

## Experiences when introducing FlexRay in a fail-operational system

2008-11-04, Automotive Electronics, Paris

# Experiences when introducing FlexRay in a fail-operational system

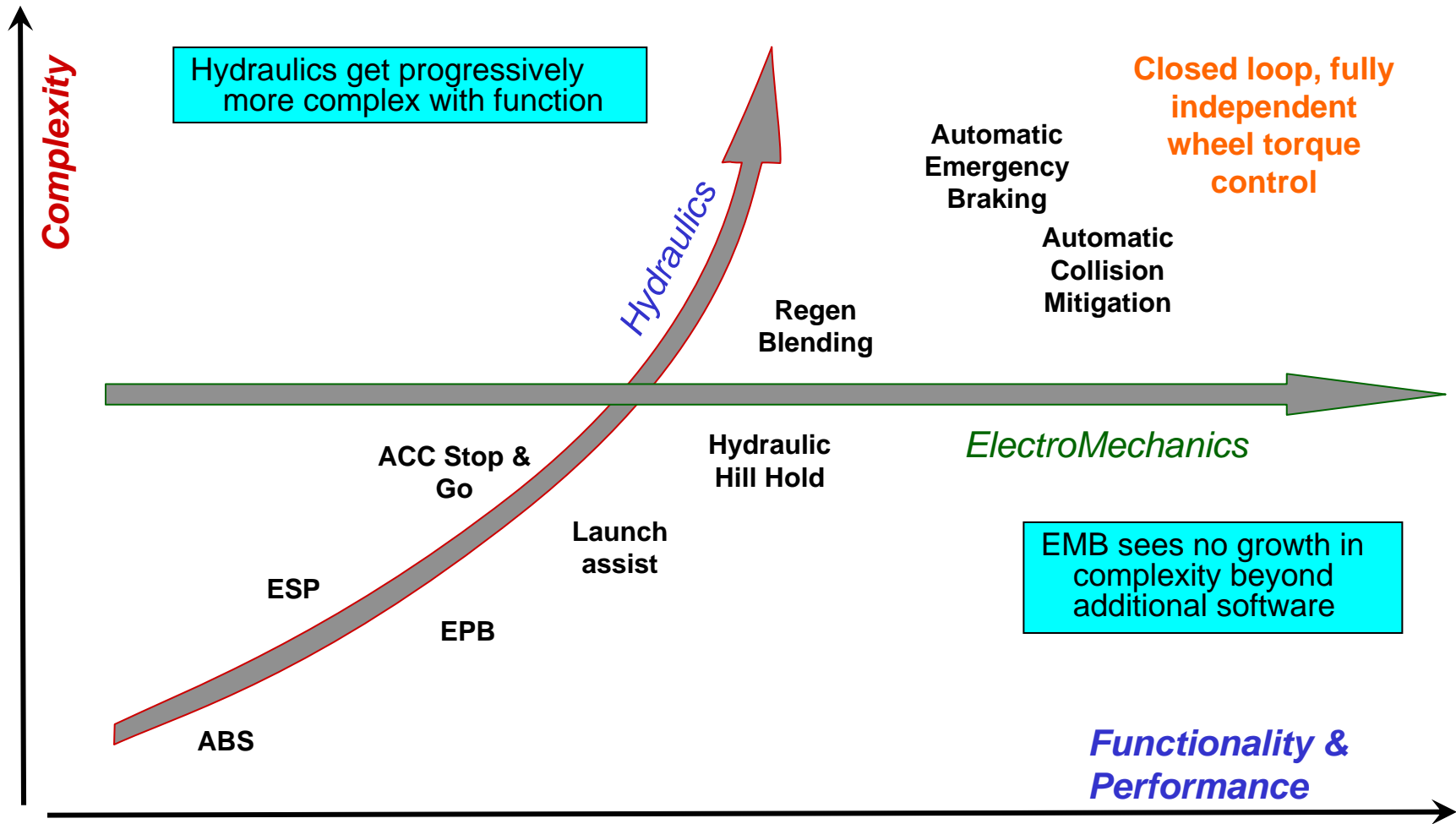
Presented by Arnaud Balmitgère, Delphi Chassis France:

- ❑ What do Distributed Full Electromechanical Brakes offer?
- ❑ Delphi experience of Brake-By-Wire Architectures
- ❑ Delphi Brake-By-Wire FlexRay Architecture
- ❑ Bus Requirements for a Distributed Brake Architecture
- ❑ Corner Brake Module Architecture
- ❑ Implementation of FlexRay Stack and Brake Software
- ❑ Outcome
- ❑ Cooperation with Vector

Presented by Dirk Großmann, Vector Informatik GmbH:

- ❑ Challenges
- ❑ Solution
  - ❑ Vector approach
  - ❑ FlexRay BSW
  - ❑ BSW Duties
- ❑ Experiences
- ❑ Future Prospects

# What do Distributed Full Electromechanical Brakes offer?



# What do Distributed Full Electromechanical Brakes offer?

## The Electromechanical Brakes offer:

- ❑ High-speed, intelligent actuation → Transparent implementation of function
- ❑ Elimination of hydraulics and fluids → Lower manufacturing, maintenance and environmental burdens.
- ❑ Decoupled Driver Interface → Simplifies autonomous functionality
- ❑ Zero engine bay packaging → More space for Hybrid PT requirements
- ❑ Software command arbitration → No centralised actuation compromises
- ❑ Corner 'closed-loop" operation → Reduces effects of efficiency variation.
- ❑ Totally open function platform → Easy implementation of OEM functions.

