



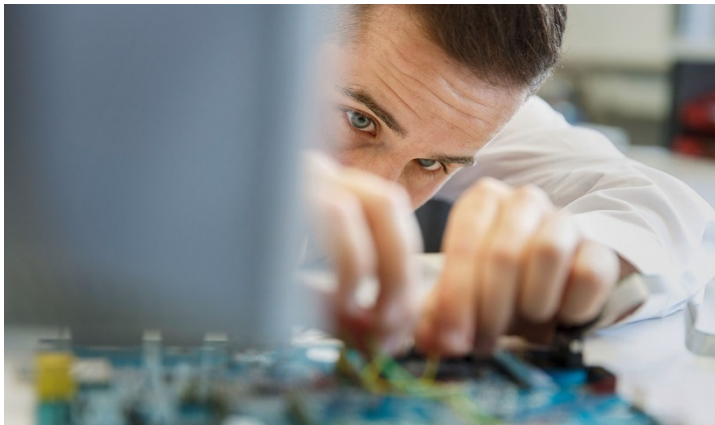
Einstiegsmöglichkeiten

Virtuelle Exkursion des Instituts für Nachrichtentechnik 2021



Aktuelle Karrieremöglichkeiten

- ▶ Werkstudenten (12 offene Stellen)
 - ▶ Praktikum (29)
 - ▶ Abschlussarbeiten (13)
 - ▶ Berufsbegleitendes Masterprogramm
 - ▶ Traineeprogramm Top-Absolventen (4)
 - ▶ Einstiegsprogramm für Naturwissenschaftler
 - ▶ Direkteinstig für Absolventen (26)
- ▶ Details → **jobs.vector.com**





Wen wir suchen

Gesuchte Studiengänge

- ▶ Hauptsächlich Fachrichtung Informatik, Elektrotechnik, Softwaretechnik, Informationstechnik und vergleichbare Studiengänge

Gesuchte Qualifikationen

- ▶ Kenntnisse und Erfahrungen in C / C++ / C# / Java
- ▶ Windows- und / oder Mikrocontroller-Plattformen

Aufgabengebiete

- ▶ Softwareentwicklung Windows / objektorientiert und hardwarenah
- ▶ Software- und Systemarchitektur
- ▶ Hardwareentwicklung
- ▶ User Interface Design, Produktmanagement, Projektmanagement IT / Automotive, Prozess- und Qualitätsmanagement, Technischer Vertrieb und Support ...



Was wir bieten

► **Super Arbeitskultur**

- > Ausgeprägter Teamgeist und Kollegialität, Verantwortung & Anerkennung, Kultur der offenen Türe, flache Hierarchien, Duz-Kultur

► **Flexibel denken, flexibel arbeiten**

- > Flexible Arbeitszeiten, Teilzeitmodelle, mobiles Arbeiten und Sabbatical, Kinderbetreuungsangebote, Vorruhestandsregelung

► **vGrow – you grow!**

- > Individuelle Weiterbildung und Entwicklungsmöglichkeiten

► **Kommunikation – das A und O**

- > Firmenversammlung, jährliche Mitarbeitergespräche, regelmäßige Team- und Abteilungsmeetings

► **Dein perfekter Einstieg**

- > Umzugskostenerstattung, Einführungsveranstaltungen, Mentoring-Programm, Schulungen, Drei- und Sechsmonatsgespräch



4,4 ★★★★★

kununu Score
347 Bewertungen



Was wir auch noch bieten

- ▶ **Bleib gesund**
 - > Höhenverstellbare Tische, Sportgruppen, Fitnessstudio vMove, Bewegungs- und Entspannungskurse, Rückensprechstunde
- ▶ **Spaß an der Arbeit**
 - > Zweitägiger Betriebsausflug, Sommerfest, Sommerparty, Weihnachtsfeier, Konzerte, Wochenausklang
- ▶ **Energieschub gefällig?**
 - > Exzellente Kantine, Barista Bar, freie Getränke
- ▶ **Du fährst – Vector unterstützt**
 - > Kostenfreie Parkplätze und Ladesäulen, Jobticket ÖPNV, JobRad, Fahrradkeller, Spinde, Einzelduschen, JobAuto
- ▶ **Angemessene Vergütung**
 - > Erfolgsprämie, Weihnachts- und Urlaubsgeld, betriebliche Altersvorsorge



Zeit für Deine Fragen!

Autor: Holger Heit (holger.heit@vector.com)

Vector – Automotive. Embedded. Engineering.



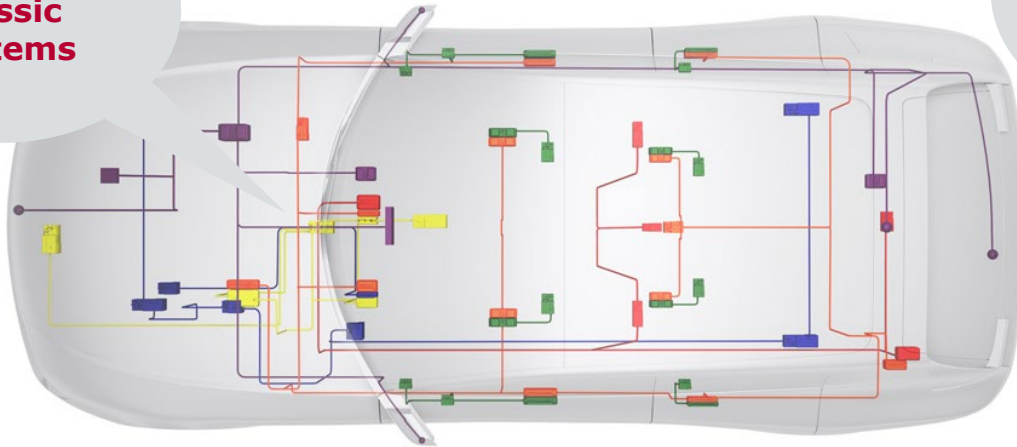
Smartphone auf Rädern

Herausforderungen in zukünftigen Fahrzeugarchitekturen

Evolution of Vehicle Architecture

Distributed ECU Architecture

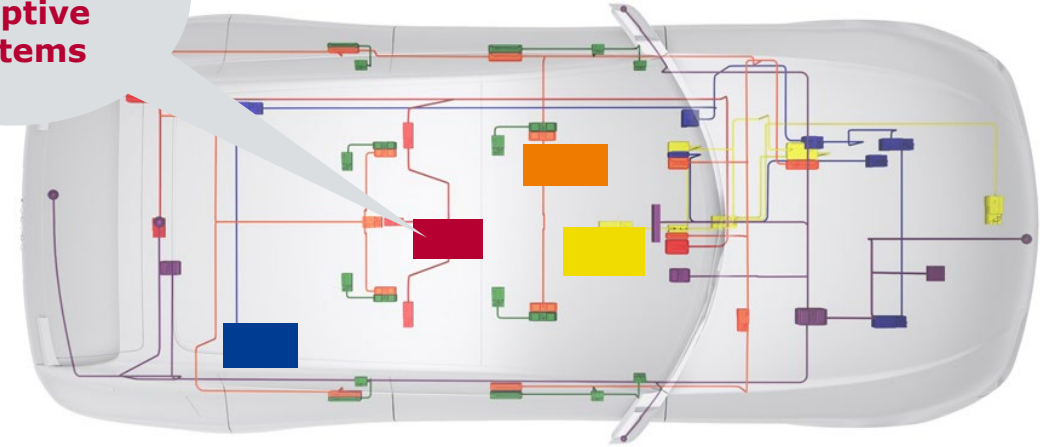
**AUTOSAR
Classic
Systems**



- ▶ ECUs implement dedicated functions
- ▶ One supplier per ECU
- ▶ Limited amount of data shared between ECUs

Domain Controller Architecture

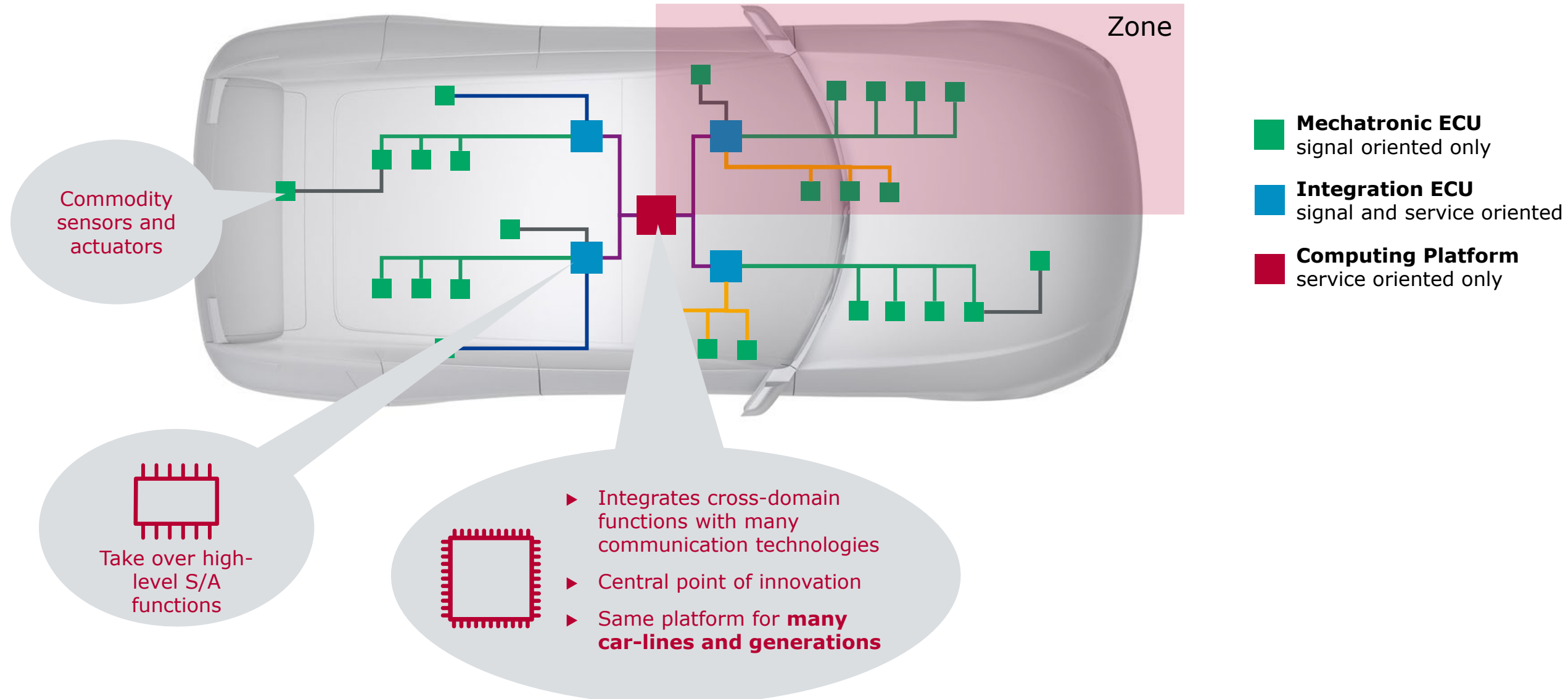
**Stand alone
AUTOSAR
Adaptive
Systems**



