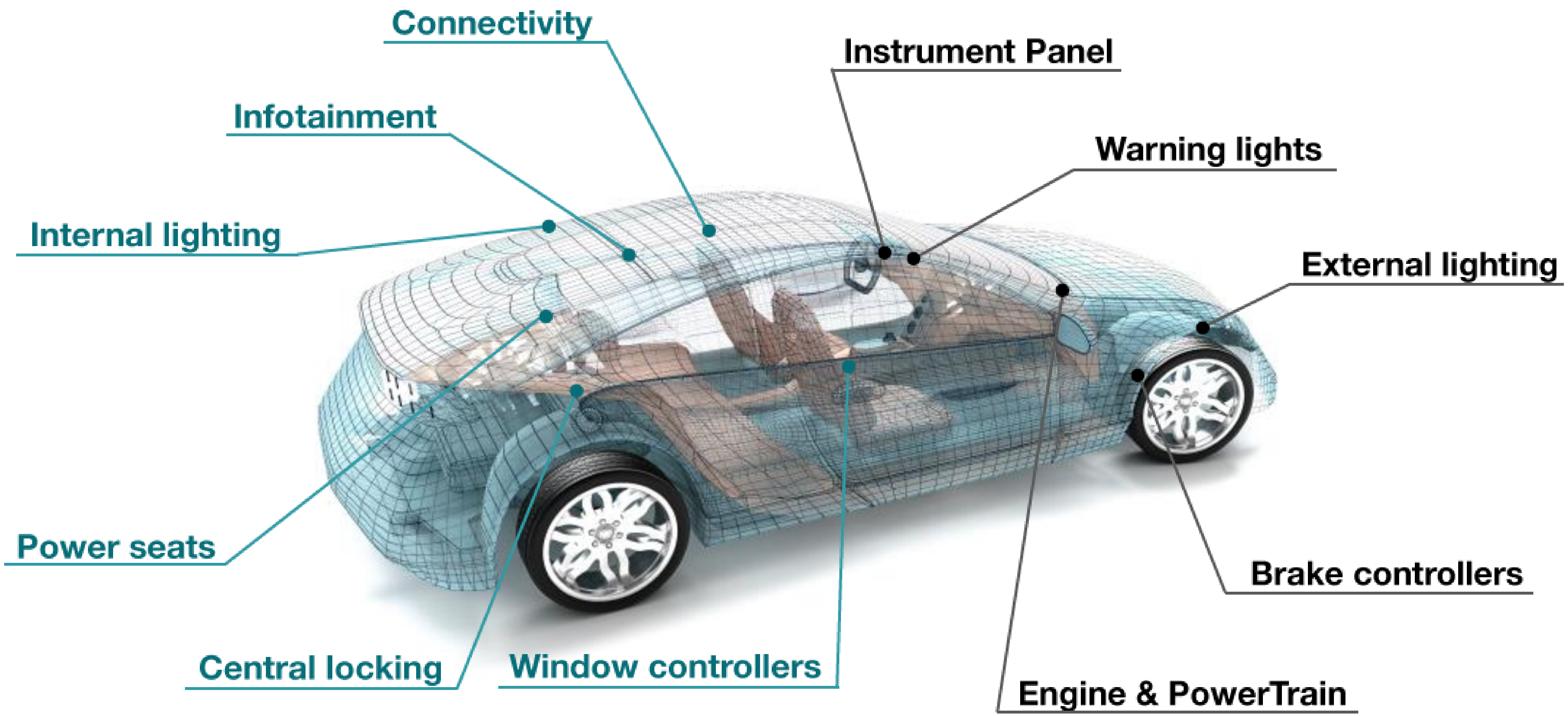
Issues Affecting Automotive Software Developers