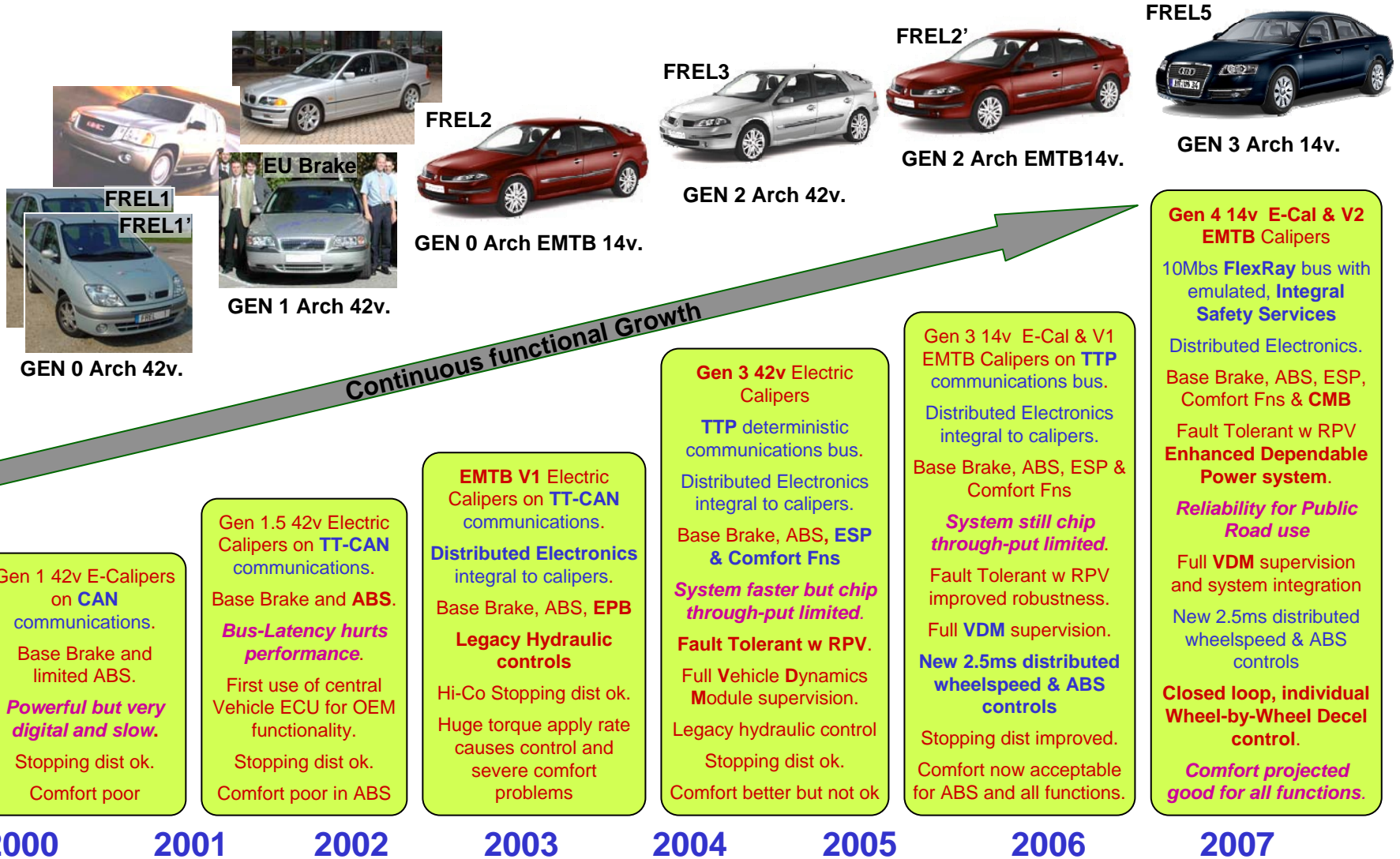
# What do Distributed Full Electromechanical Brakes offer?

Distributed Computing Architecture brings:

- ❑ Quasi-Autonomous wheel control → Reduces criticality of central supervisor
- ❑ High-speed localised control loops → Eliminates latency and resulting control instabilities
- ❑ Allows distributed sensor processing → Eliminates additional processing units
- ❑ Allows fail-silent actuation control → Inherently fault tolerant system
- ❑ Redundant action/reaction computing → An ASIL D compliant system

# Delphi experience of Brake-By-Wire Architectures

Systems (DEB4.0)



**2000**

Gen 1 42v E-Calipers on **CAN** communications.  
Base Brake and limited ABS.  
*Powerful but very digital and slow.*  
Stopping dist ok.  
Comfort poor

**2001**

Gen 1.5 42v Electric Calipers on **TT-CAN** communications.  
Base Brake and **ABS**.  
*Bus-Latency hurts performance.*  
First use of central Vehicle ECU for OEM functionality.  
Stopping dist ok.  
Comfort poor in ABS

**2002**

**EMTB V1** Electric Calipers on **TT-CAN** communications.  
**Distributed Electronics** integral to calipers.  
Base Brake, ABS, **EPB**  
**Legacy Hydraulic controls**  
Hi-Co Stopping dist ok.  
Huge torque apply rate causes control and severe comfort problems

**2003**

**Gen 3 42v** Electric Calipers  
**TTP** deterministic communications bus.  
Distributed Electronics integral to calipers.  
Base Brake, ABS, **ESP & Comfort Fns**  
*System faster but chip through-put limited.*  
**Fault Tolerant w RPV.**  
Full Vehicle Dynamics Module supervision.  
Legacy hydraulic control  
Stopping dist ok.  
Comfort better but not ok

**2004**

**Gen 3 14v** E-Cal & V1 EMTB Calipers on **TTP** communications bus.  
Distributed Electronics integral to calipers.  
Base Brake, ABS, **ESP & Comfort Fns**  
*System still chip through-put limited.*  
Fault Tolerant w RPV improved robustness.  
Full **VDM** supervision.  
**New 2.5ms distributed wheelspeed & ABS controls**  
Stopping dist improved.  
Comfort now acceptable for ABS and all functions.

**2005**

**Gen 3 14v** E-Cal & V1 EMTB Calipers on **TTP** communications bus.  
Distributed Electronics integral to calipers.  
Base Brake, ABS, **ESP & Comfort Fns**  
*System still chip through-put limited.*  
Fault Tolerant w RPV improved robustness.  
Full **VDM** supervision.  
**New 2.5ms distributed wheelspeed & ABS controls**  
Stopping dist improved.  
Comfort now acceptable for ABS and all functions.

