



Welcome to the Vector J1939 Webinar

Embedded Software for J1939
2011-15-11, starting 9:00h

- ▶ Last week: Fundamentals of the J1939 Protocol
- ▶ Today: Embedded Software for J1939
Presenters: Martin Schlodder, Holger Soehnle
- ▶ Next week: J1939 Prototype and Test Development
- ▶ All registered participants will receive a link to the slides of this webinar by email.

Vector J1939 Webinars

> **Car vs. Truck**

J1939 Basics

CANbedded J1939

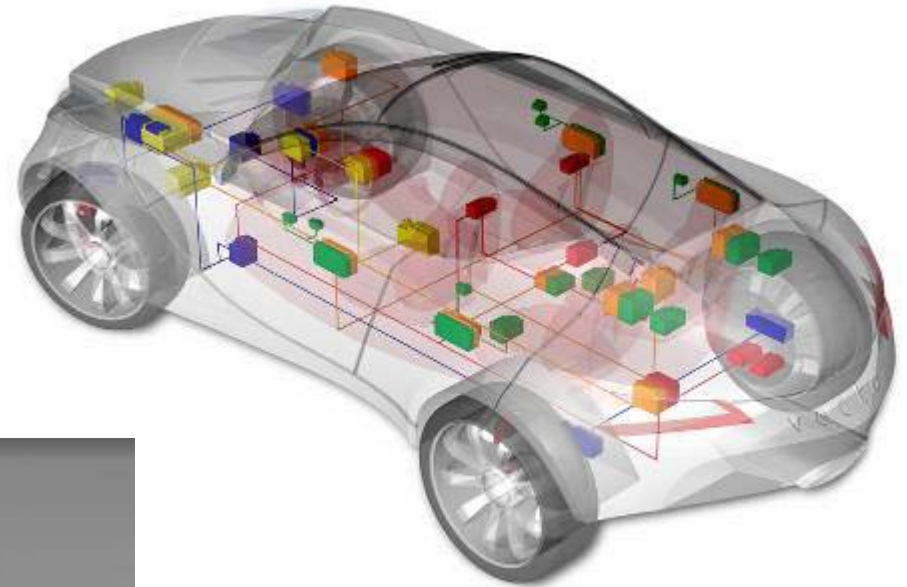
AUTOSAR & J1939

How to configure a J1939 communication stack

CANbedded API Demo

Car vs. Truck

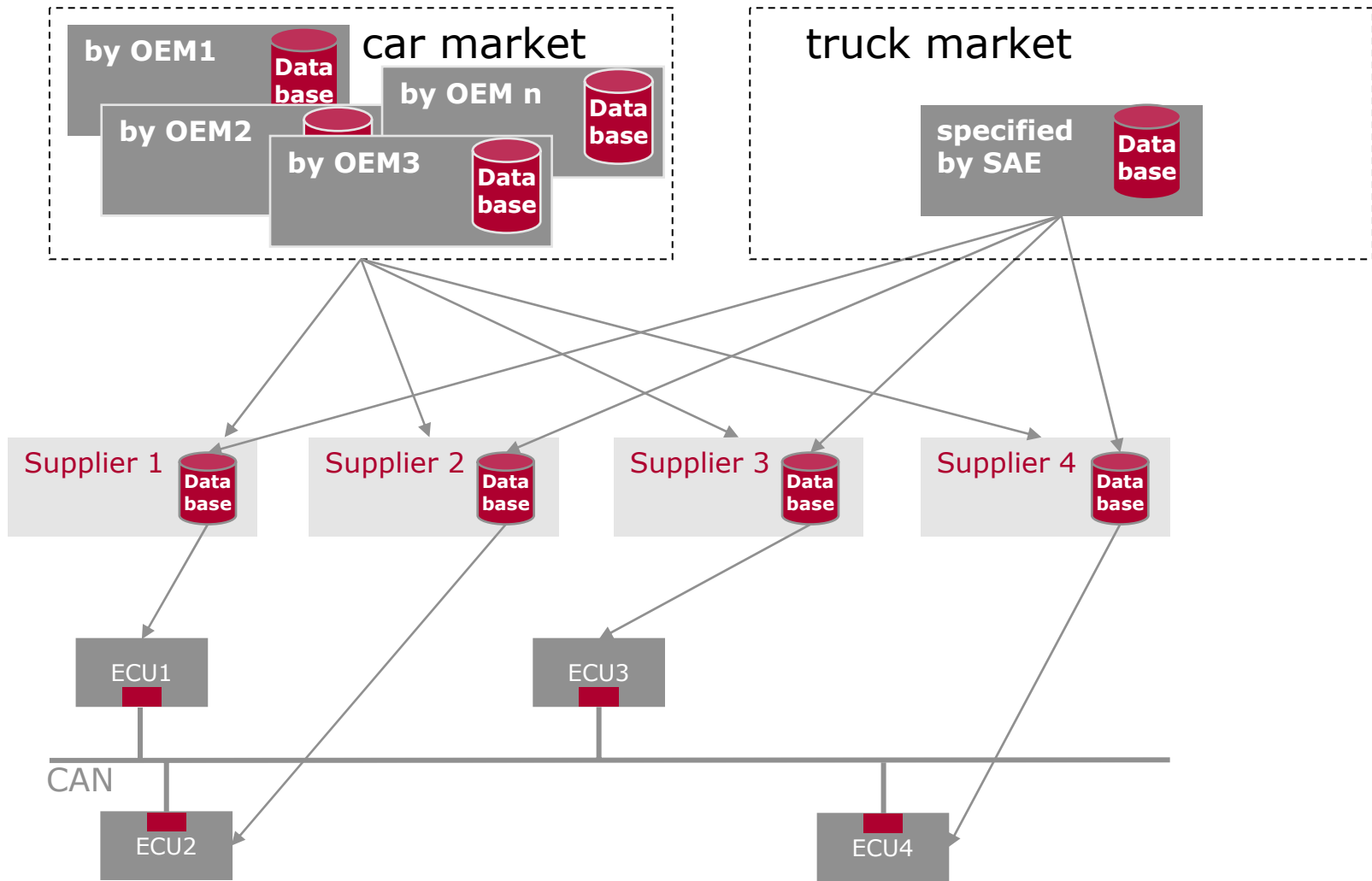
- ▶ same base for networking which is "CAN"
- ▶ but different needs by OEMs and customers