23 July, 2019

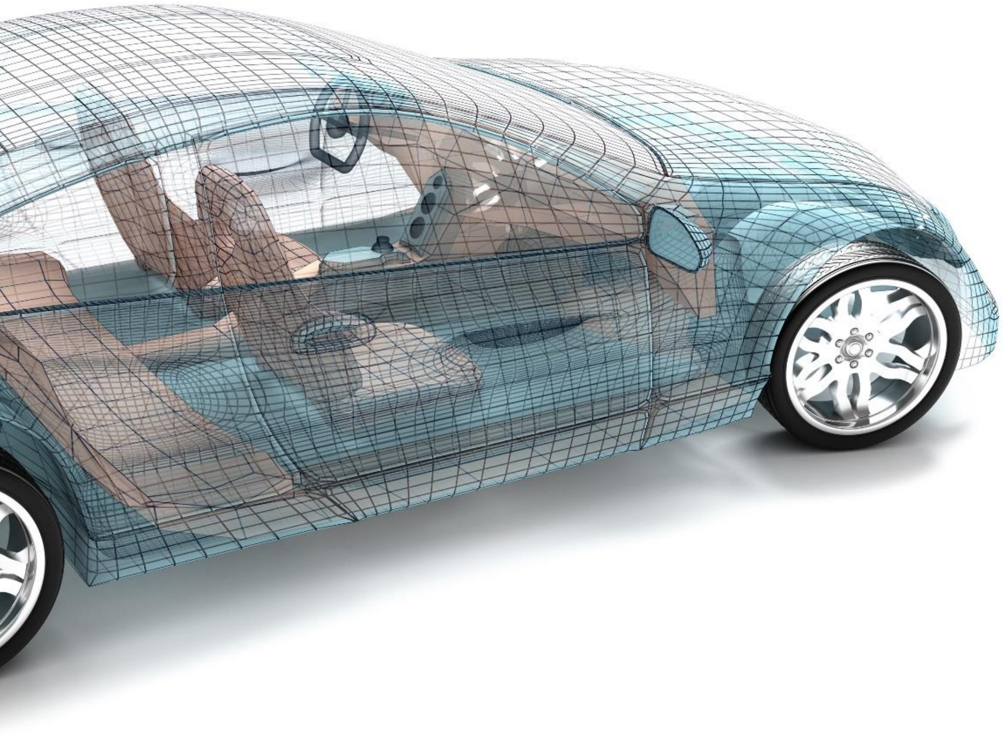
Stephen Ridley

WITTENSTEIN

Control and Infotainment Software



Advanced Driver Assistance Software

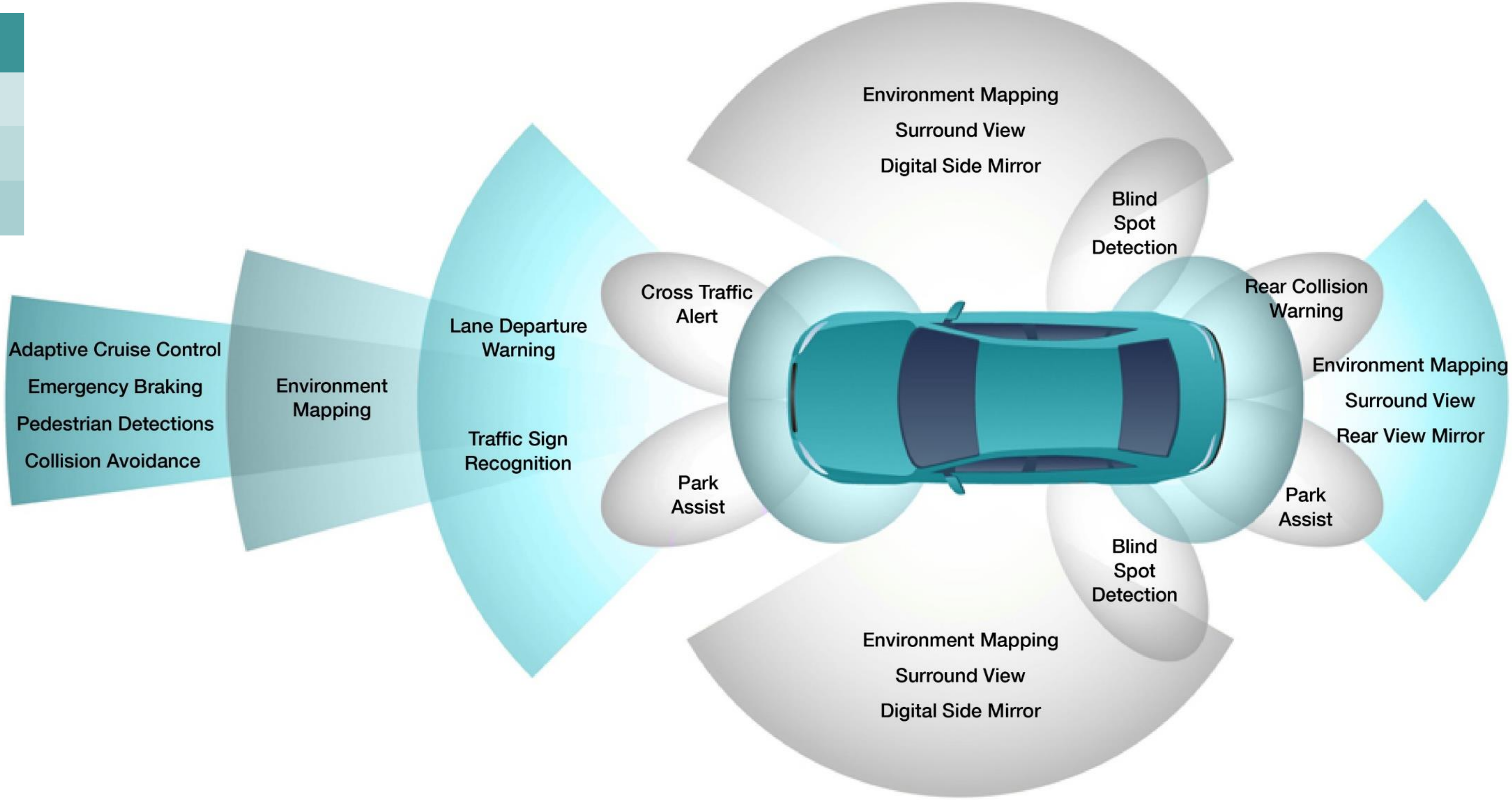


Adaptive cruise control
Glare-free high beam
Adaptive light control
Automatic parking
Automotive Navigation system
Automotive night vision
Blind Spot Monitor
Collision Avoidance system
