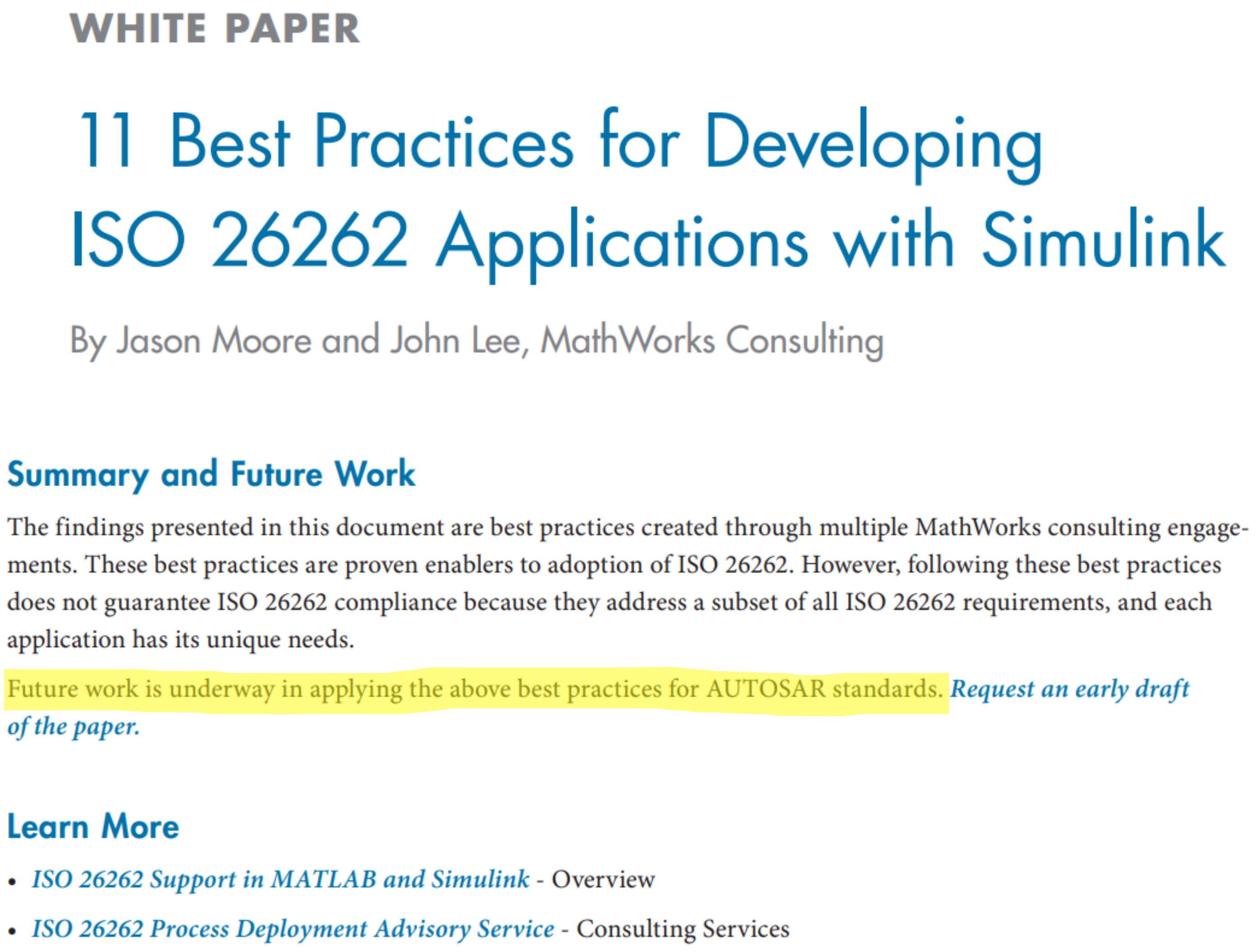
Targeting AUTOSAR and ISO 26262 with Simulink

By Michael Boyle



WHITE PAPER

11 Best Practices for Developing ISO 26262 Applications with Simulink

By Jason Moore and John Lee, MathWorks Consulting

Summary and Future Work

The findings presented in this document are best practices created through multiple MathWorks consulting engagements. These best practices are proven enablers to adoption of ISO 26262. However, following these best practices does not guarantee ISO 26262 compliance because they address a subset of all ISO 26262 requirements, and each application has its unique needs.