▶ details on next slides...

Car vs. Truck

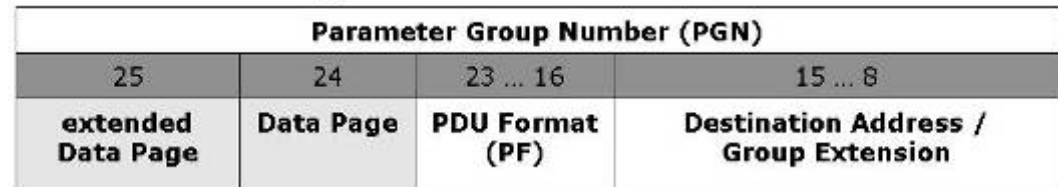
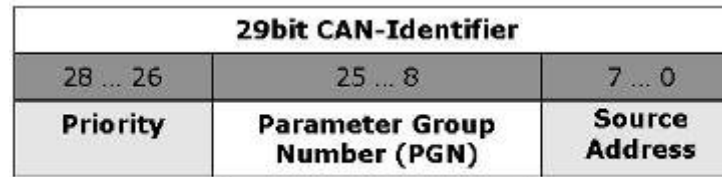
Communication Matrix



Car vs. Truck

J1939 Messages Format

- ▶ There is just one (1) communication matrix which is specified by SAE (not by OEM)
- ▶ 29-bit CAN identifier used with a specific layout
 - ▶ Parameter group
 - ▶ Source address
 - ▶ Destination addr. (depending on PGN)
- ▶ A specific ECUs supports a subset of PGNs



SAE J1939-71 Revised JAN2009 - 1110 -

PGN 44544 Tire Pressure Reference Setting TPRS

For setting the tire pressure reference values.

This message is the setpoint for the PGN 64953 Tire Pressure reference information message.

Transmission Repetition Rate: As needed
Data Length: 8
Extended Data Page: 0
Data Page: 0
PDU Format: 174
PDU Specific: DA PGN Supporting Information:
Default Priority: 6
Parameter Group Number: 44544 (0x00AE00)

Start Position	Length	Parameter Name	SPN
1	8 bits	Tire Location	3192
2	1 byte	Reference Tire Pressure Setting	3193

- ▶ NM(car) \neq NM (J1939)
- ▶ Network Management (Car)
 - ▶ Component which handles the network requests and synchronizes the sleep and wakeup of ECUs on the same bus
- ▶ Network Management (J1939)
 - ▶ Handling of ECU addresses
 - ▶ Unique network address for each ECU
 - ▶ During initialisation by “address claim”