Crosswind stabilization
Cruise control
Driver Monitoring Systems
Electric vehicle warning sounds

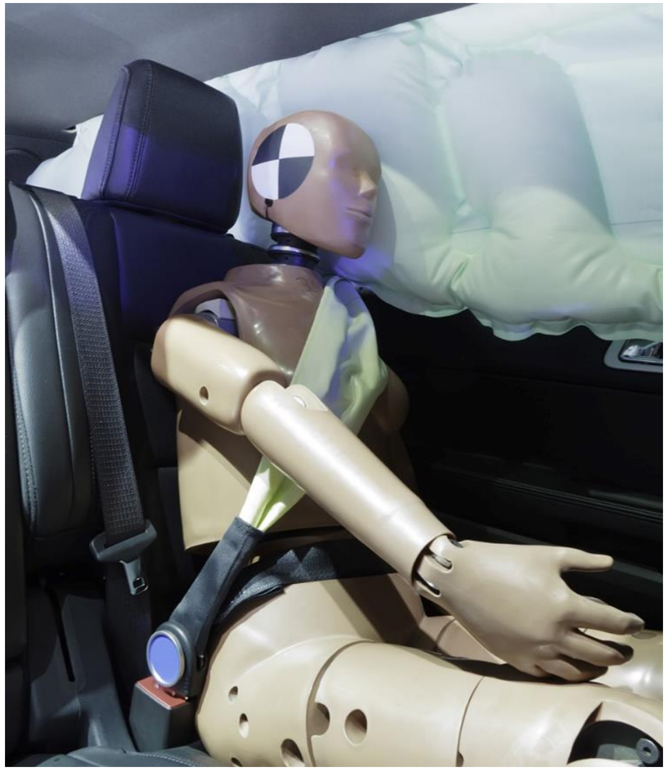
Emergency Driver Assistant
Hill Descent Control
Intelligent Speed Adaptation
Lane Departure warning system
Night Vision
Parking sensor
Pedestrian Protection System
Rain Sensor
Tire Pressure Monitor
Traffic Sign Recognition
Turning Assistant
Wrong Way Driving Warning