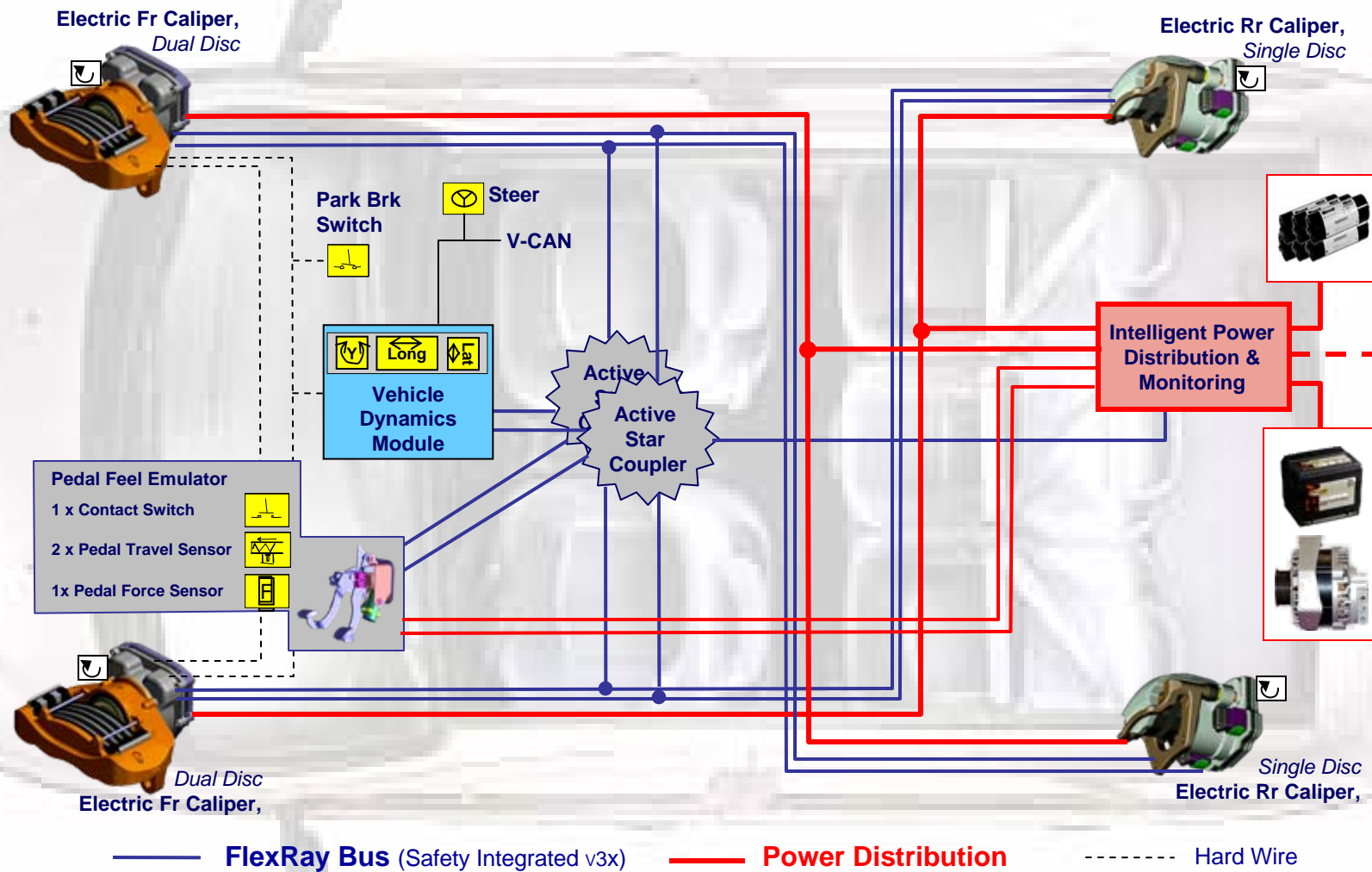
**2006**

**Gen 3 14v** E-Cal & V1 EMTB Calipers on **TTP** communications bus.  
Distributed Electronics integral to calipers.  
Base Brake, ABS, **ESP & Comfort Fns**  
*System still chip through-put limited.*  
Fault Tolerant w RPV improved robustness.  
Full **VDM** supervision.  
**New 2.5ms distributed wheelspeed & ABS controls**  
Stopping dist improved.  
Comfort now acceptable for ABS and all functions.

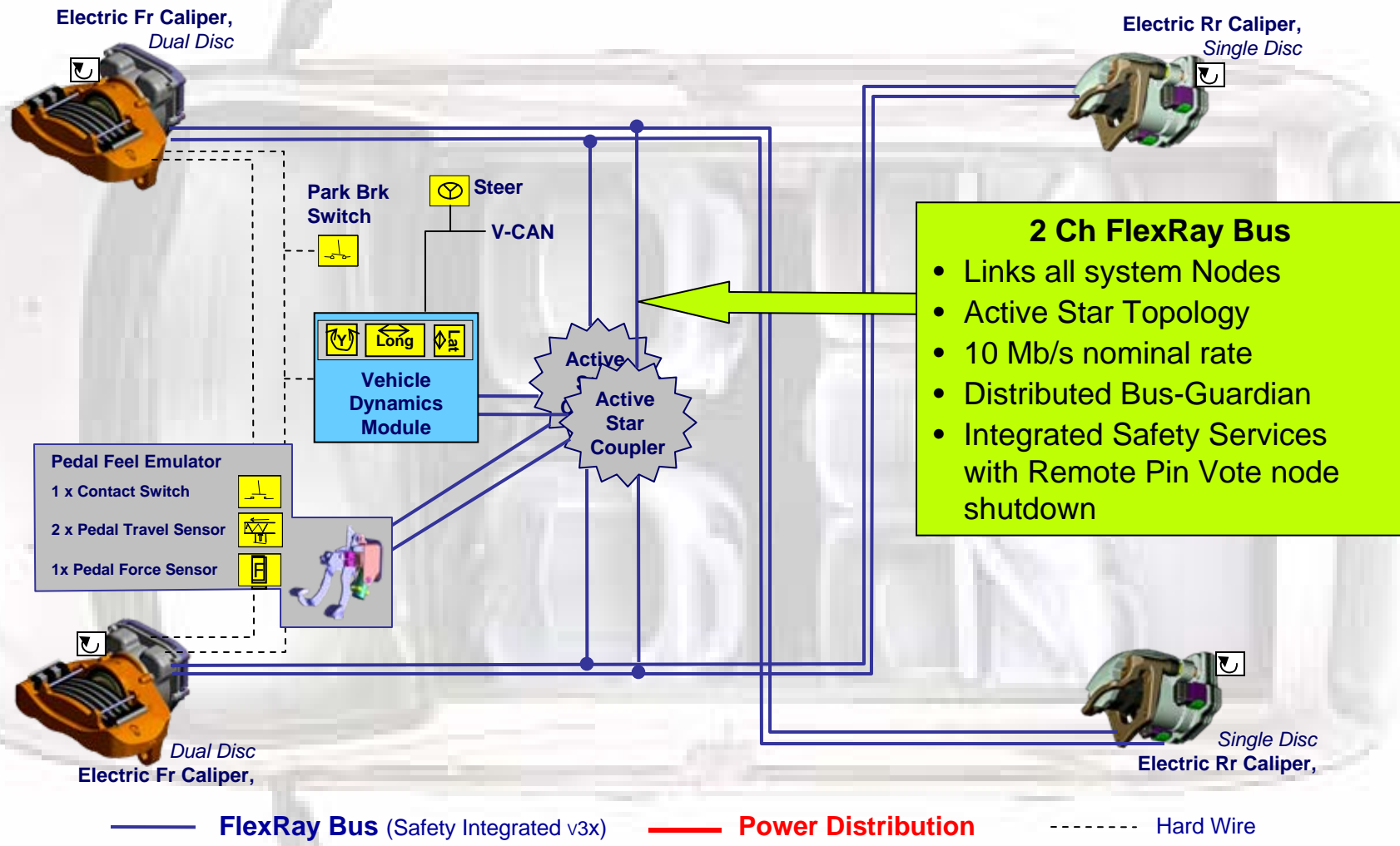
**2007**

**Gen 4 14v** E-Cal & V2 EMTB Calipers  
**10Mbs FlexRay** bus with emulated, **Integral Safety Services**  
Distributed Electronics.  
Base Brake, ABS, **ESP, Comfort Fns & CMB**  
Fault Tolerant w RPV  
**Enhanced Dependable Power system.**  
*Reliability for Public Road use*  
Full **VDM** supervision and system integration  
New 2.5ms distributed wheelspeed & ABS controls  
**Closed loop, individual Wheel-by-Wheel Decel control.**  
*Comfort projected good for all functions.*

