VECTOR

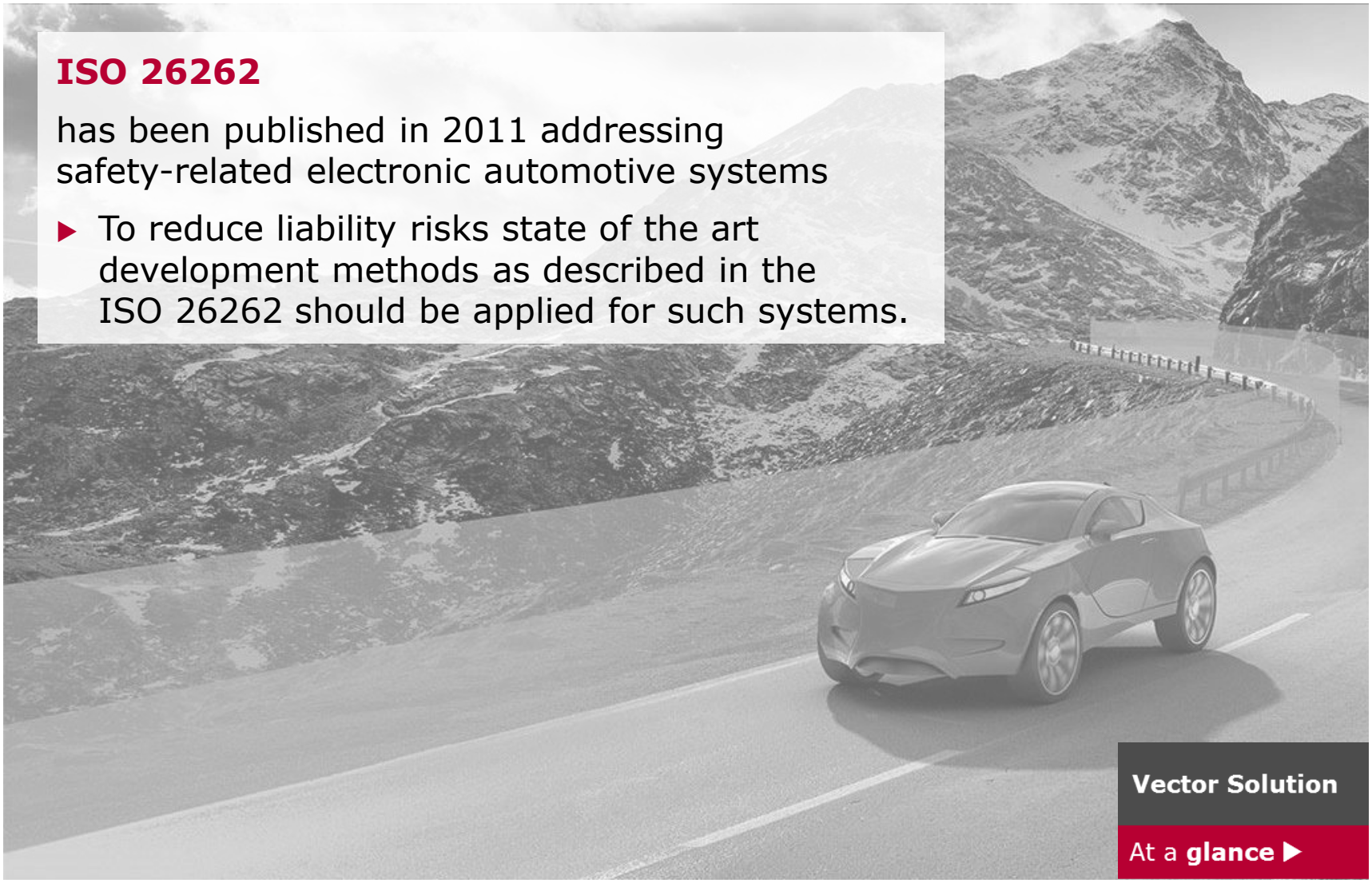
Embedded Software for Functional Safety

MICROSAR Safe

V2.02.00 | 2017-06-19

VECTOR

Functional Safety



ISO 26262

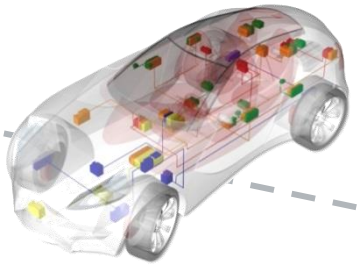
has been published in 2011 addressing safety-related electronic automotive systems

- To reduce liability risks state of the art development methods as described in the ISO 26262 should be applied for such systems.

Vector Solution

At a **glance** ▶

ISO 26262-compliant development



OEM

Item Definition

Hazard and Risk Assessment

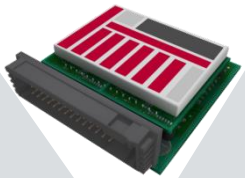
Derivation of Safety Goals

Safety Validation

Tier 1

Functional Safety Conception

Technical Safety Conception



Item Integration and Testing

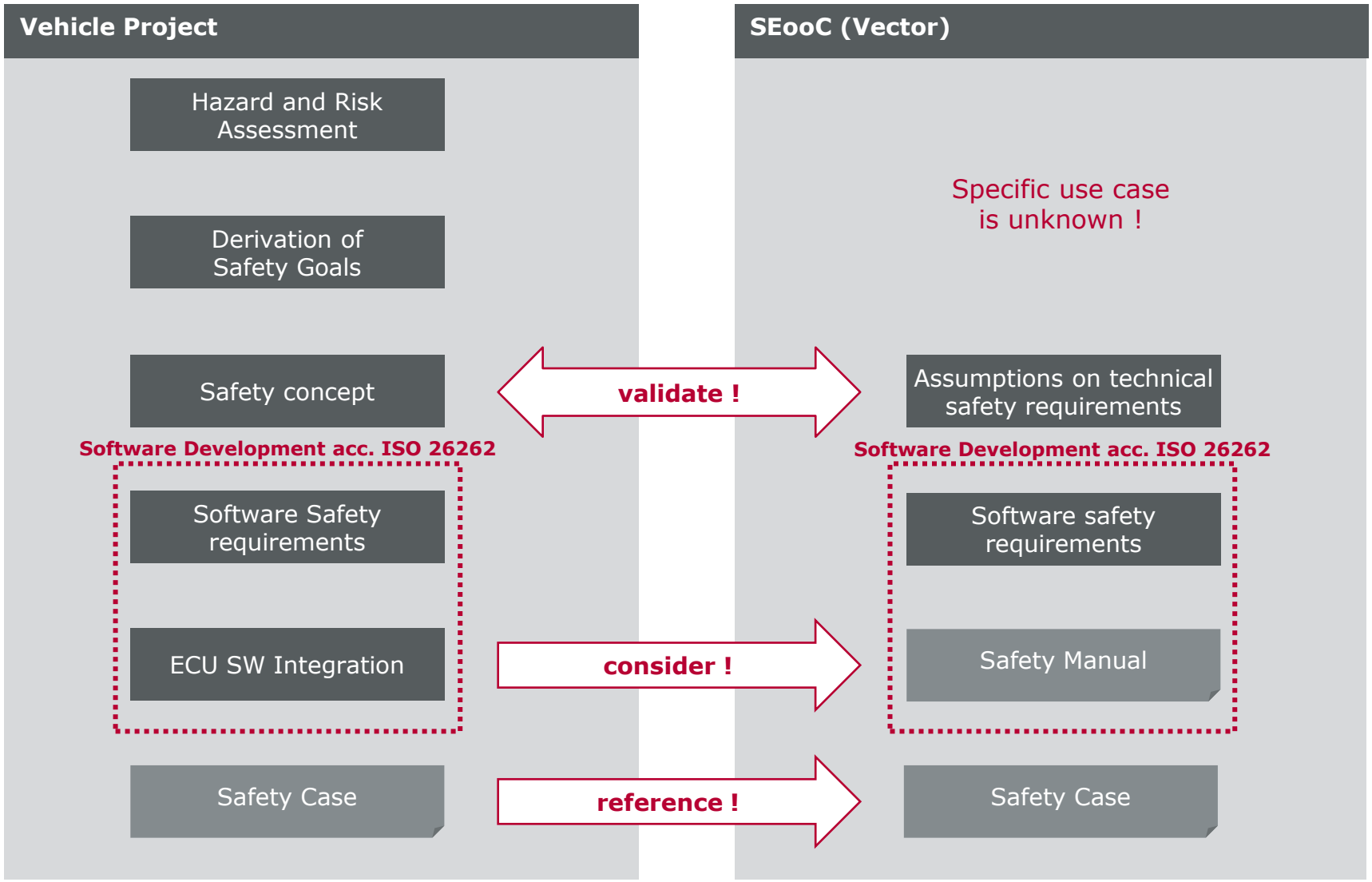
Basis SW Development

Appl. SW Development

Hardware Development

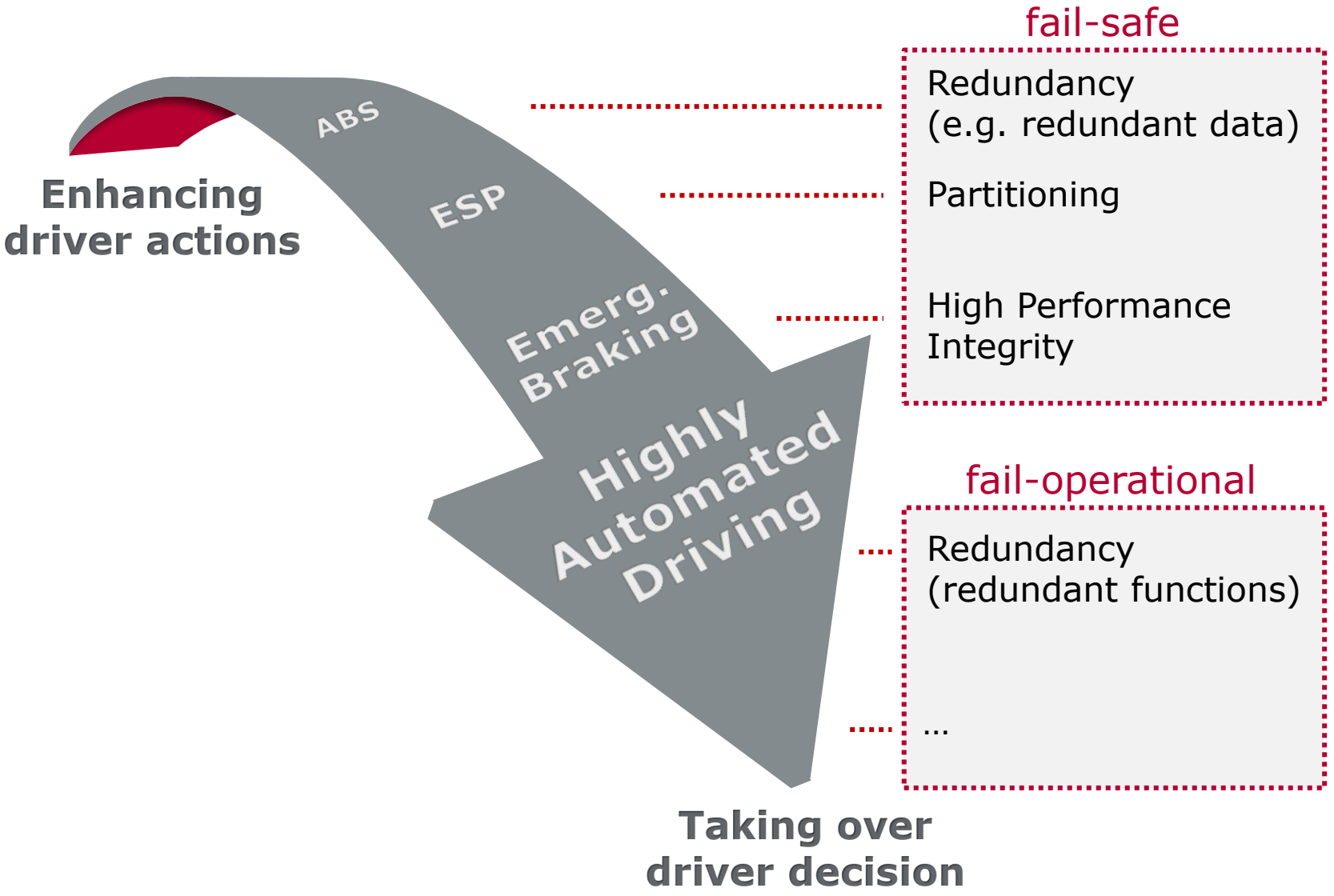
Vector

Achieving Safety Together!

