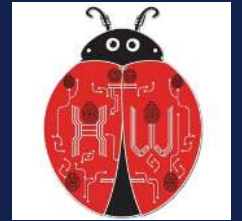


Connected Car Hacking training

01. Ignition

Hardware.io USA 2026



Quarkslab

Training Objectives



During this training, we will cover:

- ▶ What is **inside** modern vehicles
- ▶ How to build an automotive **test bench**
- ▶ Type of **memory** used in Infotainment and Telematic ECU and how to dump it
- ▶ Discover how to test the different **attack surfaces** of these ECUs
- ▶ How to create a **test basestation, capture and alter communication** between an ECU and a backend server
- ▶ **Reproduce** some real world **vulnerabilities** we faced during assessments



What this training is not

- ▶ A coding training
Most of the scripts will be premade, to focus on understanding what is going on, instead of fixing code typos
- ▶ An advanced CAN training
You'll interact with CAN buses, but we will not spend too much time digging deep into CAN, ISO-TP and UDS protocols

Philippe AZALBERT

- ▶ Joined Quarkslab in 2021 as Automotive security leader
- ▶ 10 years+ in technical support unit for LE, focusing on car
- ▶ Interests:
 - ▶ Car hacking (no, really?)
 - ▶ Fault injection
 - ▶ Having fun with radio frequencies
- ▶ Contact - mail: pazalbert@quarkslab.com / X: [Phil_BARR3TT](#) / [Linkedin](#)



- ▶ Cybersecurity company founded in **2001**, located in **Paris, Rennes** and **Buenos Aires**
- ▶ Part of **Airbus Cyberdefense** since May 2026
- ▶ Automotive dedicated team working for **OEM, Tier 1, forensic units, ...**
- ▶ Strong focus on R&D, engineers having **40% of their time** to dig into cool topics like:
 - ▶ **RH850**: emulation support with Unicorn Engine and ID code protection bypass with fault injection
 - ▶ **Telematic control unit**: Tearing down a car telematic unit (and finding an accident on Facebook)
 - ▶ **Forensic/reverse engineering**: Recovering an ECU firmware using disassembler and branches
 - ▶ **EV vehicle/charger security**: Open source EV charger security audit

What about you?



It's your turn, tell us:

- ▶ Your name
- ▶ Experience in Car Hacking
- ▶ What do you expect of this training

Feel free to skip parts you're not comfortable with



Illustration: [\[link\]](#)

Training setup: presentation



CTFd and Discord servers

- ▶ **Labs** will consist on **hands-on exercises** and **multiple choice questions** available on a CTFd server
- ▶ Solution for each challenge will be released on the CTFd server during the training
- ▶ CTFd server address: <https://training-car-hacking.quarkslab.com>
Invite code: HardwearIO-USA-26
- ▶ You can ask question or share material using the dedicated Discord server: <https://discord.gg/E4cqVzq2>
- ▶ The CTFd server instance will be up and running up to one month after the training

Virtual Machine



- ▶ Login: **user**
- ▶ Password: **password**
- ▶ On Virtualbox, it exposes port **2222** for SSH