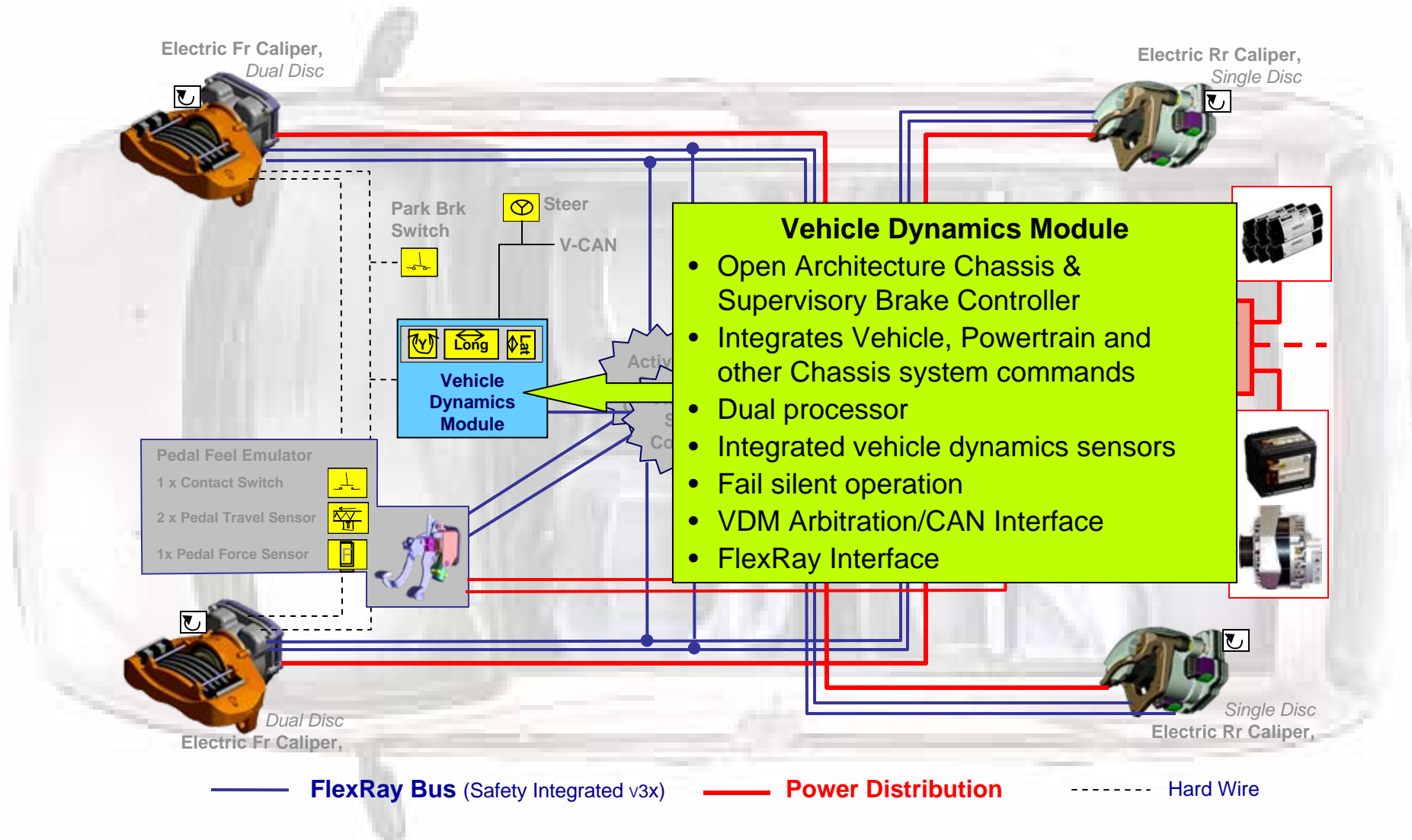
# Delphi Brake-By-Wire FlexRay Architecture (Developmental)



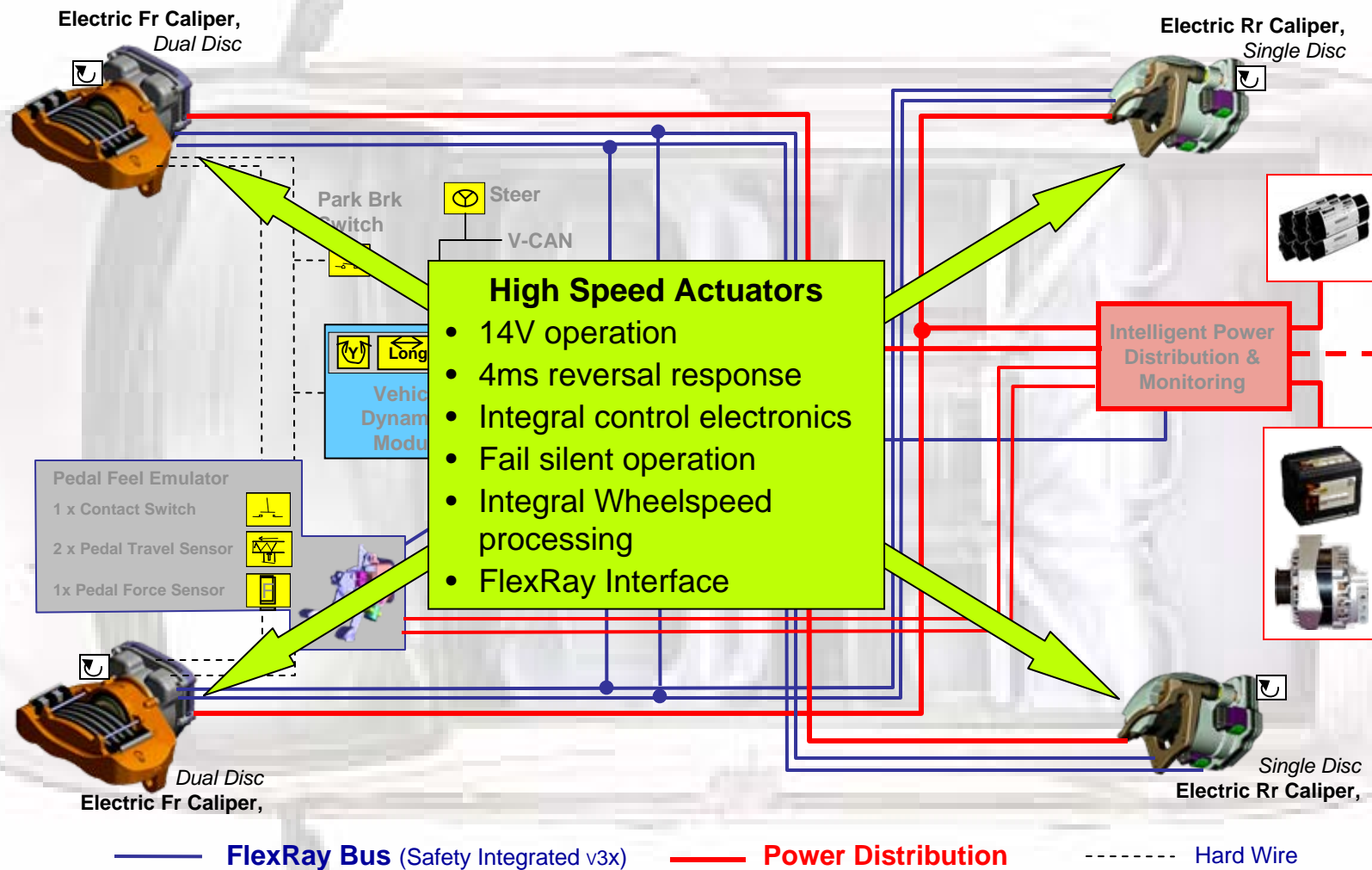
# Delphi Brake-By-Wire FlexRay Architecture: FlexRay Bus