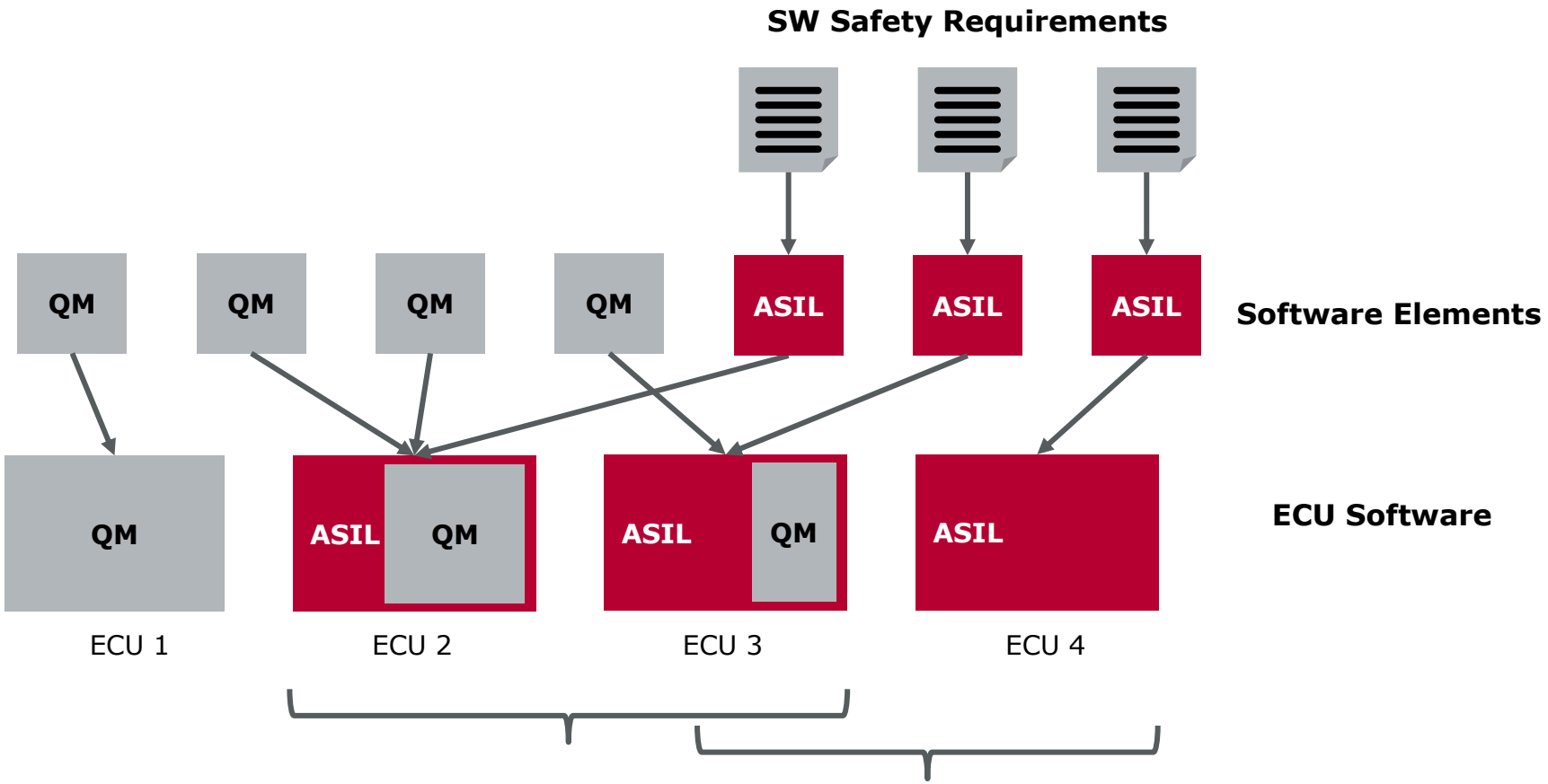


VECTOR

Evolution of Safety Concepts

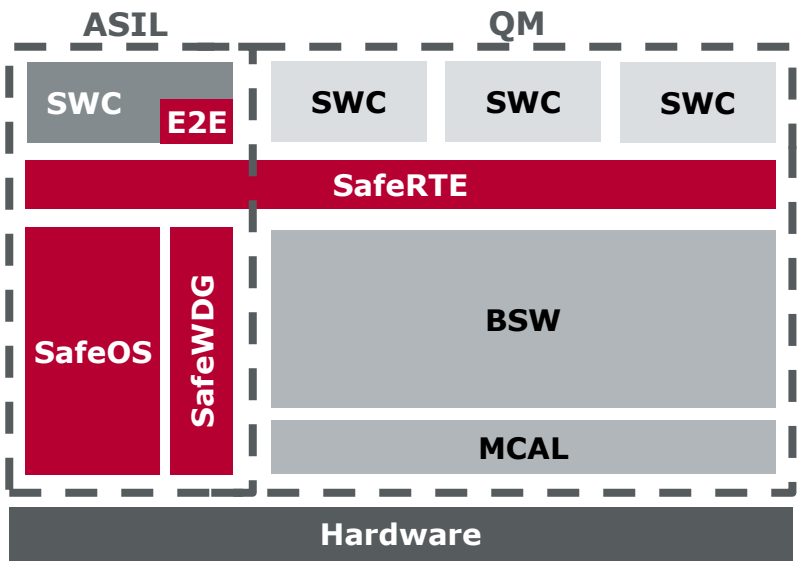


In many cases we see mixed-ASIL Systems



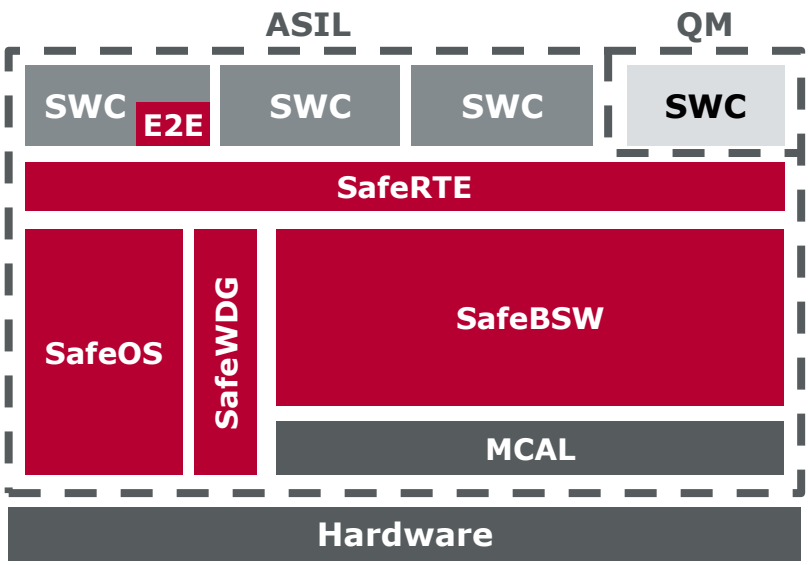
Safety Concepts and Contribution of MICROSAR Safe

Partitioning



MSR Safe Components	Rationale
▸ SafeOS ▸ SafeWDG ▸ SafeE2E ▸ SafeRTE	ISO 26262 requires that software with different ASIL in the same element do not interfere with respect to **memory**, **timing** and **communication** aspects

High Performance Integrity



MSR Safe Components	Rationale
▸ SafeOS ▸ SafeWDG ▸ SafeE2E ▸ SafeRTE ▸ **SafeBSW**	ISO 26262 requires to implement software components of different ASIL, or safety-related and non-safety-related software components, **in accordance with the highest ASIL**

Safety Building Blocks of MICROSAR Safe

MICROSAR SafeOS

- Supports memory partitioning using an MPU
- Provides safe context switch for each safety related task:
  - register settings
  - stack pointer and program counter
  - MPU settings
- Available for single- and multi-core
- OS Applications can be restarted individually

MICROSAR SafeWDG

- Detects timing and execution order faults
- Provides deadline, alive and logic monitoring
- Capable of using internal or external watchdogs as well as system basis chips (SBCs)

MICROSAR SafeE2E

- Ensures safe communication between ECUs
- Available as E2E Protection Wrapper and E2E Transformer
- All AUTOSAR profiles supported

MICROSAR SafeRTE

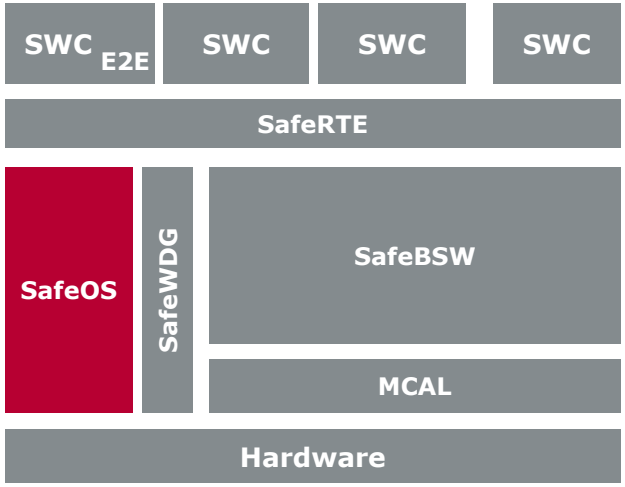
- Ensures safe communication within the ECU
- Supports safe communication across partition boundaries to exchange information between ASIL and QM applications

MICROSAR SafeBSW

- Increased performance
  - complete BSW as ASIL software
  - reduced partition switches
- Additional safety requirements, e.g.:
  - Correct initialization using an ASIL EcuM
  - Safe write/read of non-volatile data

Agenda

MICROSAR SafeOS



- Completely developed according to ASIL D
- Provides features to argue freedom from interference for application and BSW regarding
  - **Memory**: Supports memory separation using the MPU
  - **Timing**: Detection of time-budget violations
- Provides safe context switching for each safety related task:
  - register settings
  - stack pointer and program counter
  - MPU settings
- Available for single- and multi-core
- AUTOSAR 4.3 Safety Features:
  - Safe access to peripheral registers even from user mode
  - Interrupt source control API

- Features **additional to AUTOSAR OS SC3/SC4:**
  - Non-trusted function calls
  - Optimized S/R communication across different contexts

Overview SafeOS

Which faults are possible?

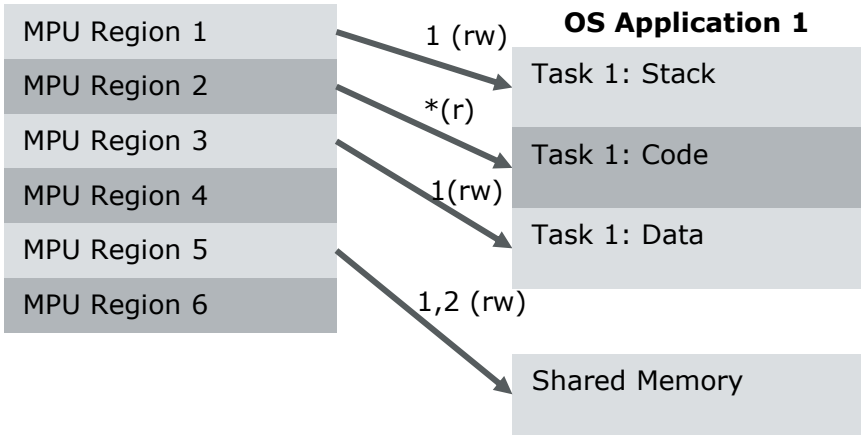
- Memory violations
  - Wild Pointers
  - Stack overflow
  - Writing outside of intended memory location of variables
- Timing violations
  - Endless loops
  - Too frequent interrupts
  - Longer calculation times due to unexpected input

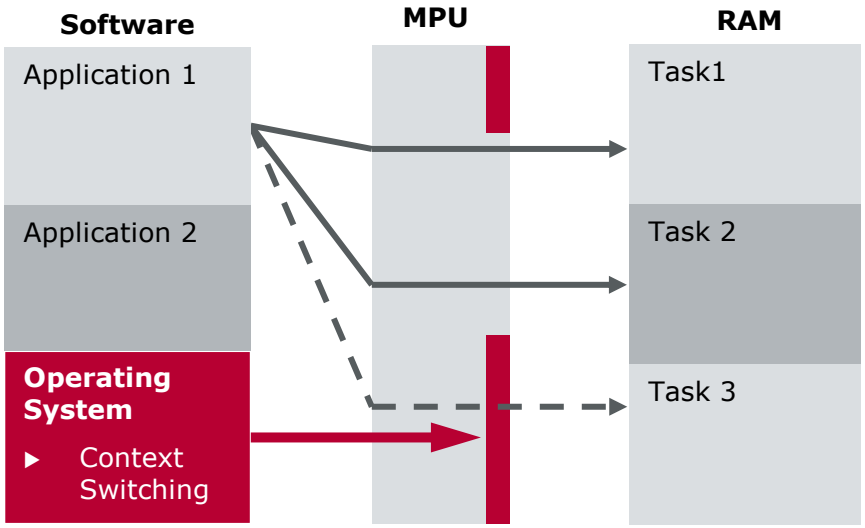
How do we address them?

- **Memory Partitioning**
  - OS Applications can be defined that are protected by the MPU
  - Protected peripheral access
- **Stack Protection**
  - Stack is separately protected by the MPU
  - Indicator values detect stack overflows also for systems without MPU
- **Timing protection**
  - Time budgets are monitored
  - Termination of applications

Partitioning

- The memory of individual OS Applications can be protected against writing by other OS Applications
- This requires dedicated hardware support (MMU/MPU)
- Reading between OS Applications is typically not restricted
- From AS 4.2 on it does not matter if the OS Applications run in supervisor mode or user mode
- OS is responsible for re-programming of the MPU if the number of available regions not sufficient.

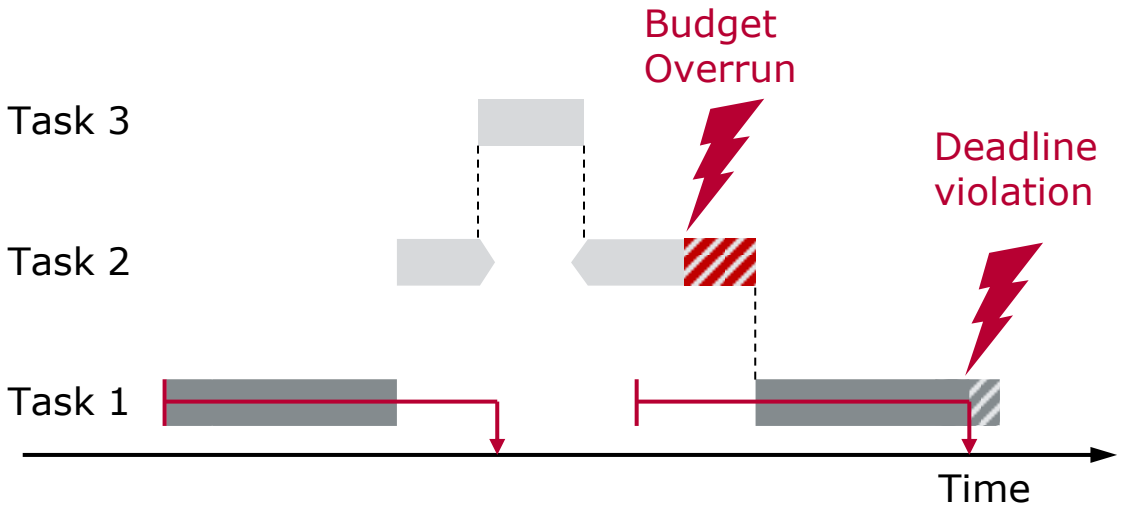




Timing Protection

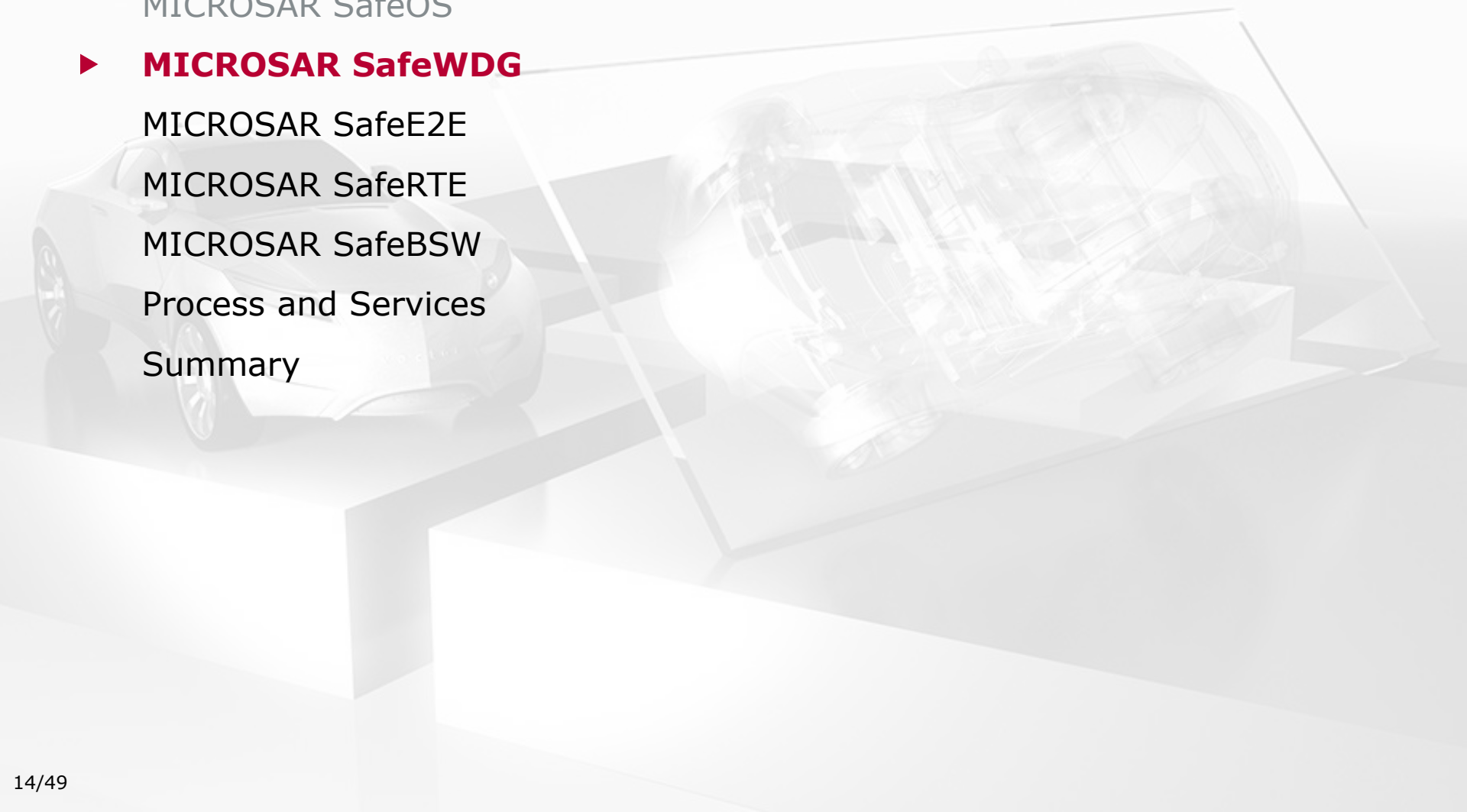
- Execution budgets are assigned to tasks and monitored
- If the budget is exceeded a protection hook is called
- Similarly, inter-arrival-times and resource locking times are monitored

