cllCreateFlag);

// Register Callback Function to created CLL to receive Events
PDURegisterEventCallback(hMod, hCLL, EventCallbackFnc);

// Set ECU specific ComParams
PDUSetComParam(hMod, hCLL, paramItem_1);
...

// Set CanIds of ECU
PDUSetUniqueRespIdTable(hMod, hCLL, UniqueResponseIdTable);

// Connect CLL to Vehicle Bus and transmit ComParams
PDUConnect(hMod, hCLL);

// Start Communication
PDUSTartComPrimitive(hMod, hCLL, PDU_COPT_STARTCOMM, 0, NULL, ...);

// Send Requests to ECU
UNUM8 data_1[] = {0x1A, 0x90};
PDUSTartComPrimitive(hMod, hCLL, PDU_COPT_SENDRECV, sizeof(data_1), data_1, ...);
...

// Stop Communication
PDUSTartComPrimitive(hMod, hCLL, PDU_COPT_STOPCOMM, 0, NULL, ...);

// Disconnect CLL, Unregister Callback Function, Destroy CLL and Deinitialize API
PDUDisconnect(hMod, hCLL);
PDURegisterEventCallback(hMod, hCLL, NULL);
PDUDestroyComLogicalLink(hMod, hCLL);
PDUDestruct();
```

1

Einleitung ASAM MCD

2

ASAM MCD 1

3

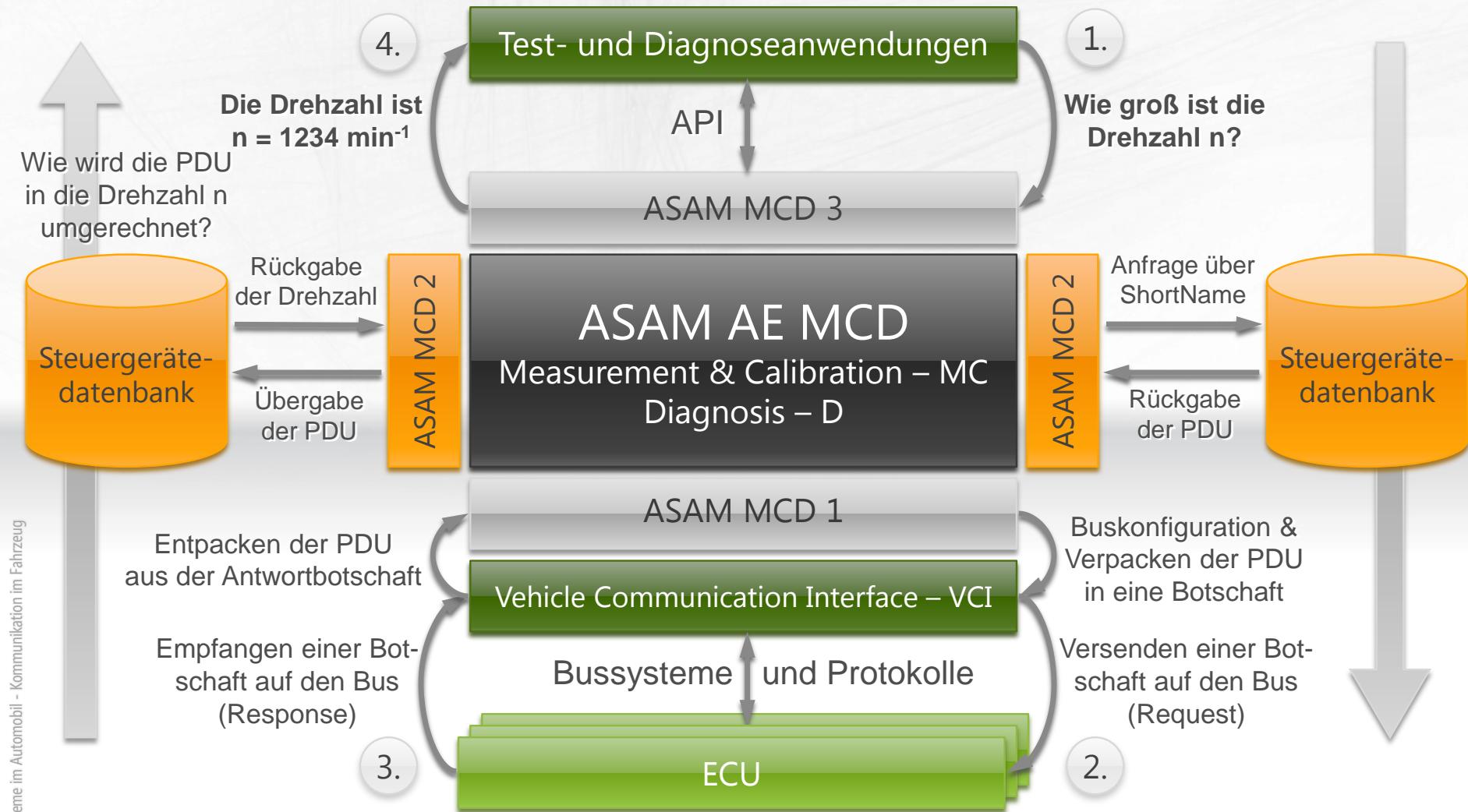
D-PDU-API

4

ASAM MCD 2 FIBEX

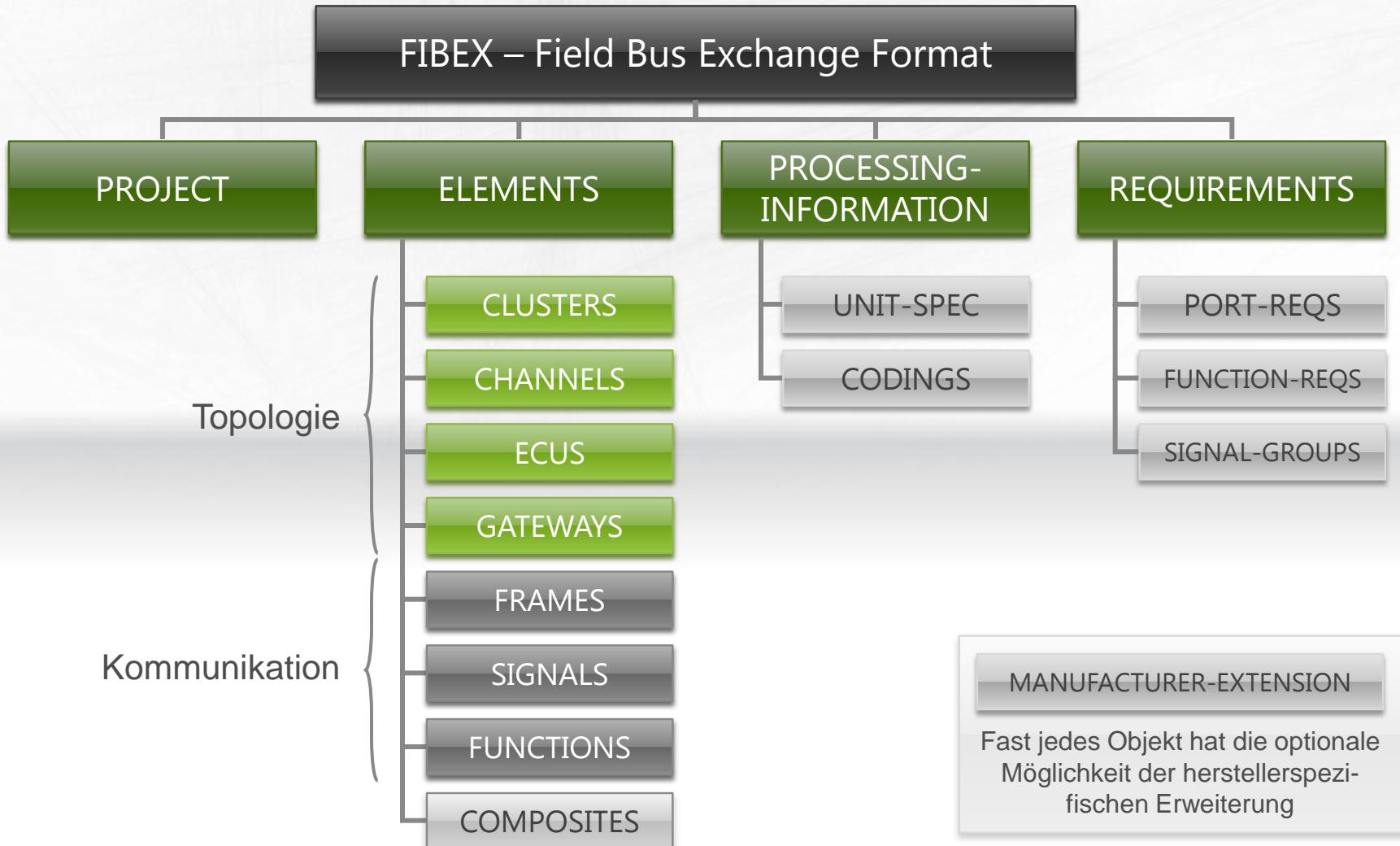
5

ASAM MCD 2MC

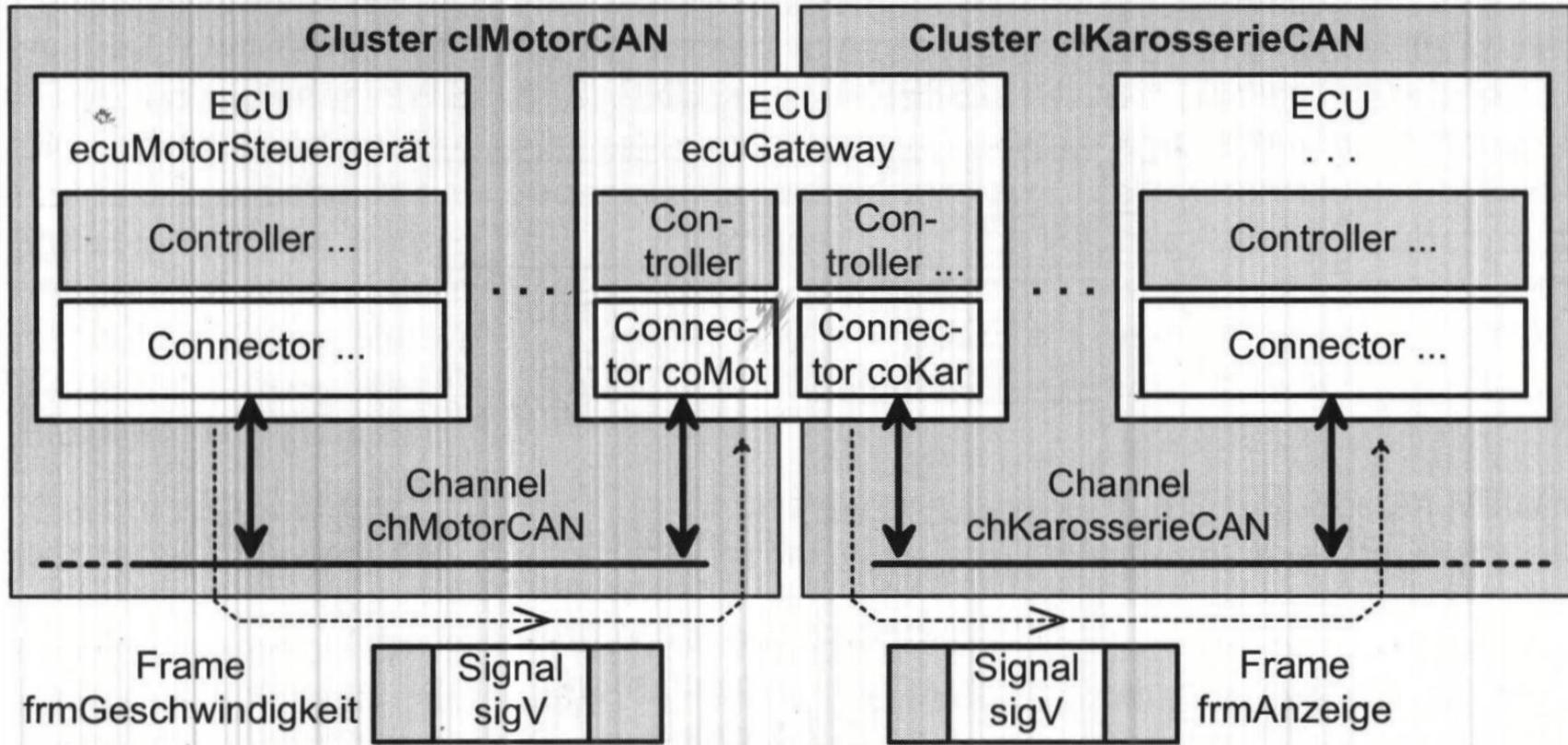