Vector J1939 Webinars

Car vs. Truck

> **J1939 Basics**

CANbedded J1939

AUTOSAR & J1939

How to configure a J1939 communication stack

CANbedded API Demo

OSI Network Structure

Application (7)

Presentation (6)

Session (5)

Transport (4)

Network (3)

Data Link (2)

Physical (1)

Physical Transmission media

J1939 documents

J1939: Protocol introduction

J1939/0X: Industry Group

J1939/01: Truck and Bus

J1939/02: Agricultural Equipment

J1939/81: Network Management

(dynamic address assignment)

J1939/7X: Application Layer

J1939/71: Common application

J1939/72: Virtual Terminal

J1939/73: Diagnosis

J1939/31: Bridge, Router, Gateway, Filter

J1939/21: Transport Protocols (DLC> 8)

J1939/1X: Physical layer

J1939/11: 250K Bits/sec., Twisted pair, shielded

J1939/12: 250KBits/sec., Twisted quad

J1939/13: Diagnostic Plug

► to get these documents please visit: www.sae.org

- ▶ J1939 specific Transport Protocols (J1939-21)
 - ▶ Broadcast Announce Message (BAM)
 - ▶ Connection Mode Data Transfer (CMDT, called RTS/CTS in J1939 std.)
 - ▶ Different to ISO 15765 TP
- ▶ J1939 Diagnostics (J1939-73)
 - ▶ J1939 specifies diagnostic messages (DM1 – DM52)
 - ▶ Different to ISO 14229/14230 (UDS/KWP2000)
- ▶ Compatibility to ISO specified TP/Diagnostics
 - ▶ J1939 reserves specific PGNs for ISO-TP
 - ▶ ISO-TP/Diagnostics can be run in parallel to J1939 diagnostics
 - ▶ Typical use case in European trucks

- ▶ J1939-81
- ▶ What's behind?
 - ▶ Each ECU needs an unique address
 - ▶ NM provides a mechanism to assign a unique address in networks with a non-static configuration (nodes not known at configuration time).
- ▶ How is it done?
 - ▶ Address claim sequence (ACL) is sent at network startup or when new nodes are connected during runtime

- ▶ Static Networks (fixed ECU addresses)
 - ▶ Constant number of ECUs
 - ▶ Fixed communication matrix
 - ▶ No Address Claim needed
 - > but Address Claim message can be sent (as 'information')
- ▶ Simple Dynamic Networks (changing ECU addresses)
 - ▶ Change of own ECU address during reboot
 - ▶ No tracking of other ECU addresses
 - ▶ Suitable for networks using mainly broadcast PGNs
- ▶ Fully dynamic network management
 - ▶ Used in ISO 11783 applications
 - ▶ E.g. tractor with different implements