- Timing Protection vs. Watchdog
  - Timing protection does not detect deadline violation!
  - Detects causes of deadline violations earlier than watchdog
  - Timing protection is limited to tasks / ISR 2 (ISRs of type 1 bypass the timing protection)

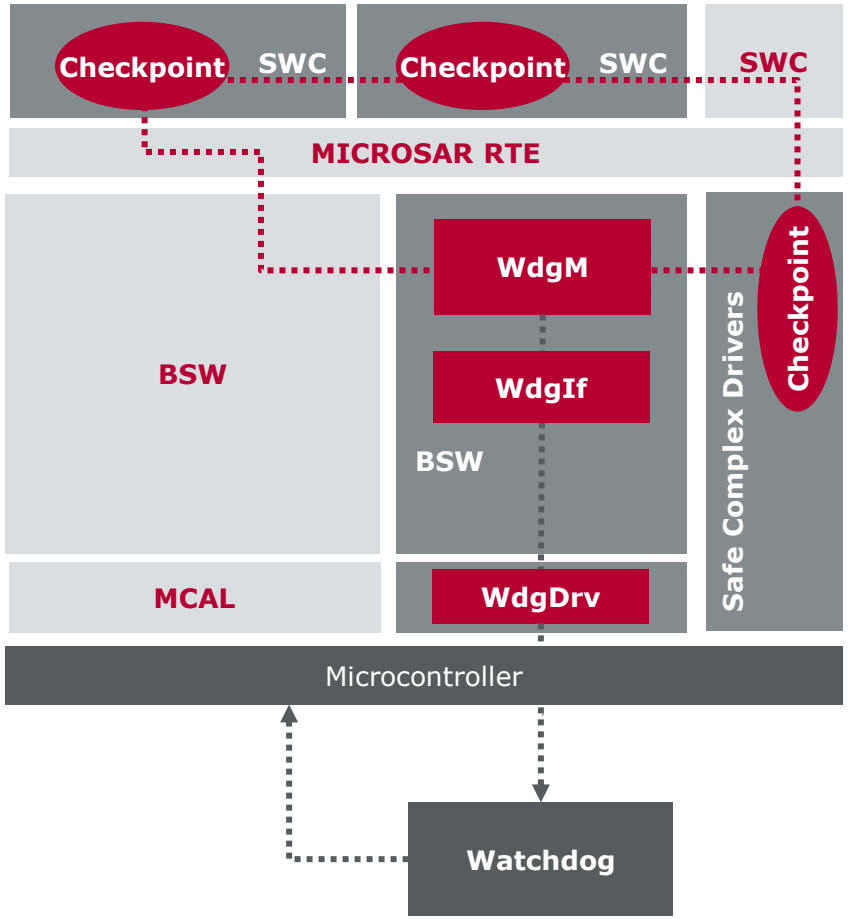


Agenda

- Introduction
- MICROSAR SafeOS
- **MICROSAR SafeWDG**
- MICROSAR SafeE2E
- MICROSAR SafeRTE
- MICROSAR SafeBSW
- Process and Services
- Summary



Overview Watchdog



- A watchdog can detect timing violations in the Application and BSW
- Also provides Program Flow Monitoring up to ASIL D
- Is supervised by independent HW-Watchdog
- On detected violations in a supervised entity the ECU can be reset
- Acknowledgements of a checkpoint are performed without context switch

VECTOR

Overview SafeWDG

Which faults are possible?

- Execution of code without request
- Code not executed although requested
- Execution of code started too early or too late
- The execution time of an code is longer or shorter than expected
- The program flow of an code differs from the expected behavior

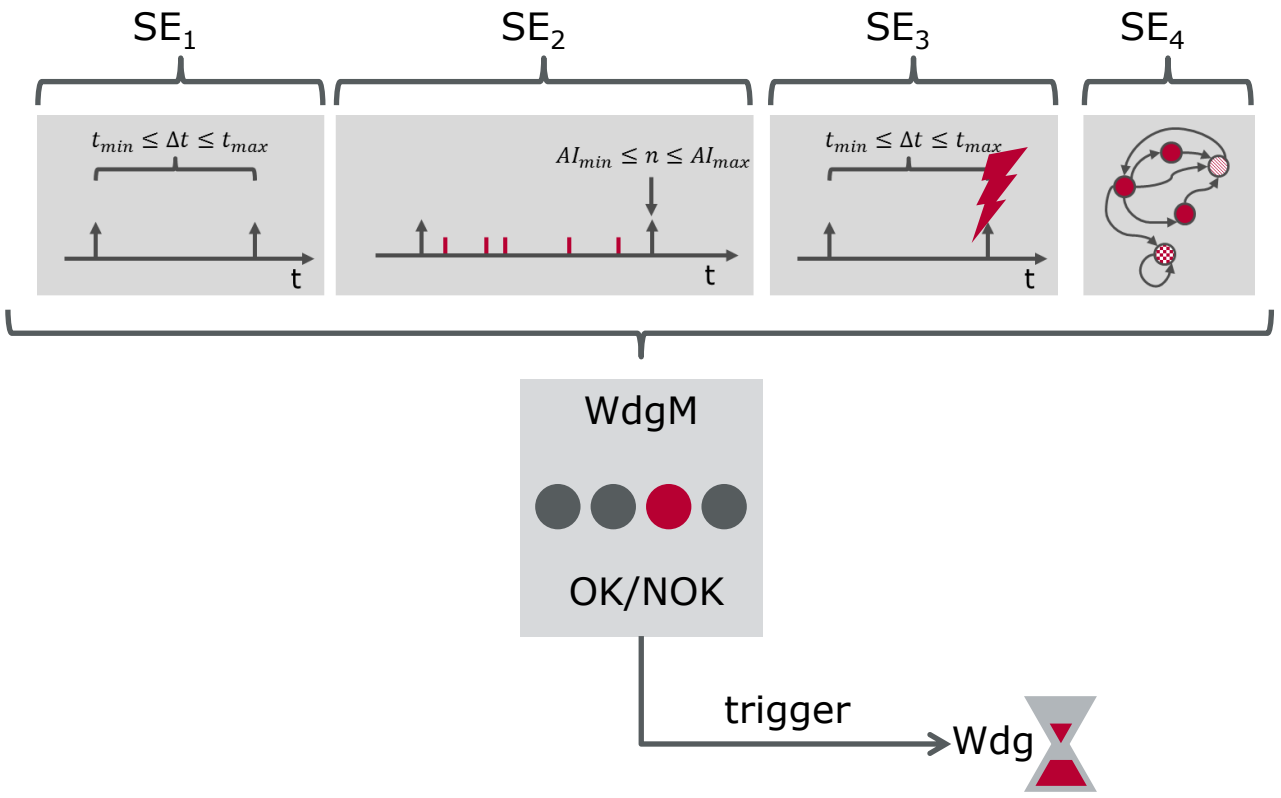
How do we address them?

- **Deadline Monitoring**
  - Applicable for aperiodic entities
  - Time between two checkpoints is compared to min/max values
- **Alive Monitoring**
  - Applicable for periodic entities
  - Number of checkpoints in interval is monitored
- **Logic Monitoring**
  - Detect wrong execution order
  - Validate checkpoint activation sequence against preconfigured execution graphs

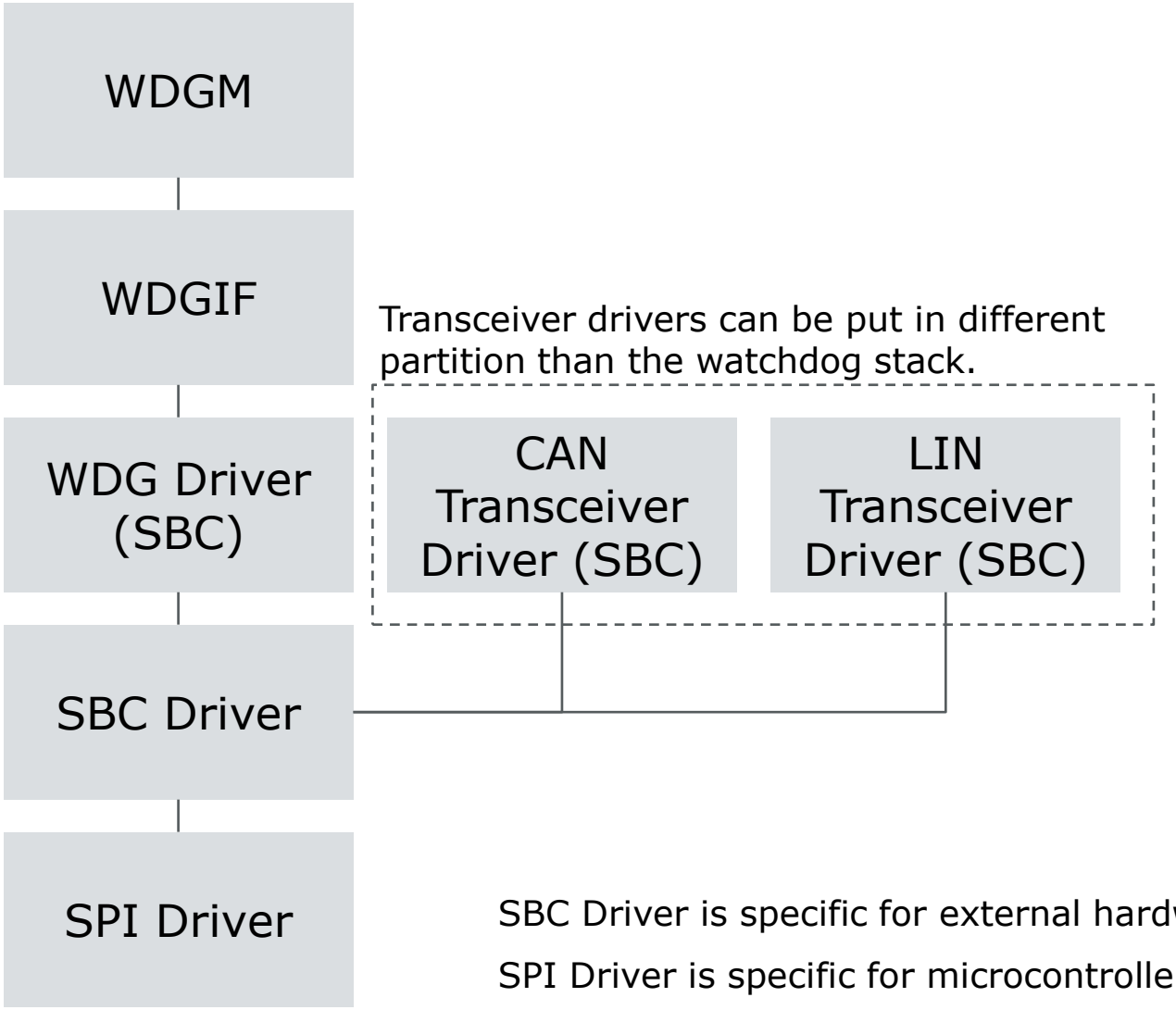
- It is assumed for all MICROSAR Safe components, that timing faults are handled using a watchdog.

Global Watchdog State

- Watchdog Manager combines all Supervised Entity states to a system state
- Depending on the system state the Watchdog Manager triggers the Watchdog Driver