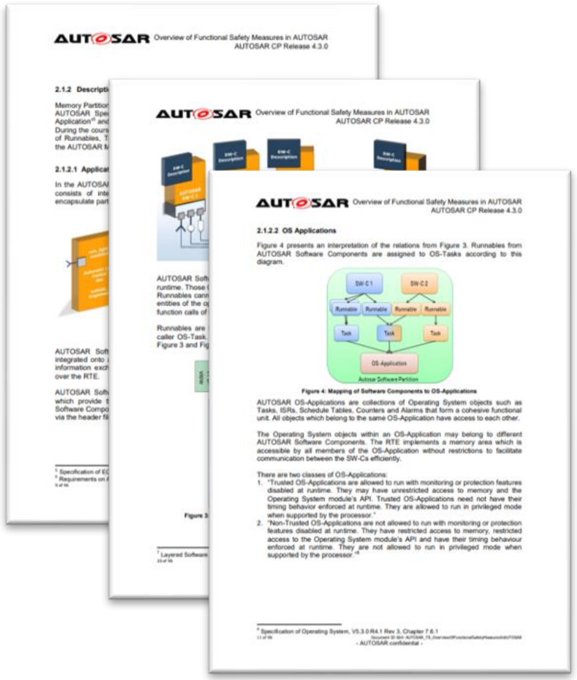
Future work is underway in applying the above best practices for AUTOSAR standards. *Request an early draft of the paper.*

Learn More

- *ISO 26262 Support in MATLAB and Simulink* - Overview
- *ISO 26262 Process Deployment Advisory Service* - Consulting Services

AUTOSAR and ISO 26262 are complementary

- AUTOSAR was constructed with functional safety in mind
- AUTOSAR has a release overview of functional safety features
  - https://www.autosar.org/fileadmin/user_upload/standards/classic/21-11/AUTOSAR_EXP_FunctionalSafetyMeasures.pdf
- The features AUTOSAR has implemented help with
  - Encapsulation
  - Reuse
  - Freedom from interference
  - Memory partitioning
  - Interoperability



AUTOSAR Overview of Functional Safety Measures in AUTOSAR
AUTOSAR CP Release 4.3.0

2.1.2.2 OS Applications

Figure 4 presents an interpretation of the relations from Figure 3. Runnables from AUTOSAR Software Components are assigned to OS-Tasks according to this diagram.

Figure 4: Mapping of Software Components to OS-Applications

AUTOSAR OS-Applications are collections of Operating System objects such as Tasks, ISRs, Schedule Tables, Counters and Alarms that form a cohesive functional unit. All objects which belong to the same OS-Application have access to each other.

The Operating System objects within an OS-Application may belong to different AUTOSAR Software Components. The RTE implements a memory area which is accessible by all members of the OS-Application without restrictions to facilitate communication between the SW-Cs efficiently.

There are two classes of OS-Applications:

1. "Trusted OS-Applications are allowed to run with monitoring or protection features disabled at runtime. They may have unrestricted access to memory and the Operating System module's API. Trusted OS-Applications need not have their timing behavior enforced at runtime. They are allowed to run in privileged mode when supported by the processor."
2. "Non-Trusted OS-Applications are not allowed to run with monitoring or protection features disabled at runtime. They have restricted access to memory, restricted access to the Operating System module's API and have their timing behaviour enforced at runtime. They are not allowed to run in privileged mode when supported by the processor."[8]