Autonomous Driving Software

Stages
Hands Off
Eyes Off
Mind Off



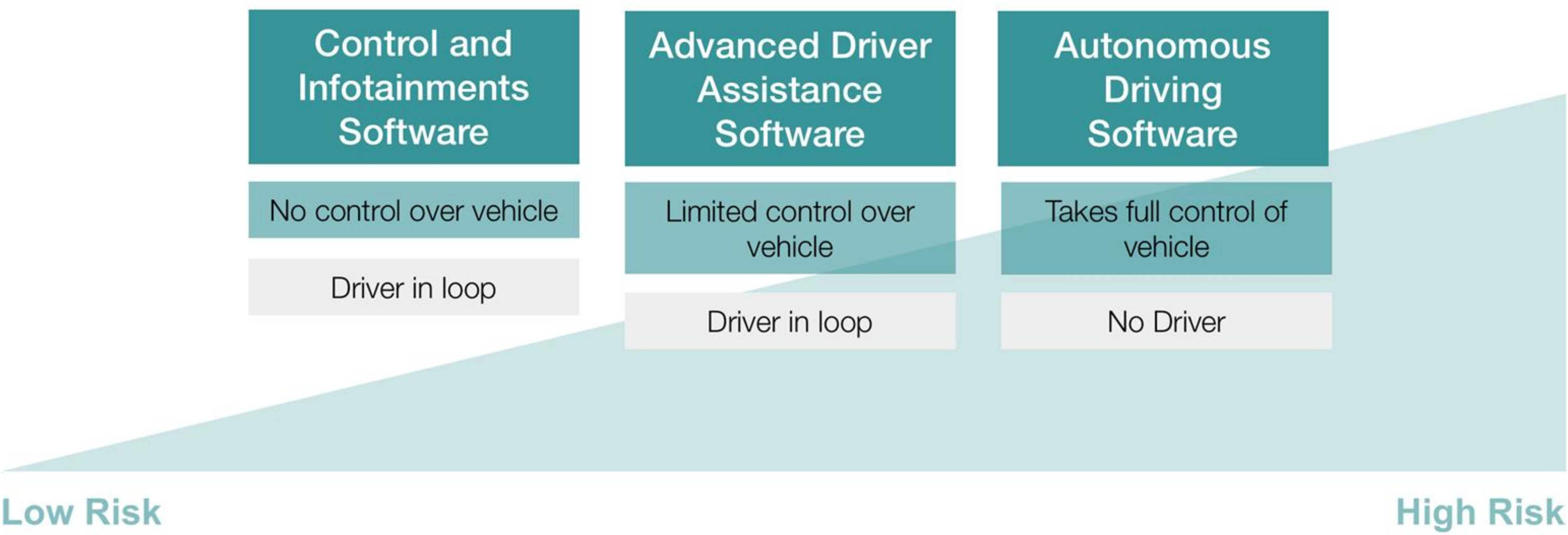
Software Enabled Vehicle & Pedestrian Safety



"Our vision is that by 2020 no one should be killed or seriously injured in a new Volvo car."