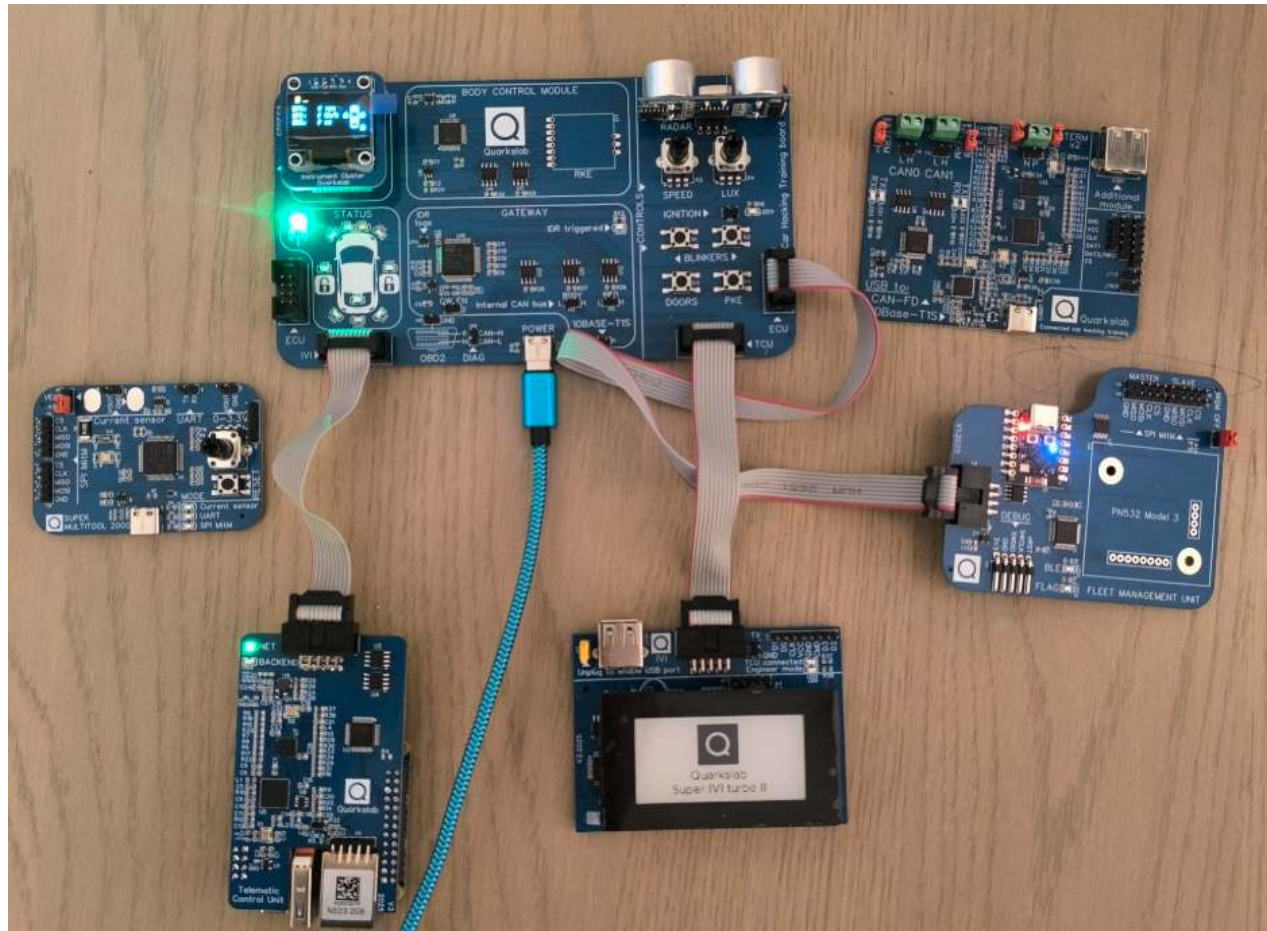
Virtual Machine - details



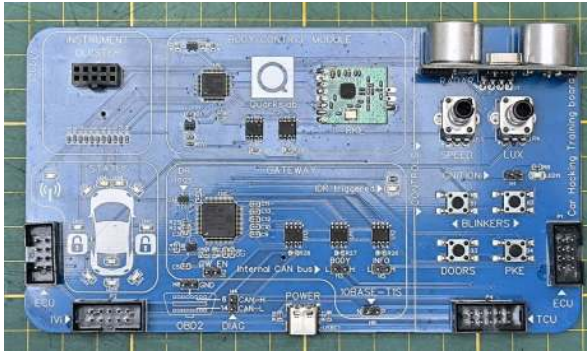
Ubuntu based VM, including:

- ▶ **Can-utils & Scapy**: CAN tools
- ▶ **srsRAN & Open5GS**: LTE network emulation tools (with BladeRF support)
- ▶ **Sysmo-isim-tools**: programmable SIM management
- ▶ **imHex**: Hex editor
- ▶ **Binary Ninja**: reverse engineering software
- ▶ **Saleae Logic 2**: logic analyser
- ▶ **JADX**: APK reverse engineering
- ▶ **Terminator**: multi-window terminal
- ▶ **Facedancer**: USB emulation
- ▶ **Frida**: binary instrumentation
- ▶ **Wireshark, Nmap, Termshark, Bettercap**: network analysis

Training ECUs and tools



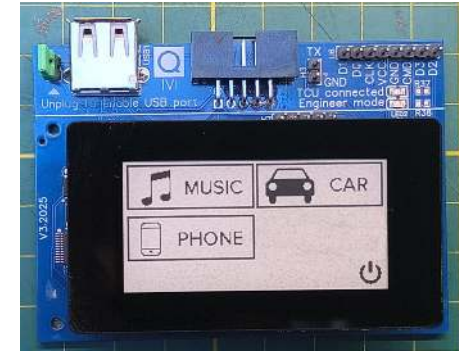
Body Control Module



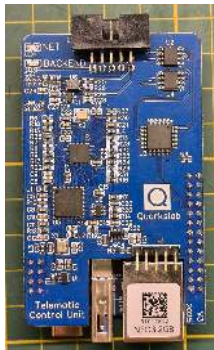
Instrument Cluster



In-Vehicle Infotainment



Telematic Control Unit



To interact with the training ECUs, the following tools will be used:

- ▶ **2xCAN-FD + 10Base-T1S USB adapter**
- ▶ **Super Multitool 2000**
- ▶ **Facedancer21**
- ▶ **SD card reader**
- ▶ **PCSC SIM card reader**

And few little “souvenirs”



To allow you to practice your new knowledge at the end of the training, the following items will be given to you:

- ▶ Dual CAN-FD adapter
- ▶ Sysmocom programable SIM card
- ▶ eSIM adapter
- ▶ SD card adapter



Playing with real ECUs



- ▶ One of our **Car in a Box** will be available during the training
- ▶ A **Raspberry Pi** with a **PiCAN** hat is connected to one of the CAN bus, giving access at least to the **Instrument Cluster**
 - ▶ **Wi-Fi SSID:** Quarkslab_CarHacking
 - ▶ **Wi-Fi passphrase:** HackMyC4r!
 - ▶ **IP:** 192.168.11.254
 - ▶ **Login:** student
 - ▶ **Password:** canihack



“My Car” application



- ▶ For some labs, a fake mobile application is available at <http://192.168.11.254:8080>
- ▶ It displays **information** sent by your setup
- ▶ You can also remotely **lock/unlock the doors**



Application availability

Due to the limited capabilities of the Raspberry board 1.3W used, communication with the backend and the Telematic Unit may have temporary disconnection, please be patient.

Introduction to Automotive Security

Detail of a modern car

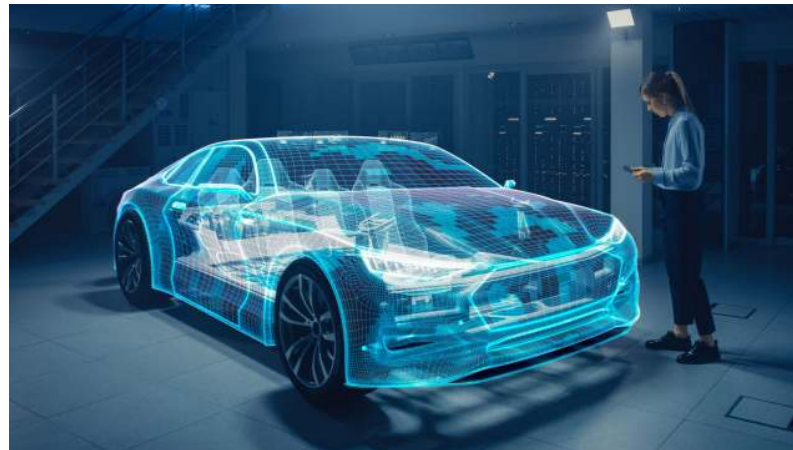


Same base as many decades: 4 wheels around a motor.