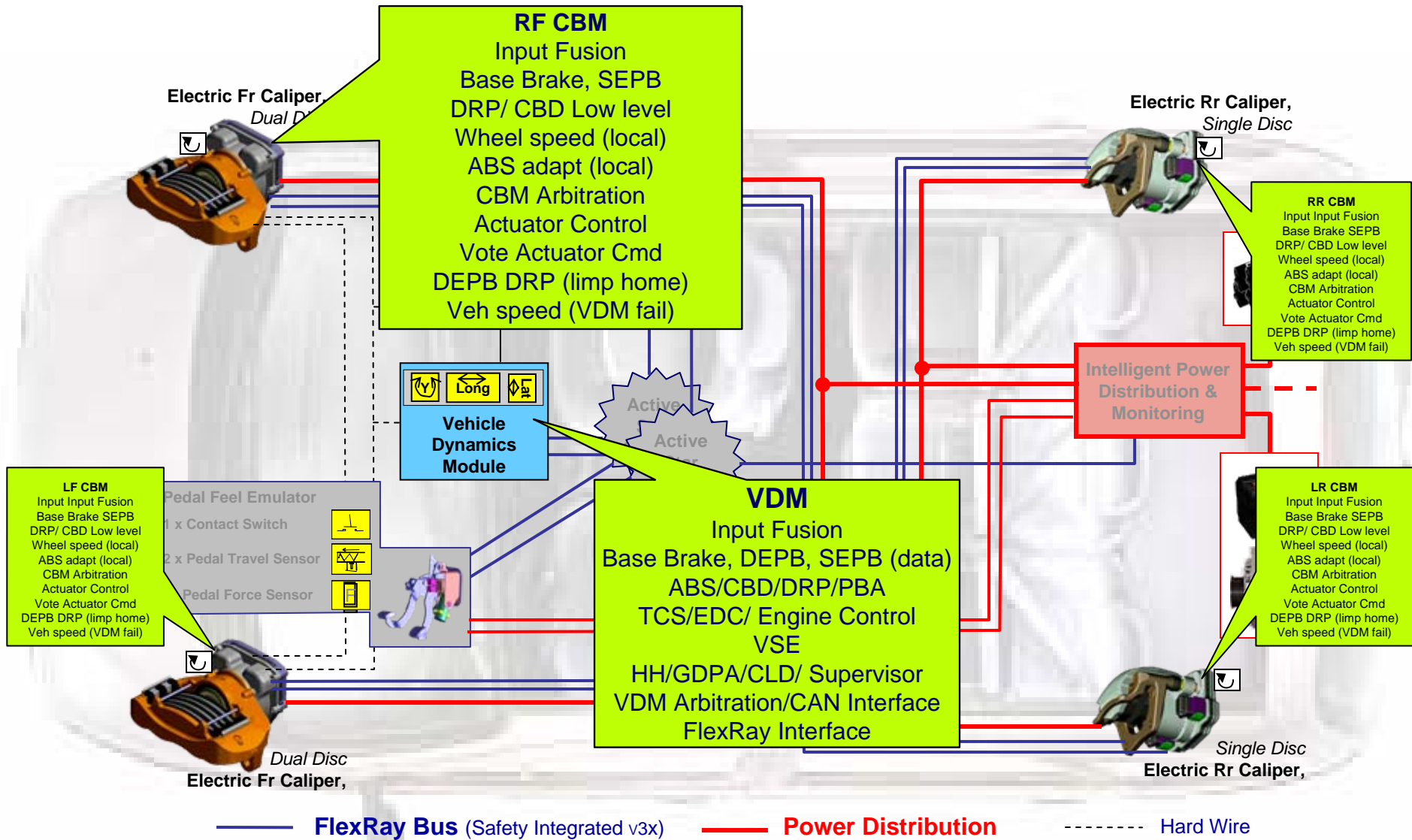
# Delphi Brake-By-Wire FlexRay Architecture: Central VDM