Håkan Samuelsson, President and CEO, Volvo Cars

Overview of Automotive Software



Control and Infotainments Software

No control over vehicle

Driver in loop

Advanced Driver Assistance Software

Limited control over vehicle

Driver in loop

Autonomous Driving Software

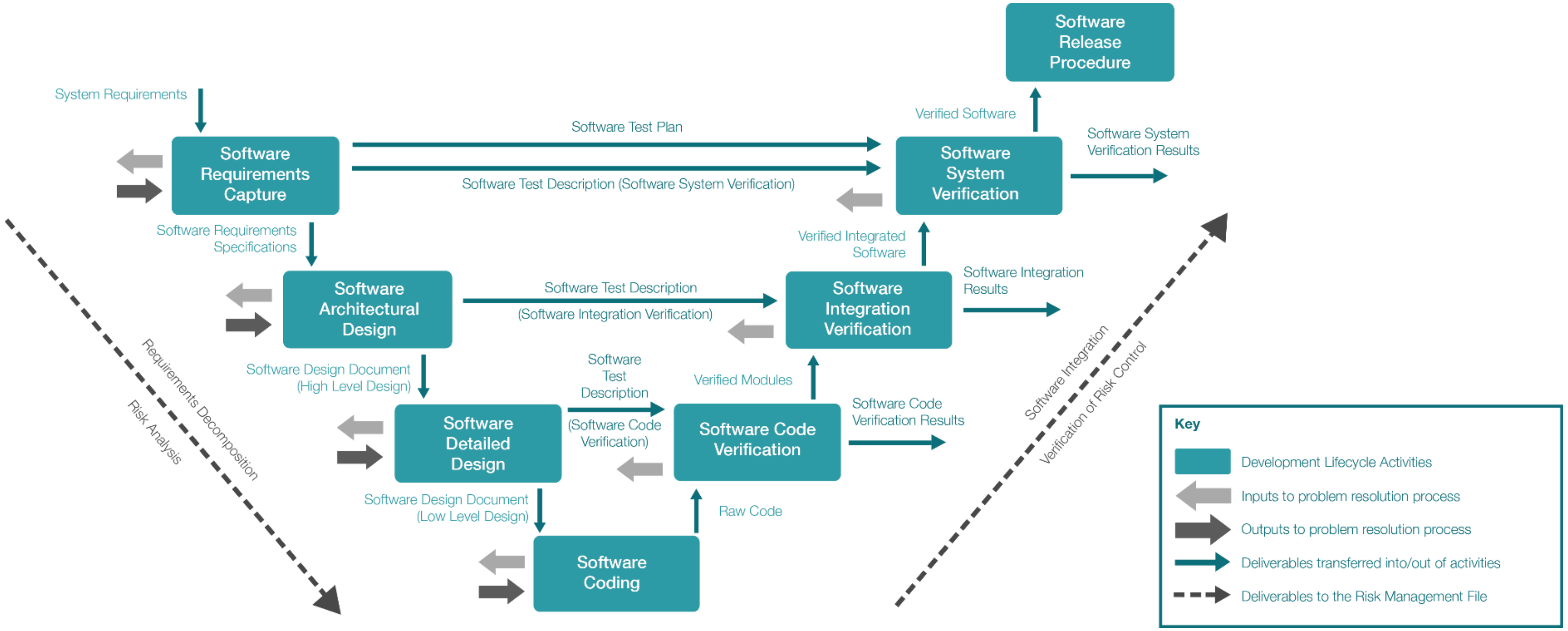
Takes full control of vehicle

No Driver

Automotive Safety Software Development

ISO 26262 has 9 parts covering:

- Management of functional safety
- System Development
- Hardware Development
- Software Development
- Production and Operation



System Requirements

Software Requirements Capture

Software Test Plan

Software Test Description (Software System Verification)

Software System Verification

Verified Software

Software Release Procedure

Software System Verification Results

Software Requirements Specifications

Software Architectural Design

Software Test Description

(Software Integration Verification)

Software Integration Verification

Verified Integrated Software

Software Integration Results

Software Design Document (High Level Design)

Software Detailed Design

Software Test Description

(Software Code Verification)

Software Code Verification

Verified Modules

Software Code Verification Results

Software Design Document (Low Level Design)

Software Coding

Raw Code

Requirements Decomposition

Risk Analysis

Software Integration

Verification of Risk Control

Key

- 🟦 Development Lifecycle Activities
- ⇦ Inputs to problem resolution process
- ➡ Outputs to problem resolution process
- → Deliverables transferred into/out of activities
- ⇢ Deliverables to the Risk Management File

ISO 26262 ASIL D is the highest possible safety rating, with ASIL A being for the Lowest Risk.

Automotive Self Monitoring Software

ISO 26262 6.4.1 The software safety requirements shall address each software-based function whose failure could lead to a violation of a technical safety requirement allocated to software.

NOTE 1 *These include both the self-monitoring of the software in the operating system and application-specific self monitoring of the software to detect, indicate and handle systematic faults in the application software.*