- ▶ Functions integrated per domain
- ▶ Multiple application software supplier per ECU
- ▶ High-level functionality of sensors and actuators already reduced and moved to Domain Controllers

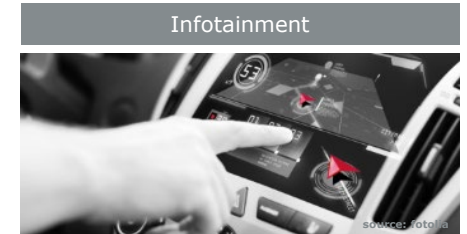
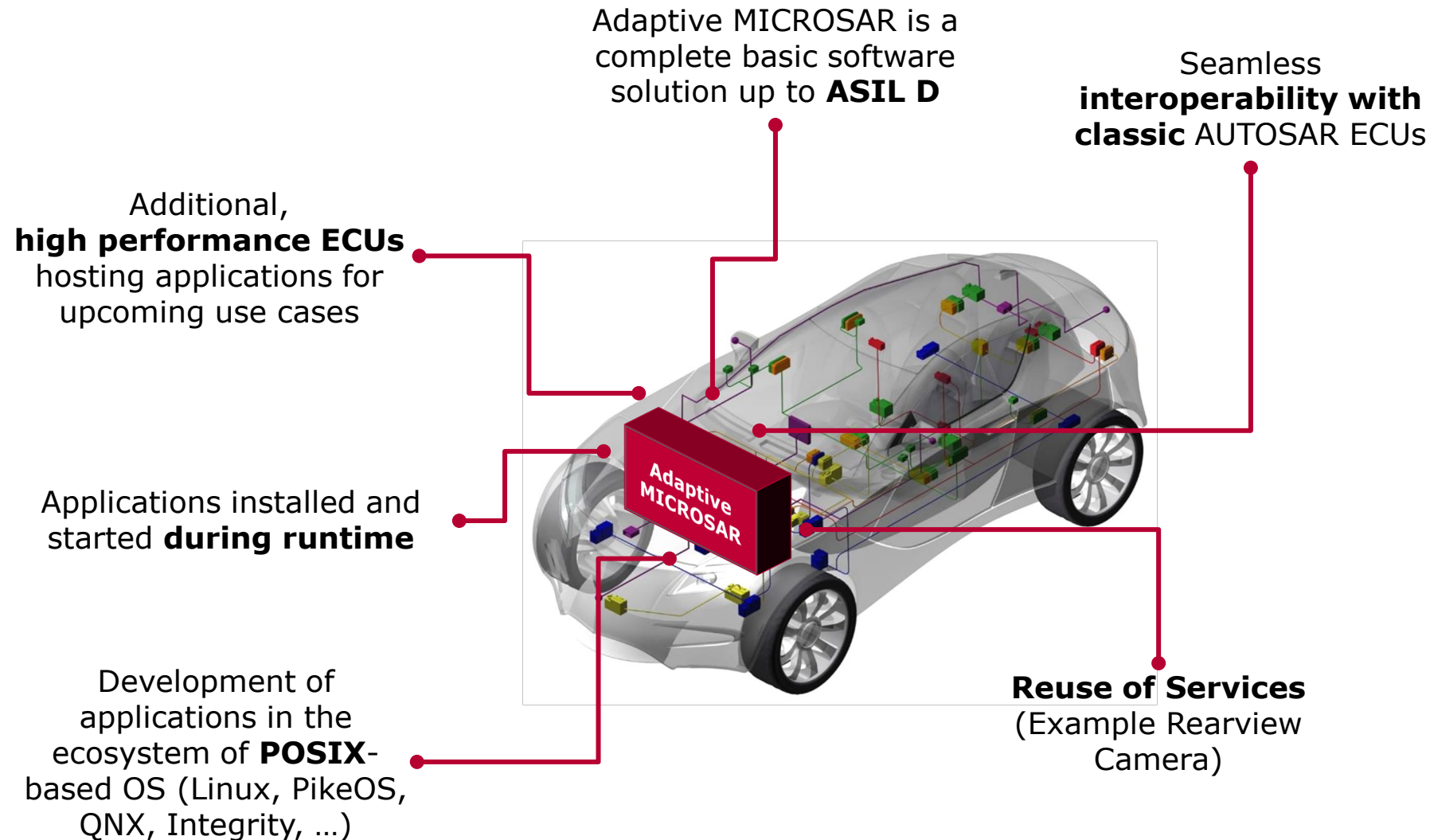
Will your next car be a “Smartphone on Wheels”?

Central Computing Platform (Zonal Architecture)



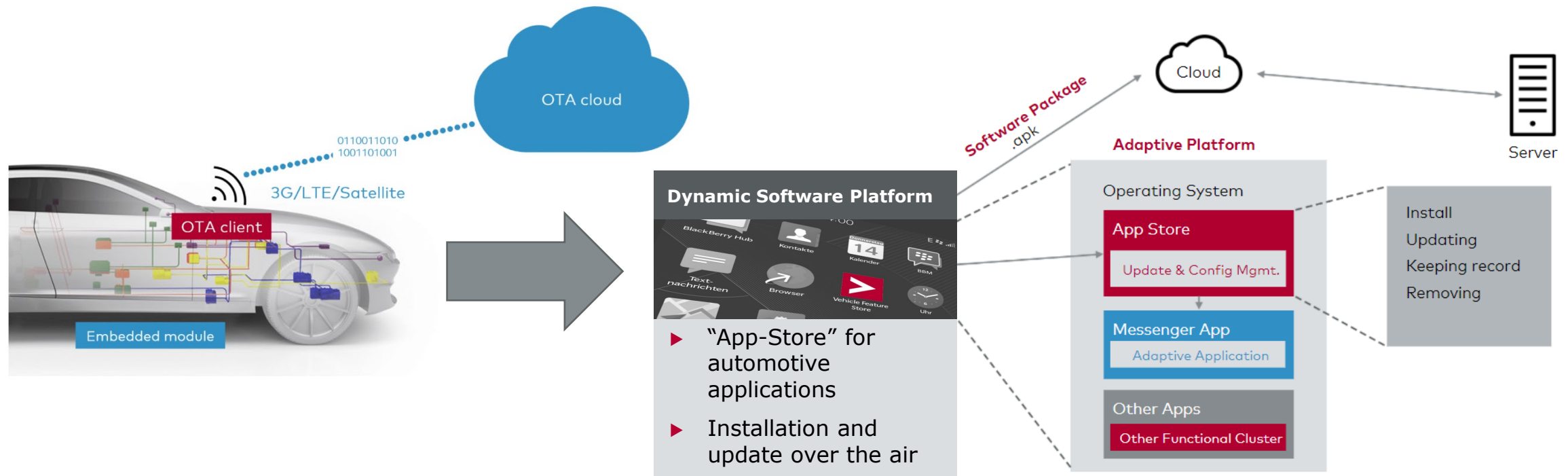
Will your next car be a “Smartphone on Wheels”?

Being Prepared for the Next-Generation of Basic Software



Update and Configuration Management (UCM)

- ▶ One of the goals of Adaptive MICROSAR is the ability to flexibly update the software and its configuration through over-the-air updates
 - ▶ The Update and Config Management (UCM) is responsible for updating, installing, removing and keeping a record of the software on an AUTOSAR Adaptive Platform
- ▶ Installing, uninstalling and updating applications will be just as easy as with a mobile phone





Find communication partner: Service Discovery

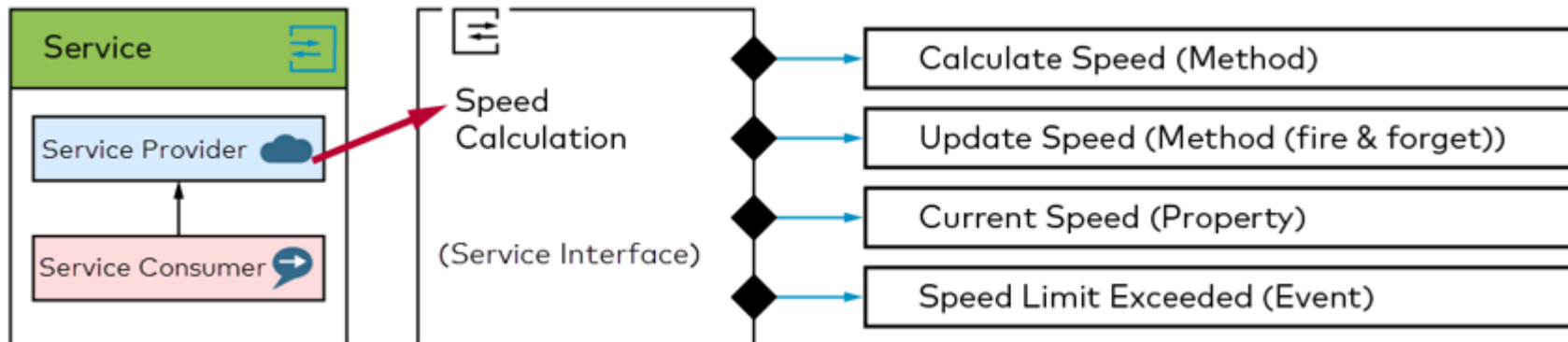


(1) Service Provider

(2) Service Consumer(s)