But with significant improvements:

- ▶ **Driving assistances** (anti-collision, line detection, ESP/ABS, road sign id, self-park...)
- ▶ **Connectivity** (GPS/LTE/Wi-Fi/Bluetooth...)
- ▶ **Onboard services** (remote control, localisation, auto-diagnostic...)
- ▶ **Autonomous driving**

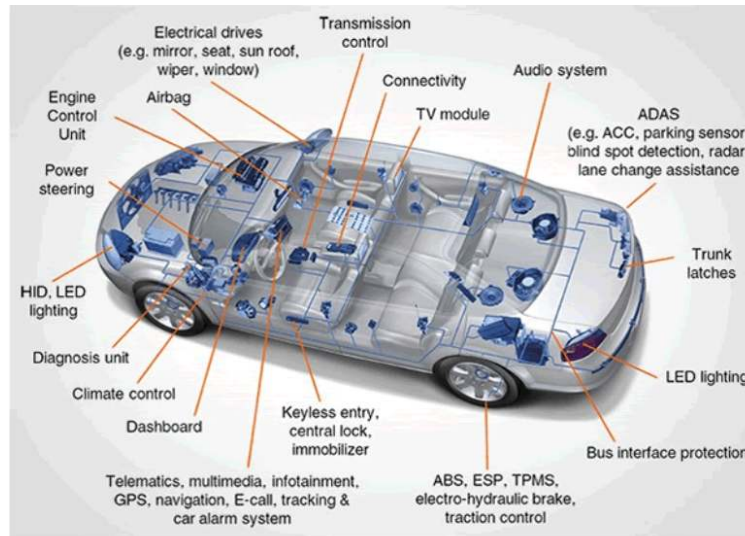
Illustration: [\[link\]](#)



Cores of a car: ECUs

ECU: Electronic Control Unit

It reads **SENSORS**, manages **ACTUATORS**, communicates with other **ECUs/DEVICES/BACKEND** through wired or wireless networks. Could be one or several **MCU/SOC**.



ECUs communications: multiple internal networks

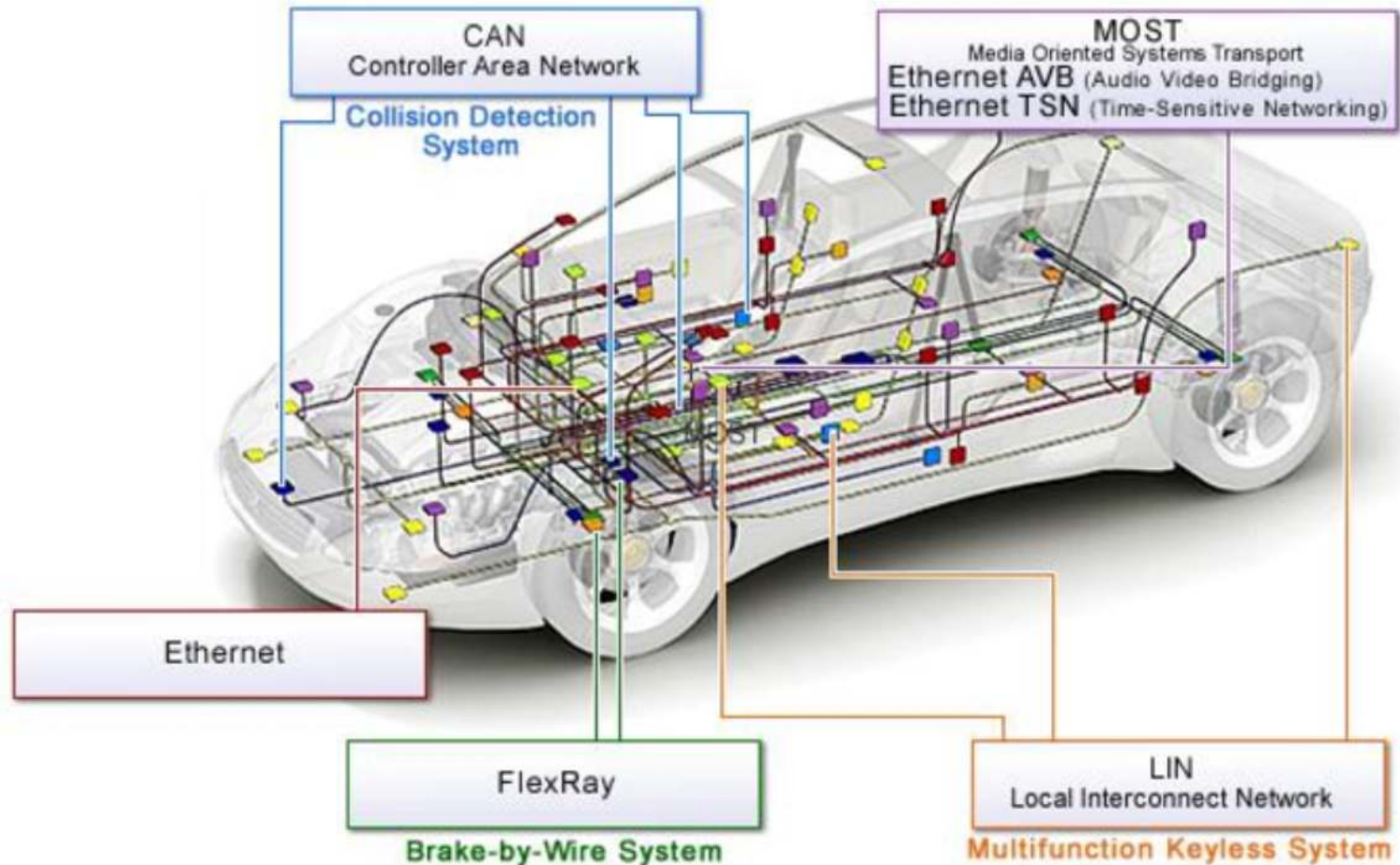
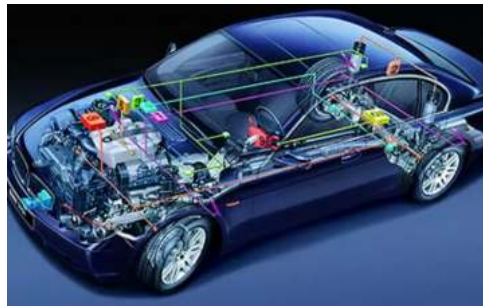