Vector J1939 Webinars

Car vs. Truck

J1939 Basics

> CANbedded J1939

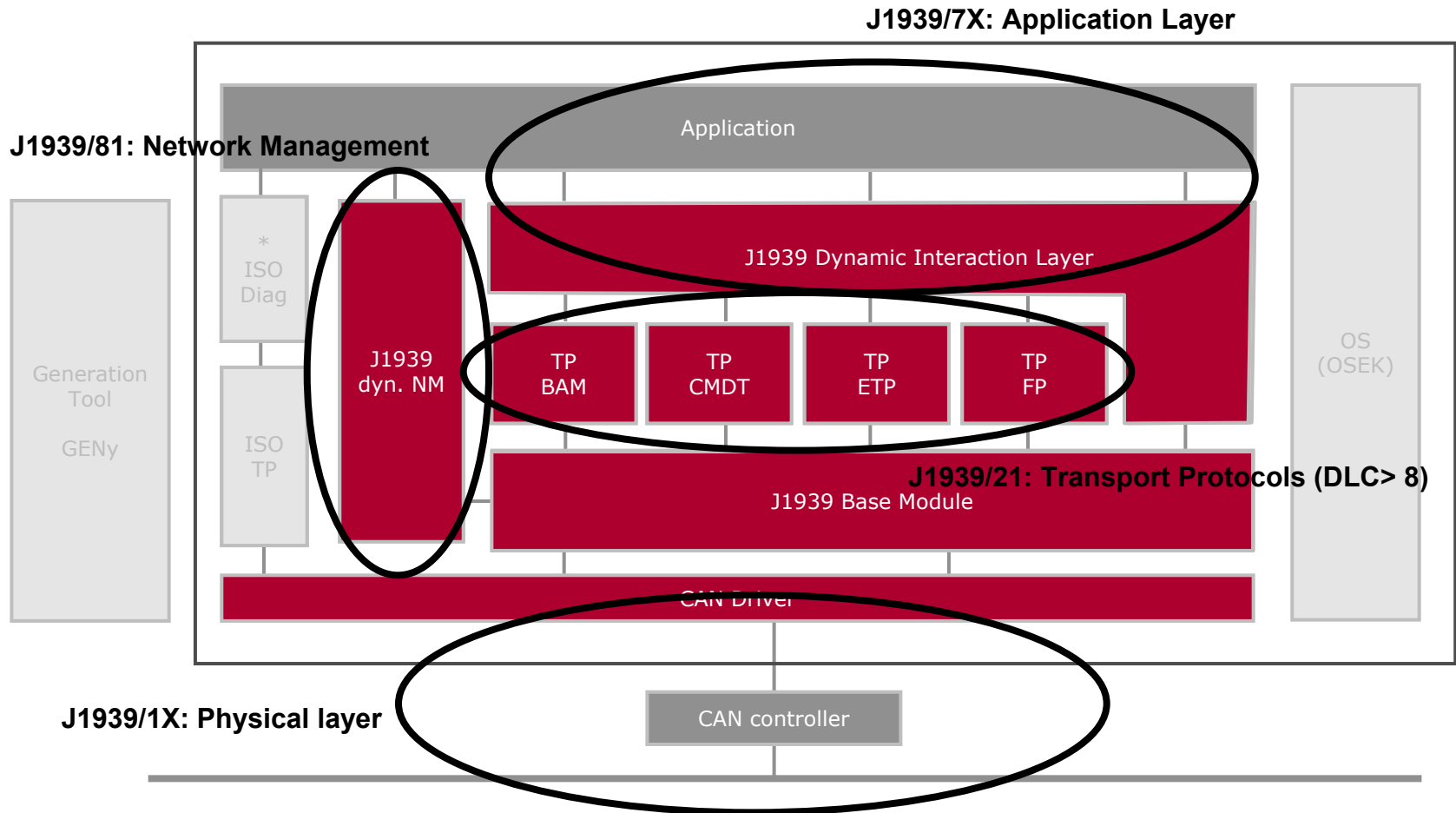
AUTOSAR & J1939

How to configure a J1939 communication stack

CANbedded API Demo

CANbedded J1939

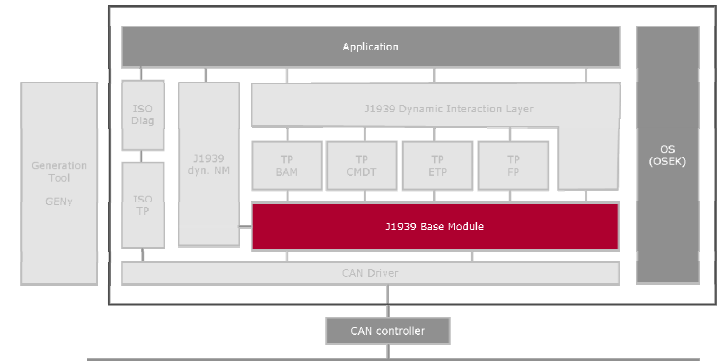
Features



CANbedded J1939

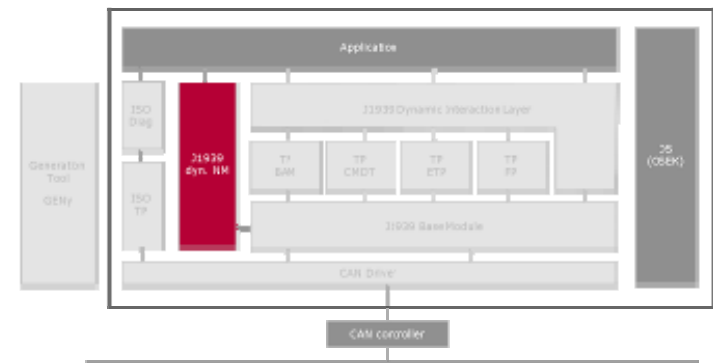
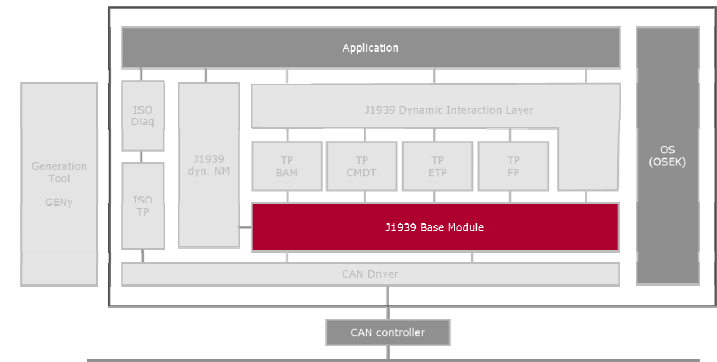
Technical Details: Base Module