*PDU = Protocoll Data Unit

- FIBEX beschreibt die Onboard-Kommunikation eines gesamten Fahrzeugnetzes (nicht die Diagnose → ODX)
- XML-Beschreibungsformat für die Kommunikation auf folgenden Bussysteme:
 - CAN
 - TTCAN
 - LIN
 - FlexRay
 - Byteflight
 - MOST
- Ziel: Ersetzen der proprietären Formate
 - CANdb
 - LDF und NDF (LIN)
- Format der Wahl für FlexRay
- Verwendung für
 - Spezifikation
 - Test und Validierung
 - Simulation
- Sehr gut für herstellerspezifische Aufgaben erweiterbar

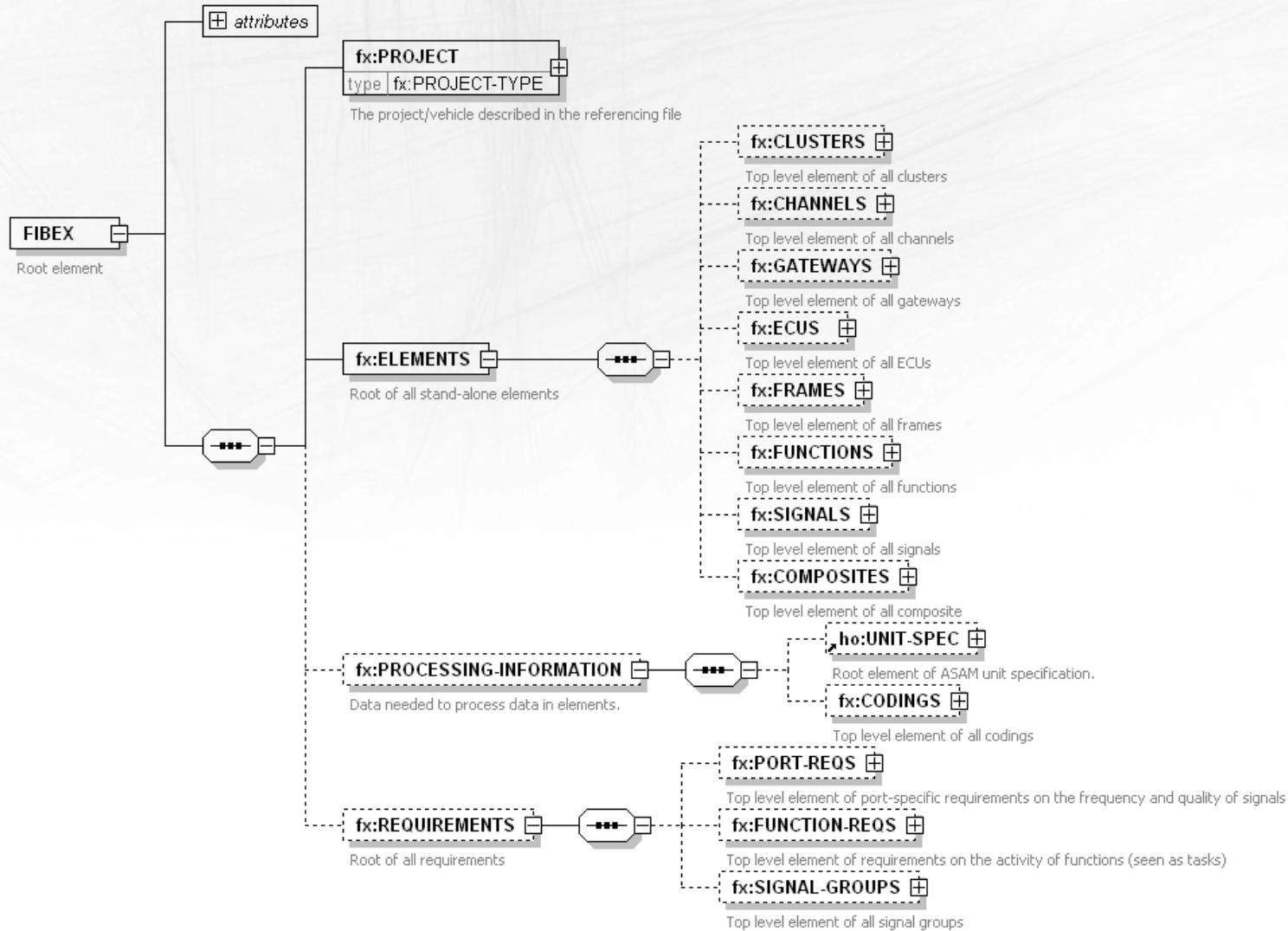


Zwei über ein Gateway gekoppelte Bussysteme



Quelle: Zimmermann

Schema



1

Einleitung ASAM MCD

2

ASAM MCD 1

3

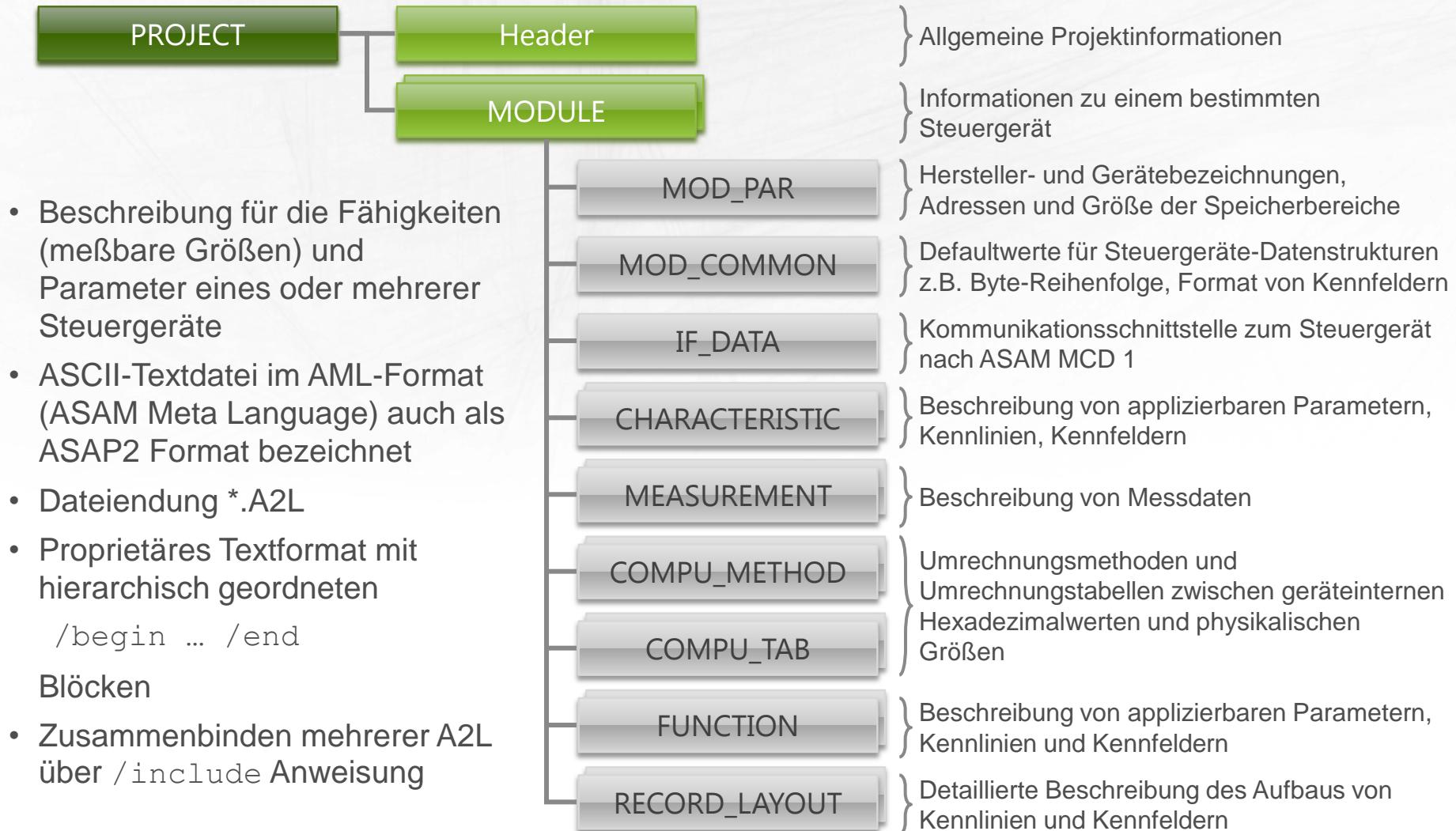
D-PDU-API

4

ASAM MCD 2 FIBEX

5

ASAM MCD 2MC



Bedatungsbeispiel