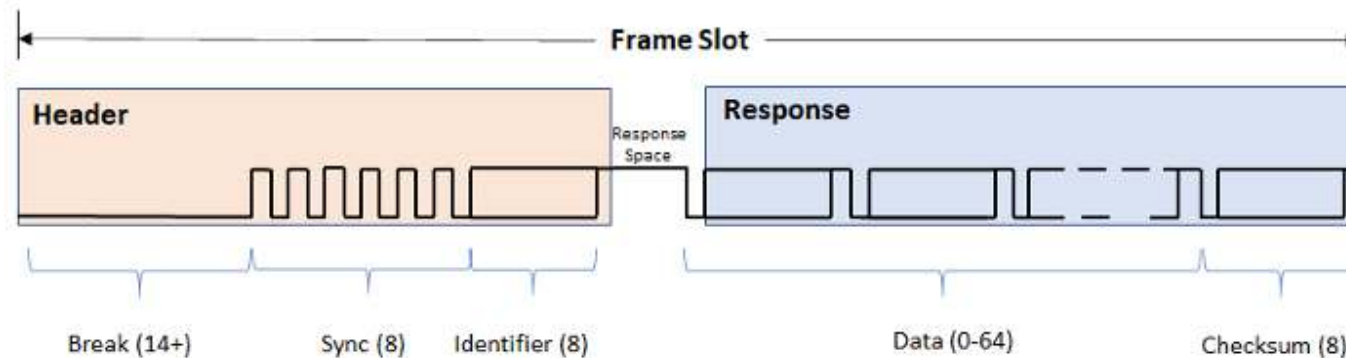
Illustration: [link]

- ▶ **Two twisted wires** half-duplex network protocol with speed up to **1 Mbit/s** and **8 bytes** payload (ISO 11898)
- ▶ Fault resistant protocol
- ▶ **CAN-FD** (Flexible Data Rate) allows speed up to **8 Mbit/s** and **64 bytes** payload
- ▶ Latest version, CAN-XL: **2048** bytes payload and speed up to **20Mbit/s**
- ▶ Most commonly used in-vehicle network to connect ECUs

Illustration: [\[link\]](#)



- ▶ **Single wire** serial network protocol with speed up to **19.2 Kbit/s (ISO 17987)**
- ▶ **Broadcast** protocol allowing up to **16 nodes**
- ▶ **Master - slave** communication system
- ▶ **Low cost** network for **non-critical** applications (locking system...)

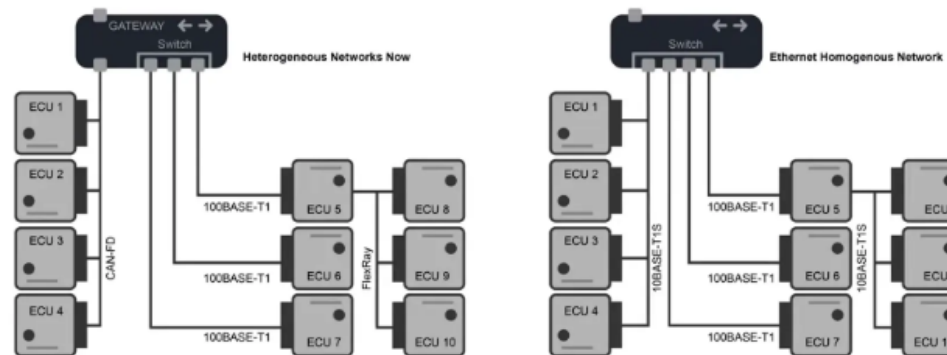


Automotive Ethernet



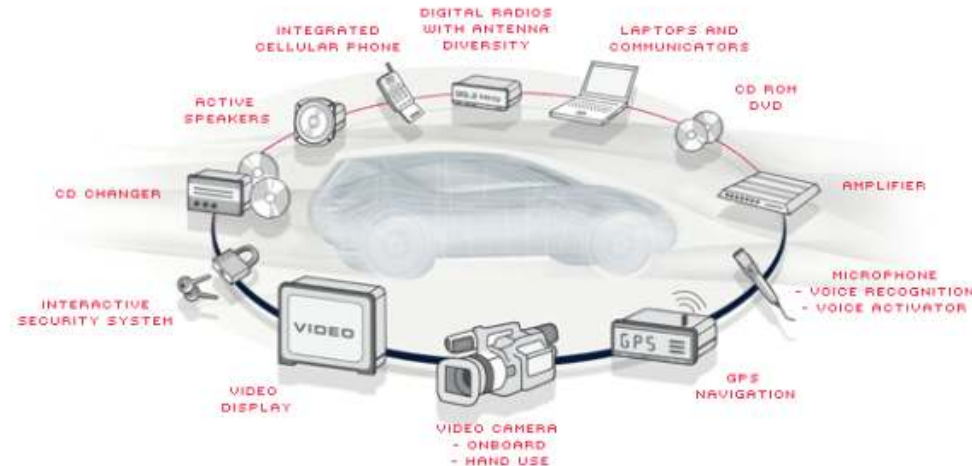
- ▶ **High speed** two twisted wires network with speed up to **1 000 Mbit/s**
- ▶ **Physical layer** different from traditional ethernet (100/1000Base-T1, BroadR-Reach)
- ▶ Handle **generic network protocols** and **automotive specifics**
- ▶ Also **10Base-T1S**, **multidrop ethernet** network with speed up to **10 Mbits/s** for **Ethernet homogeneous network**