# Delphi Brake-By-Wire FlexRay Architecture: Calipers

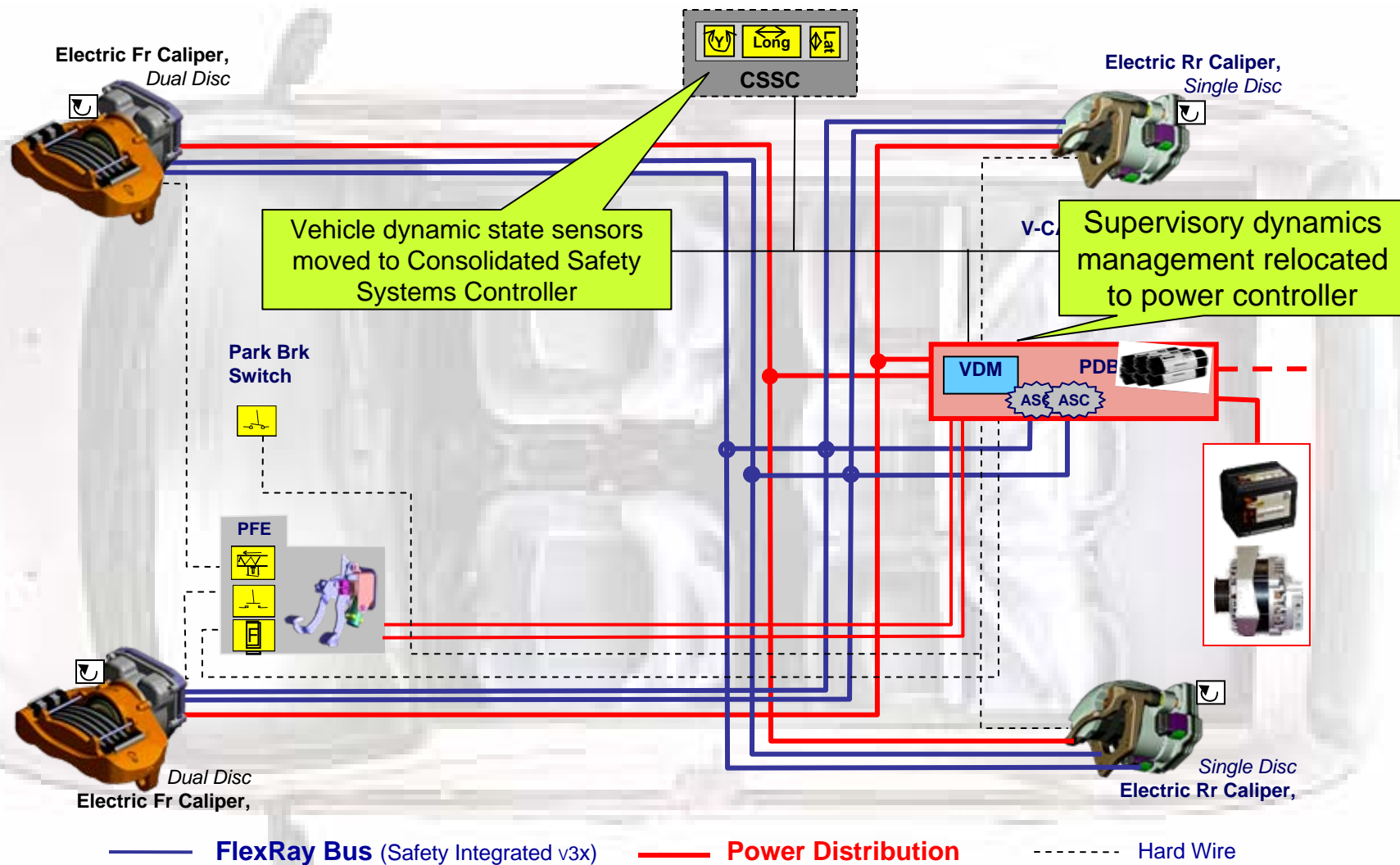


# Delphi Brake-By-Wire FlexRay Architecture: Distributed Functions



# Delphi Brake-By-Wire FlexRay Architecture Final Evolution

- Projected production state: 4 very smart actuators and 1 multifunctional ECU



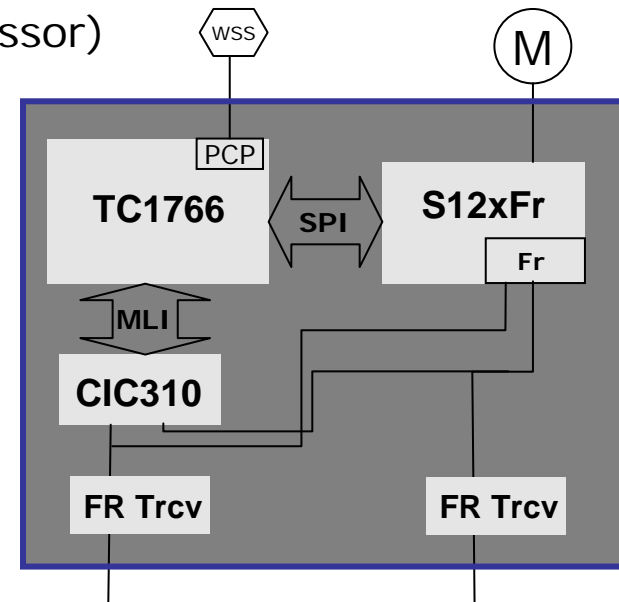
# Bus Requirements for a Distributed Brake Architecture

- ❑ Must allow complete synchronisation of all system ECUs
  - ❑ Fundamental to the distributed computing and RPV safety concept
- ❑ Must have high bandwidth to permit exchange of computed action/reaction data between Corner Brake Modules
- ❑ Must be capable of inherent consistency checking.
- ❑ Must be inherently robust to EM interactions
- ❑ Must allow independent and fail silent dual operation
- ❑ Must be topologically flexible with minimum subsidiary development
- ❑ Must utilise affordable and readily available implementation hardware

# Corner Brake Module (CBM - Caliper Controller) Architecture

Dual micros: Infineon TriCore 1766 + Freescale Star 12

- ❑ Selected to ease re-use of validated legacy code and not for “electronic efficiency”
- ❑ TC1766 usage: (in combination with CIC310 Flexray chip)
  - ❑ Mid-level brake application software: Base Brake, ABS (local)... expressing force demands
  - ❑ Local Wheel speed processing (processed by the TC Peripheral Control Processor)
  - ❑ Functional FlexRay frames processing
- ❑ Star 12xFR usage
  - ❑ Motor control (very low level) – legacy code
  - ❑ Actuator control (low level)
  - ❑ Debug FlexRay frames processing
  - ❑ Bus guardian
- ❑ SPI link between the two  $\mu$ C...
  - ❑ ... with the synchronization constraints it implies...



# Implementation of FlexRay Stack and Brake Software

- ❑ Before Vector FlexRay driver:
  - ❑ Prototype driver developed by Delphi on a TC1766 EVB
  - ❑ Selection of the FR Stack supplier, based on Delphi specification
  - ❑ Vector driver delivery:
    - ❑ for TC1766 + external FR controller CIC310
    - ❑ for Star 12 (incl. internal FR controller)
  
- ❑ “Basic” schedule implementation:
  - ❑ Setup of FlexRay communication with Vector Stack
  - ❑ Basic Base Brake integration

# Implementation of FlexRay Stack and Brake Software

- ❑ “Complete” schedule implementation
  - ❑ Increase of FlexRay frames numbers...
    - ❑ due to actuator redundancy: each node computes what all other nodes compute for their own local control needs
    - ❑ this redundancy strategy already existed with TTPC
  - ❑ ... leading to increased timing constraints, making the DMA (vs regular MLI) integration necessary
    - ➔ automatic transfer (of RX frames only)
  
- ❑ “Improved” schedule implementation
  - ❑ System lifecycle states management
    - ❑ together with the **V**ehicle **D**ynamics **M**odule, and the **P**ower **D**istribution **B**ox, both connected to the FlexRay bus
  - ❑ FlexRay communication errors management
    - ❑ Errors provided by the FlexRay Stack, and processed by the Delphi software

- ❑ System integration
  - ❑ The distributed FlexRay architecture was successfully tested on HIL
  - ❑ The system was implemented and tested on an A6 development vehicle
    - ❑ Extensive development of synchronised but local ABS and closed loop wheel deceleration was conducted with greatly improved stopping distance and comfort!
    - ❑ Excellent reliability and robustness of Flexray network.
- ❑ New distributed FlexRay project ongoing:  
Distributed MagneRide™ controlled suspension system
  - ❑ **Brake By Wire** specifics:
    - ❑ 2 channels (redundancy)
    - ❑ Star coupler
    - ❑ Safety time critical
    - ❑ Data integrity
  - ❑ Distributed **MagneRide™** specifics:
    - ❑ 1 channel
    - ❑ High bandwidth
    - ❑ Bus mode

# Cooperation with Vector

- ❑ FlexRay Stack initially provided by Vector
- ❑ Delphi asked for some improvements:
  - ❑ Access to FlexRay communication errors:
    - ❑ an interface was provided by Vector
    - ❑ at first, non AUTOSAR
  - ❑ DMA support:
    - ❑ Regular MLI was leading to a high CPU consumption
    - ❑ DMA support was then provided by Vector
  - ❑ Usage of Star12 XGate:
    - ❑ High Star12 CPU throughput due to Motor control timing constraints
    - ❑ Attempt to use XGate for FlexRay frames encoding and decoding  
FlexRay on XGate support then provided by Vector

Thank you for your attention.