[8] Specification of Operating System, V5.3.0 R4.1 Rev 3, Chapter 7.6.1

11 of 96

- AUTOSAR confidential -

AUTOSAR Functional Safety Measures

Overview

- AUTOSAR with ISO 26262 using Simulink
- Best Practices based on Consulting Engagements
- Focus areas for today:
  - AUTOSAR Architecture
  - AUTOSAR Data Transfer

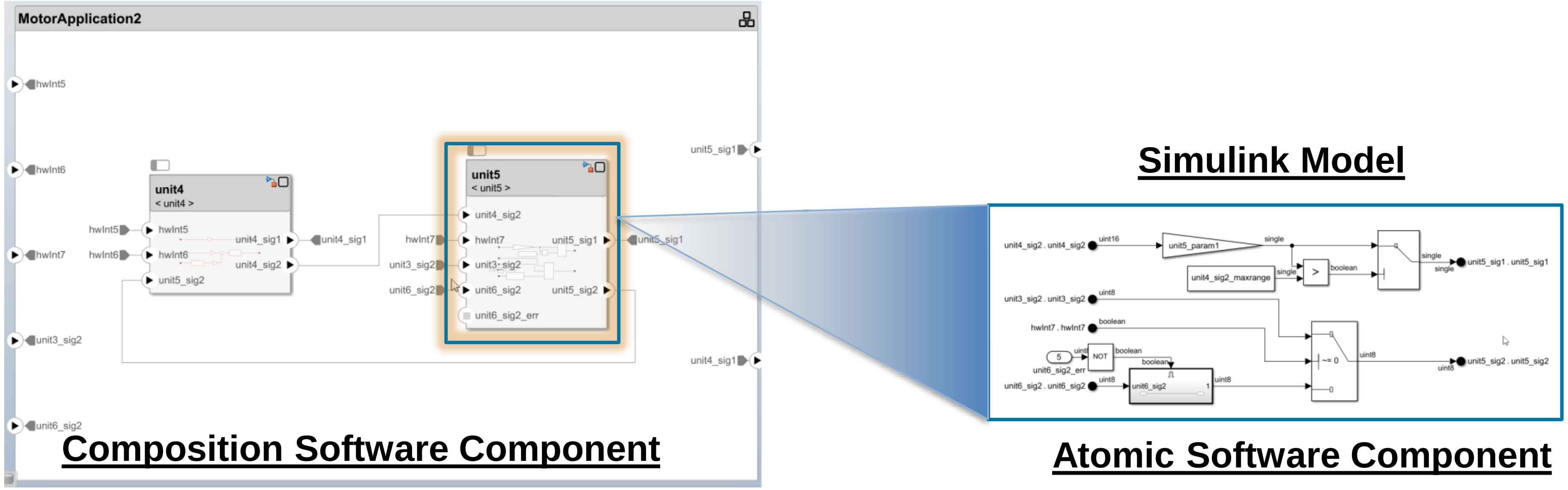
- Paper
  - Contact mboyle@mathworks.com

Atomic Software Component as Unit

- Issues:
  - How do I unit test an AUTOSAR application?
- Best Practice
  - Define Atomic Software Components as the unit boundary
    - Map to Simulink models
    - Generate encapsulated reusable code