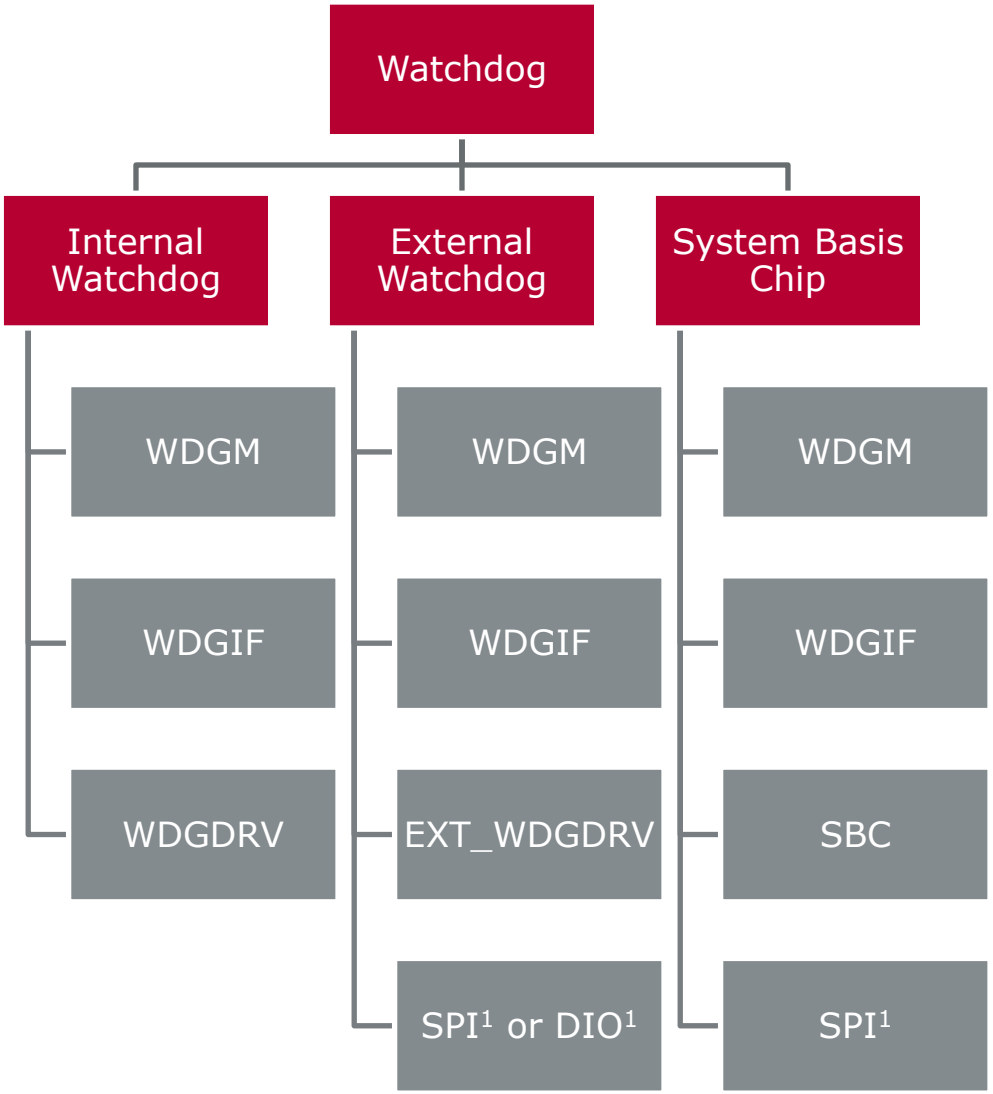
System Basis Chips in the AUTOSAR Stack



SBC Driver is specific for external hardware unit

SPI Driver is specific for microcontroller

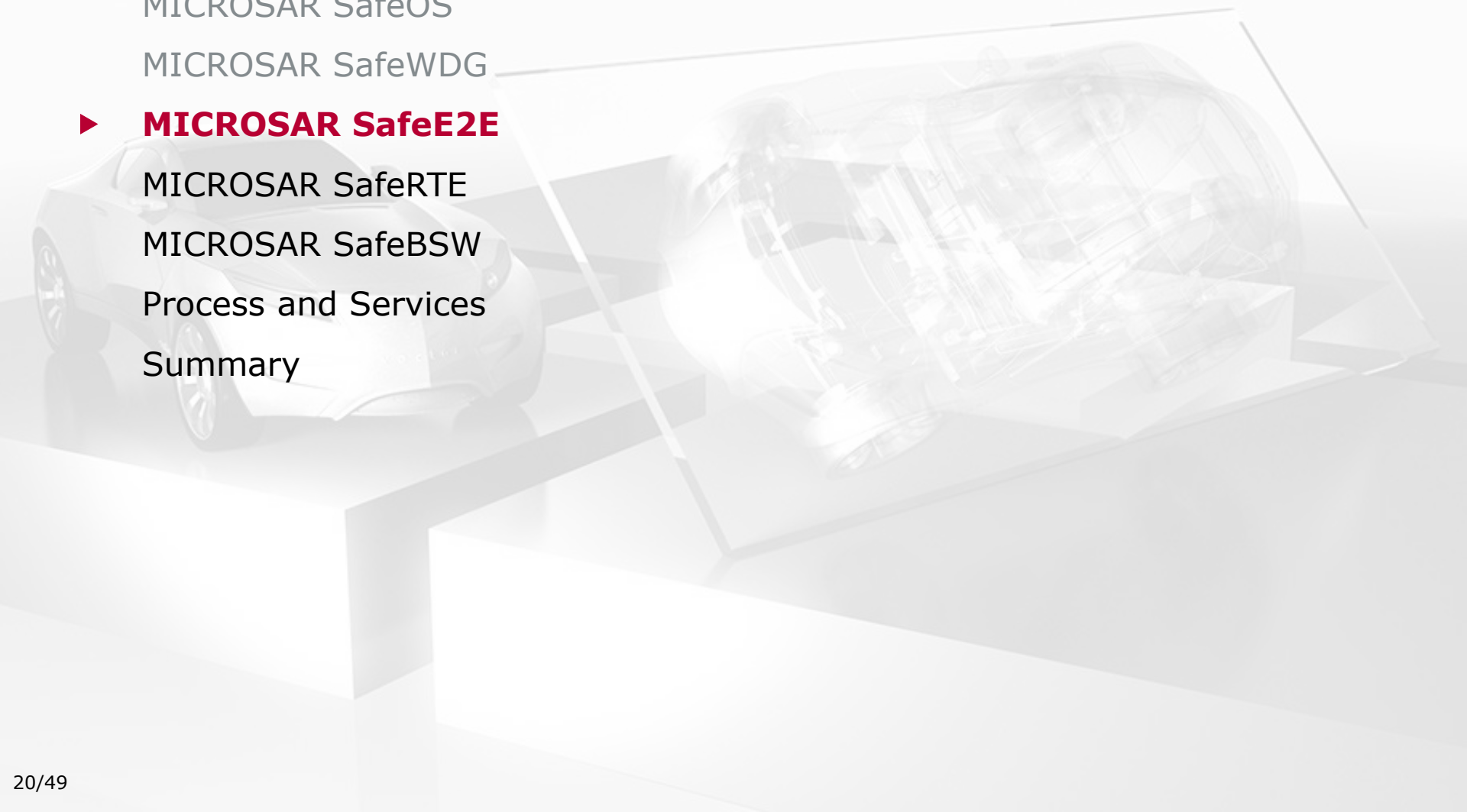
Configurations



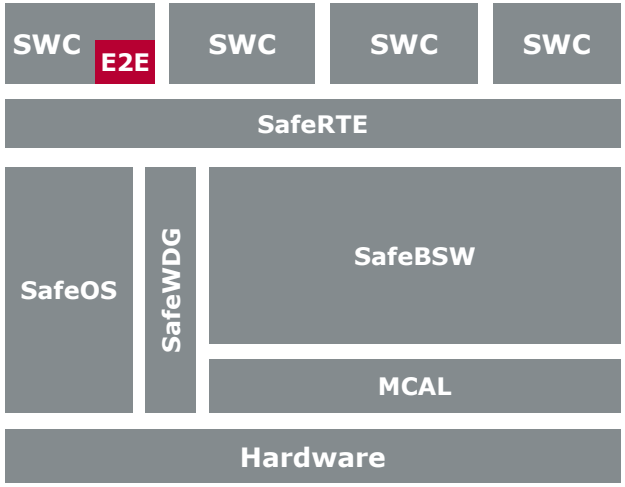
[1] Digital I/O (DIO) and Serial Peripheral Interface (SPI) drivers can be developed by Vector or used from another source (e.g. MCAL) if available with the required ASIL.

Agenda

- Introduction
- MICROSAR SafeOS
- MICROSAR SafeWDG
- **MICROSAR SafeE2E**
- MICROSAR SafeRTE
- MICROSAR SafeBSW
- Process and Services
- Summary

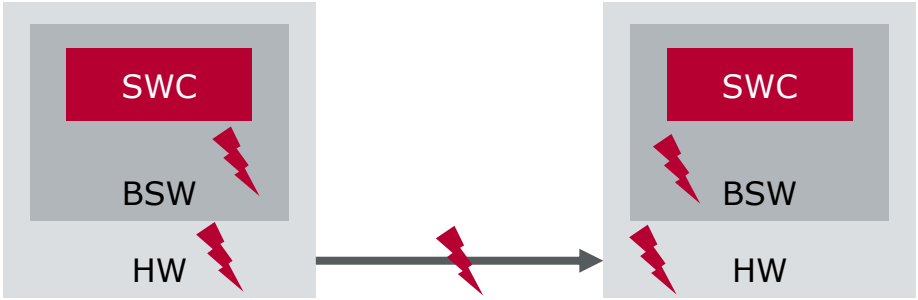


E2E Overview



- Communication from one SWC to another SWC on a different ECU over an unsafe channel
- This channel comprises:
  - RTE
  - Com-Stack
  - Bus Controller
  - Cabling

- E2E is able to **detect** if fault occurred during the transmission
- The application needs to use this information to react on the faults



VECTOR

Overview E2E

Which faults are possible?

- Failure of communication peer (even in lower software layers)
- Message masquerading
- Message corruption
- Unintended message repetition
- Insertion of messages
- Re-sequencing
- Message loss
- Message delay

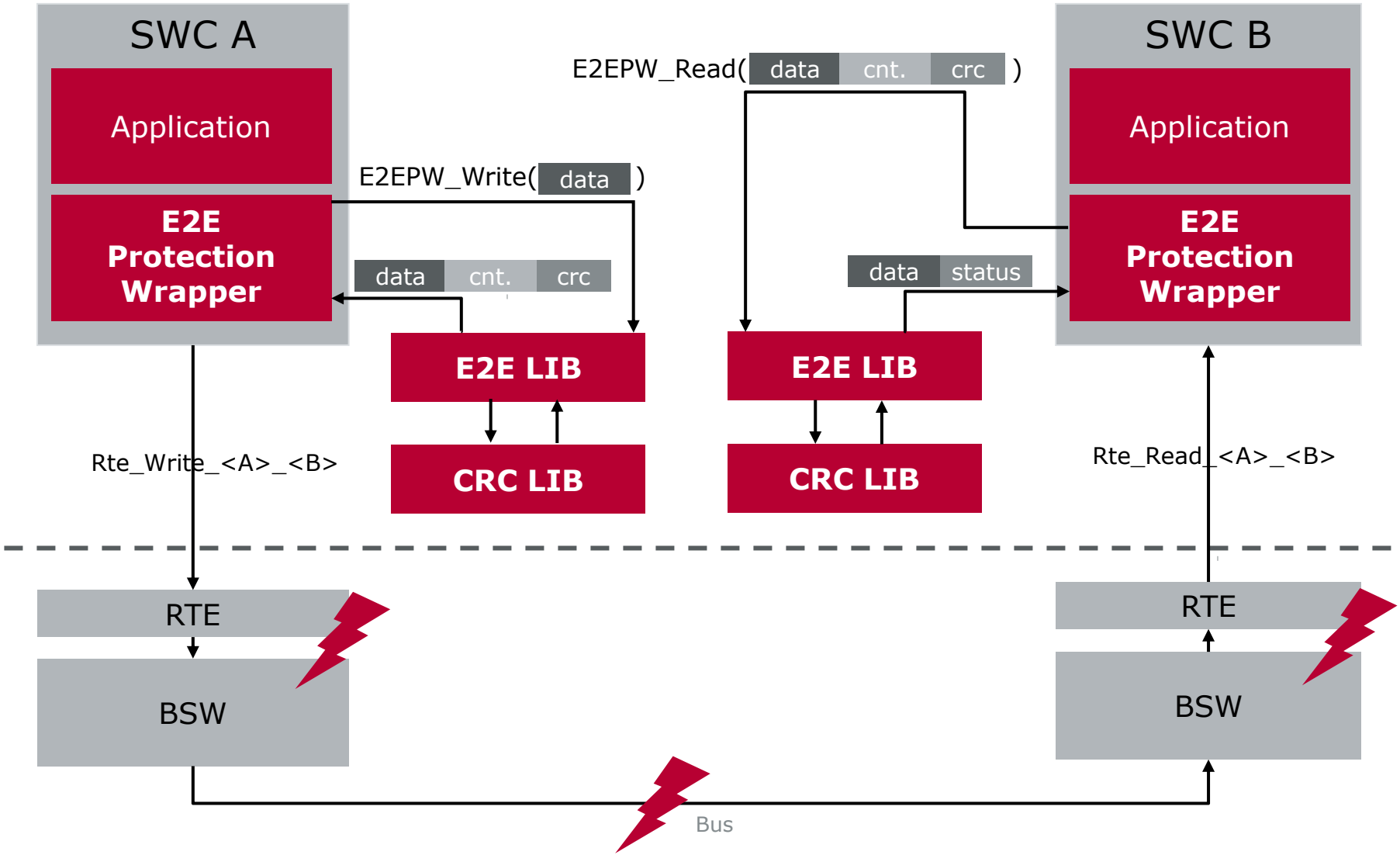
How do we address them?

- **CRC** over data, data ID and sequence counter
  - Allows to detect corruption and masquerading of the signal
- **Sequence counter**
  - Allows to detect faults in the order of messages
  - Allows to detect repeated/inserted messages
- **Timer** on receiver side
  - React on lost messages
  - React on delayed messaged

See also: ISO 26262 Part 6 D.2.4 Exchange of information

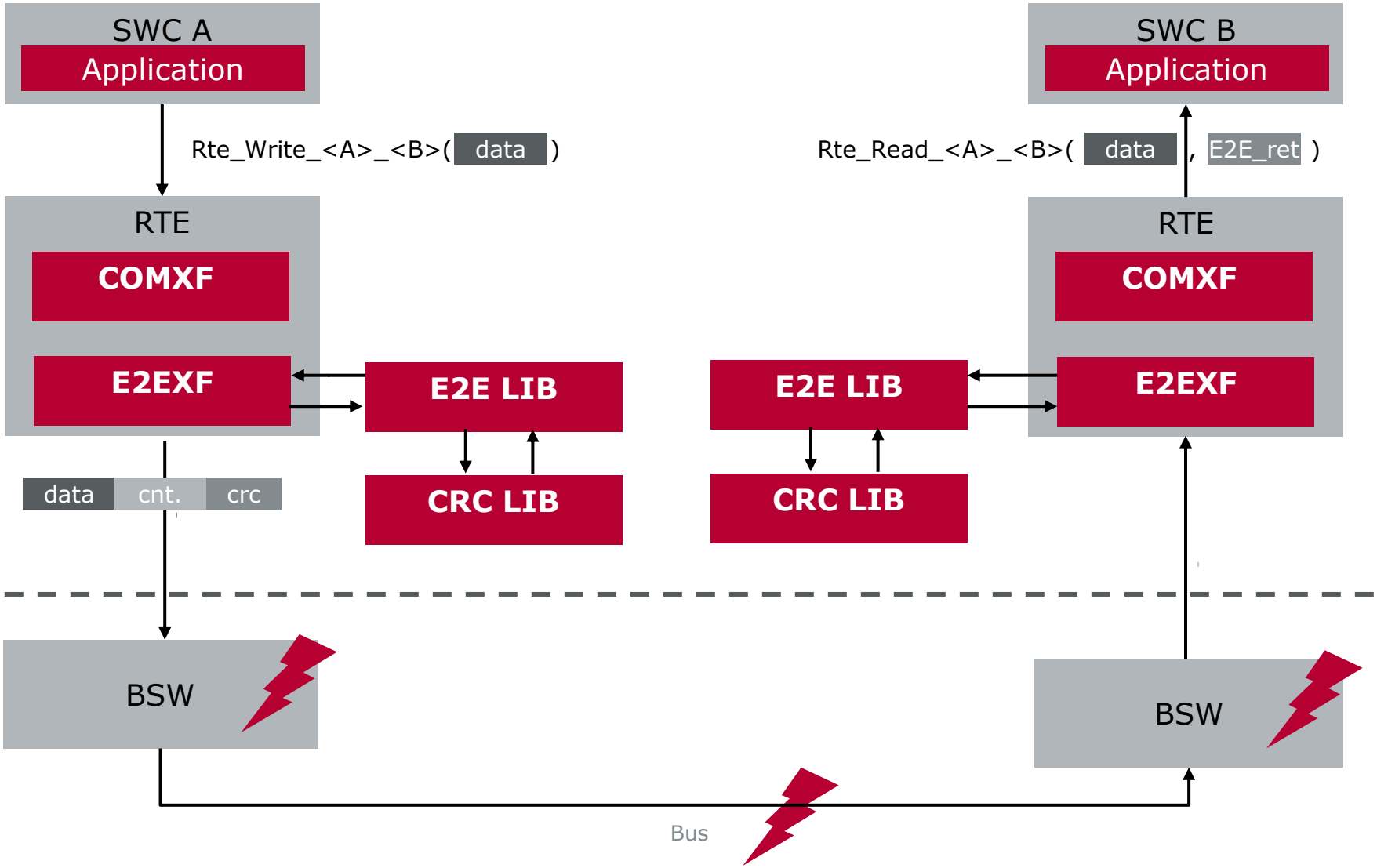
VECTOR

SafeE2E using E2E Protection Wrapper



VECTOR

SafeE2E using E2E Transformer



VECTOR

E2E Profile Overview

Profile	CRC	CRC Length	Counter	Data ID	Explicit[1]	Dynamic[2]	Msg. Length[3]
1 A[4]	0x1D	8 Bit	4 Bit	16 Bit	0	No	32 Byte
1 B[4]	0x1D	8 Bit	4 Bit	16 Bit	0	No	32 Byte
1 C	0x1D	8 Bit	4 Bit	16 Bit	4	No	32 Byte
2	0x2F	8 Bit	4 Bit	8 Bit	0	Yes	32 Byte
4	0x1F4A CFB13	32 Bit	16 Bit	32 Bit	32	No	4096 Byte
5	0x1021	16 Bit	8 Bit	16 Bit	0	No	4096 Byte
6	0x1021	16 Bit	8 Bit	16 Bit	0	No	4096 Byte
7	0x42F0E 1EBA9E A3693	64 Bit	32 Bit	32 Bit	32	No	4 MByte