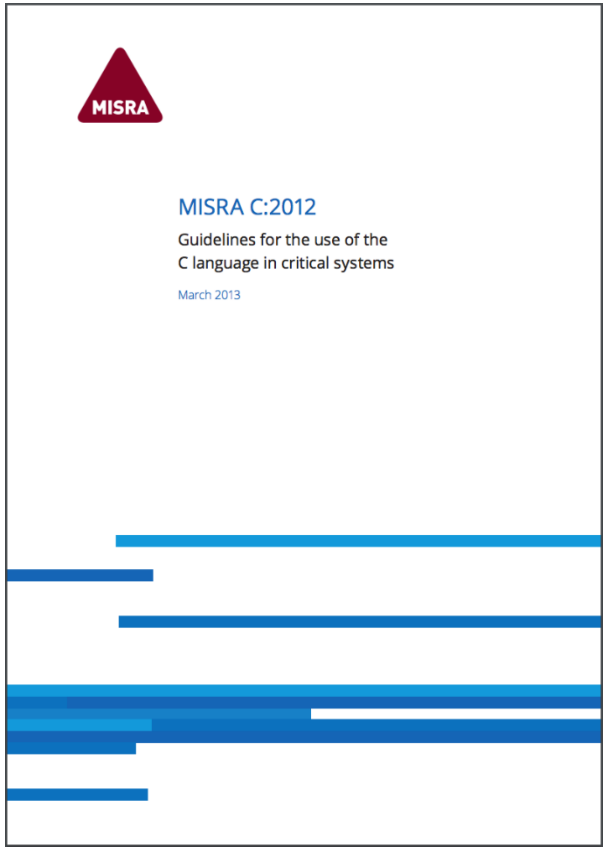
NOTE 2 *On-board tests can be carried out by the system itself or through other systems within the vehicle network during operation and during the pre-run and post-run phase of the vehicle.*

Automotive Coding Standards

MISRA C: a set of software development guidelines for the **C** programming language.

MISRA C facilitates:

- Code safety
- Code security
- Code Portability
- Code Reliability



MISRA

MISRA C:2012

Guidelines for the use of the
C language in critical systems

March 2013

No Safety Without Security

Software Security Design Methods

- Verified Boot
- Authentication
- Public/Private
- Encryption/Decryption
- Isolation

Mixing Safety and Security Software?

- Safety – Long design/test cycles, Once proven rarely updated
- Structured environment
- Security – Frequently updated to address new attack vectors
- Changing environment