Illustration: [\[link\]](#)



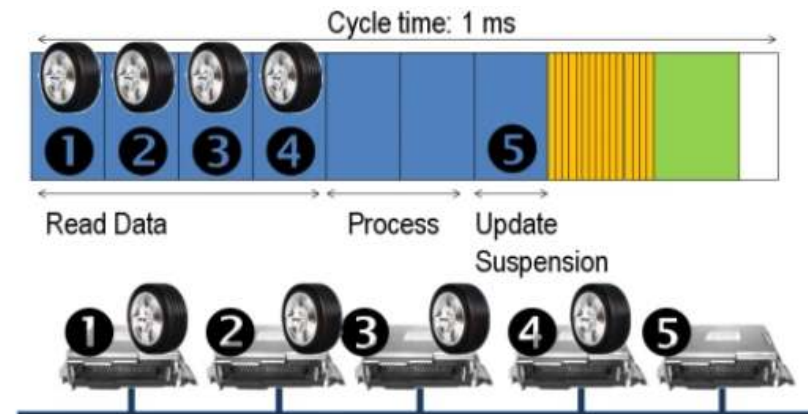
MOST: Media Oriented System Transport

- ▶ **High speed** multimedia network with speed up to **150 Mbit/s (ISO 21806)**
- ▶ **Ring network** topology to transport audio, video, voice and data signals
- ▶ **Expensive network** using optical fibre



- ▶ **High performance** 2/4 wires network protocol with speed up to **10 Mbit/s (ISO 17458)**
- ▶ **Fault resistant** and **deterministic**
- ▶ **TDMA** (Time Division Multiple Access): Flexray node communicates during a **scheduled time slot**
- ▶ Used for **high-performance applications** (Steer-by-wire, ADAS, ...)

Illustration: [\[link\]](#)



Attack surfaces

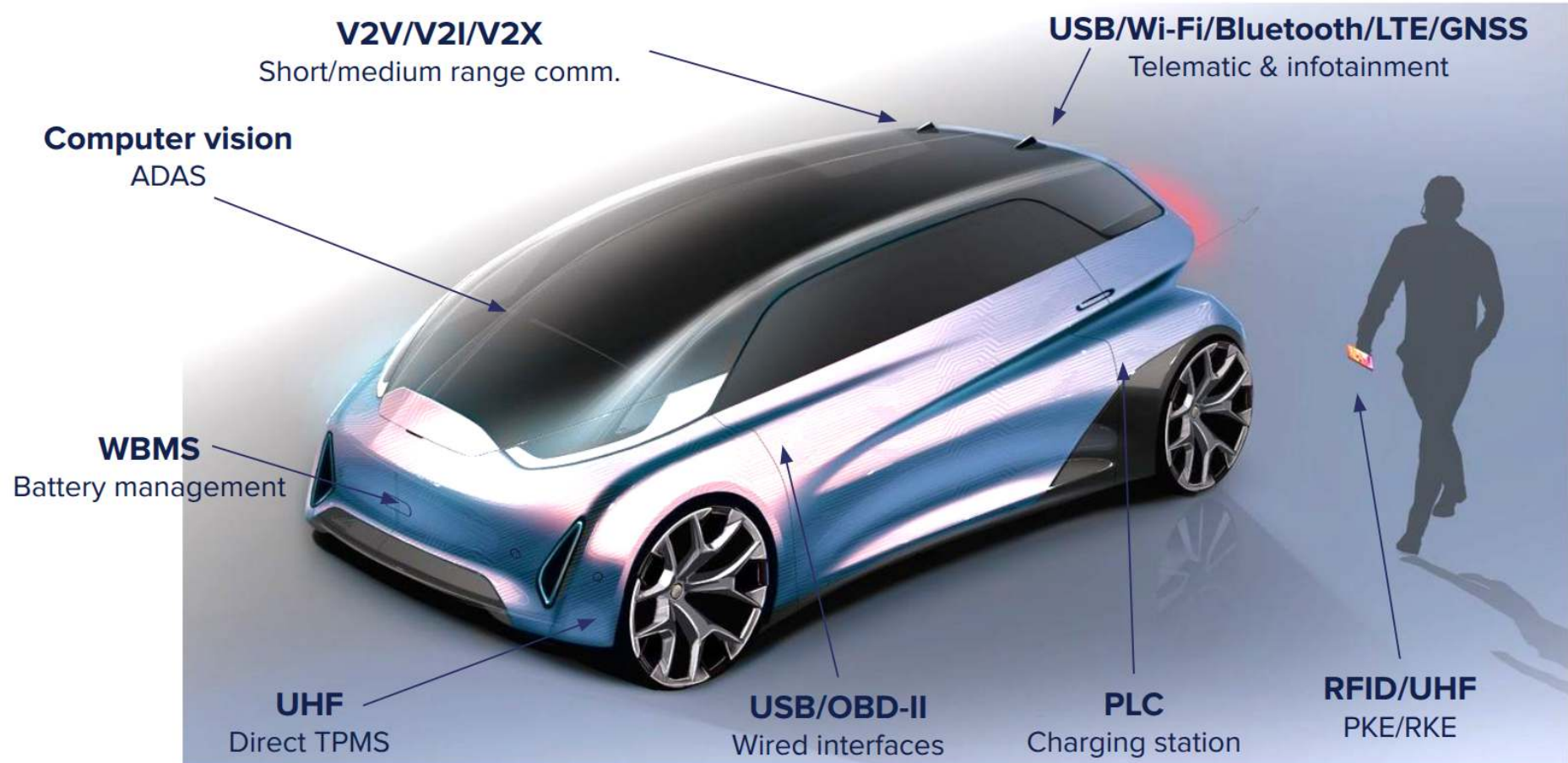


Illustration: [link]

Automotive & cybersecurity: milestone



In **2015** security researchers demonstrated several **major vulnerabilities** in a connected vehicle, causing the recall of **1.5M** of vehicles in US.