- ▶ General J1939 support
- ▶ Message Handling
 - ▶ ID filtering (hash / linear)
 - ▶ Buffered/queued
- ▶ Basic Network Management functionality
 - ▶ Details on next slide
- ▶ Option: virtual ECU
 - ▶ Integrate several standardized J1939 ECUs into one real ECU



- ▶ Basic NM functionality
 - ▶ Static Networks (fixed ECU addr.)
 - > No Address Claim
 - > Initial Address Claim
 - ▶ Simple Dyn. Networks (changing ECU addr.)
 - > Initial Address Claim
 - > Change of address via appl. callouts

- ▶ Fully dynamic NM (e.g. ISO-11783)
 - ▶ Dynamic address claiming



CANbedded J1939

Technical Details: Transport Protocols

- ▶ Broadcast Announce Message (BAM)

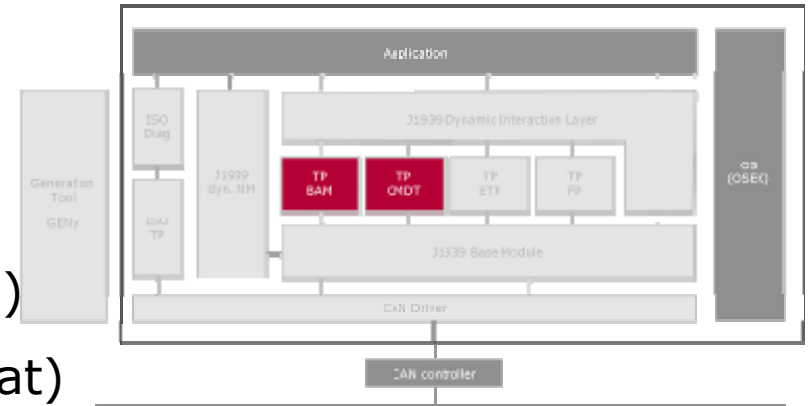
- ▶ e.g. DM1 (active errors, heart beat)

- ▶ Connection Mode Data Transfer (CMDT)

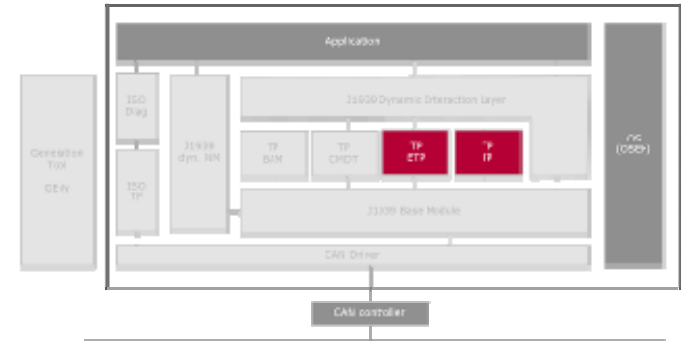
- ▶ e.g. DM2 (prev. active errors)

- ▶ J1939 TP Features

- ▶ Timeout supervision of BAM and CMDT
- ▶ Support of PGNs with ≤ 1785 bytes (max. defined by J1939)
- ▶ Support for variable size PGNs



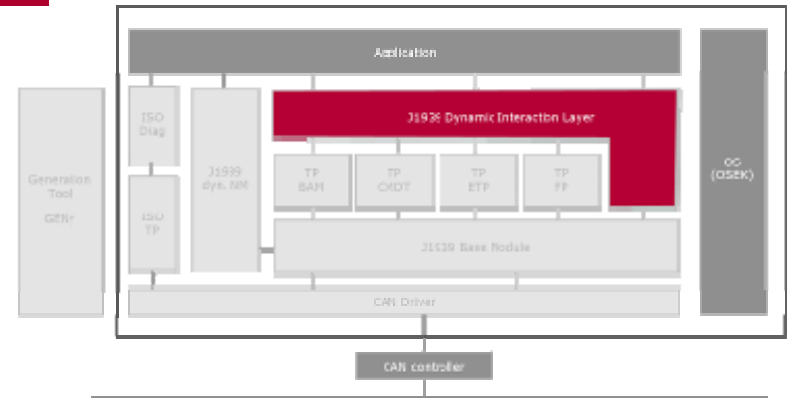
- ❑ ISO11783 extension
 - ❑ ETP: Extended Transport Protocol
 - ❑ Based on J1939/CMDT, but supports longer data
 - ❑ Main purpose: object pools for Virtual Terminal (1785 ... >100MB)
 - ❑ FP: Fast-Packet TP (NMEA2000)
 - ❑ Transport of position data (GNSS, size <= 233bytes)
- ❑ Working Set support
 - ❑ Passive listening to CMDT and ETP