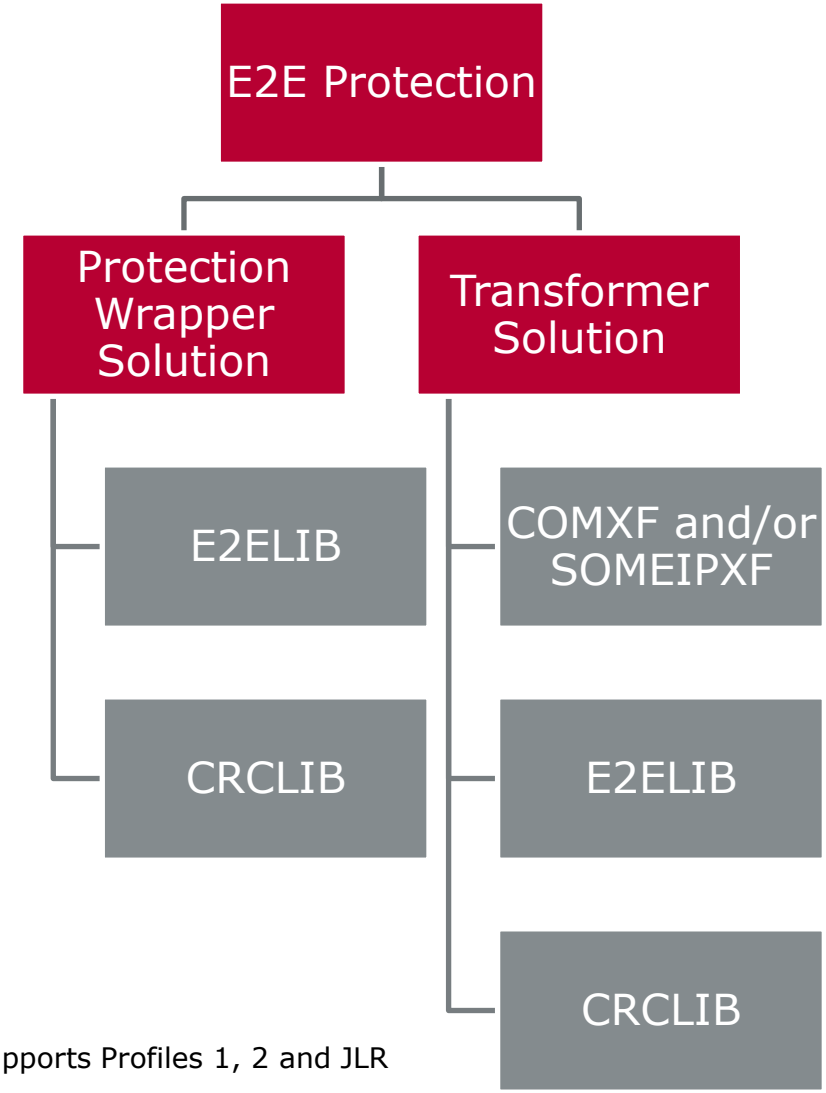
[1] How many bits of the data ID are explicitly transmitted

[2] Different data IDs for different counter values exist

[3] Maximum possible message size

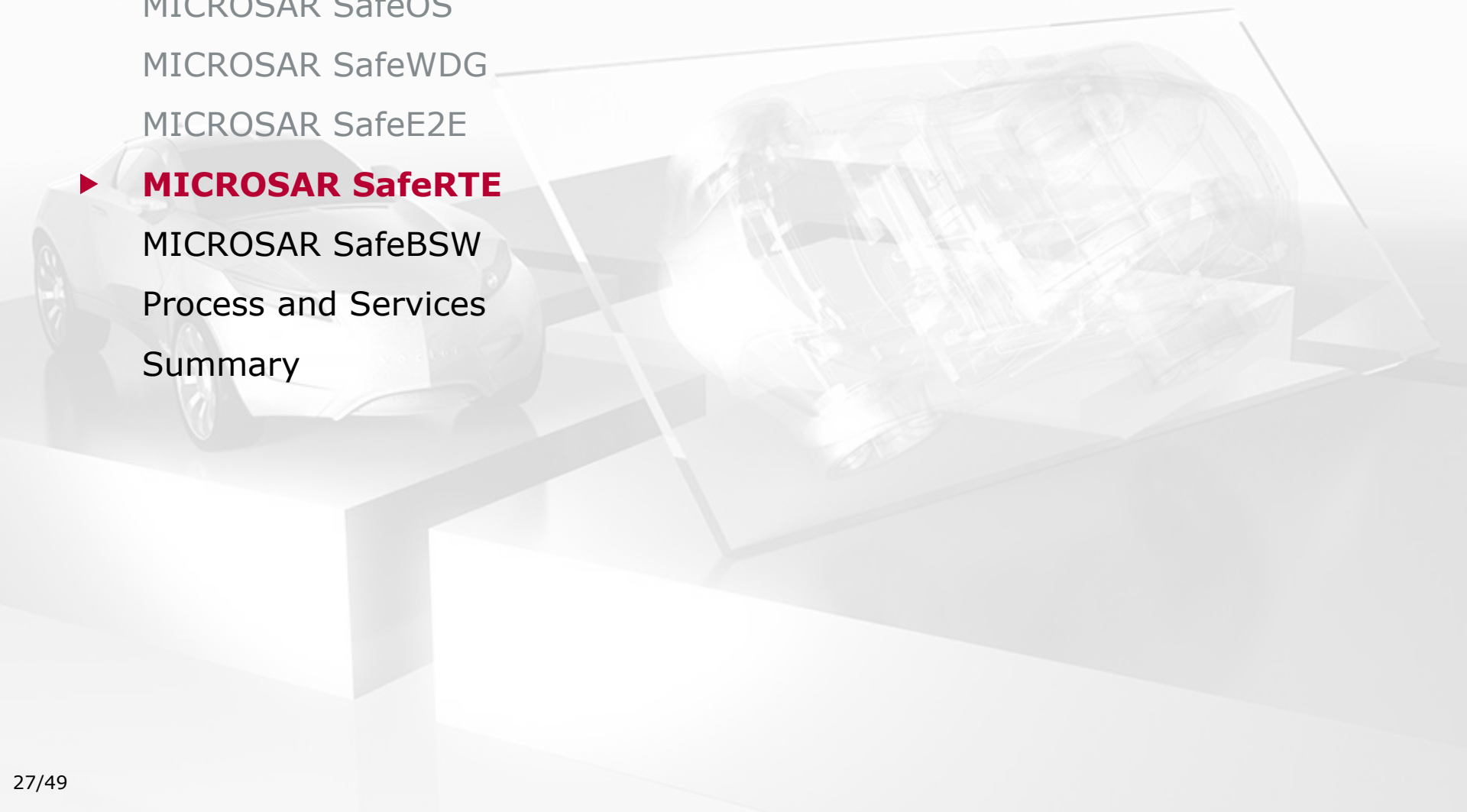
[4] Difference between 1A and 1B is the inclusion of different parts of the data ID in the CRC

Configurations



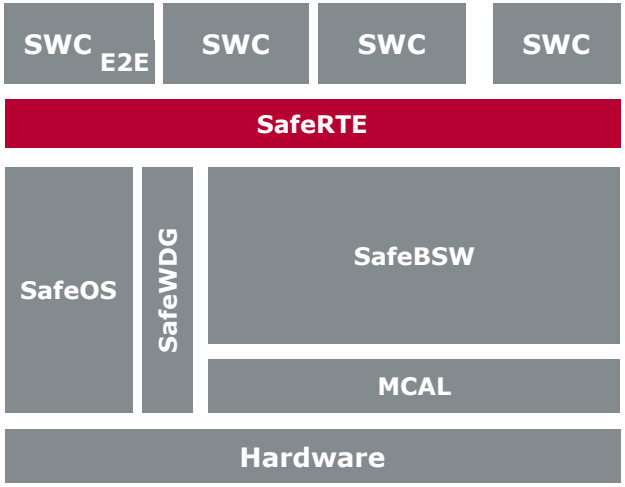
Protection Wrapper only supports Profiles 1, 2 and JLR

Agenda



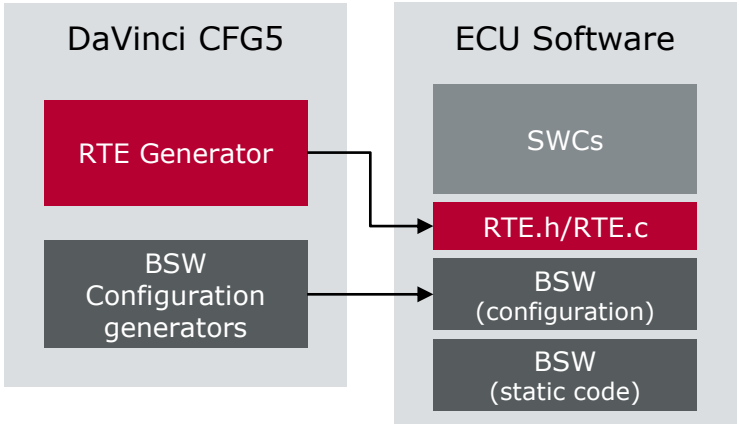
- Introduction
- MICROSAR SafeOS
- MICROSAR SafeWDG
- MICROSAR SafeE2E
- **MICROSAR SafeRTE**
- MICROSAR SafeBSW
- Process and Services
- Summary

Overview RTE

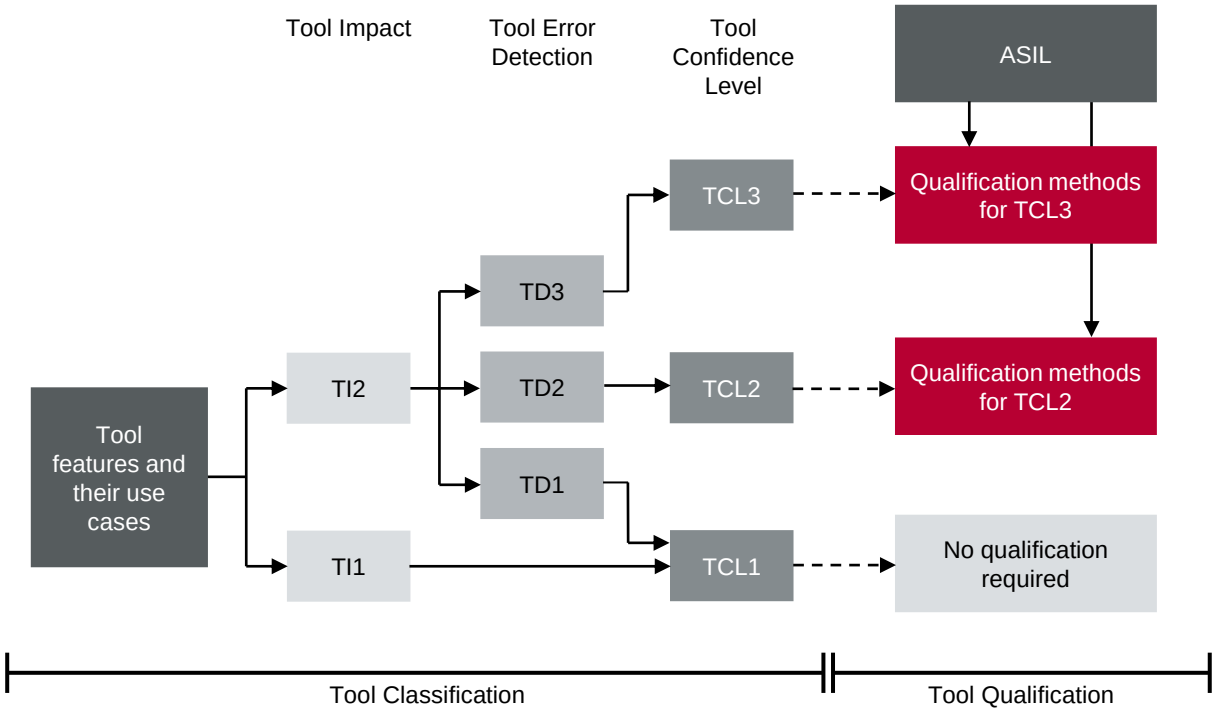


- The RTE can be used to communicate between application software components.
- The RTE can provide communication between different memory partitions and connects QM and ASIL software.
- The RTE is usually required as ASIL if it is used within the same partition as ASIL application Software Components.

- Using the RTE in safety relevant ECUs:
  - The RTE is completely generated using DaVinci Configurator PRO (CFG5) and a corresponding generator.
  - Thus, ISO 26262 Part 8 Clause 11 (Confidence in the use of software tools) applies.



Derivation of Tool Confidence Level (TCL)



Tool classification and qualification are usually performed by the user of the tool.

VECTOR

RTE vs. SafeRTE

RTE

- Classification of RTE:
  - TI2: "a malfunction of a particular software tool can introduce or fail to detect errors in a safety-related item or element being developed"
  - TD2: "there is a medium degree of confidence that a malfunction and its corresponding erroneous output will be prevented or detected"

Classification of CFG5 as TCL2 necessary

Manual qualification of CFG5 or the generated software by the user is necessary!

SafeRTE

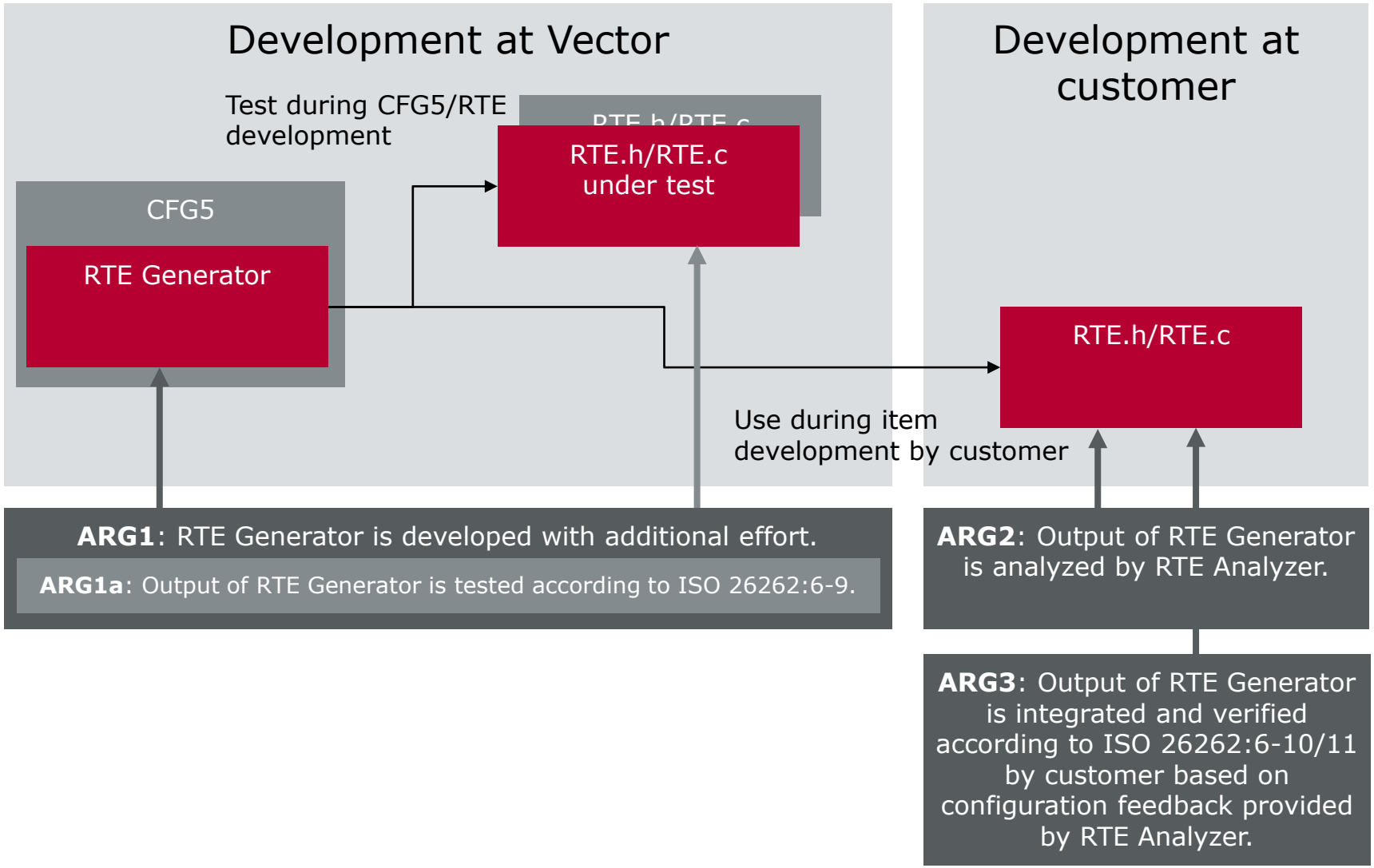
- Classification of RTE:
  - TI2: "a malfunction of a particular software tool can introduce or fail to detect errors in a safety-related item or element being developed"
  - TD1: "high degree of confidence that a malfunction and its corresponding erroneous output will be prevented or detected"

Vector provides argumentation to classify CFG5 as TCL1.