For detailed information about Delphi  
and our products please refer to:  
[www.delphi.com](http://www.delphi.com)

Author:

Arnaud Balmitgere

Delphi France SAS

64 avenue de la Plaine de France

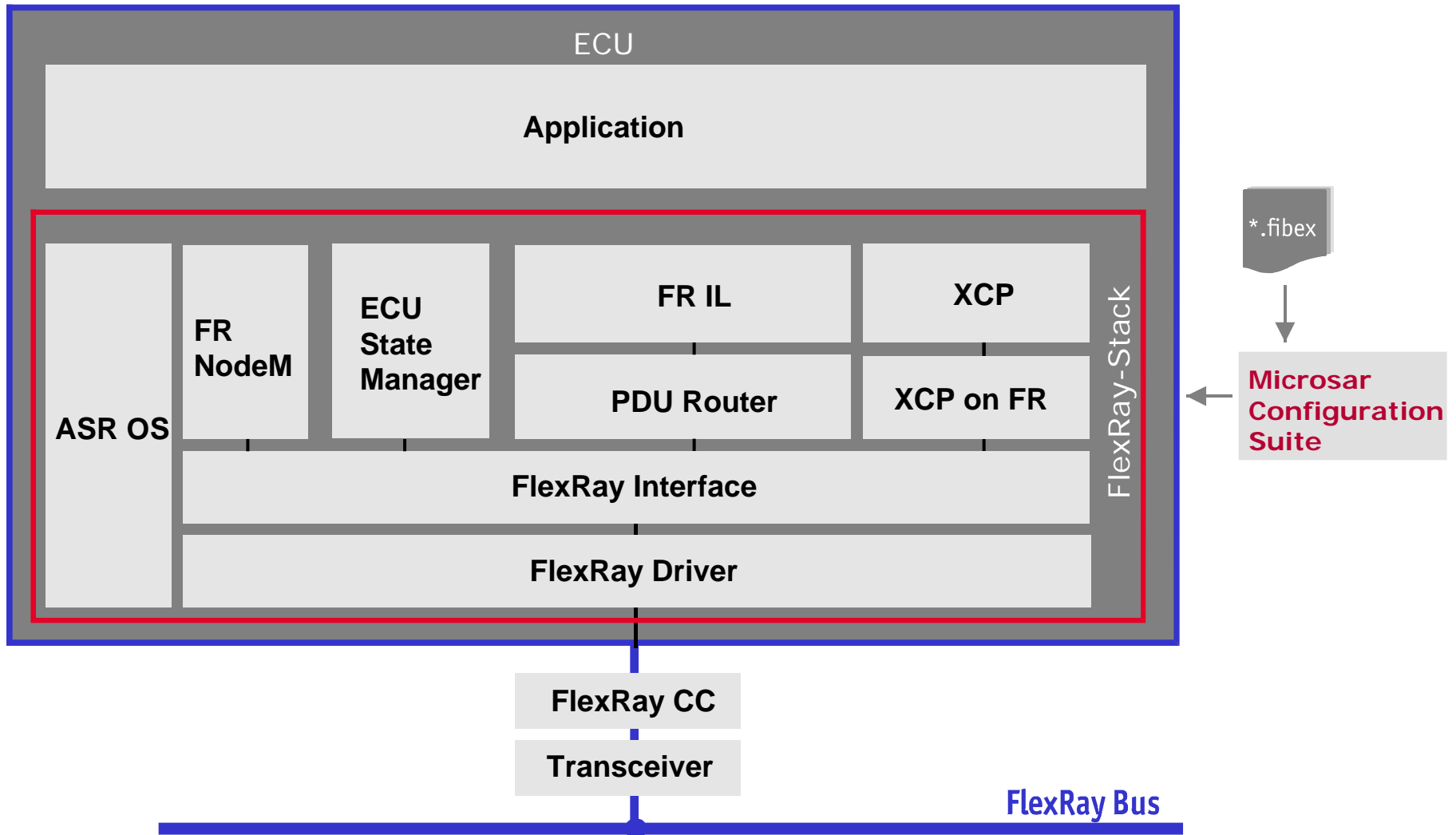
93290 Tremblay en France

- ❑ Customer Request
  - ❑ FlexRay software with
    - ❑ No, very little interrupt blocking time
    - ❑ Minimal CPU load
    - ❑ Support of XCP for measurement and calibration
    - ❑ Configuration via Fibex
  - ❑ Synchronization of tasks to FlexRay
  - ❑ Inter-ECU synchronization of tasks (accuracy 50µs)
  
- ❑ Hardware
  - ❑ “Dual Core” ECU
  - ❑ S12FR128, internal FlexRay CC
  - ❑ Infineon TriCore1766, external CIC310 FlexRay CC via MLI

- ❑ Vector approach
  - ❑ AUTOSAR BSW 2.1, incl. XCP (see BMW X5)
  - ❑ AUTOSAR OS
    - ❑ HighResolutionTimer
    - ❑ ScheduleTables
  - ❑ Project specific integration work
    - ❑ XGATE on S12X
    - ❑ DMA Controller on TC1766

# Solution

## FlexRay BSW



- ❑ AUTOSAR OS:
  - ❑ ScheduleTables
  - ❑ High Resolution Timer
  
- ❑ FlexRay Driver:
  - ❑ Initialize the FR Controller
  - ❑ Send and Receive Frames
  - ❑ Detection of FR Controller Errors
  
- ❑ FlexRay Interface
  - ❑ Assemble PDUs to Frames and vice versa
  - ❑ Handling of PDU Update Bits
  - ❑ Indication/Confirmation