AUTOSAR Architecture Model

Simulink Model

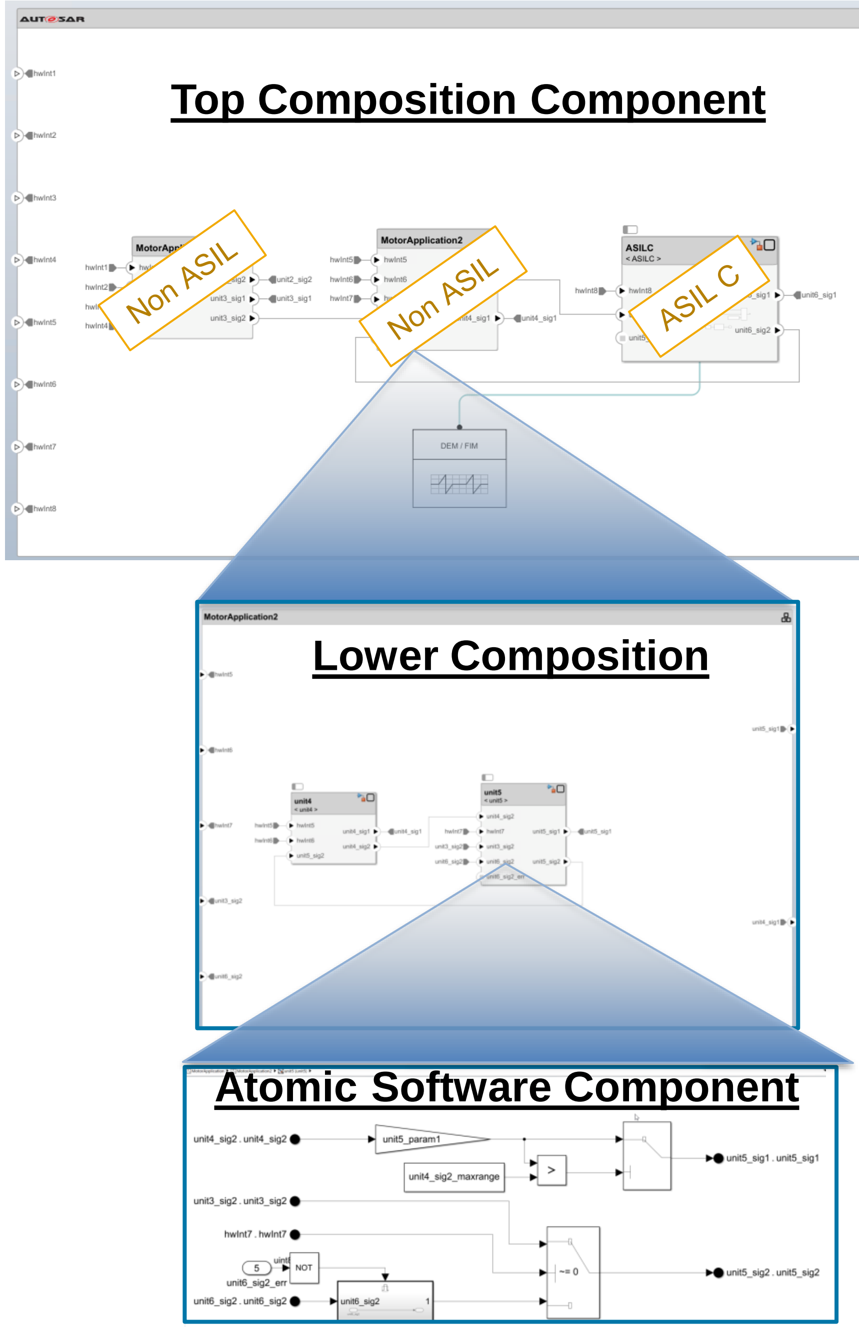


Composition Software Component

Atomic Software Component

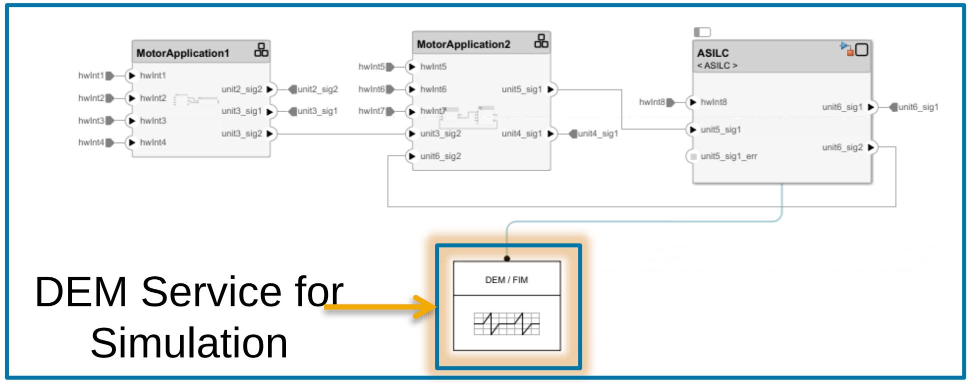
Segment like ASIL Components

- Issues:
  - How should I architect my AUTOSAR software for ISO 26262?
- Best Practice
  - Split non ASIL and ASIL sections into separate AUTOSAR compositions
    - Eases managing freedom from interference concerns
  - Add composition hierarchy to split features in each ASIL section