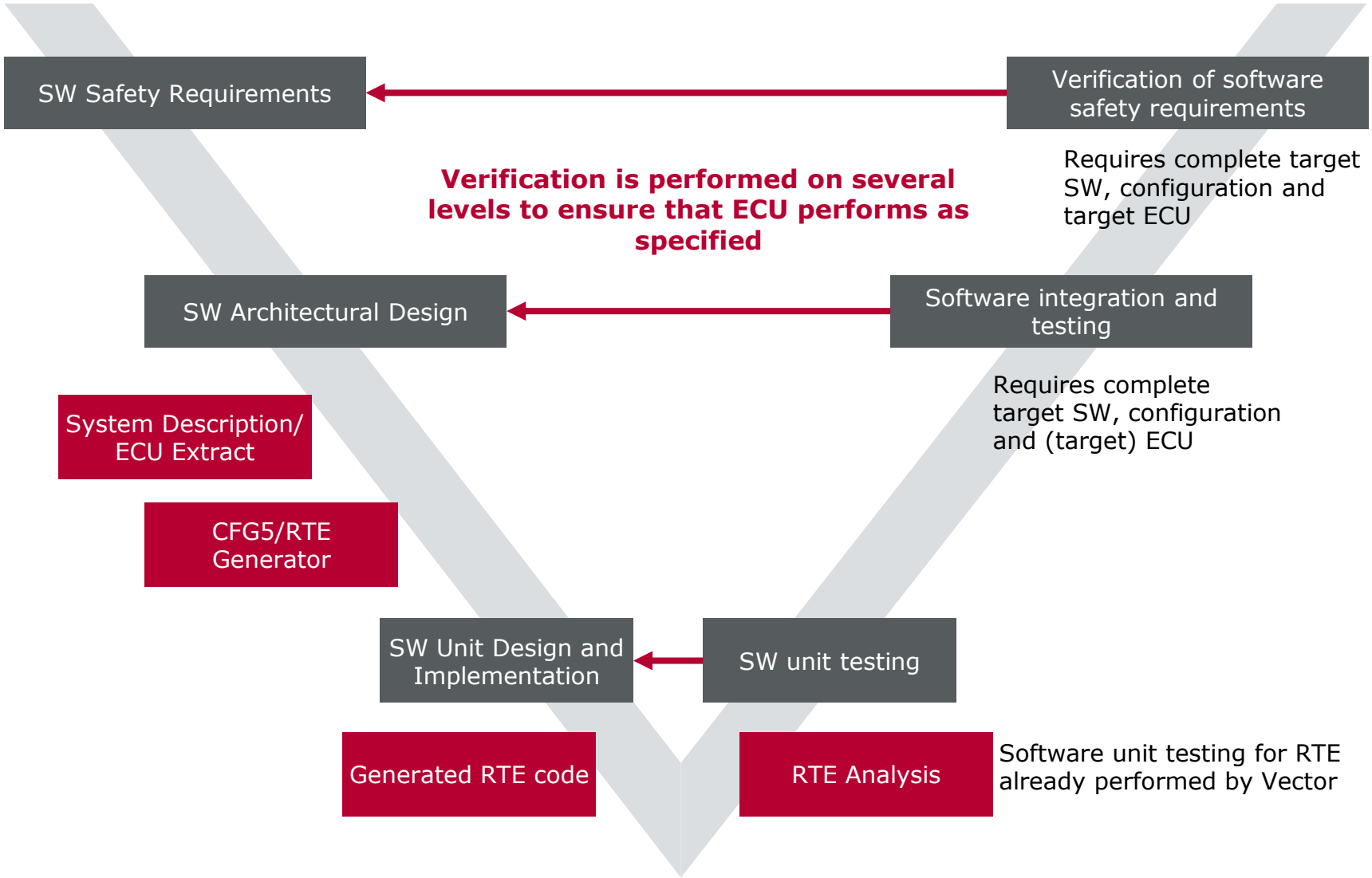
No qualification for TCL 1 tools needed!

VECTOR

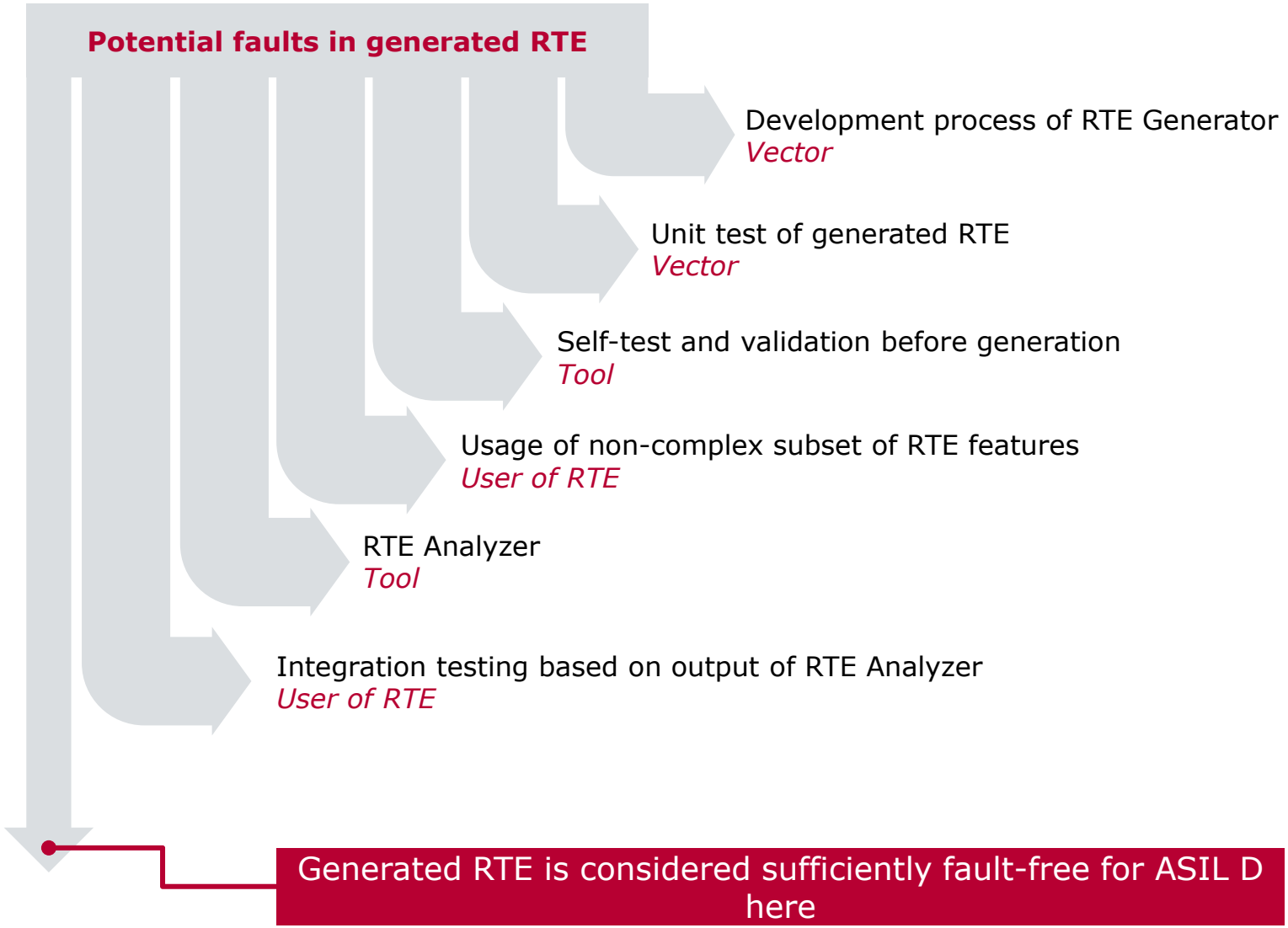
Arguments for TD1 in Detail



ISO 26262-compliant Software Development with SafeRTE



Fault detection and prevention



VECTOR

Features of the RTE Analyzer

General

- Compilation check for RTE code
- Detection of recursive call sequences
- Analysis report generation
- Configuration feedback report for guided integration testing

Out-of-bounds Access

- Detection of out-of-bounds write accesses within RTE APIs
- Detection of non type-safe interfaces to the BSW and SWCs where a call with a wrong parameter might cause out of bounds writes by the RTE or a called runnable/BSW API

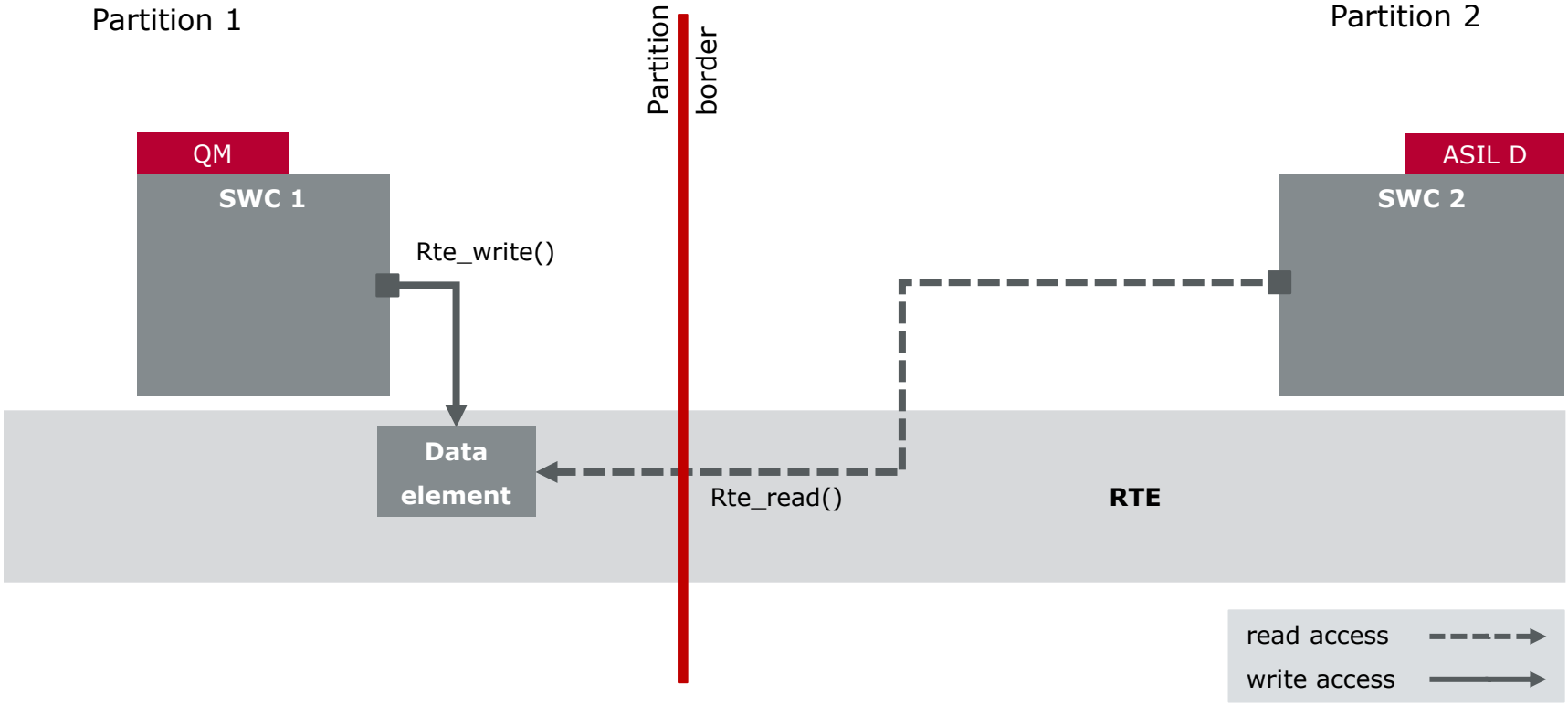
Concurrent Access

- Detection of RTE variables that are accessed from concurrent execution contexts without protection
- Detection of concurrent calls to non-reentrant APIs within the RTE
- Detection of variables that are accessed from multiple cores and that are not mapped to non-cacheable memory sections

RTE API Usage

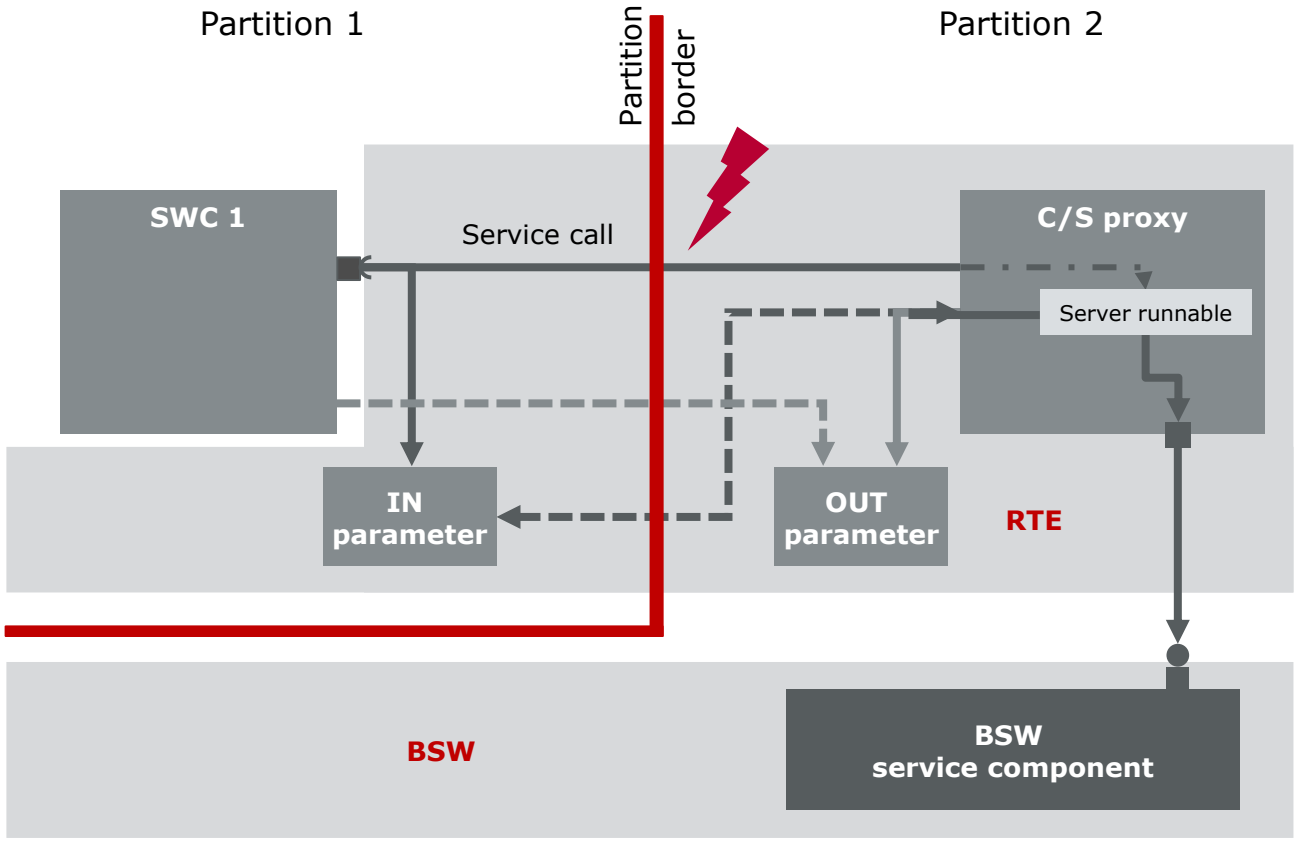
- Detection of interrupt lock API sequence mismatches within RTE APIs
- Detection of unreachable RTE APIs and runnables
- Detection of RTE APIs for which a call from a wrong context might cause data consistency problems