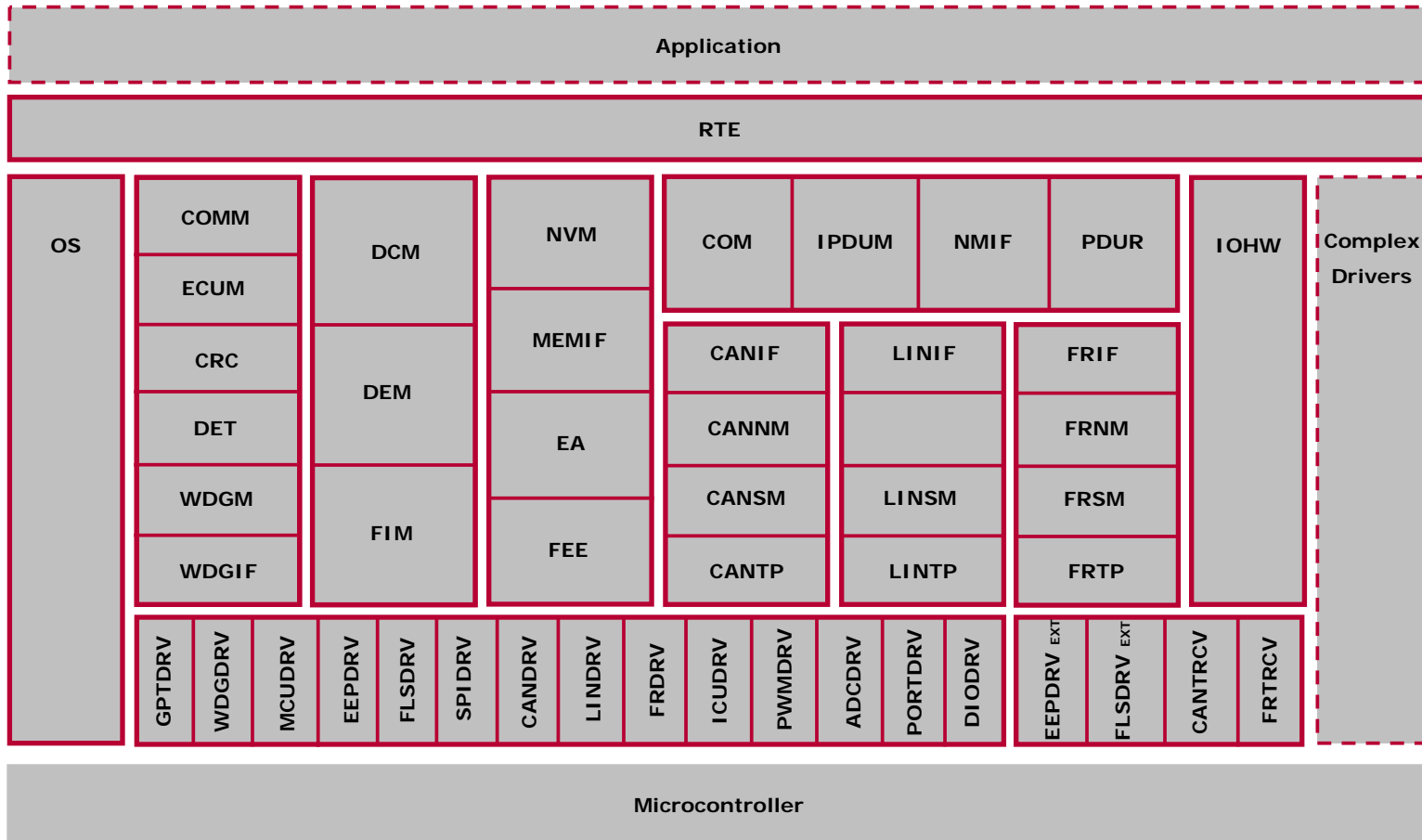
- ❑ PDU Router
  - ❑ Dispatching PDUs to COM and DCM
  - ❑ Might contain PDU-Gateway
  
- ❑ FlexRay Interaction Layer
  - ❑ Extraction of signals out of PDUs
  
- ❑ ECU State Manager:
  - ❑ Handling of ECU States (OFF, RUN, SLEEP, STARTUP/ SHUTDOWN)
  - ❑ Initializing of OS and all BSW modules
  
- ❑ FlexRay Node Manager:
  - ❑ Startup handling for FlexRay

- ❑ FlexRay software
  - ❑ Execution on Task Level
  - ❑ XGATE no help in case internal FR-CC and SystemRam is used
  - ❑ DMA big help for MLI
- ❑ Task synchronization
  - ❑ Achieved accuracy 5 $\mu$ s

# Future Prospects

## MICROSAR – Basic Software for AUTOSAR 3.0

Full range of basic software modules offered by Vector

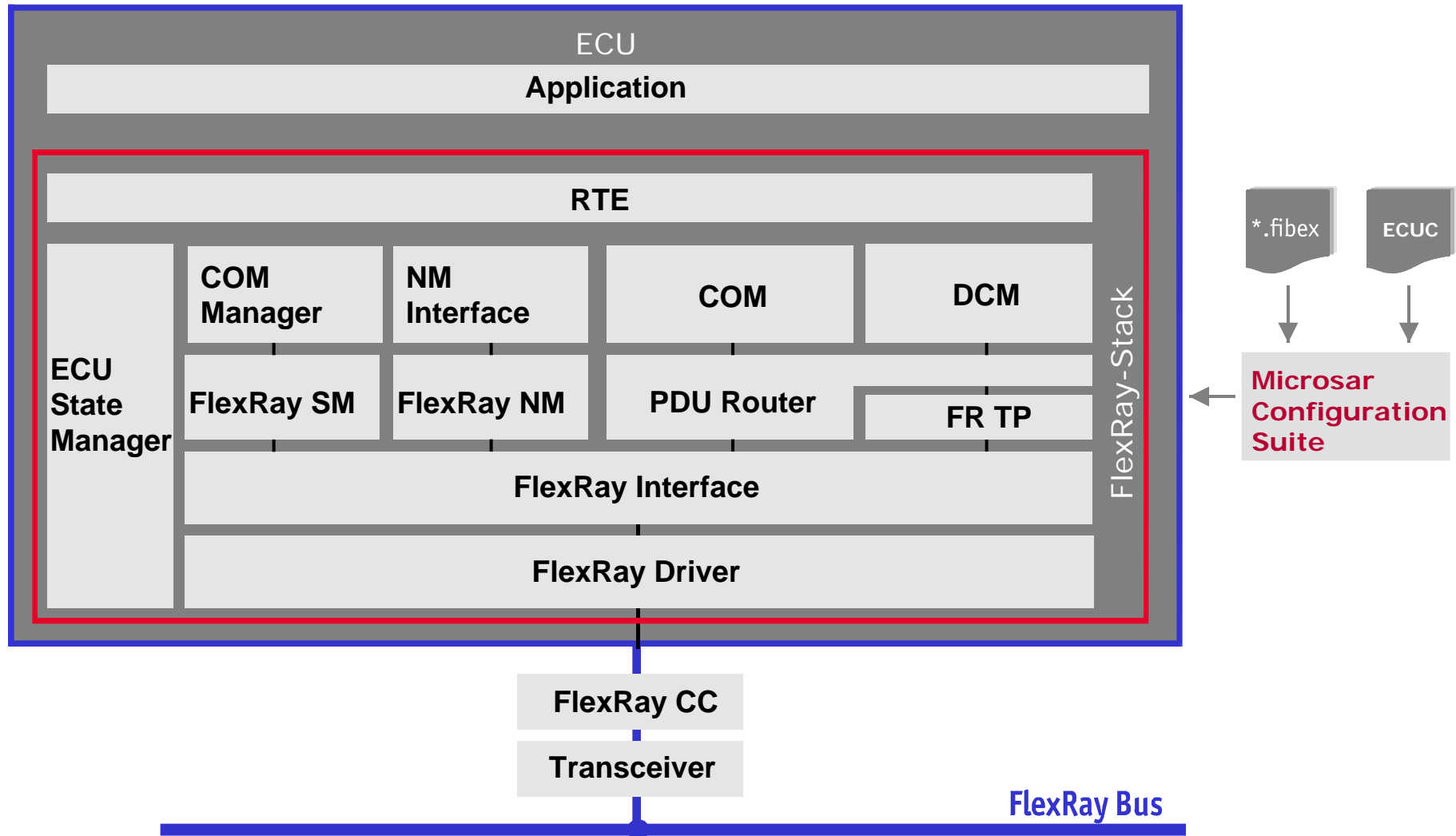


 Vector MICROSAR product

 Service by Vector

# Future Prospects

## Overview



Thank you for your attention.

For detailed information about Vector  
and our products please have a look at:

[www.vector-informatik.com](http://www.vector-informatik.com)

Author:

Dirk Großmann

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart