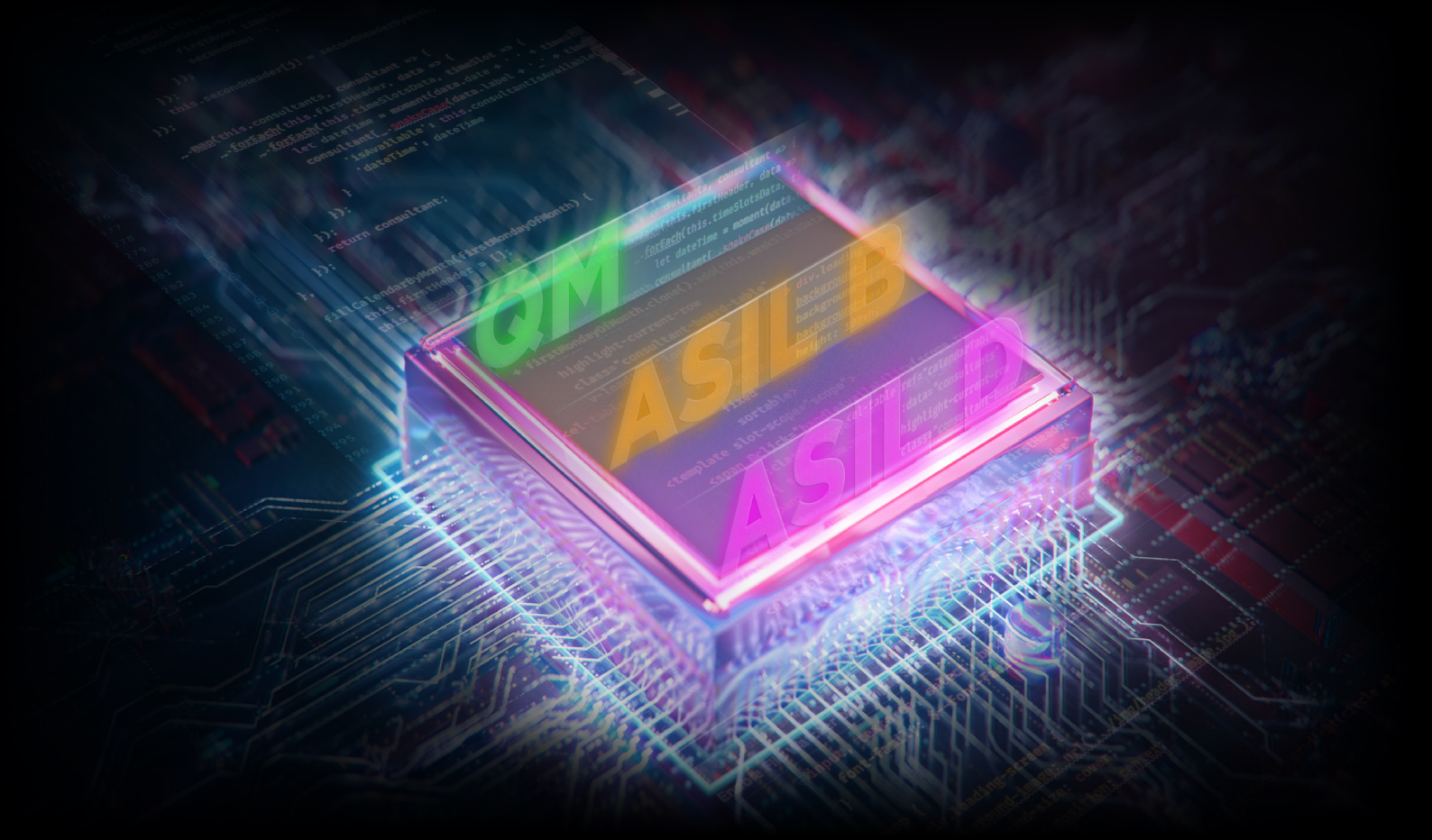
TASKING®

TASKING SAFETY CHECKER

SAFETY CHECKER GETS A WELL-DESERVED UPGRADE!



www.tasking.com

The concept of Freedom from Interference is illustrated in the following diagram. The ASIL D Software Component 2 can access both the memory locations as well as memory mapped peripherals of the ASIL D and QM functions. However, the converse is not true.

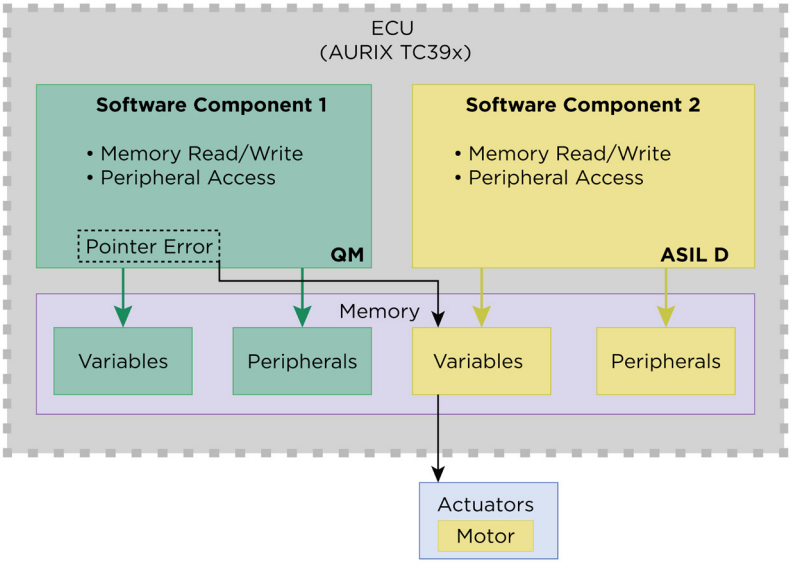


Figure 1: Freedom from Interference

ISO 26262 forbids a lower ASIL rated software component from writing to a higher rated safety feature/function. In today's ECUs, it is common practice to have software functions with different safety requirements and ASIL levels integrated within the same microcontroller. In this scenario, ISO 26262 requires that either all software components be developed to the highest ASIL level -OR- the software components be partitioned and freedom from interference between the software partitions be guaranteed.