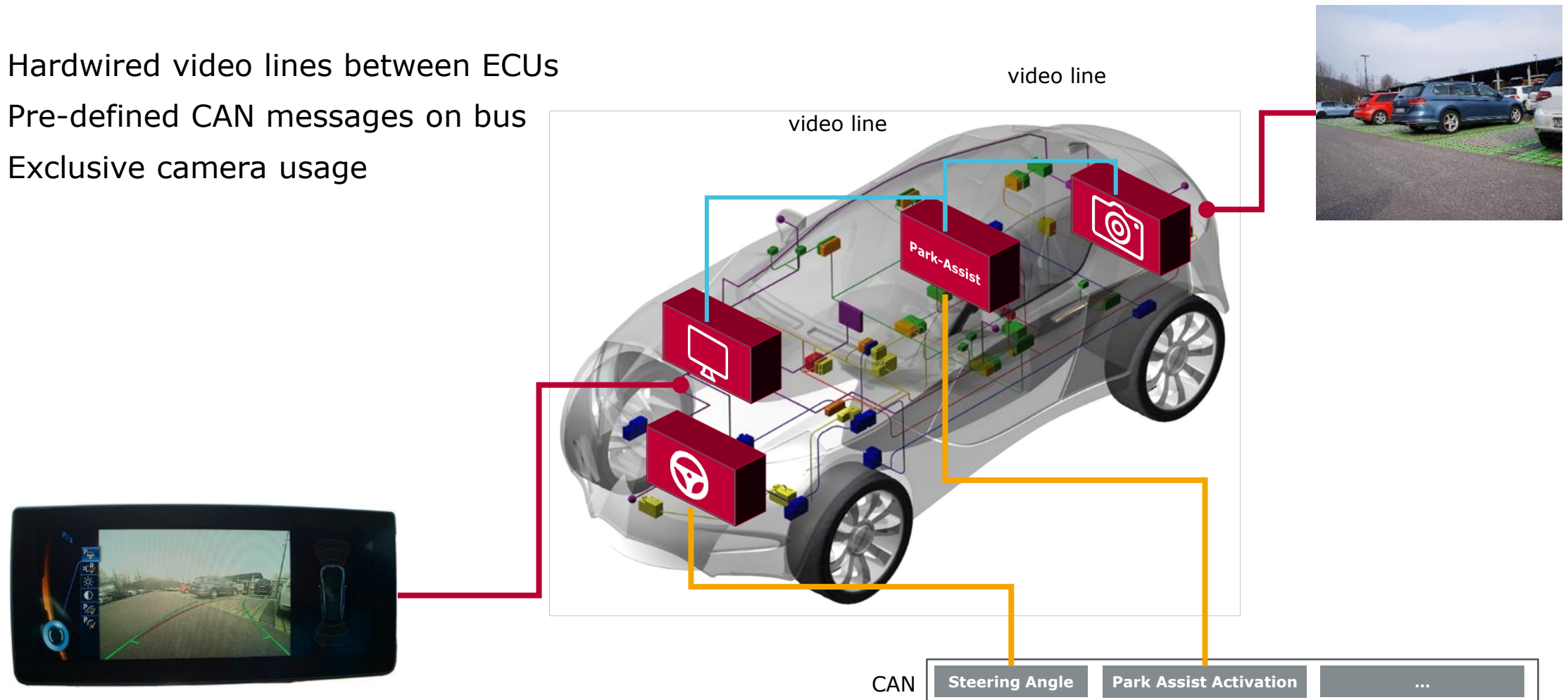
→ Service provider and service consumer need to find each other at run time

► Communication must take place via a well-defined and specified service interface:



Fixed Design: Statically connected HW Ressources

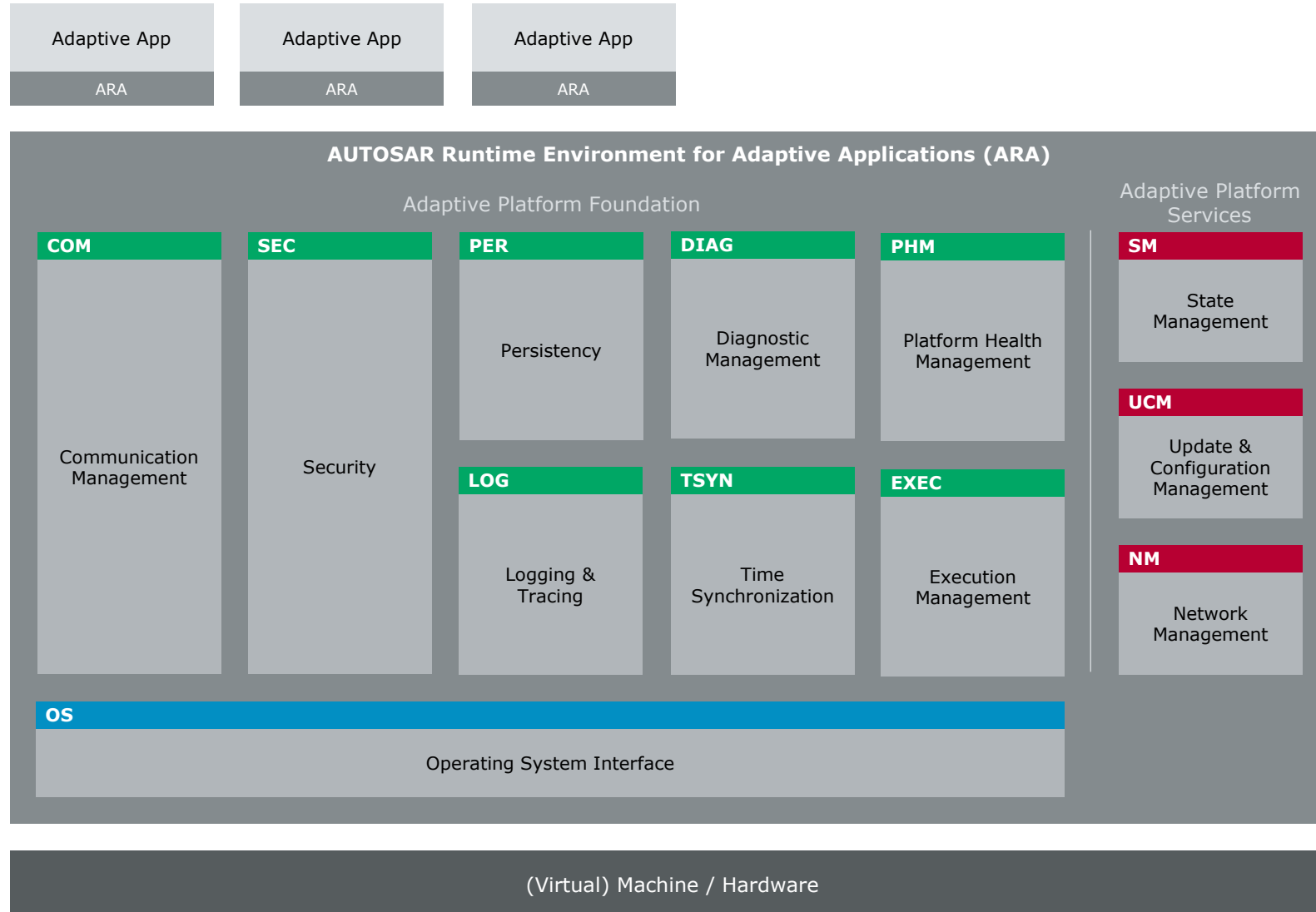
- ▶ Hardwired video lines between ECUs
- ▶ Pre-defined CAN messages on bus
- ▶ Exclusive camera usage



- ▶ Smart sensors/actuators provide HW over service interface
- ▶ All ECUs connected via Ethernet
- ▶ Compound service, using base services as lower layer
- ▶ Applications can provide services for e.g. HMI integration
- ▶ No function oriented wiring



Adaptive MICROSAR – Platform Architecture Overview



Platform Support?

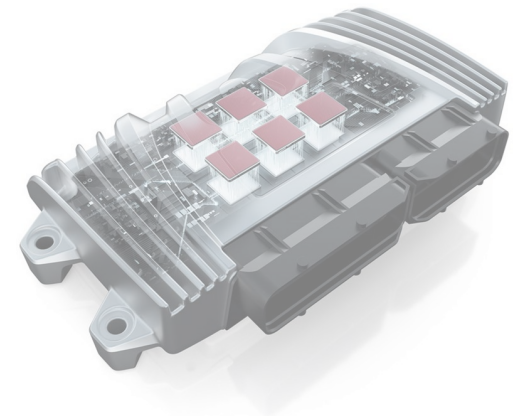
Wide range of supported Operating Systems

- ▶ MICROSAR Adaptive is designed to support all major OS on the market
- ▶ OS Abstraction Layer reduces effort to adapt to new OS
- ▶ Currently supported: QNX (Blackberry), PikeOs (SYSGO), Linux
- ▶ Prototypes: VxWorks (Wind River), Integrity (Green Hills), Android

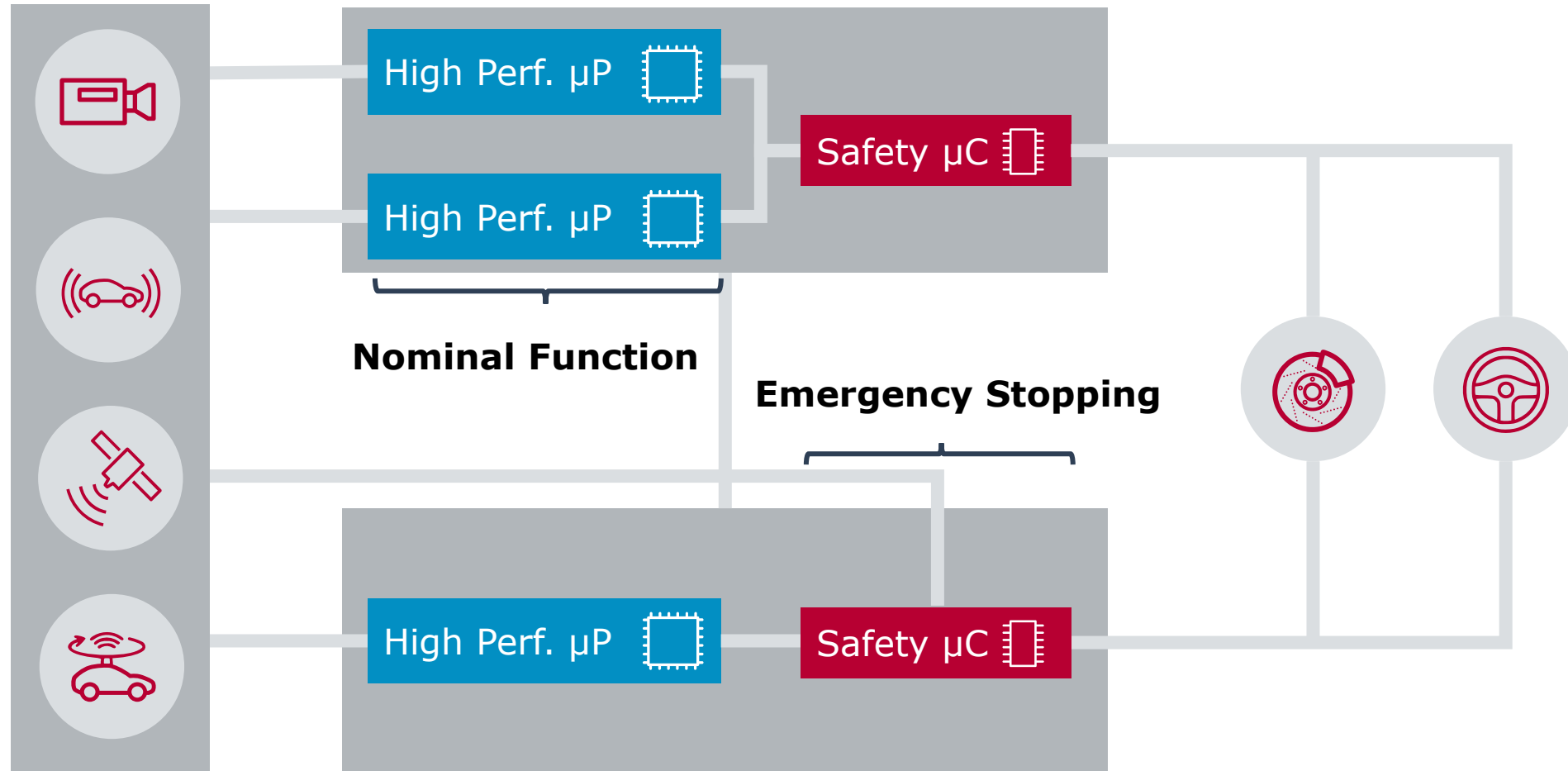


Hardware independent

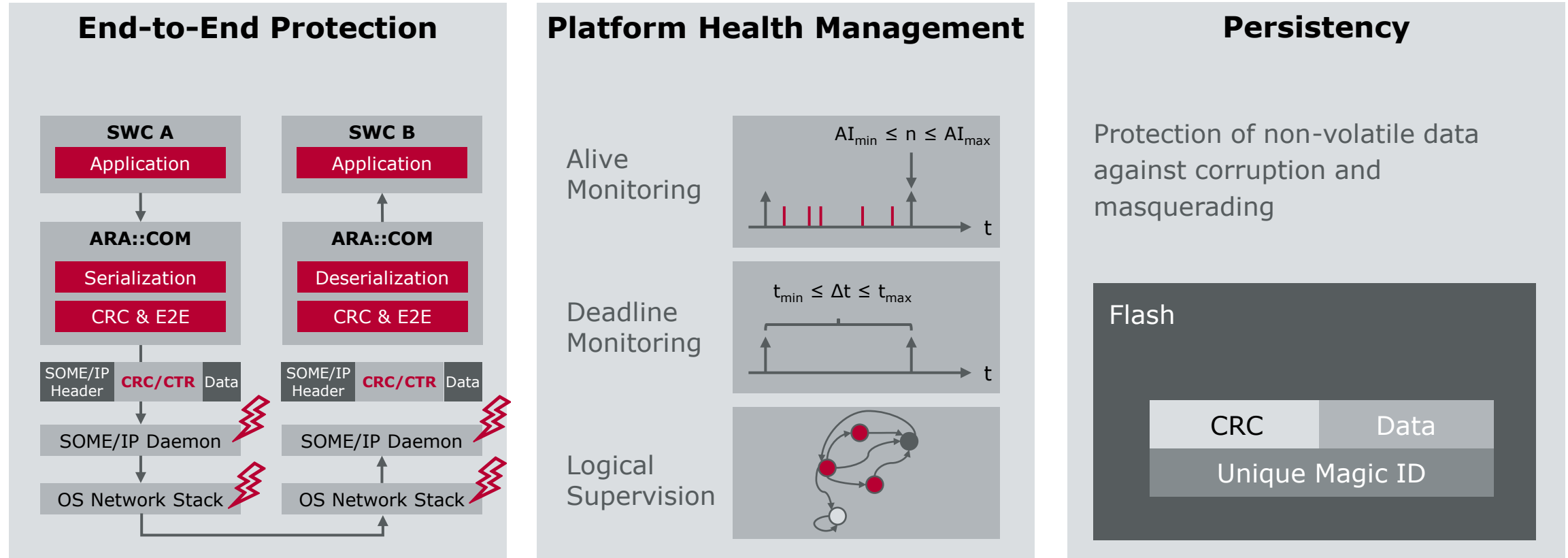
- ▶ OS provides HW abstraction for many devices
- ▶ Where required we discuss driver design with silicon vendors
 - ▶ E.g.: HSM, TSYN, IPC on SoC
- ▶ Exemplary silicones:
 - ▶ Renesas (R-Car)
 - ▶ Qualcomm (Snapdragon A)
 - ▶ Nvidia (Xavier)
 - ▶ Samsung (Exynos)



Typical Automated Driving Safety Architecture (SAE Level ≤ 3)



Safety Mechanisms in Adaptive AUTOSAR



► Provides safety mechanisms to support safety-related applications

► Allows detections of certain faults in the Application and the OS

► **For fail-operational systems this detection is not sufficient**

- > **Timing + Memory determinism needed**
- > **No "normal" OS/Adaptive AUTOSAR usable.**

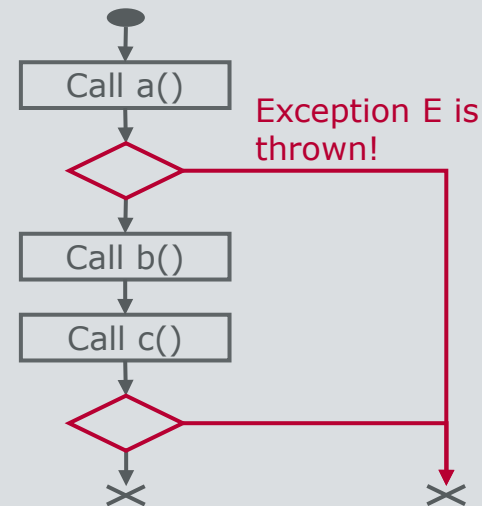
Challenges with C++

Dynamic Memory

- ▶ **Fail-Safe**
 - ▶ Double use of memory
 - ▶ Use after free()/Dangling Pointers
 - ▶ Double Free/Double Allocation
 - ▶ **Coding Guidelines**
 - ▶ **Static Code Analysis/Reviews**
- ▶ **Fail-Operational**
 - ▶ Out-of-memory conditions
 - ▶ Fragmentation
 - ▶ Bad performance
 - ▶ Non-deterministic runtime
 - ▶ **Use constant time memory allocators and pre-allocate memory from OS**
 - ▶ **Bound memory usage**

Exceptions

- ▶ ISO 26262-6:2018 Table 6
 - ▶ One entry and exit-point of functions
 - ▶ No dynamic objects
 - ▶ No hidden data or control flow



- ▶ **Don't use Exceptions, AUTOSAR provides exceptionless API variant**

Templates

- ▶ Templates can be used to write code once for different types.
- ▶ **Additional tool supported needed for testing the software**
- ▶ **Additional testing processes needed**
- ▶ **Additional care of developing software that eases testing**

```

template<typename T>
T calculate(T input) {
    T output = input / 255;
    return output;
}
    
```

Android Auto vs. Android Automotive



- ▶ Android Automotive (first vehicle is Volvo's Polestar 2 electric car)
 - ▶ Natively integrated into car
 - ▶ display apps, entertainment and messages on the car's dashboard

- ▶ Android Auto (available in > 500 vehicles of different brands)
 - ▶ needs a connected (tethered) smartphone
 - ▶ display apps, entertainment and messages on the car's dashboard

Will your next car be a “Smartphone on Wheels”?

Conclusion

Architecture of Automated Driving Systems

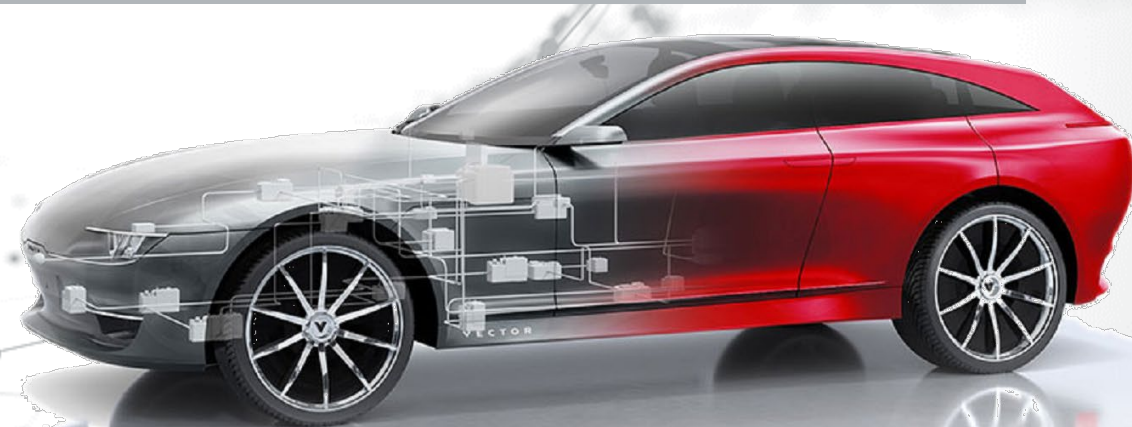
- ▶ Important where to locate microcontroller and microprocessors
- ▶ Fail-Operational requirements can be fulfilled using Classic AUTOSAR on microcontrollers right now
- ▶ Microprocessors can fulfill safety requirements in the fail-safe paths of the software using AUTOSAR Adaptive

Service Oriented Architecture

- ▶ Enabler for new service applications
 - > Service provider and service consumer need to find each other at run time
- ▶ Communication must take place via a well-defined and specified service interface

Design of high performance ECUs

- ▶ More components need to be safe: Standard C and C++ libraries, Inter Process Communication
- ▶ For systems > *SAE Level 3* additional requirements exist:
 - > OS developed for safety is needed to fulfill availability requirements
 - > Scheduling and communication needs to be correct
- ▶ Hypervisors ease the system partitioning to install safety related Applications



More Information!
Visit our Website for:

www.vector.com

Author:
Felix Pistorius
Vector Germany



Mein Einstieg in die Embedded-Software-Entwicklung

Andreas Kühefuß



Über mich

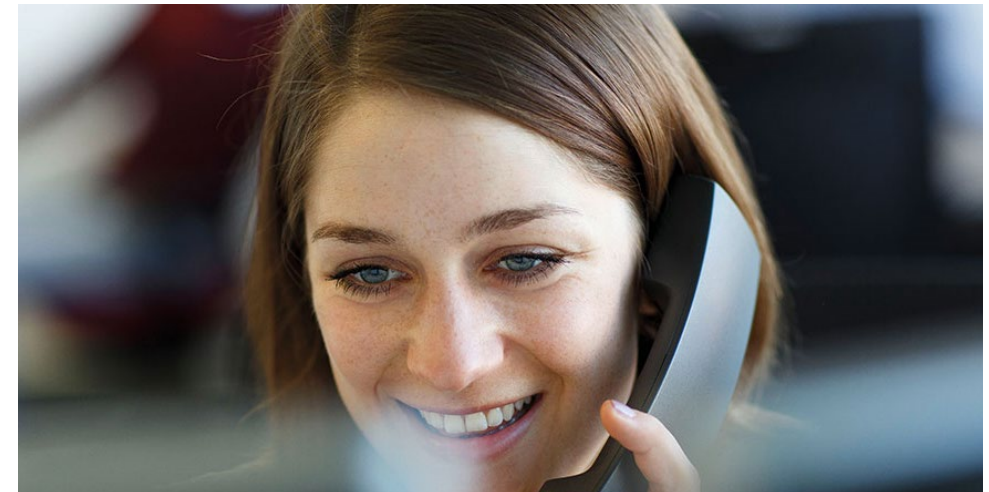
Andreas Kühefuß, geb. 13.12.1989

- ▶ 2013 – 2019 Elektrotechnikstudium an der Hochschule Karlsruhe
- ▶ Seit 01. März 2019 bei Vector
- ▶ 2015 – 2018 Deutschlandstipendium durch Vector Informatik



Warum zu Vector?

- ▶ Kennen gelernt durch das Deutschlandstipendium & Study TechNight
- ▶ Spannende Produkte
- ▶ Das Trainee-Programm
- ▶ Great Place to Work





Das Trainee-Programm

Zusammenfassung

- ▶ Vier verschiedene Trainee-Programme
- ▶ Fachliche Entwicklung
 - ▶ Einsatz in mehreren Funktionsbereichen in der gewählten Vertiefungsrichtung
- ▶ Entwicklung der persönlichen Kompetenzen
 - ▶ Bausteine zur Entwicklung von Fach-, Methoden- und Anwendungswissen
- ▶ Persönlich zugeschnitten
 - ▶ Flexibler Programmstart
 - ▶ Individuelle Funktionsbereiche
- ▶ Regelmäßiger Austausch mit anderen Trainees
 - ▶ Trainee Round Table
 - ▶ Kollegiale Fallberatung

VECTOR

Region: Deutschland [Ändern](#) | Deutsch English

```
01 using System;
02
03
04
05 public static class Jobs
06 {
07     public static void Main(string[] args)
08     {
09         Console.WriteLine("Mache einen der besten Arbeitgeber noch eine Idee besser.");
10         VectorJobs.ShowJobDetails("myJob");
11     }
12 }
13
14
15
```

Bewirb Dich initiativ!

Du bist hier: [Home](#) > [Jobs & Bewerbung](#) > [Jobangebote](#)

Ergebnisse (4)

Trainees (m/w/d) Softwareentwicklung Research & Development für innovative Softwaretools Top-Absolventen-Programm	Stuttgart TP-SW-RDI-2021
Trainees (m/w/d) Softwareentwicklung Embedded Top-Absolventen-Programm	Stuttgart oder Regensburg oder Karlsruhe TP-SWE-2021
Trainees (m/w/d) Softwareentwicklung Top-Absolventen-Programm	Stuttgart oder Karlsruhe TP-SW-2021
Trainees (m/w/d) Softwareentwicklung Research & Development Top-Absolventen-Programm	Stuttgart TP-SW-RDE-2021

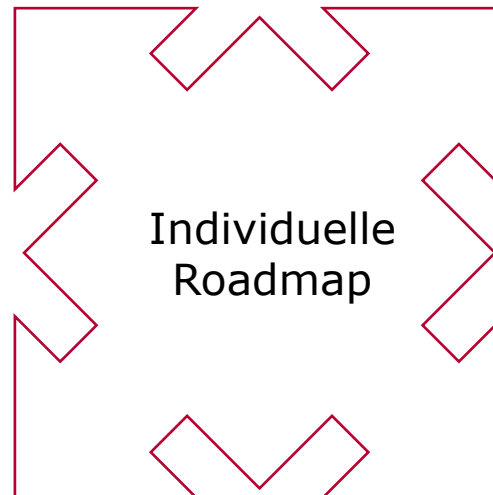
Das Trainee-Programm

Matching *jederzeit*

- ▶ 2-stufiger Auswahlprozess mit Interviews & Fallstudien
- ▶ Unbefristeter Arbeitsvertrag

Speed *je 4-8 Monate*

- ▶ 3 - 4 Rotationen in verschiedene Funktionsbereichen
- ▶ Rotationen werden durch Coaches aus Funktionsbereichen begleitet



Warm-up *ca. 6 Monate*

- ▶ Start in Stammabteilung
- ▶ Mentoring
- ▶ Mitwirken in der Produktentwicklung

Touchdown *nach ca. 18-24 Monate*

- ▶ Beendigung des Programms
- ▶ Einstieg in Zielposition innerhalb unserer Fach- und Führungslaufbahn

Meine Trainee-Stationen

▶ **Microsar Classic Entwicklung (10 Monate im Team: Basic Runtime Environment)**

- ▶ Komponenten Entwicklung mit der Programmiersprache C und embedded Compiler
- ▶ Startup verhalten (Assembly startup code, peripheral initialization, memory initialization)
- ▶ Unterstützung für jeden denkbaren automotive controller

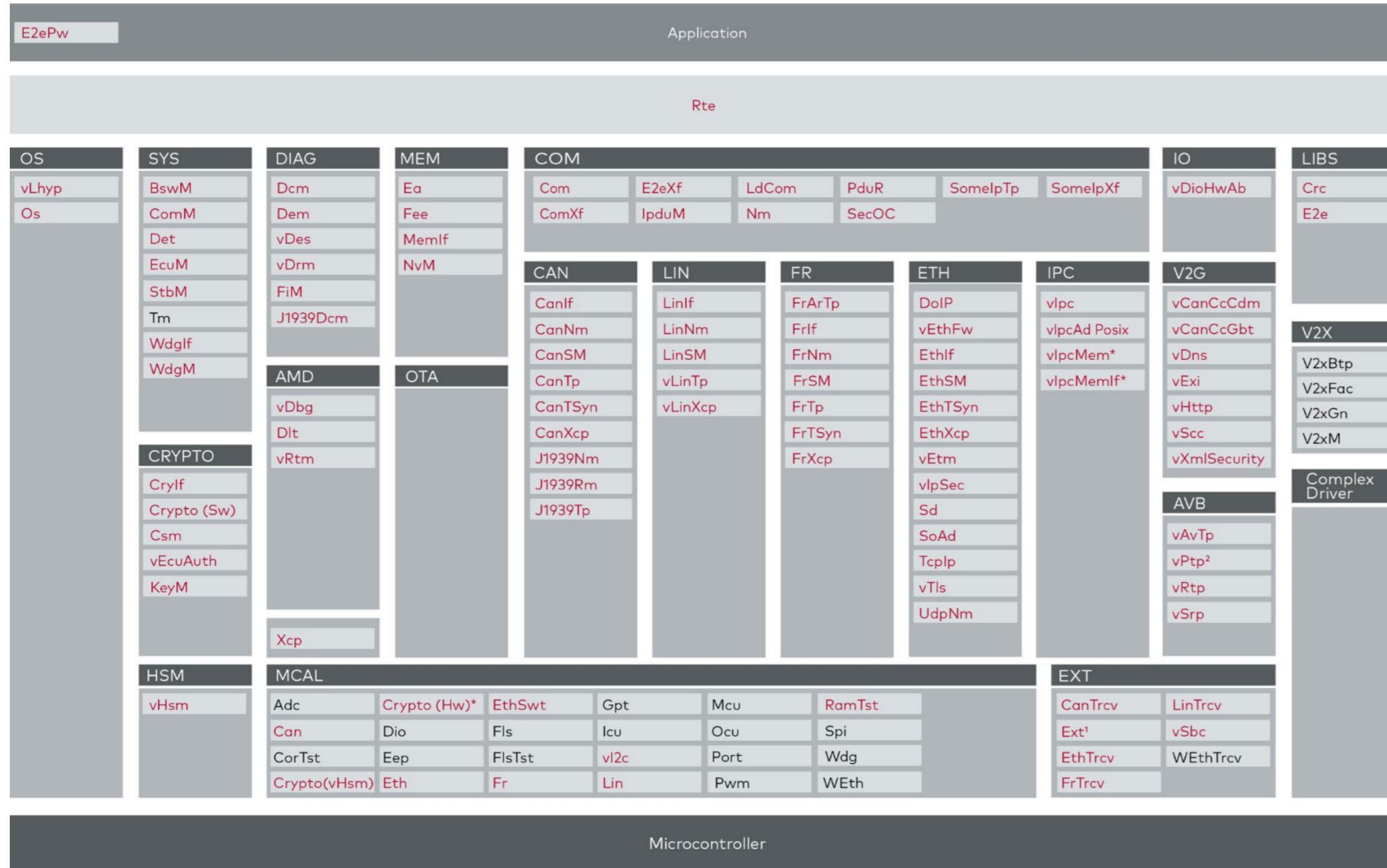
▶ **Microsar Classic Integration (6 Monate im Team: Deployment für Kundenprojekte)**

- ▶ Deployment mit DaVinci Configurator, der Programmiersprache C und embedded Compiler
- ▶ Kundenspezifische Projektkonfiguration, Integration von sämtlichen Vector Komponenten aus dem Autosar Stack

▶ **Microsar Adaptive Entwicklung (8 Monate im Team: Kommunikation Stack)**

- ▶ Komponenten Entwicklung mit der Programmiersprache C++ in der Unix Welt
- ▶ Entwicklung einer Service orientierten Kommunikation-Lösung

Meine Trainee-Stationen



Resümee des Trainee-Programmes

- ▶ Rotationen
 - ▶ Super Einarbeitung durch Mentoring
 - ▶ Spannende & unterschiedliche Tätigkeiten
 - ▶ Mitarbeit an Produktentwicklung
 - ▶ Angenehmes Arbeitsklima
 - ▶ Hilfsbereite Kollegen
- ▶ Trainee Programm
 - ▶ Round Table bieten Austausch zur Geschäftsführung und Management
 - ▶ Austausch mit anderen Trainees, z.B.: Kollegiale Fallberatung
- ▶ Social Life bei Vector
 - ▶ Herausragende Cantine
 - ▶ Vecathon
 - ▶ Wochenausklang
 - ▶ Sommerparty, Sommerfest, Weihnachtsfeier
 - ▶ Betriebsausflüge



Zeit für Fragen!

Autor:
Kühefuß, Andreas
Vector Informatik GmbH



Test und Simulation von V2X/ADAS-Systemen

Wie man das Kommunikationschaos mit CANoe beherrschen kann

Agenda

► **Warum Vector**

CANoe ADAS Feature Set

CANoe Car2x

Aufgaben im Arbeitsalltag



Warum Vector?



Warum Vector

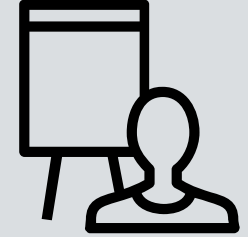
Versorgung

- ▶ CANtine
- ▶ Kostenlose Getränkeversorgung



Weiterbildungen

- ▶ Fachliche Schulungen
- ▶ Soft Skill Schulungen
- ▶ Personenbezogene Förderung



Arbeitsalltag

- ▶ Spannende Aufgaben
- ▶ Gutes Miteinander
- ▶ Abwechslungsreich
- ▶ Interne Wechsel werden gut gehandhabt
- ▶ Gleizeit



Work Life Balance

- ▶ Wochenausklang
- ▶ Events & Feiern
- ▶ Kostenloses Fitnessstudio mit Kursen
- ▶ Duz Kultur



Agenda

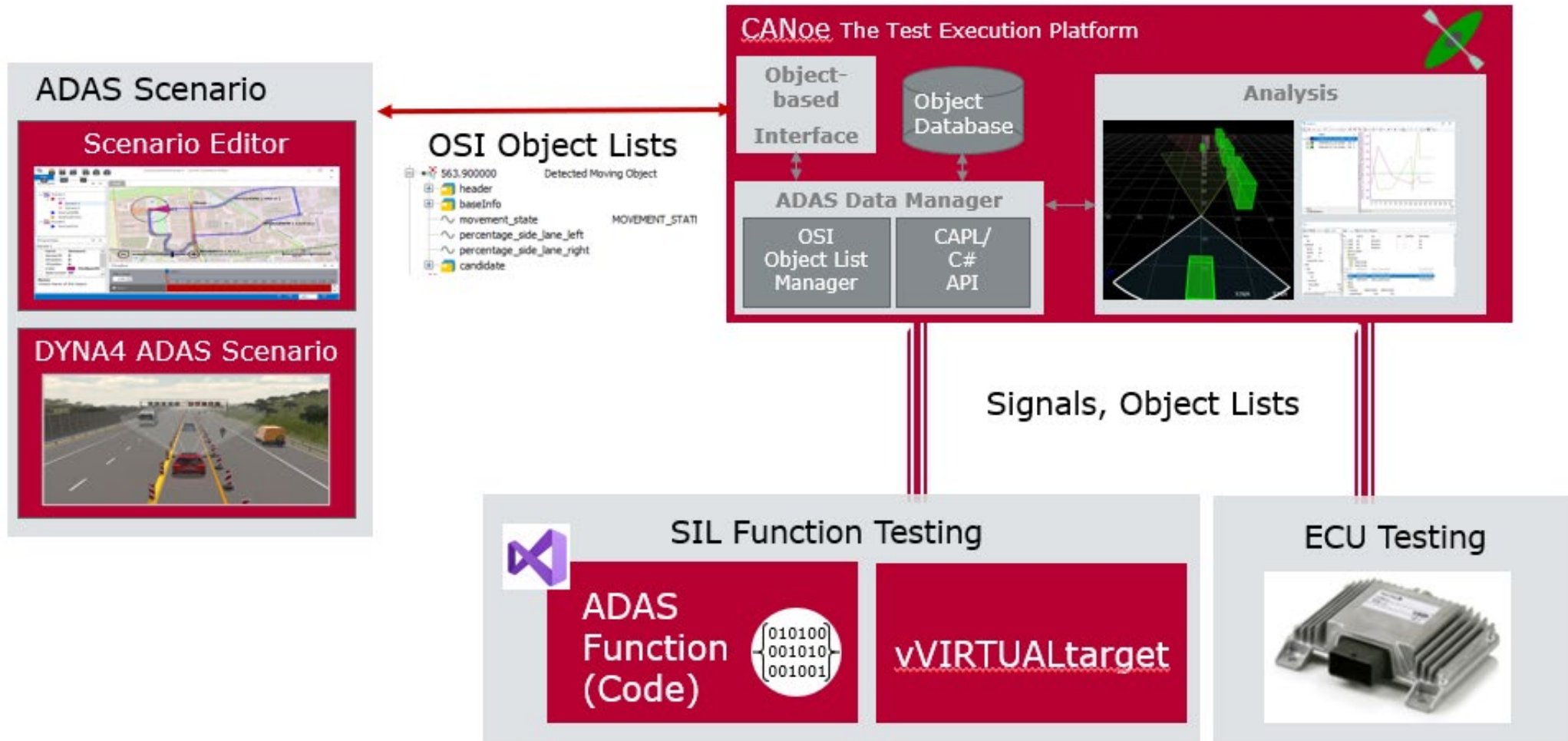
Warum Vector

► **CANoe ADAS Feature Set**

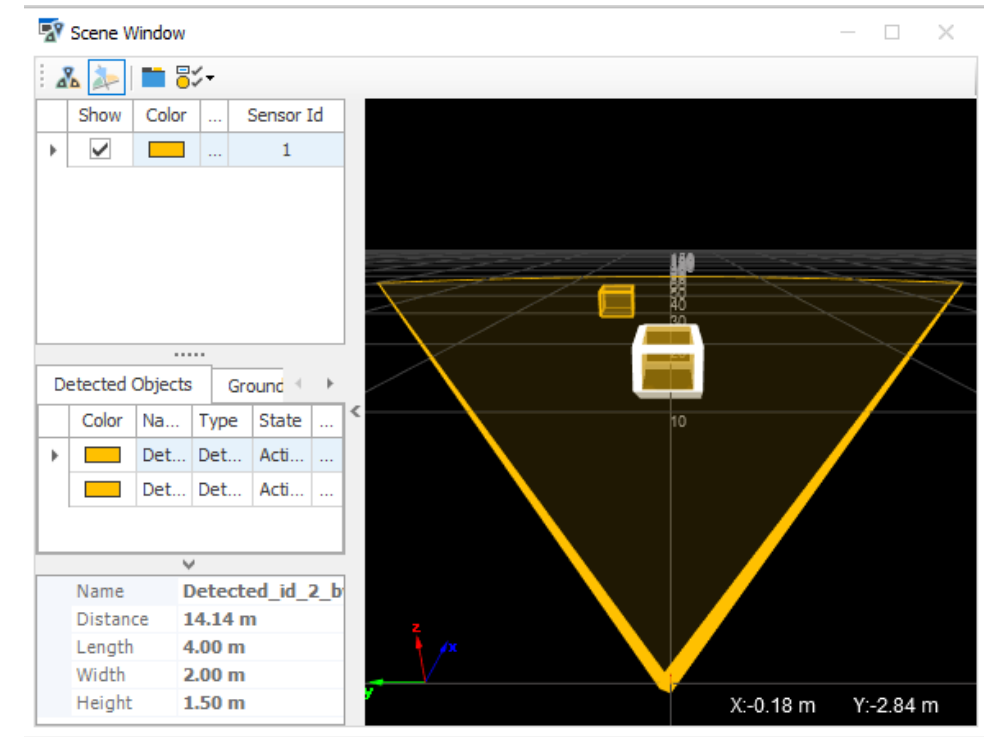
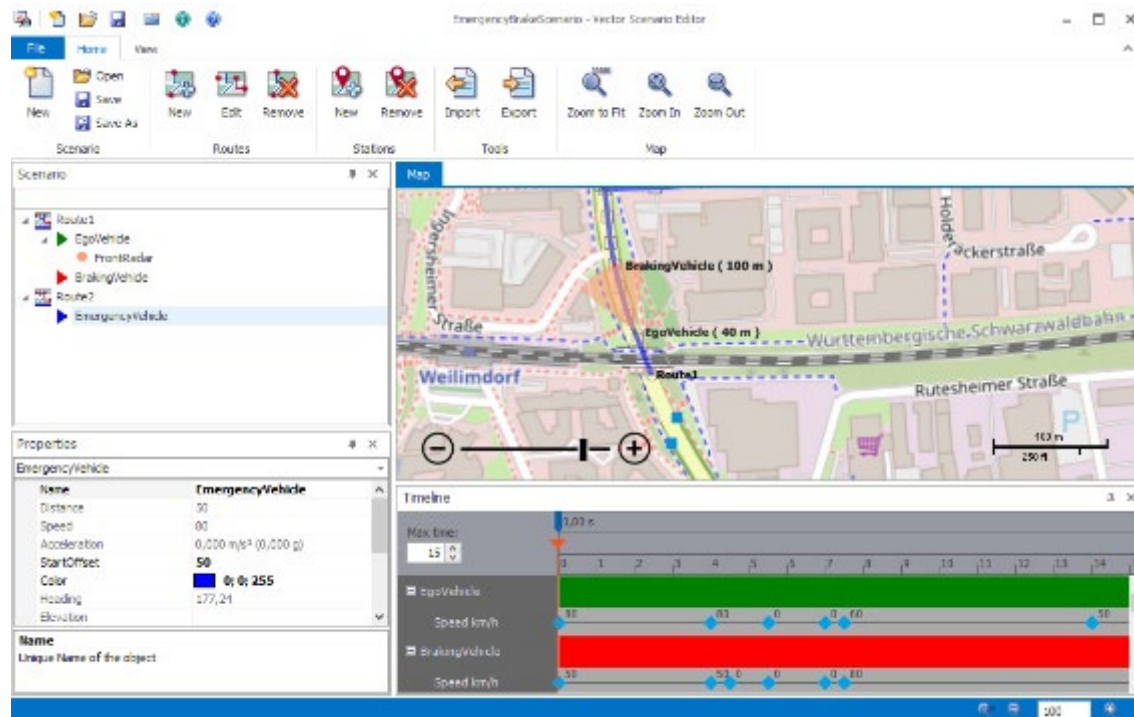
CANoe Car2x

Aufgaben im Arbeitsalltag



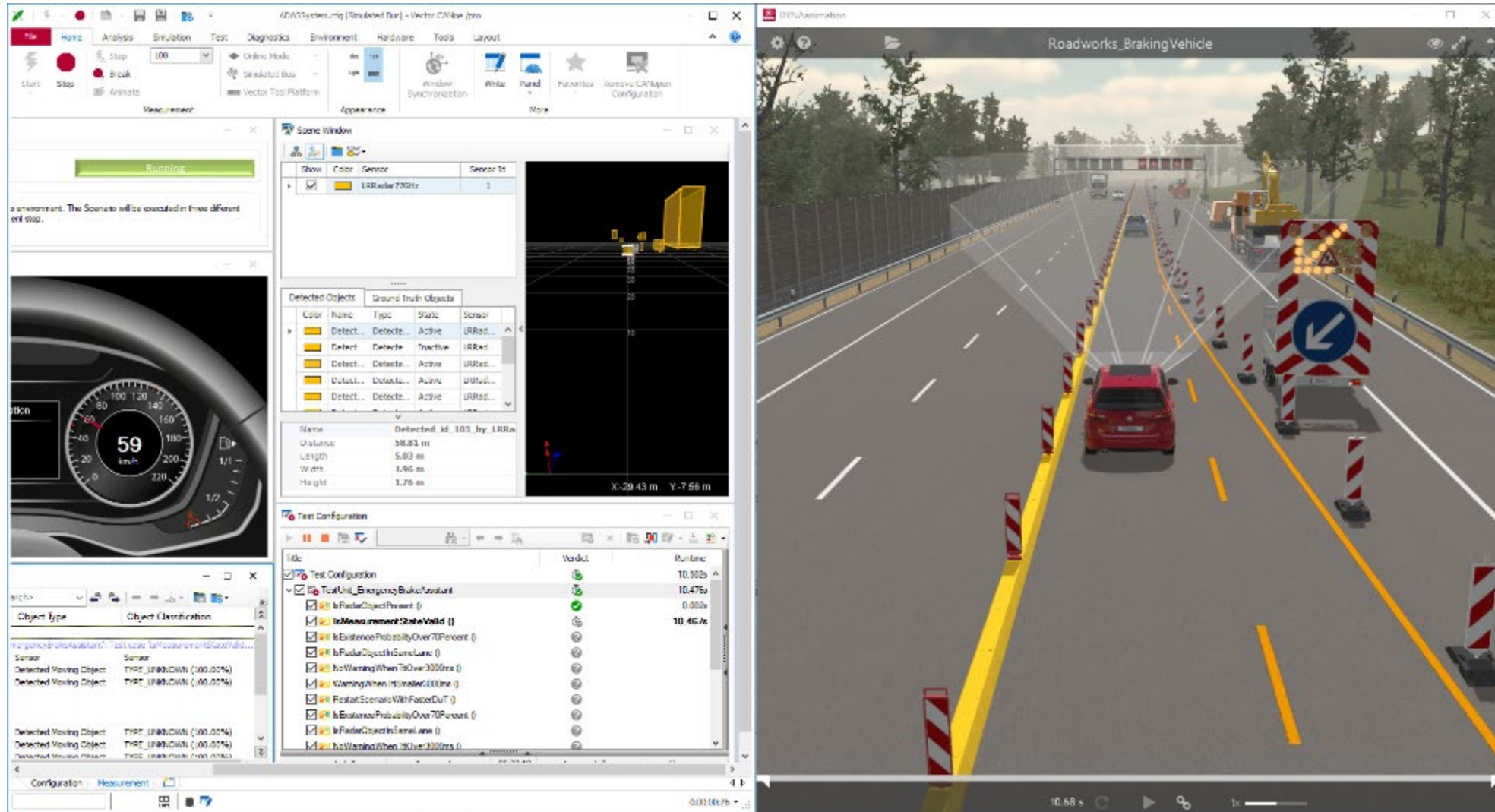


Testdaten erstellen

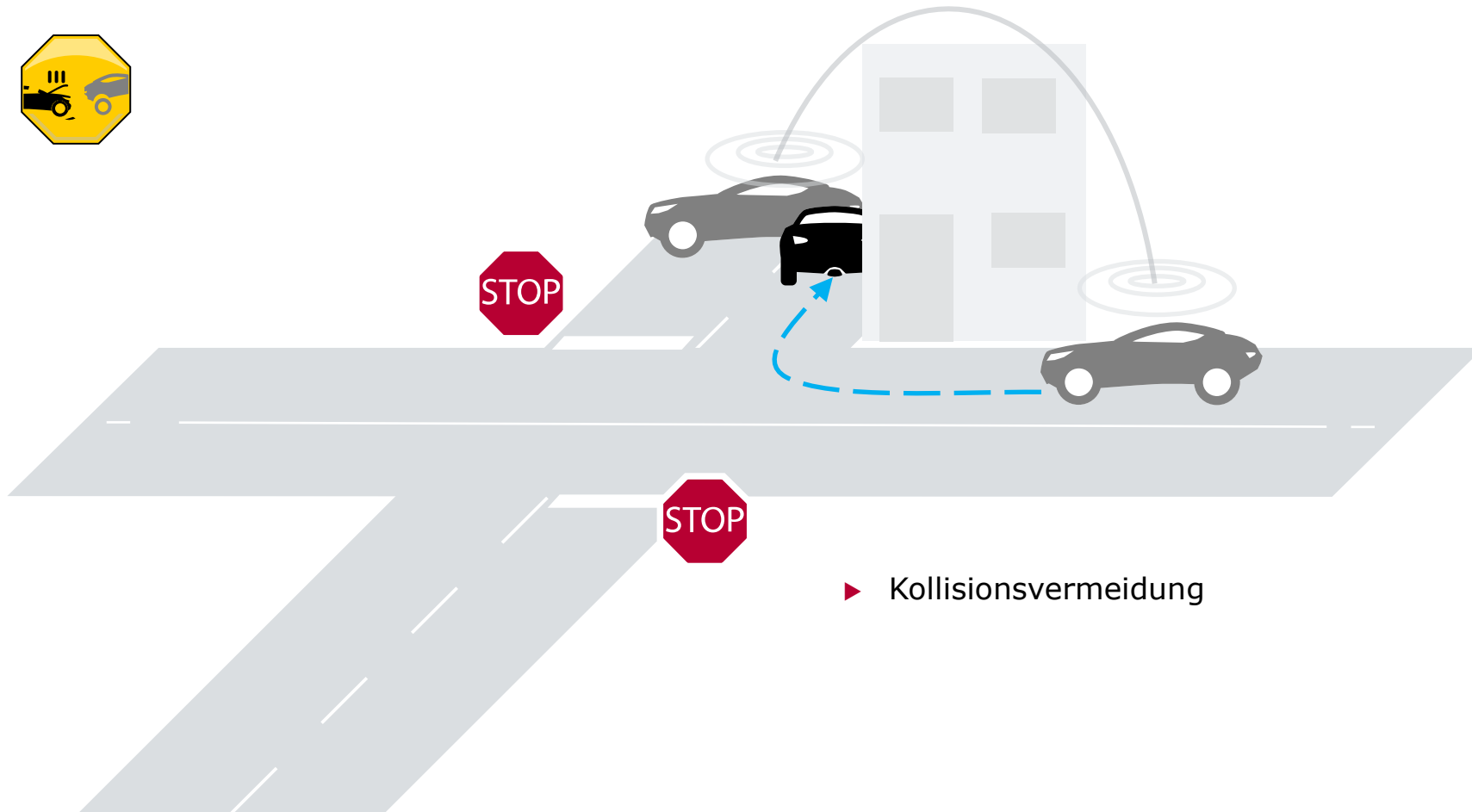


- ▶ ADAS Szenarien
 - ▶ Einfache Szenarien mit dem Szenarien Editor erstellen
 - ▶ OSI (Open Simulation Interface) konforme Szenarien einlesen
 - ▶ Mithilfe von DYNA4 Closed Loop Testing

CANoe und DYNA4



Active road safety – Stationary Vehicle (V2V / V2I)

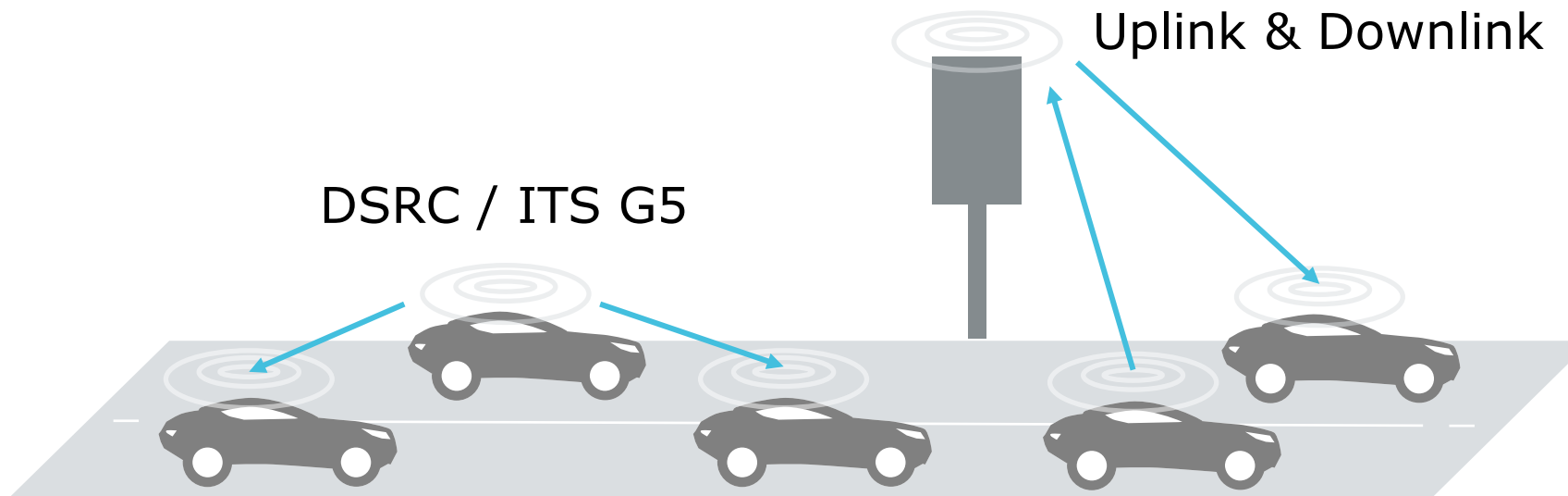


► Kollisionsvermeidung

Technologie - WLAN 802.11p

► ITS G5 / DSRC

- WLAN 802.11p
- Ad-Hoc Kommunikation
- Broadcast Kommunikation
- Signiert mit Zertifikaten, aber nicht verschlüsselt
- Frequenzband 5,9 GHz
 - > 7 Channels, 10 MHz bandwidth
- Reichweite bis zu ~1km