Illustration: [\[link\]](#)



Network segregation



Nowadays, it is common that in-vehicle networks are **segregated** regarding their usage, a **gateway** managing a **secure bridge** between them.

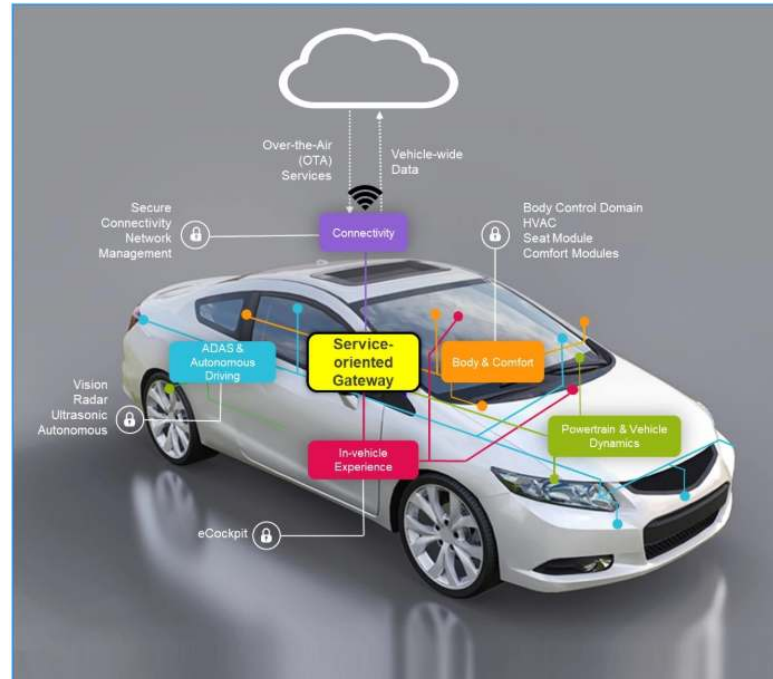
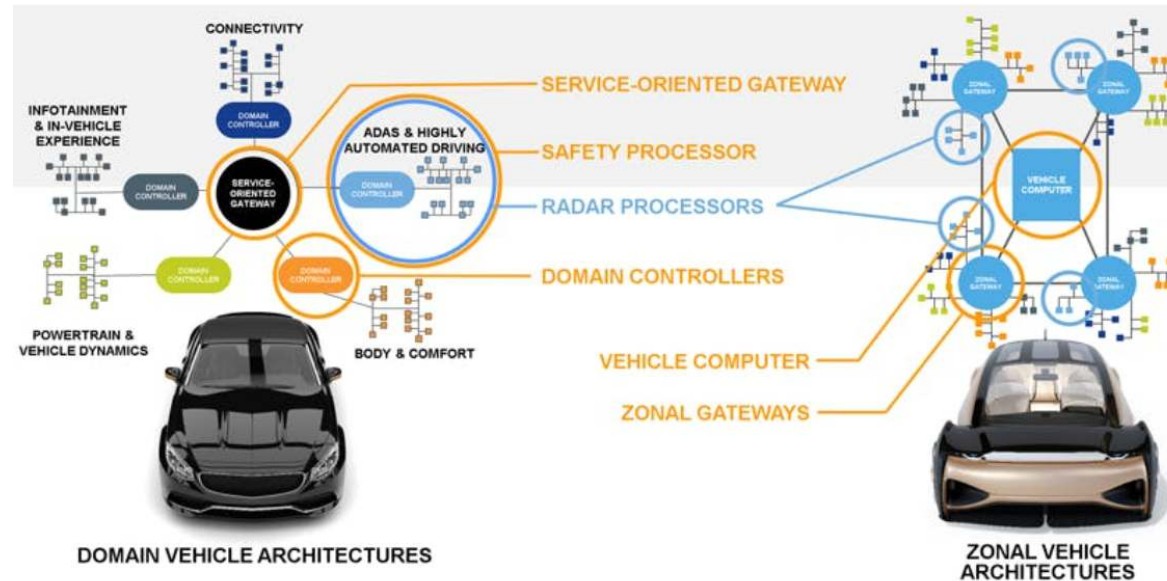


Illustration: [link]

From Domain to Zonal controller

OEMs tend to switch from **Domain architectures** to **Zonal architectures**, where a **central computer** handles data and actuators from different isolated zones, providing better scalability/reliability, **bringing software-defined vehicles**.

Illustration: [\[link\]](#)

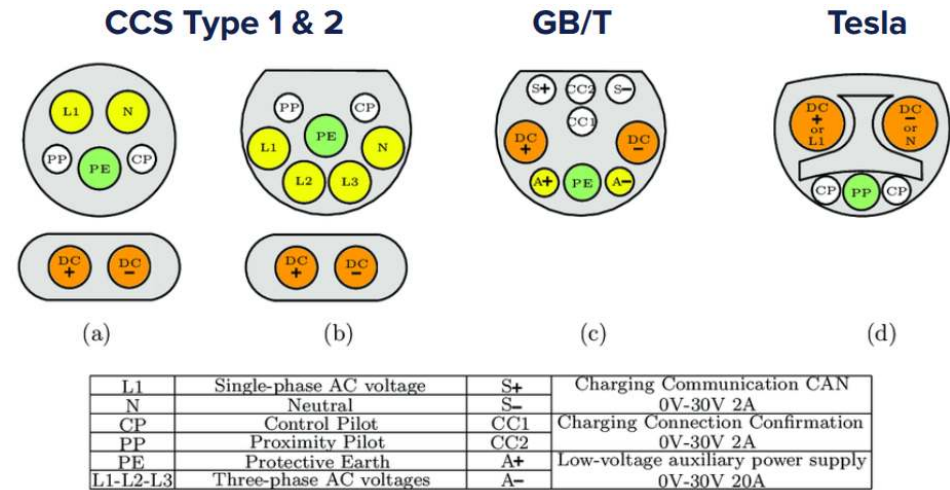


Electrical vehicle plugs



Illustration: [link]