```
ASAP2_VERSION 1 3
/begin PROJECT Samtec "Samtec Demo A2L"
  /begin HEADER ""
    VERSION      "0.0.0.0"
  /end HEADER

  /begin MODULE SAMTEC ""
    /begin CHARACTERISTIC
      cur_speed
      "Engine speed"
      VALUE
      0x12345
      RLT_UWORD16
      655350.0
      CM_1/min
      0.00
      655350.0
      FORMAT "%8.1"
      EXTENDED_LIMITS 0.00 655350.0
    /end CHARACTERISTIC

    /begin RECORD_LAYOUT RLT_UWORD16
      FNC_VALUES      1 UWORD COLUMN_DIR DIRECT
    /end RECORD_LAYOUT

    /begin COMPU_METHOD
      CM_1/min
      "comp. meth. speed"
    /end COMPU_METHOD
    /begin MEASUREMENT
      uBatt
      "Battery voltage"
      SWORD
      CM_Voltage
      1
      100
      -640625,6
      640606,1
      FORMAT "%8.1"
      ECU_ADDRESS 0x7F8BAE
    /end MEASUREMENT

    /begin COMPU_METHOD
      CM_Voltage
      "comp.meth voltage"
      RAT_FUNC
      "%6.1"
      "V"
      COEFFS 0 5 0 0 0 1
    /end COMPU_METHOD
  /end MODULE
/begin PROJECT
```

Danke für Ihre Aufmerksamkeit!



28



**Sprechen Sie
mit uns!**

Wir helfen Ihnen gern.

www.emotive.de