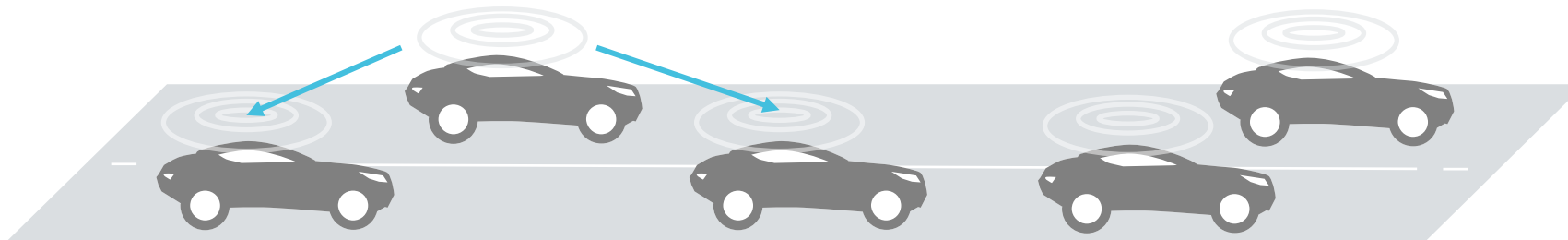


Technologie - C-V2X (Cellular V2X)

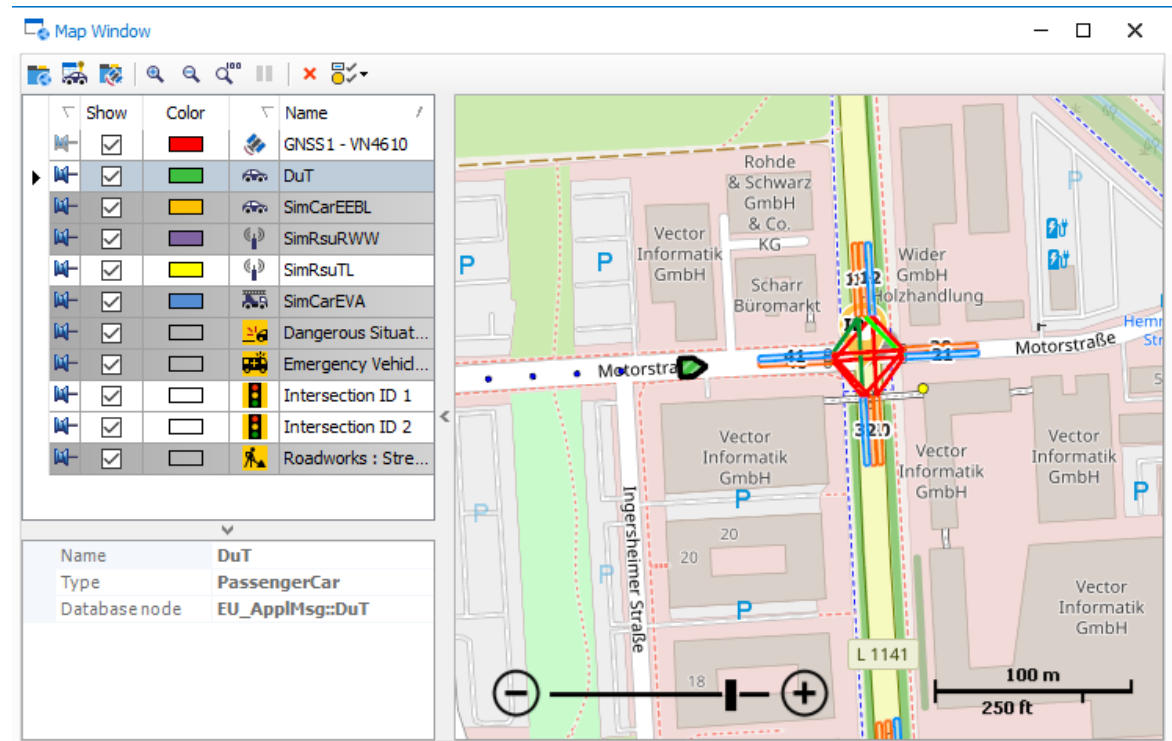
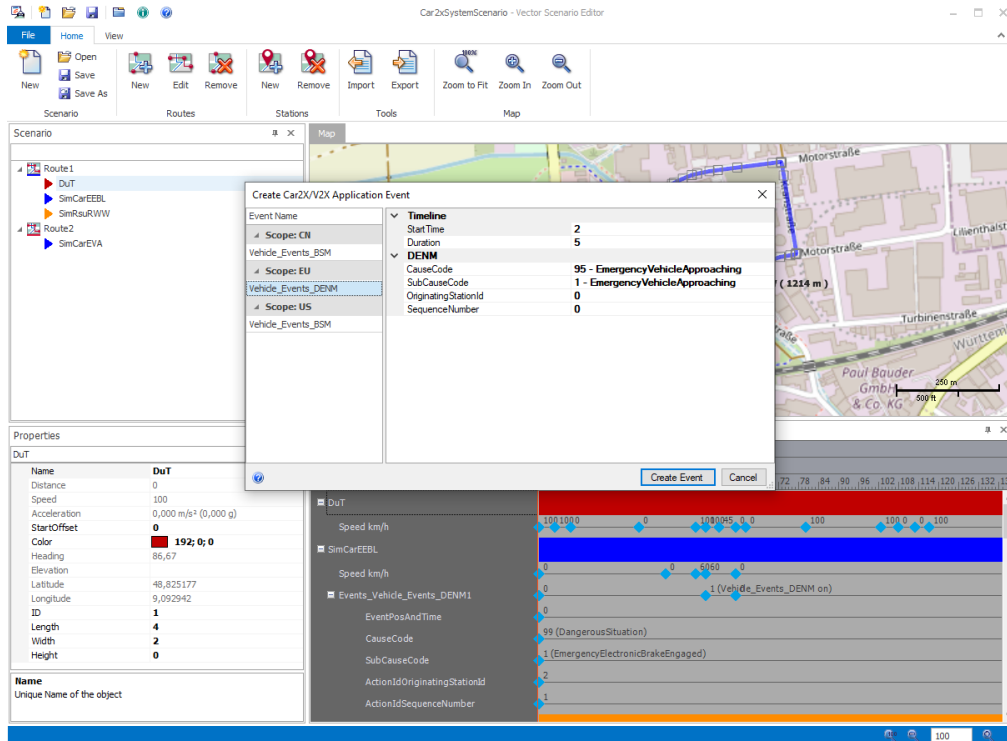
► C-V2X

- LTE-PC5 link – **direkte Kommunikation** zwischen Fahrzeugen/Infrastruktur
 - > Cellular V2X sidelink
 - > Transmission Mode 4 (3GPP Rel.14 as LTE-V2X, PSCCH, PSSCH)
 - > Möglichkeit weitere Nutzergruppen wie Fußgänger & Radfahrer über Smartphones einzubeziehen

Sidelink PC5



Testdaten erstellen



- ▶ Car2x Szenarien
 - ▶ Einfache Szenarien mit dem Szenarien Editor erstellen
 - ▶ Einfache Erstellung von DENM/BSM über GUI im Editor
 - ▶ Automatische Berechnung von CAMs zur Laufzeit

Aufgaben

► Requirementsanalyse

- Zukünftige Anforderungen erkennen, spezifizieren und in Features aufteilen
- Verwendete Tools: JIRA (Ticketing System, Sprintplanung), Microsoft Tools (z.B. Word)

► Implementierung

- Umsetzung neuer Features, Bugfixes und Refactoring
- Programmiersprachen: C# & C++
- Verwendete Tools: Visual Studio, Versionierungssystem (z.B. SVN, Git), beliebiger Browser für StackOverflow

► Test & Erprobung

- Erstellung von Test-spezifikationen & -konfigurationen sowie Erprobung aktueller Stände durch kleinere Projekte
- Verwendete Tools: Eigene automatisierte Testumgebung

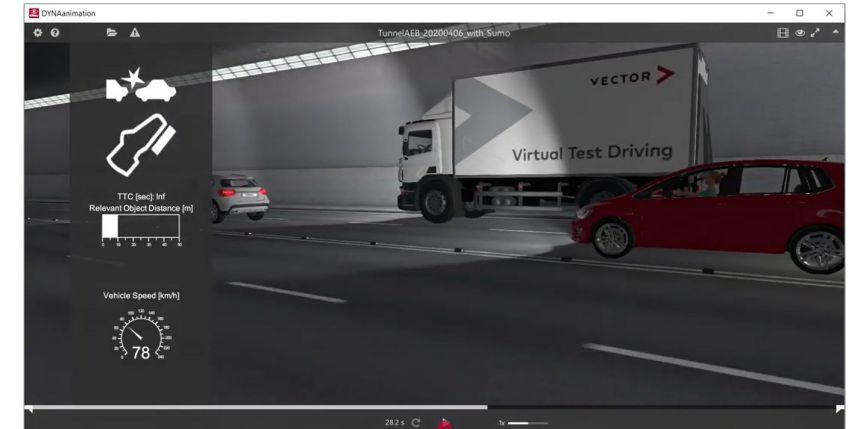
Zeit für Fragen!

Autorin:
Stupnik, Sonja (sonja.stupnik@vector.com)
Vector Informatik GmbH

Weiterführende Infos

- ▶ Virtueller Fahrversuch: ADAS-Steuergeräte mit DYNA4 und CANoe wie in realen Testfahrten stimulieren:

- ▶ <https://www.youtube.com/watch?v=1byiePHWUK4>



- ▶ Car2X Anwendungen und Technologien bei Mercedes-Benz:

- ▶ <https://www.youtube.com/watch?v=fwUS91aO-Jk>