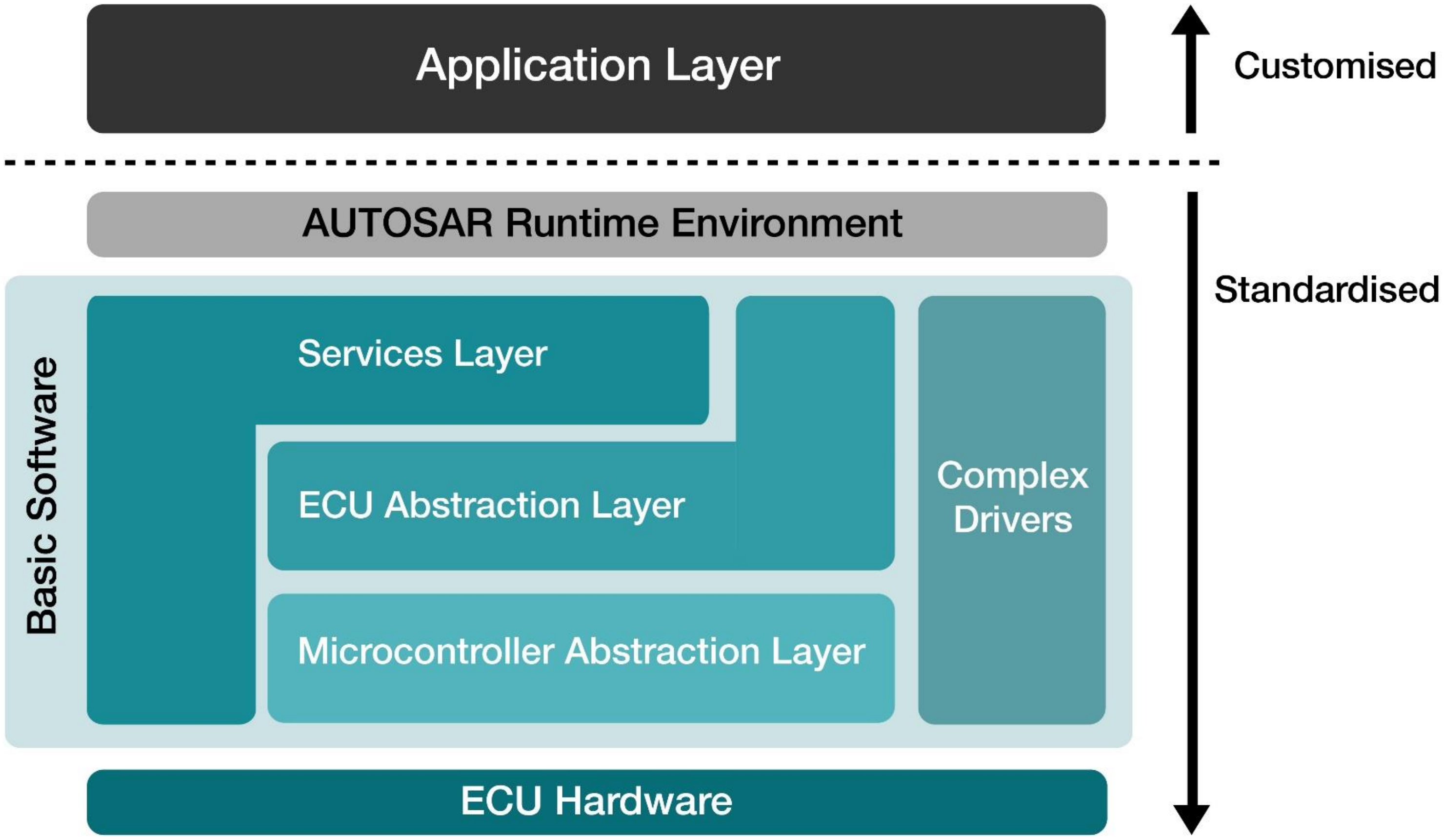
Security Standards

- Secure Hardware Extension (SHE)
- EVITA project
- PRESERVE
- Trusted Platform Module (TPM)
- Cert C Coding Standard