- Electrical vehicles are able to communicate with charging stations using **PLC** or **CAN**
- **PLC** communication is made via **Control Pilot** pin



Vehicle-to-grid



- ▶ **Plug & charge**, defined by **ISO-15118**, allows drivers to simply plug their vehicle and start charging without needing to authenticate
- ▶ It provides a standardized **secure** communication protocol, relying on several certificates (OEM, MO, CPO...) using several **PKIs**

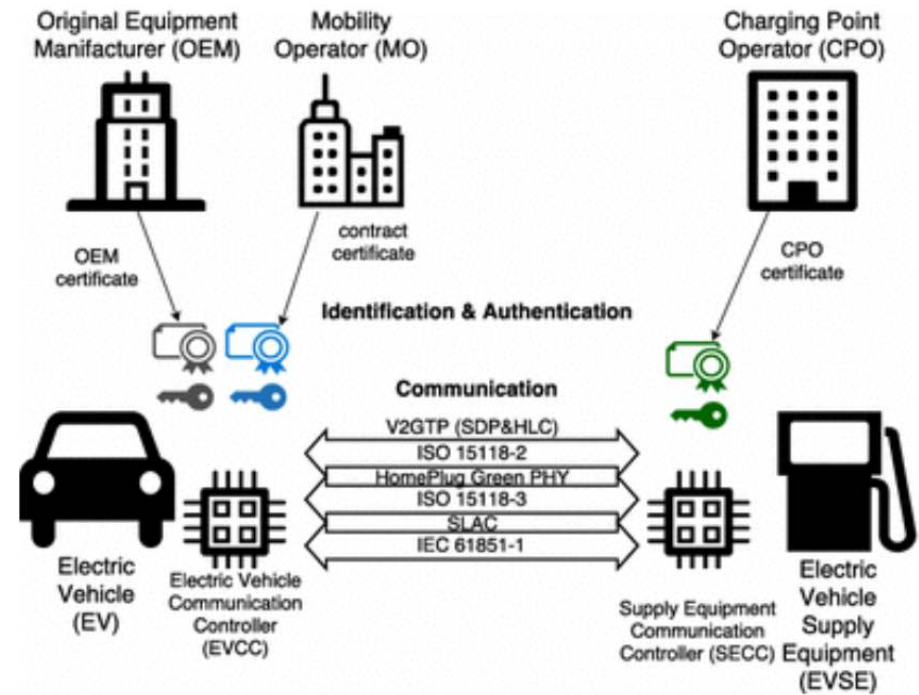


Illustration: [\[link\]](#)

Security

UNECE WP.29 - Regulations no. 155 & 156 (annex 1958 agreement since 22 January 2021)

- ▶ Manage vehicle cybersecurity risks and secure vehicles by design
- ▶ Detect /respond to security incidents across a vehicle fleet
- ▶ Provide safe, secure software updates

Safety

EU 2015/758 - April 2015: mandatory e-Call system in new cars

- ▶ Data/voice connection plus GNSS

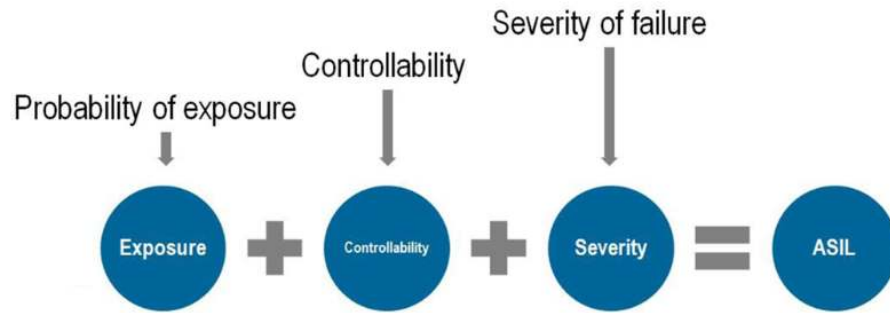
UNECE R64 - 2009 & **UNECE R141** - 2017: mandatory tire pressure monitoring system

> **Safety** brings more attack surface and complexity, requiring more **Security** for automotive <

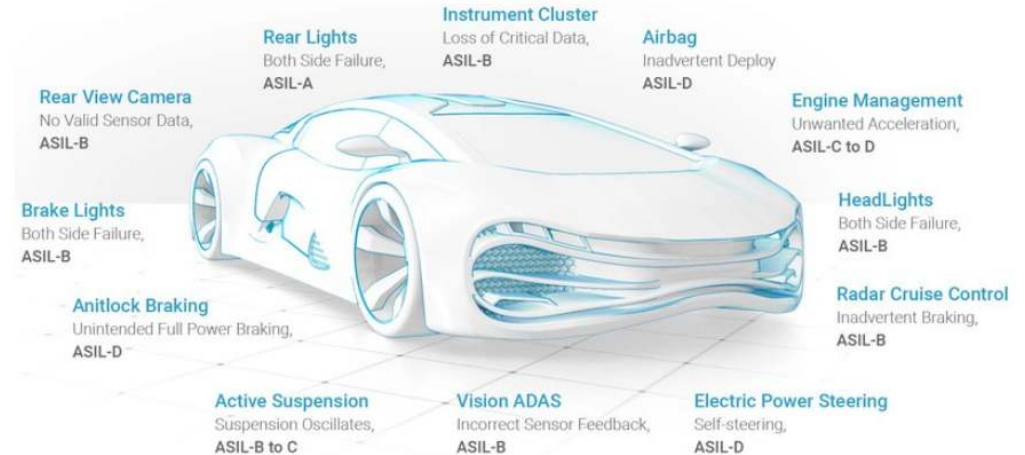
From ISO 26262 (ASIL)



Illustration: [link]