CANbedded J1939

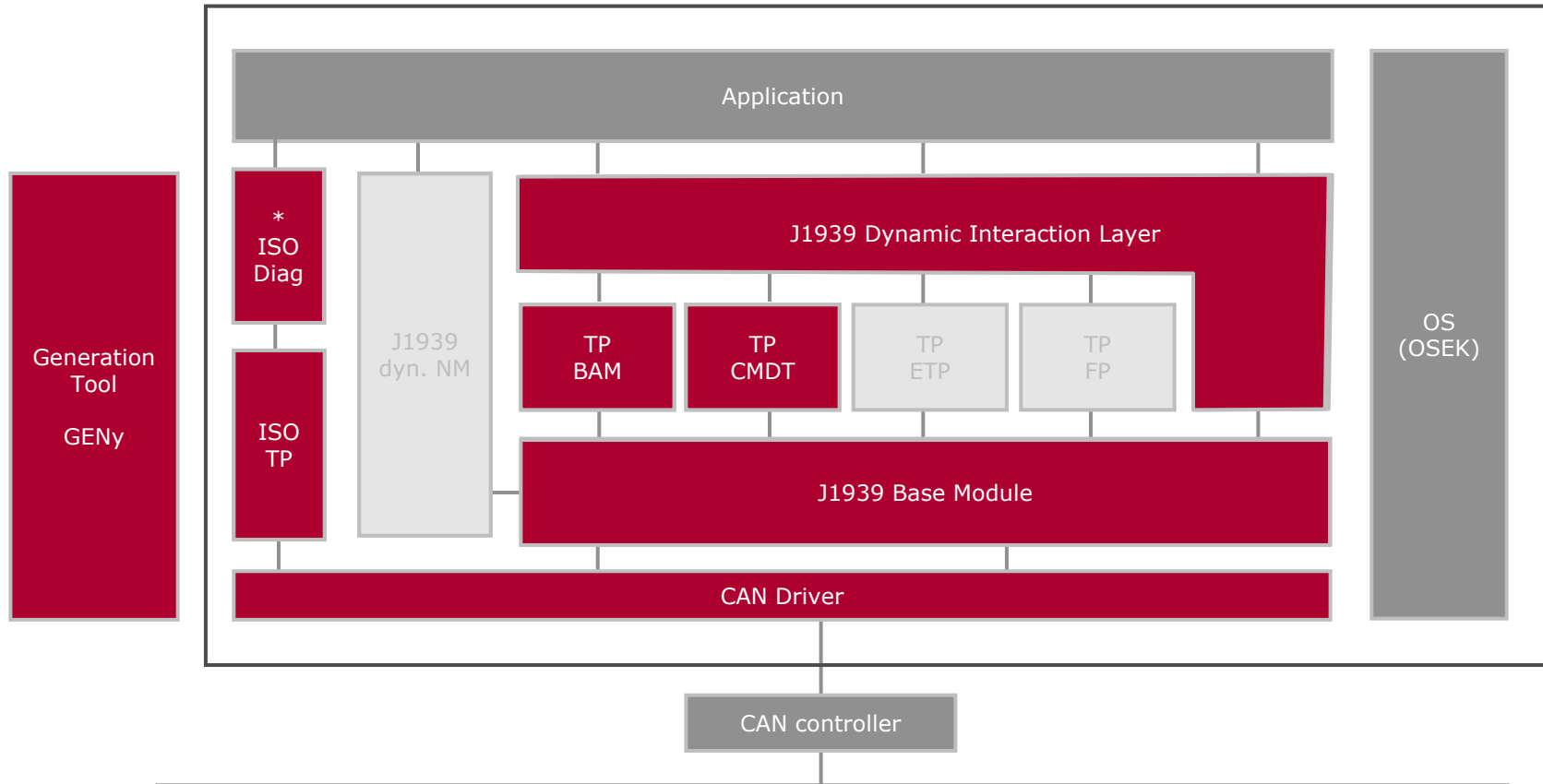
Technical Details: Interaction Layer

- ▶ Dynamic IL
 - ▶ Supports variable CAN IDs
 - ▶ Macros to access PGs and Parameters of PGs
 - ▶ Send types
 - > Cyclic and spontaneous
 - ▶ Timeout supervision (Rx)