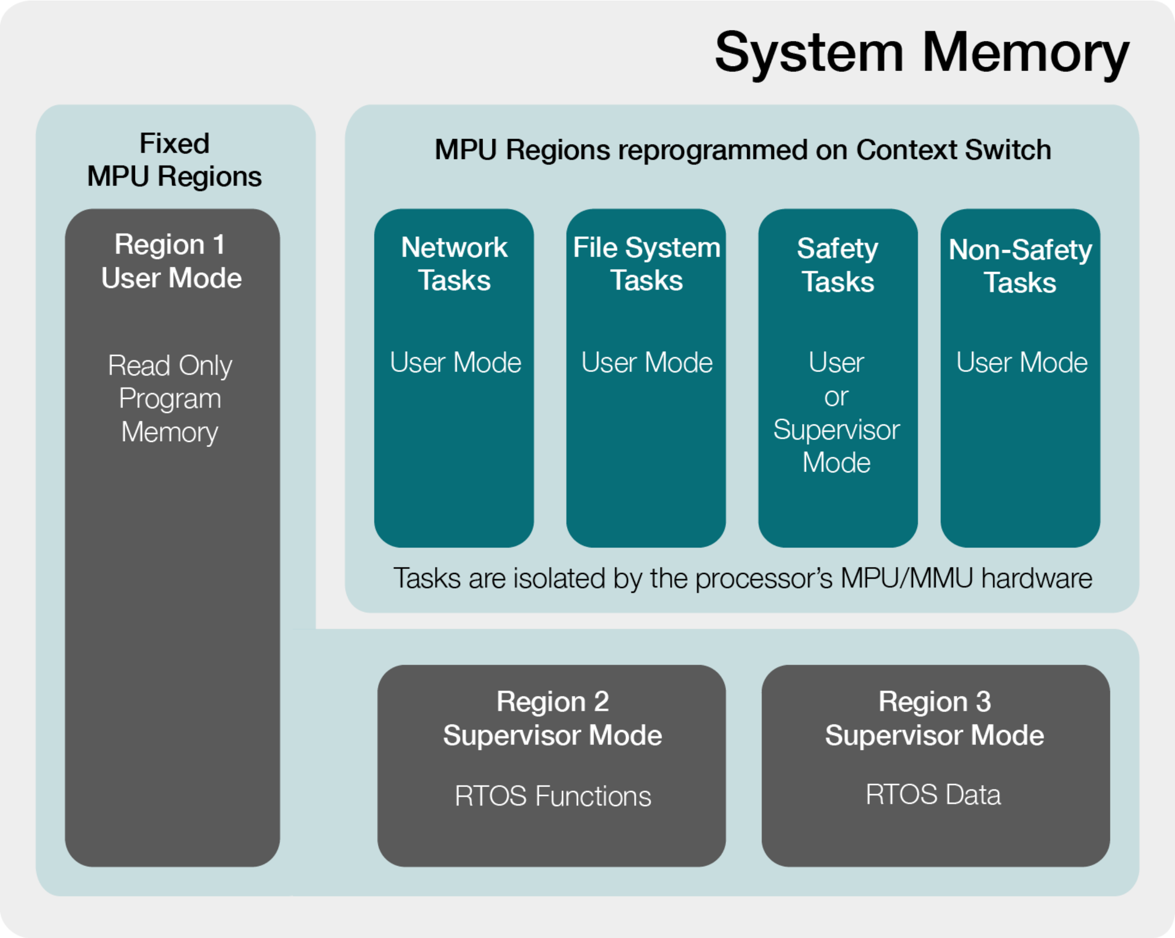
Documented Car Hacking Attack Vectors

- Cellular Network Connection
- Tire Pressure Monitoring System
- On board diagnostics

Standardised Automotive Software Architectures



Automotive Software Design Issues



System Memory

Fixed MPU Regions

Region 1
User Mode

Read Only Program Memory

MPU Regions reprogrammed on Context Switch

Network Tasks

User Mode

File System Tasks

User Mode

Safety Tasks

User or Supervisor Mode

Non-Safety Tasks

User Mode

Tasks are isolated by the processor's MPU/MMU hardware

Region 2
Supervisor Mode

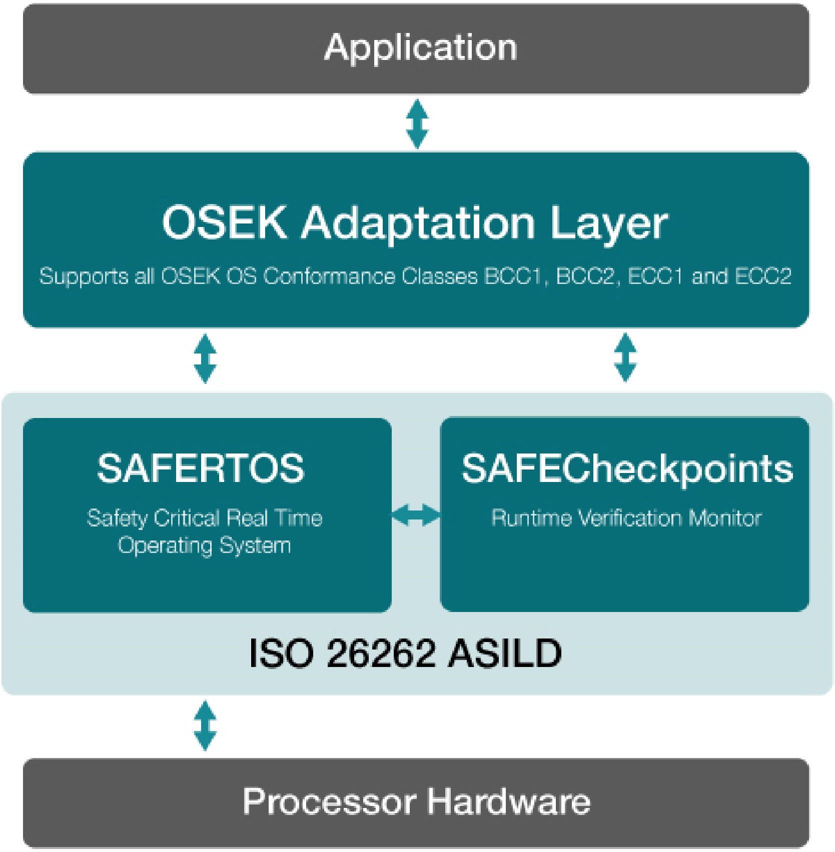
RTOS Functions

Region 3
Supervisor Mode

RTOS Data

SAFE**RTOS**®: RTOS for Automotive

- An ISO 26262 ASIL D pre-certified Real Time Operating System
- Supports the isolation and separation of tasks as standard
- Quick boot time
- Supports a sophisticated Task Monitoring plugin called SAFE**Checkpoints**
- Available with an OSEK OS wrapper
- Widely used across the automotive industry


Application
OSEK Adaptation Layer
Supports all OSEK OS Conformance Classes BCC1, BCC2, ECC1 and ECC2
SAFERTOS
Safety Critical Real Time Operating System
SAFECheckpoints
Runtime Verification Monitor
ISO 26262 ASILD
Processor Hardware