    - Improves design readability, feature segmentation, and testability

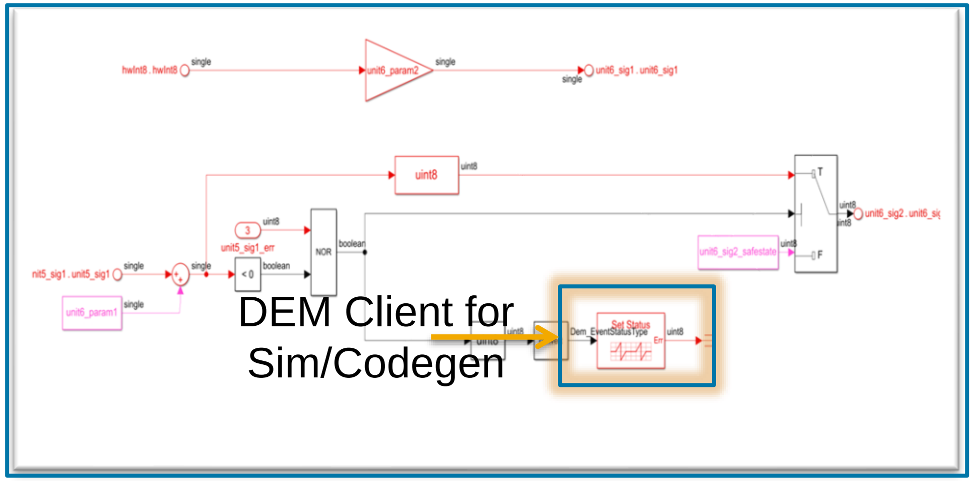


Utilize AUTOSAR Standard Services

- Issues:
  - How can I reduce my development and verification effort for my AUTOSAR software?
- Best Practice
  - Use AUTOSAR Standardized Interfaces to leverage Basic Software Services
    - AUTOSAR Blockset provides service components for simulation
    - Enables reuse and exchange of application software
    - Off-the-shelf BSW packages available from vendors