Usually, this partitioning of software components with different ASIL levels is done by the microcontrollers Memory Protection Unit (MPU). One of the shortcomings of using the MPU is when a Freedom from Interference violation is detected, the MPU will generate a trap or memory exception which requires a lot of time and effort to resolve. The TASKING Safety Checker provides a nice alternative. As a static code analysis tool, the Safety Checker can detect memory exceptions during application development or a merge verification step, thus protecting the main development branch with no additional effort.

What if your microcontroller doesn't have an MPU or doesn't have enough MPU regions to address all the Freedom from Interference concerns?

In this case, the TASKING Safety Checker can be used to provide evidence that software components with different safety requirements are free from interference in the spatial domain up to ASIL B.

PRODUCT FEATURES

- Detects lower level ASIL rated software components attempting to 'modify the memory' or 'preempt the execution' of higher level ASIL rated software components.
- Improves the efficiency of safety software development by statically analyzing the source code for access violations that would trigger MPU violations.
- A new graphical user interface improves ease of use by providing the capability to define safety classes, access rights and map files/functions to the defined safety classes.
- The tool is compiler and hardware architecture independent and can be integrated into continuous integration build environments like Jenkins.
- Supports import of AUTOSAR R4.0.3 and/or R4.4 configuration files
- Integrated MISRA, CERT C and FFI Rule Checker
- Project reports can be HTML, XML or PDF format and include information like the function call graph, MISRA/CERT C results, code metrics and access violation log.

TASKING'S EMBEDDED SOFTWARE SAFETY ECO-SYSTEM

TASKING offers a complete Eco-System of safety and security compliant development solutions for embedded multi-core software development.

The TASKING® Safety Checker is one part of a safety eco-system which additionally includes,

- TÜV Nord Safety Certified toolsets for the TriCore and selected Arm architectures
- TriCore Inspector
- Qualified C Libraries.

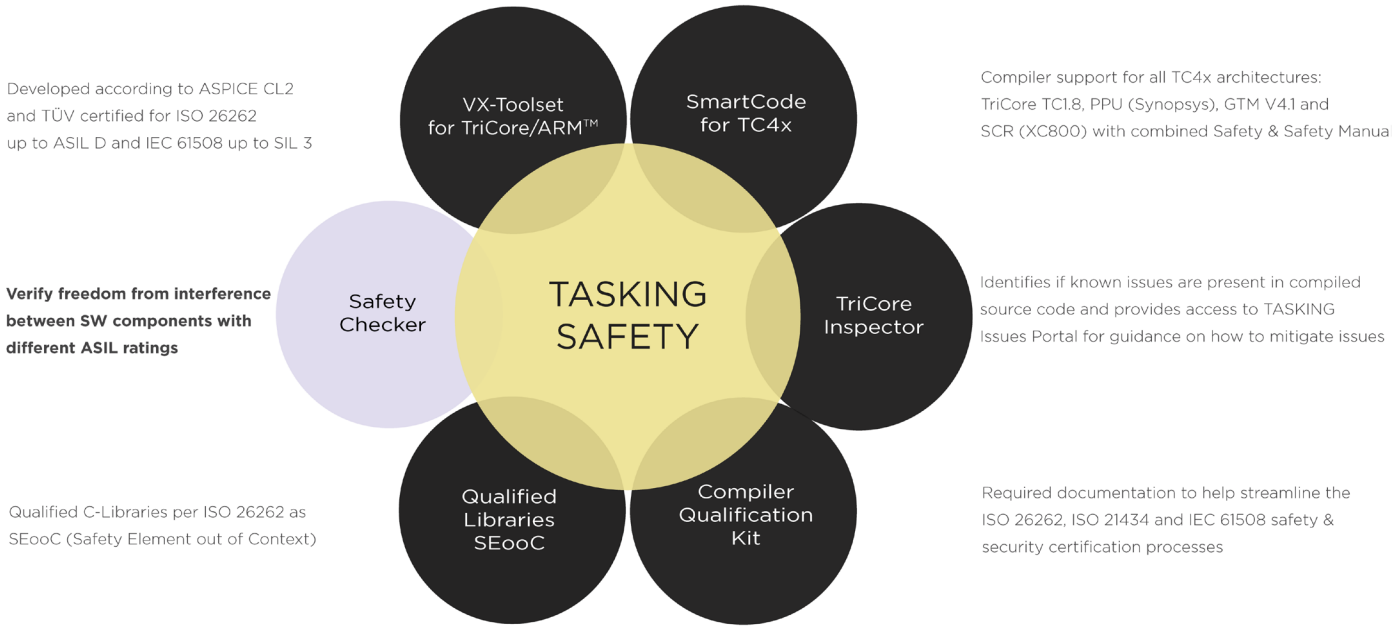


Figure 2: TASKING Safety Eco-System

LICENSING

A flexible licensing model supports the requirements of all parties in the automotive software supply chain.

Time-based floating licenses are available in combination with a floating CI license to be used in a build system.