Example – Inter-partition Sender/Receiver Communication



- MPU restricts write access from other partitions put may allow read access
- Data is stored in own partition and event notifies availability of data
- Principle is identical for ASIL ➔ QM communication

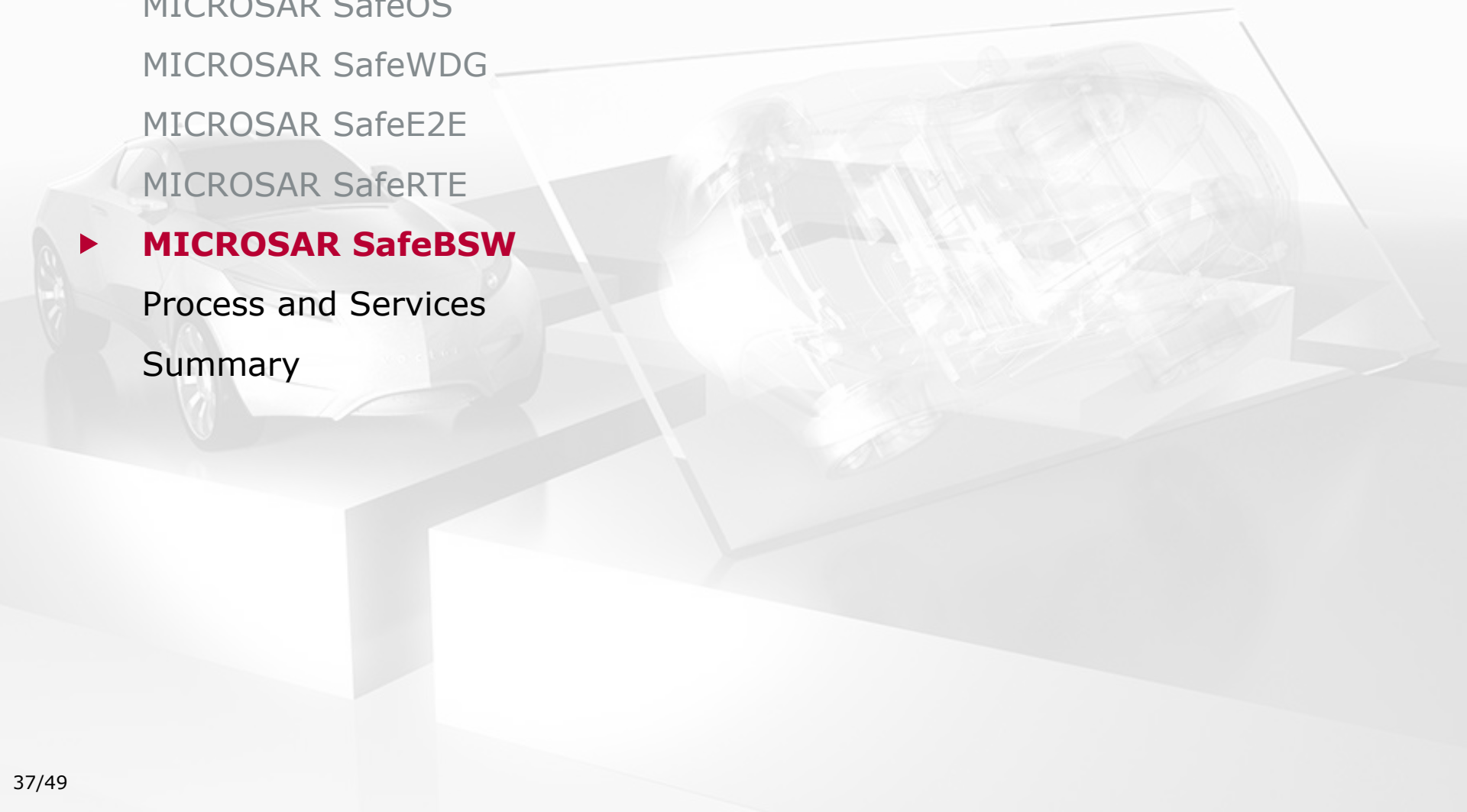
Example – Inter-partition Client/Server Communication



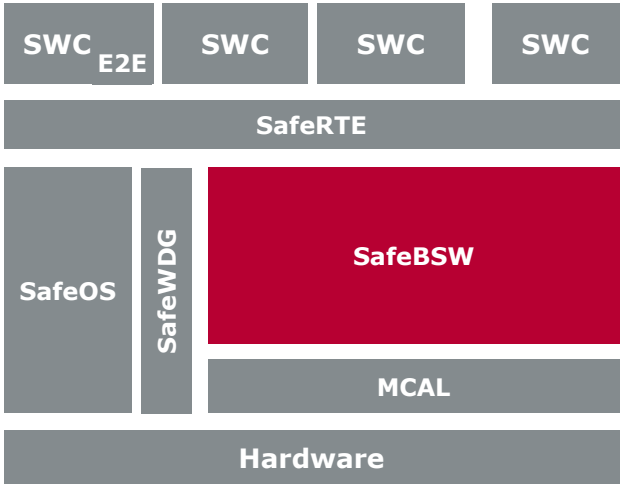
- Vector's SafeRTE automatically generates proxies to allow inter-partition client/server communication.

read access
write access

Agenda

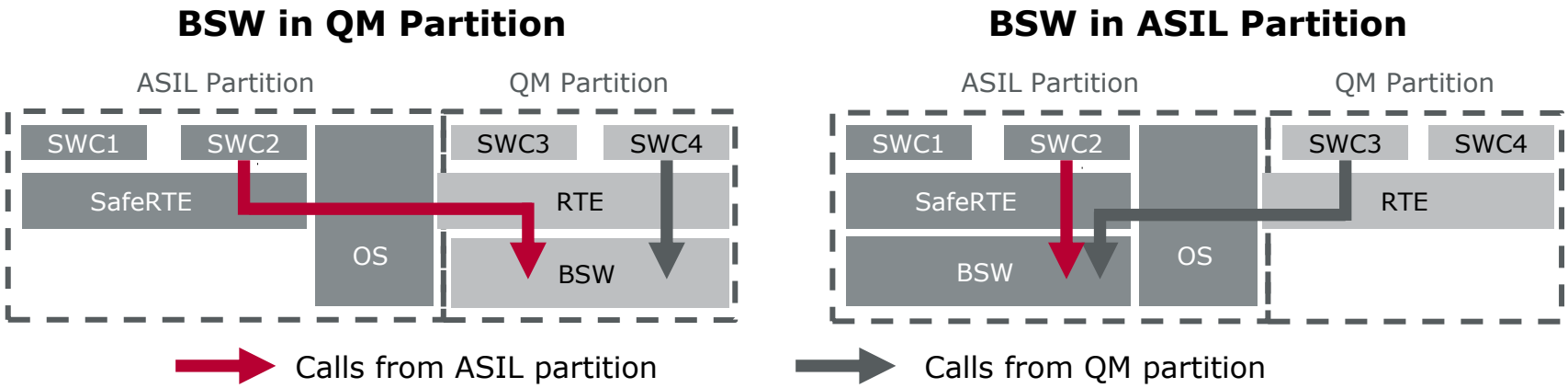


MICROSAR SafeBSW

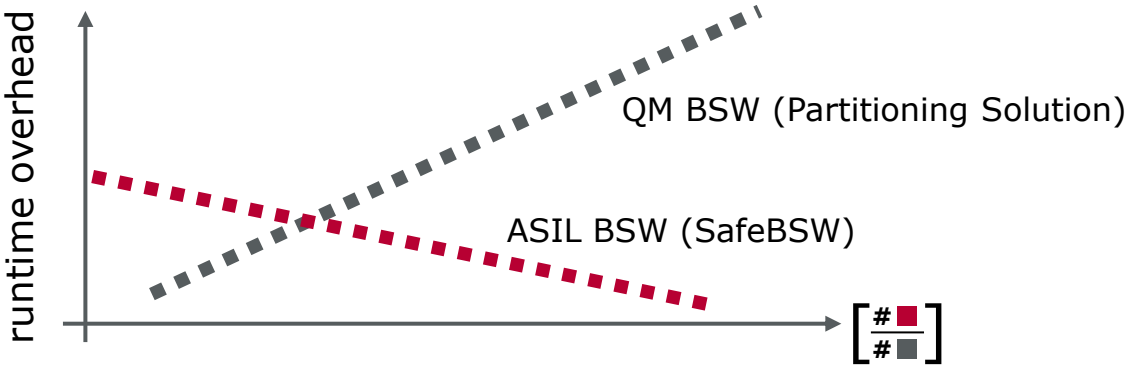


- BSW modules developed according to ASIL D
- SafeBSW can be run in the **same OS application** as your ASIL SW components.
  - There is **no need for context switches** for calls to SafeBSW.
- SafeBSW can **implement (parts of) safety mechanisms**.

Choosing the Right Approach

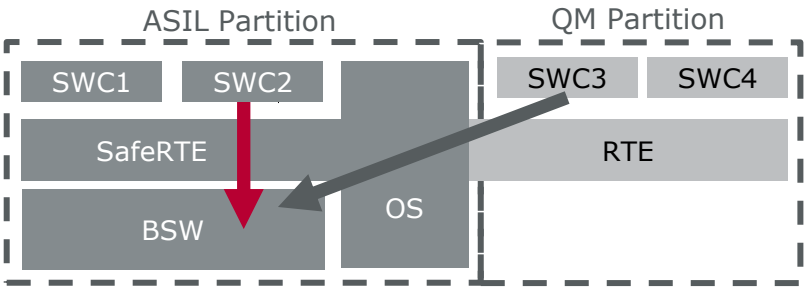


- Calls to BSW are necessary for e.g. external communication, notifications, …



- If the majority of application software has the same ASIL, performance can be boosted by having an ASIL BSW that allows to coexist in the same partition.

Improving Performance



Speedup of ASIL SWC Comm.

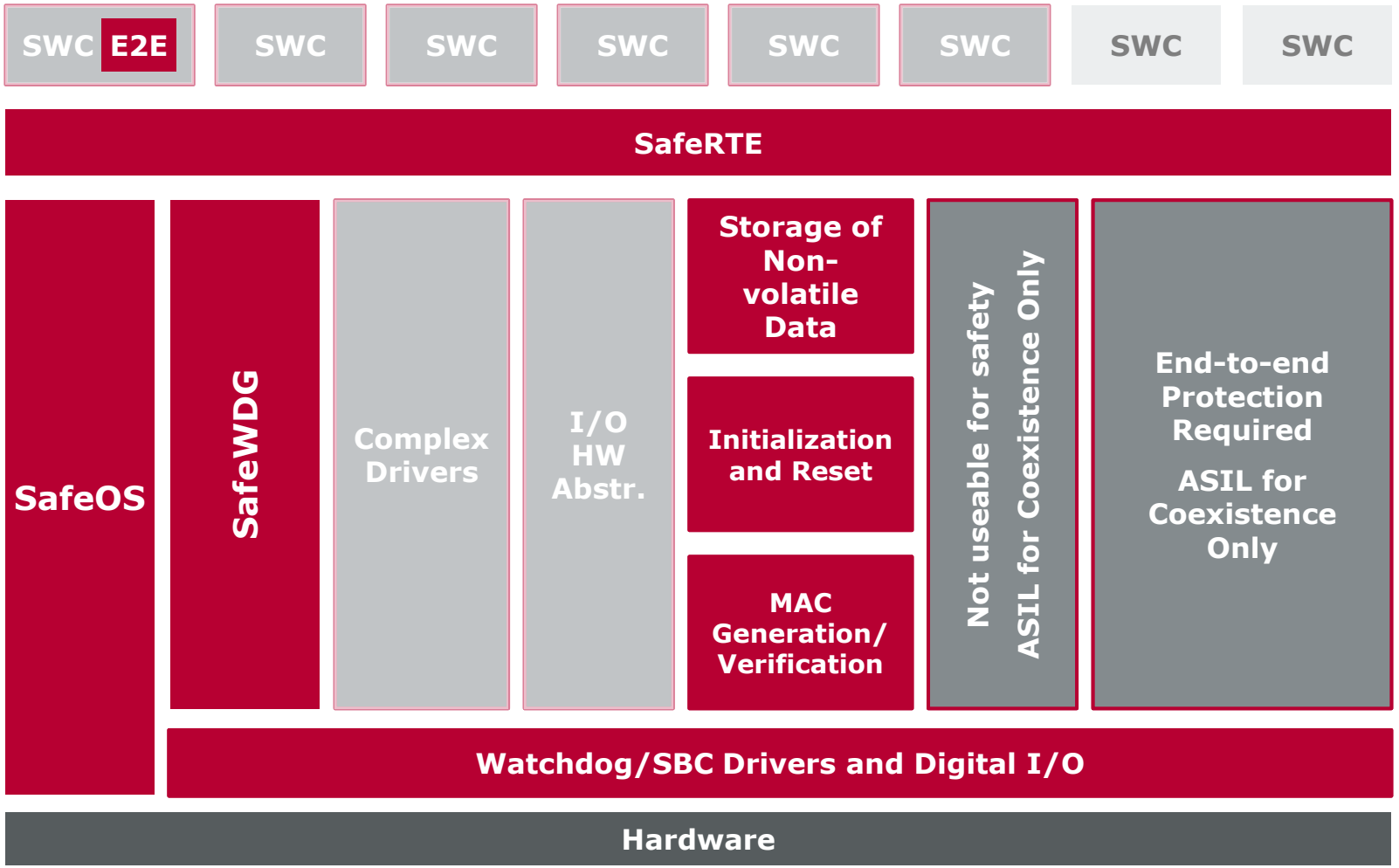
- BSW can run in same partition as ASIL SWCs
- No partition switch necessary if ASIL SWCs communicate with BSW
- Reduced overhead for scheduling of ASIL tasks
- Direct access to protected registers possible from ASIL drivers

Speed-up of QM SWC Comm.