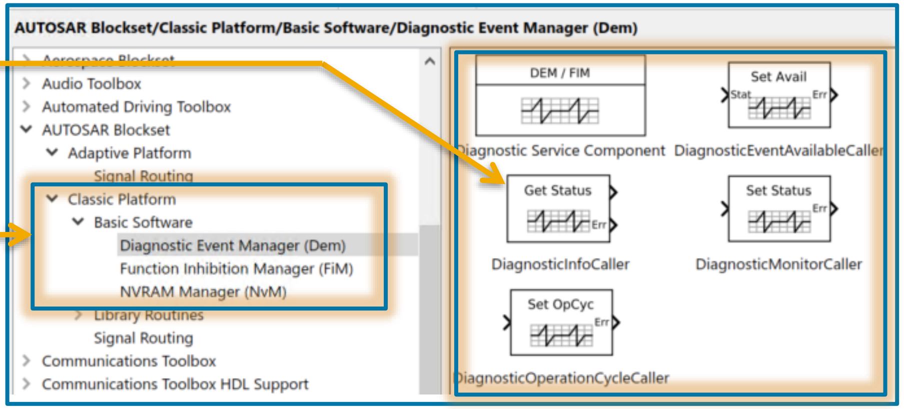
DEM Service for Simulation



DEM Client for Sim/Codegen

Block Available

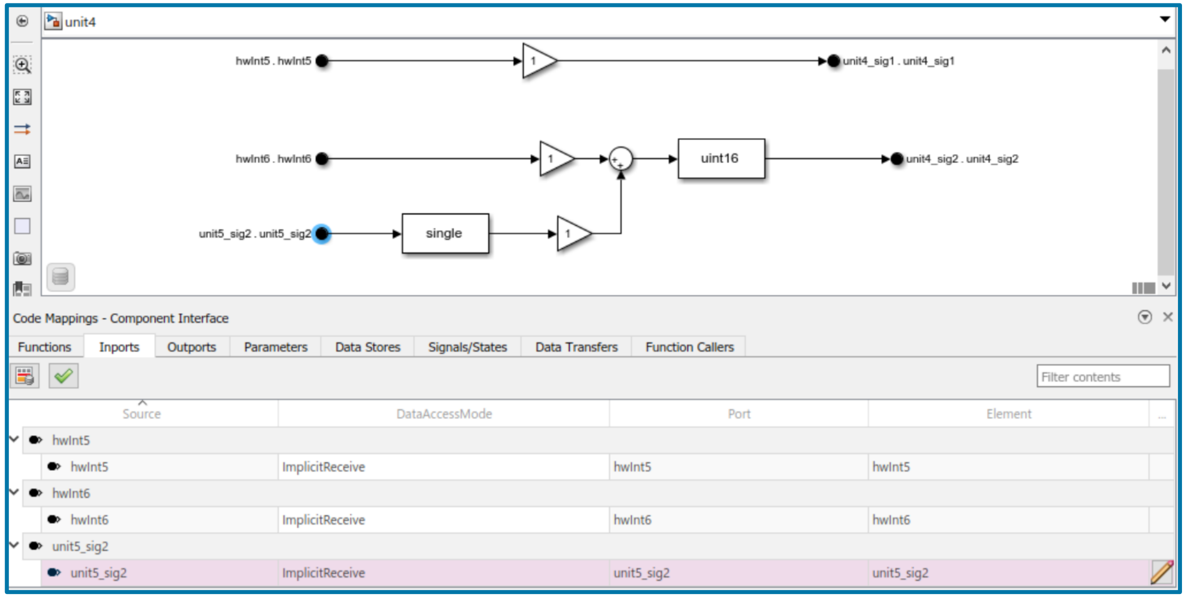
Services Supported


AUTOSAR Blockset/Classic Platform/Basic Software/Diagnostic Event Manager (Dem)

- Aerospace Blockset
- Audio Toolbox
- Automated Driving Toolbox
- AUTOSAR Blockset
  - Adaptive Platform
    - Signal Routing
  - Classic Platform
    - Basic Software
      - Diagnostic Event Manager (Dem)
      - Function Inhibition Manager (FiM)
      - NVRAM Manager (NvM)
    - Library Routines
    - Signal Routing
- Communications Toolbox
- Communications Toolbox HDL Support

RTE for Safety Critical Data

- Issues:
  - How do I manage data transfers throughout the AUTOSAR software?
- Best Practice
  - Safety critical data should go through the RTE
    - Supports encapsulation of software component implementations
    - Easy to configure for model elements in Code Mappings editor