ASIL: Automotive Safety Integrity Levels



To ISO 21434



UN Regulations R155 & R156 refer to ISO 21434

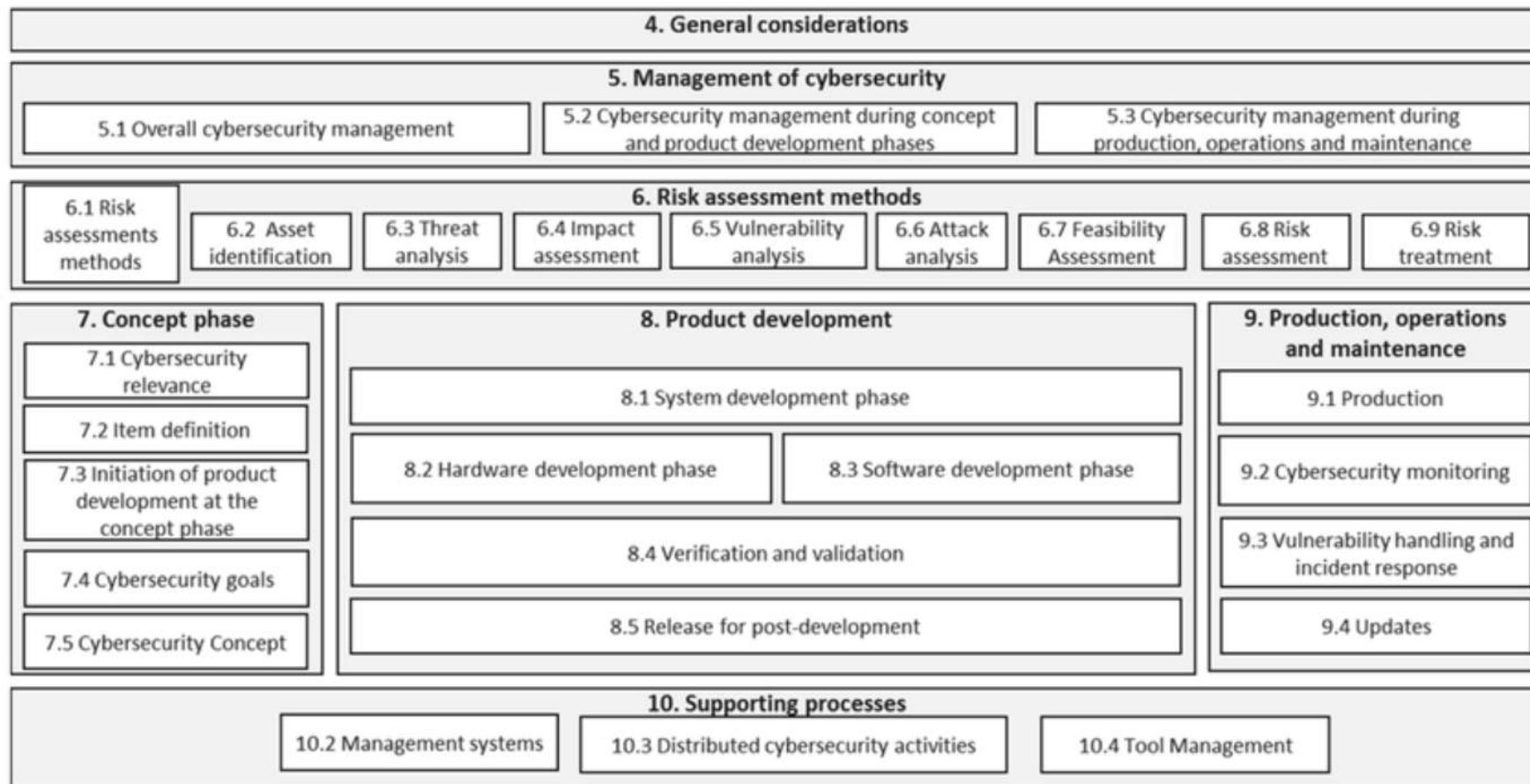
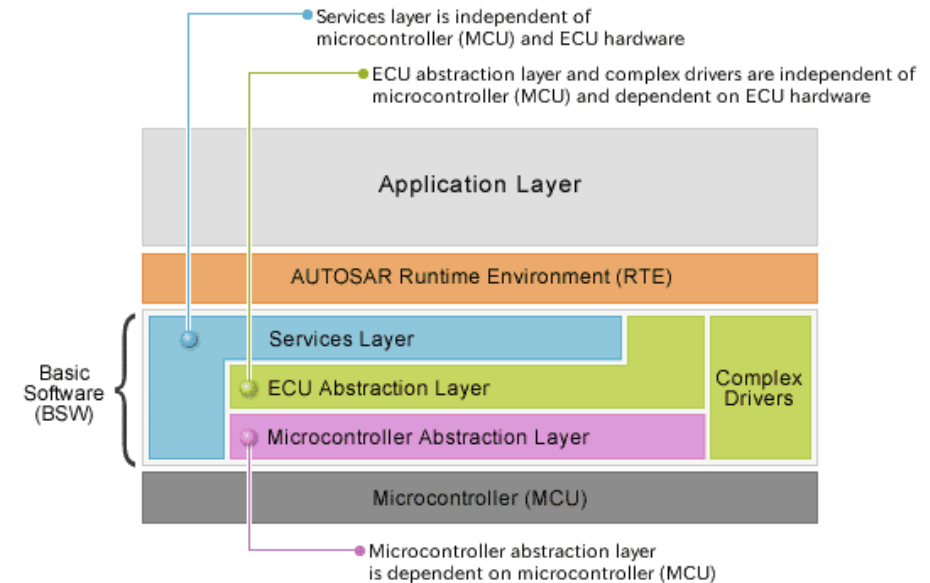


Illustration: [\[link\]](#)

- ▶ Development partnership of **automotive parties** (manufacturers, Tier 1 suppliers) founded in 2003
- ▶ Defines a **standardized software architecture** for ECUs, **methodology** and **procedures**
- ▶ **Simplifies** development phases across various actors



Thank you

Contact information

Email:

contact@quarkslab.com

Phone:

+33 1 58 30 81 51

Website:

<https://www.quarkslab.com>