CANbedded J1939

Example: Typical Truck Configuration



*Diagnostic options: UDS or KWP

Vector J1939 Webinars

Car vs. Truck

J1939 Basics

CANbedded J1939

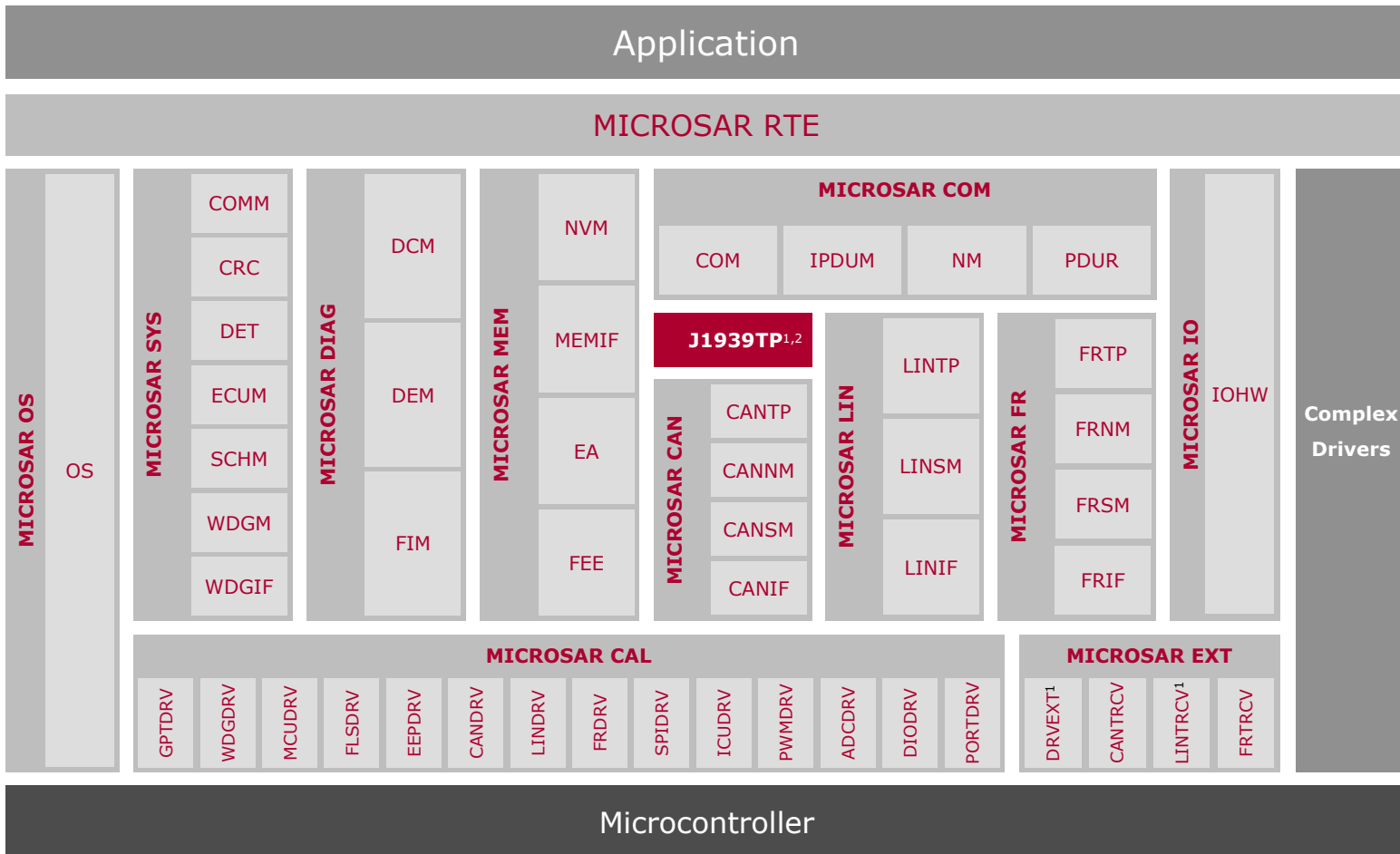
> **AUTOSAR & J1939**

How to configure a J1939 communication stack

CANbedded API Demo

AUTOSAR & J1939

MICROSAR – Vector's full range of Basic Software Modules



²BAM and CMDT Option available

- ▶ AUTOSAR: due to static approach there are limitations concerning usage of J1939
 - ▶ E.g. CAN identifiers fix at configuration time (→ SA, DA)
 - ▶ Limitations often acceptable for (european) truck use
- ▶ Vector already provides an AUTOSAR solution for truck ECUs
- ▶ Vector solution already available for series production
 - ▶ First customer SOP very soon