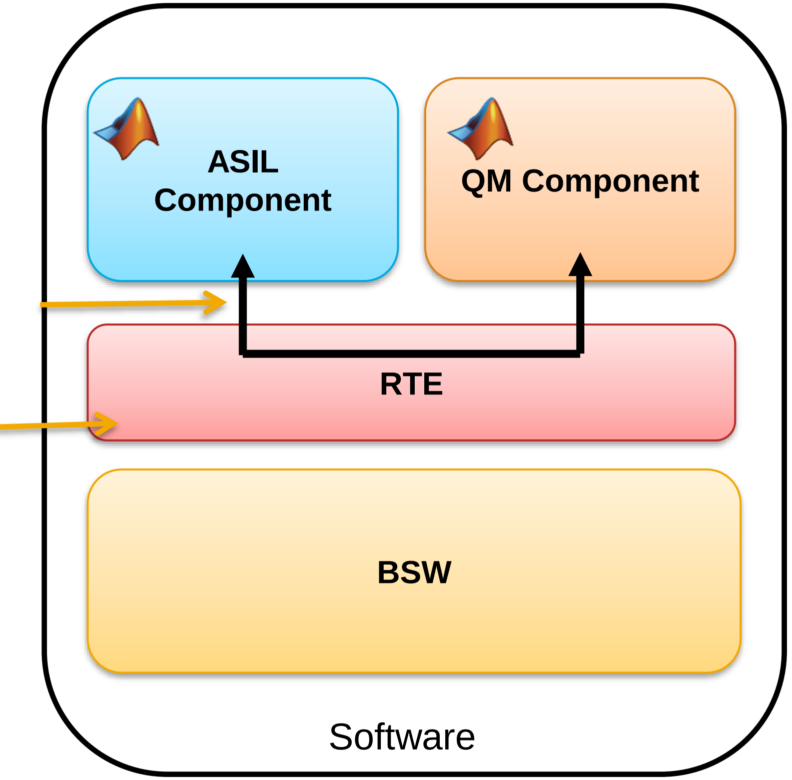


Code Mappings - Component Interface

Source	DataAccessMode	Port	Element
hwInt5			
hwInt5	ImplicitReceive	hwInt5	hwInt5
hwInt6			
hwInt6	ImplicitReceive	hwInt6	hwInt6
unit5_sig2			
unit5_sig2	ImplicitReceive	unit5_sig2	unit5_sig2

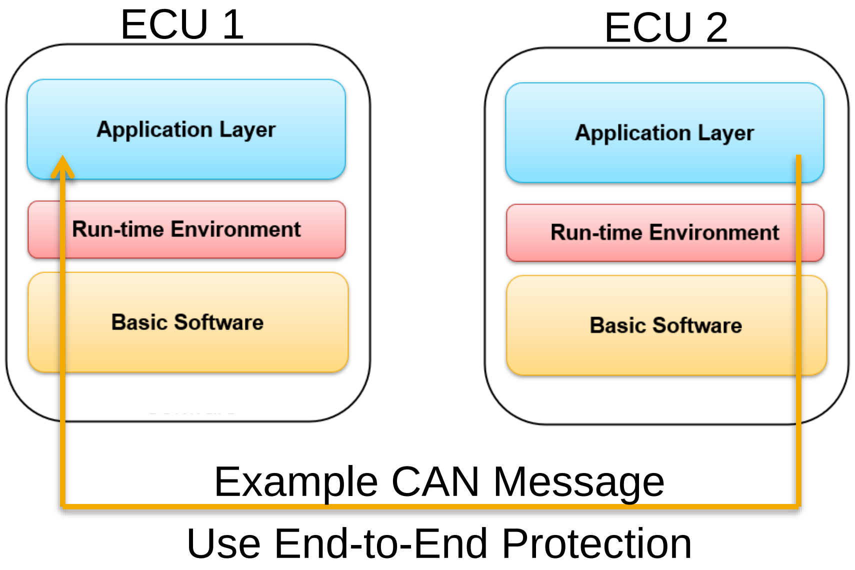
AUTOSAR Data Exchanged Through RTE

RTE Layer Generated by an RTE Generation Tool



End to End Protection for Critical Signals

- Issues:
  - How can I protect critical signals during transmission?
- Best Practice
  - Use End-to-End protection for high integrity signals
    - End-to-End protection can detect errors throughout the signal transmission chain