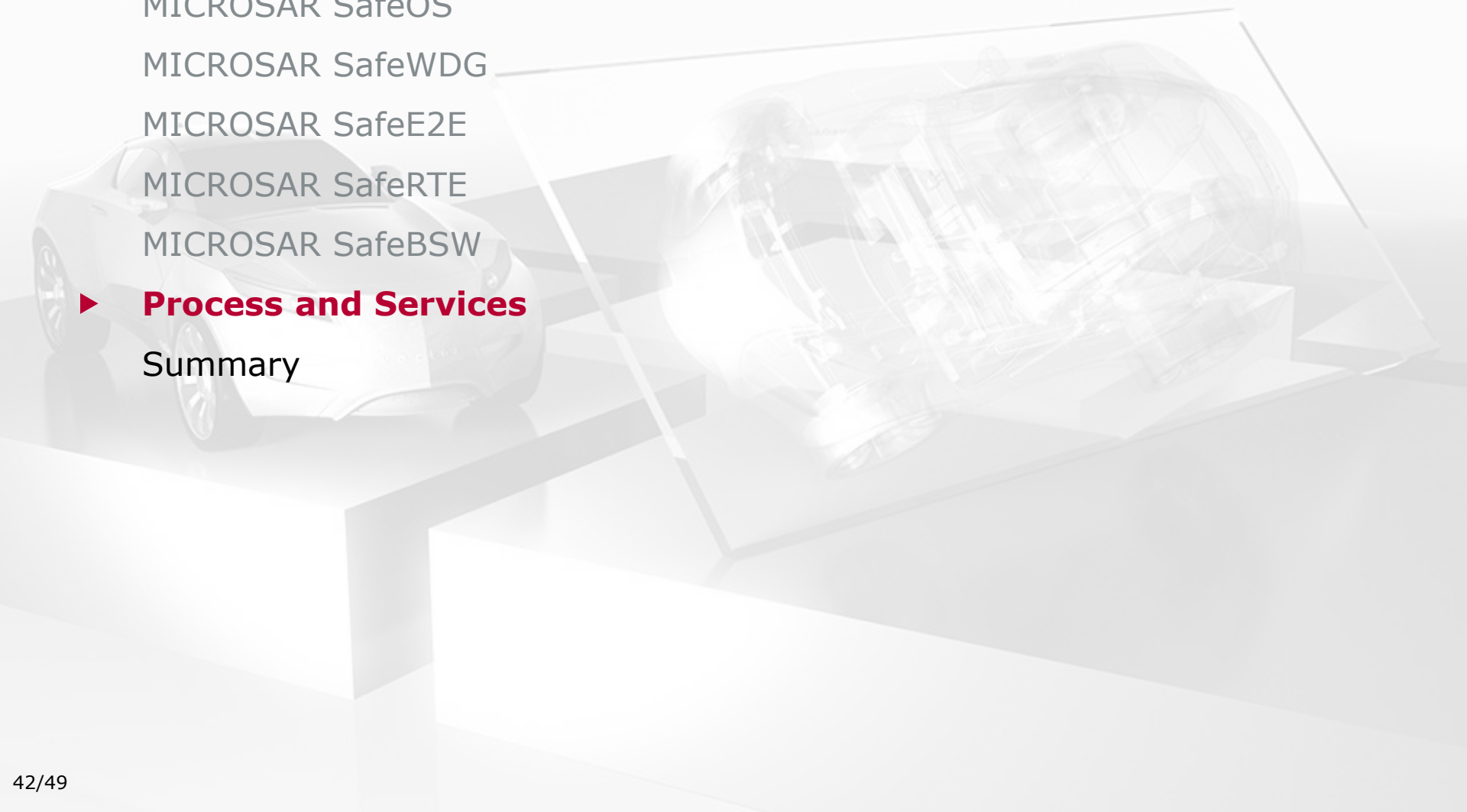
- QM SWC can use trusted-function calls to call BSW functions
- There is a mode-switch, but no context switch ➔ The code is executed on the stack of the caller
- Resulting time to cross partition boundary is reduced
- Trusted Function stubs can be automatically generated for BSW services

VECTOR

Safety Requirements

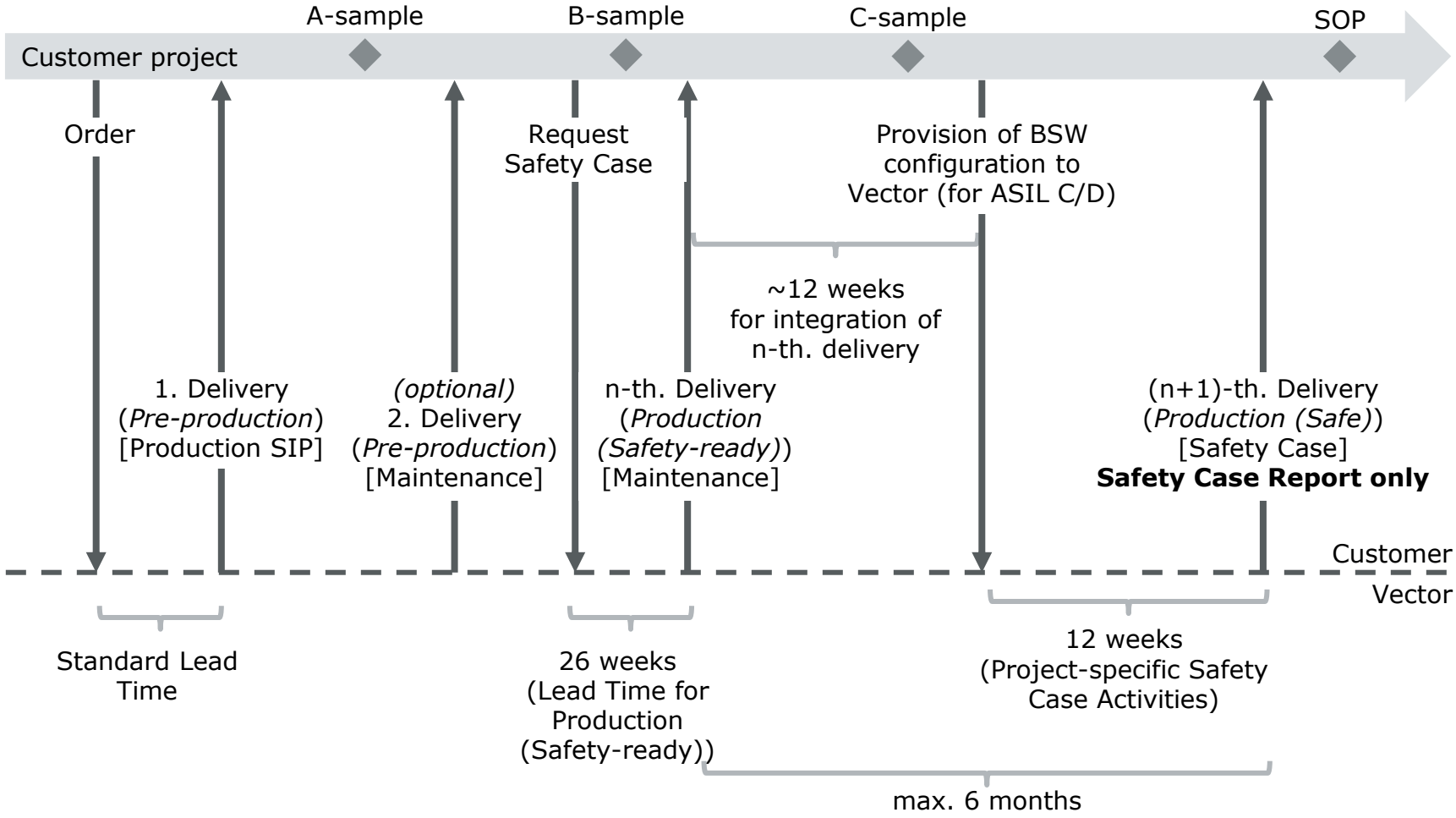


Agenda



VECTOR

Example timeline for Safety Case



Project-specific Safety Case Activities

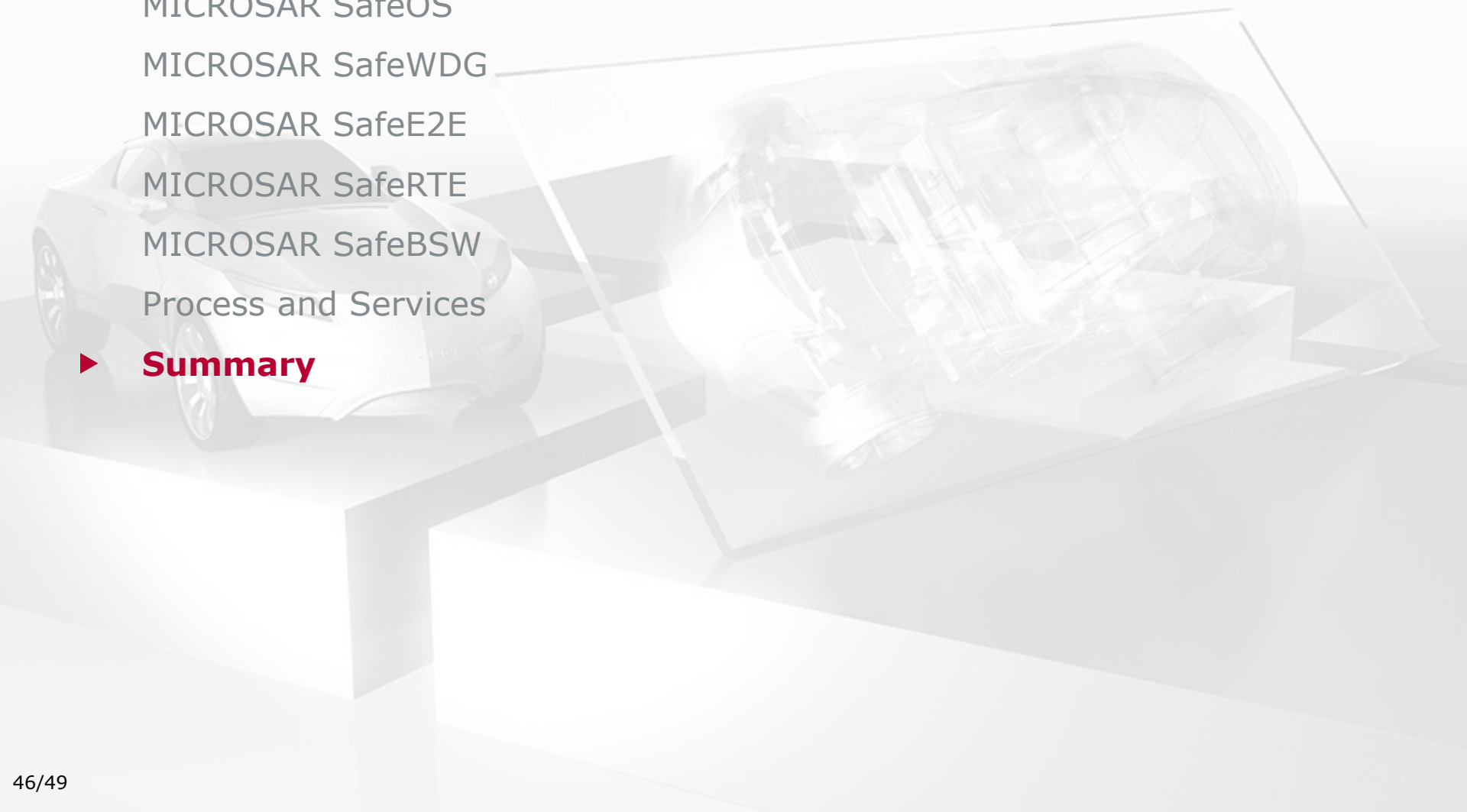
- The following activities are performed to create a Safety Case by Vector
  - **Validation of component tests**
    - Based on your configuration we examine if Vector's component tests match your configuration. In doubt Vector performs additional tests on your configuration.
    - Necessary for ASIL C/D only
  - **Confirmation report for completeness by quality department**
    - Our quality department is considered sufficiently independent (I3).
  - **Creation of safety case report**
    - The safety case report summarizes the relevant information, i.e.:
      - Set of MICROSAR Safe components
      - Configuration
      - Microcontroller (and external hardware) derivative
      - Compiler and compiler options
    - The safety case report is your confirmation from Vector that the basic software has the requested ASIL.

Services in context of SafeBSW

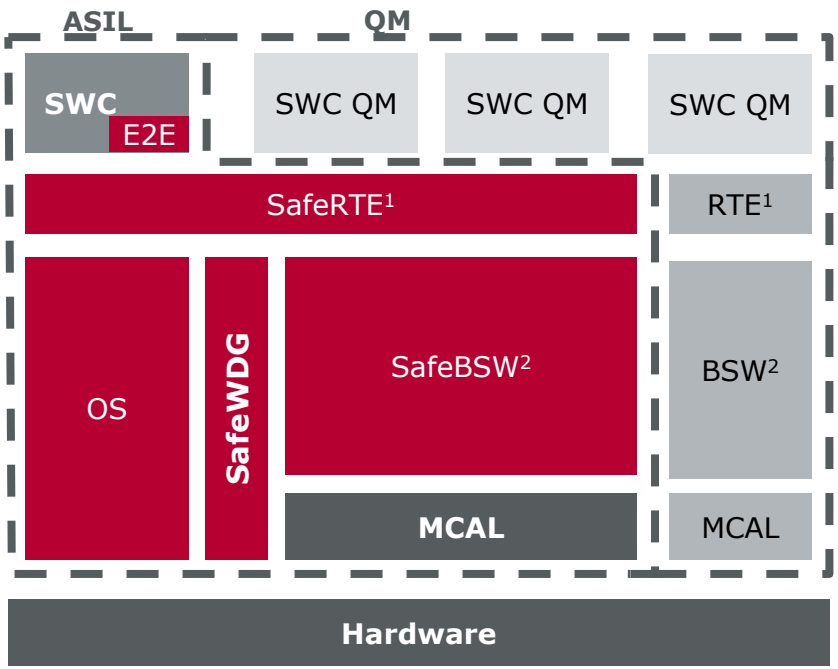
- Additional project specific services can be ordered from Vector:
  - Development of test specifications and tests to verify the SafeBSW Software Safety Requirements within your context.
  - Execution of tests using VT System.
  - Qualification of the I/O Hardware Abstraction.
  - Qualification of generated executable code, callouts and hooks from BSW.
  - Qualification of MCALs.
  - Support for SafeBSW and VT System.

Agenda

- Introduction
- MICROSAR SafeOS
- MICROSAR SafeWDG
- MICROSAR SafeE2E
- MICROSAR SafeRTE
- MICROSAR SafeBSW
- Process and Services
- **Summary**



Overview and Availability



ASIL (Tier1/OEM)

QM (Tier1/OEM)

MICROSAR Safe

MICROSAR QM

ASIL (3rd party)

[1],[2]: RTE and BSW are either ASIL or QM

- Partitioning:
  - **SafeOS** available
  - **SafeE2E** available
  - **SafeWatchdog** available
- RTE: **SafeRTE** available
- BSW **SafeBSW** available for most modules (e.g. ETH will follow)

Safety Building Blocks of MICROSAR Safe

MICROSAR SafeOS

- Supports memory partitioning using an MPU
- Provides safe context switch for each safety related task:
  - register settings
  - stack pointer and program counter
  - MPU settings
- Available for single- and multi-core
- OS Applications can be restated individually

MICROSAR SafeWDG

- Detects timing and execution order faults
- Provides deadline, alive and logic monitoring
- Capable of using internal or external watchdogs as well as system basis chips (SBCs)

MICROSAR SafeE2E

- Ensures safe communication between ECUs
- Available as E2E Protection Wrapper and E2E Transformer
- All AUTOSAR profiles supported

MICROSAR SafeRTE

- Ensures safe communication within the ECU
- Supports safe communication across partition boundaries to exchange information between ASIL and QM applications

MICROSAR SafeBSW

- Increased performance
  - complete BSW as ASIL software
  - Reduced partition switches
- Additional safety requirements, e.g.:
  - Correct initialization using an ASIL EcuM
  - Safe write/read of non-volatile data

For more information about Vector
and our products please visit

www.vector.com

Author:

Günther Heling, Jonas Wolf, Markus Oertel

Vector Germany