Vector J1939 Webinars

Car vs. Truck

J1939 Basics

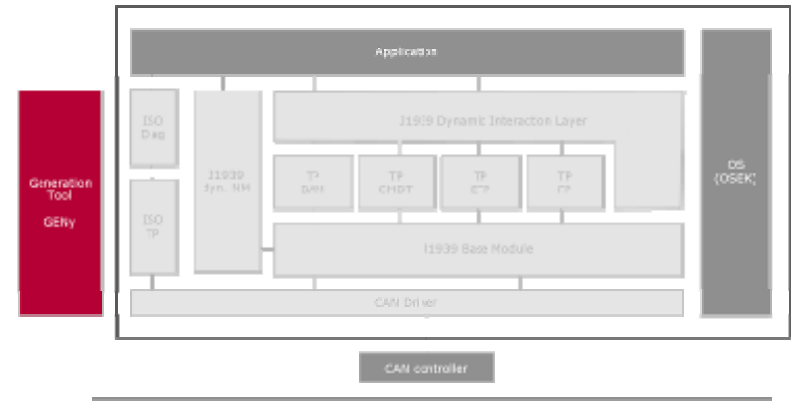
CANbedded J1939

AUTOSAR & J1939

> How to configure a J1939 communication stack

CANbedded API Demo

How to configure a J1939 communication stack



- ▶ GENy Demo
 - ▶ Generates all J1939 sources
 - ▶ Live demonstration

Vector J1939 Webinars

Car vs. Truck

J1939 Basics

CANbedded J1939

AUTOSAR & J1939

How to configure a J1939 communication stack

> CANbedded API Demo

Transmission API

▶ Query transmission state

```
state = GET_TX_PGSTATE_<CANdbName>_<PGN>_<Sender>_<Receiver>()  
state: PG_TX_FREE, PG_TX_REQ, PG_TX_XMT, PG_TX_TX (defined in j1939cfg.h)
```

▶ Query and set transmission type

```
SET_TX_PG_REPTYPE_<CANdbName>_<PGN>_<Sender>_<Receiver>(type)  
type: TT_ASYNC, TT_CYCLIC, TT_BAM_ASYNC, TT_BAM_CYCLIC, TT_CMDT_ASYNC, TT_CMDT_CYCLIC  
(defined in j1939cfg.h)  
SET_TX_PG_ENABLED_<CANdbName>_<PGN>_<Sender>_<Receiver>(boolean)
```

▶ Check and update signal values

```
GET_TX_<CANdbName>_<Signalname>_<PGN>_<Sender>_<Receiver>();  
SET_TX_<CANdbName>_<Signalname>_<PGN>_<Sender>_<Receiver>(data);
```

▶ Trigger transmission

```
SEND_PG_<CANdbName>_<PGN>_<Sender>_<Receiver>()
```

Reception API

❑ Query attributes (sender, DLC, ...)

```
pgn = GET_RX_PGN(pg)
```

```
sender_address = GET_RX_SOURCE_ADDRESS(pg)
```

```
priority = GET_RX_PRIORITY(pg)
```

```
dlc = GET_RX_DATALENGTH(pg)
```

❑ Read signal values

❑ Buffered mode

```
data = GET_RX_<CANdbName>_<Sig>_<PGN>_<Sndr>_<Rcvr>(pg)
```

```
ptr = GET_RX_DATAPTR_<CANdbName>_<Sig>_<PGN>_<Sndr>_<Rcvr>(pg)
```

❑ Queued mode

```
data = GET_RX_<CANdbName>_<Sig>_<PGN>_<Sndr>_<Rcvr>()
```

```
ptr = GET_RX_DATAPTR_<CANdbName>_<Sig>_<PGN>_<Sndr>_<Rcvr>()
```

Please visit us at: www.vector.com

For details about embedded software
please contact us directly:

embedded@vector.com

Thank you for your attention.

▶ Please also consider our next J1939 webinar:

„J1939 Prototype and Test Development with CANoe.J1939 –
Use the Standard Efficiently“

in english: Wednesday May, 18th 9:00h (GMT+2h)

in deutsch: Mittwoch 18.Mai, 14:00h