ECU 1

Application Layer

Run-time Environment

Basic Software

ECU 2

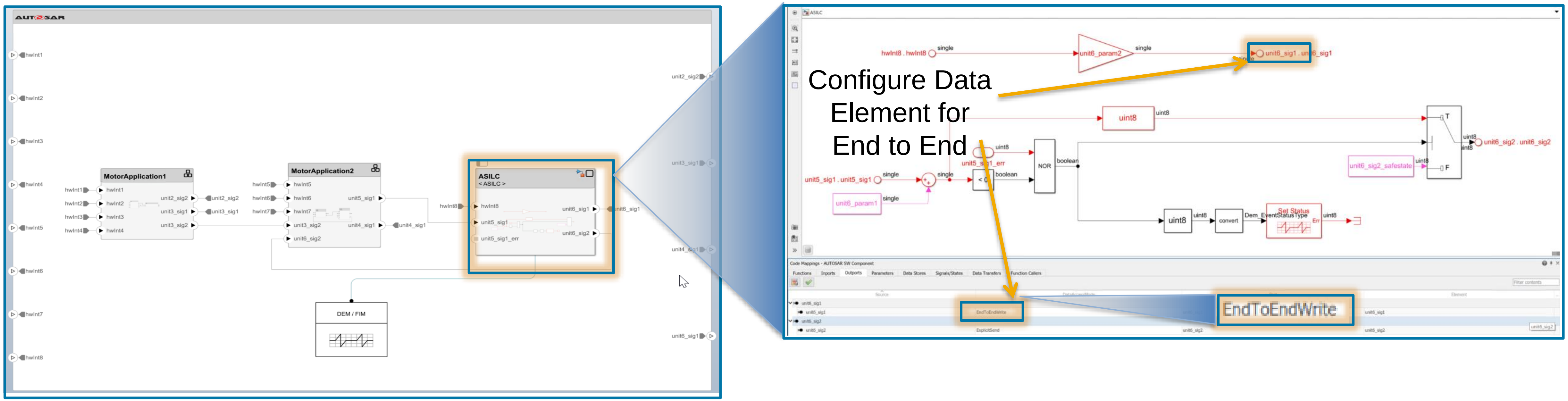
Application Layer

Run-time Environment

Basic Software

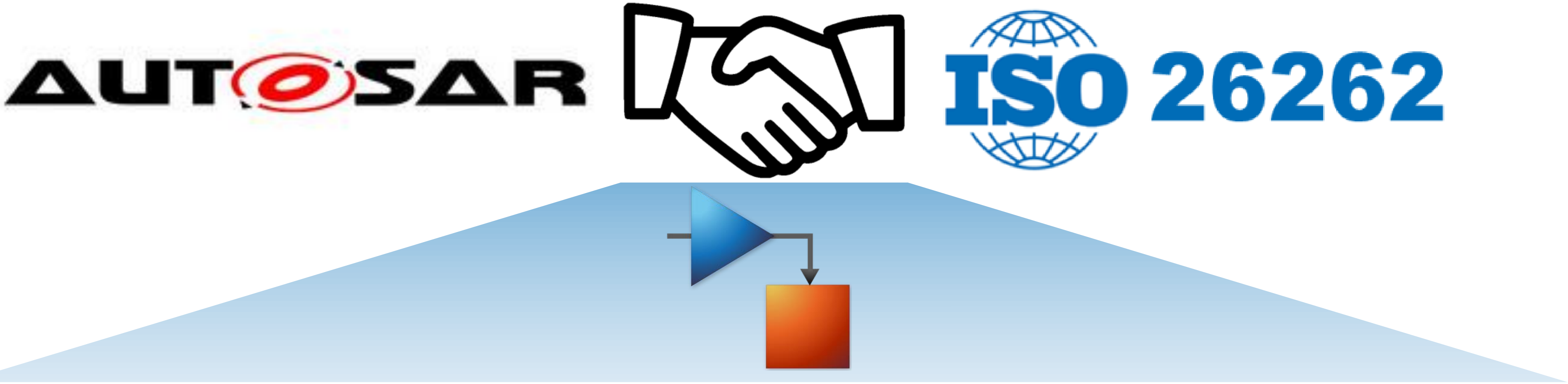
Example CAN Message

Use End-to-End Protection



Configure Data Element for End to End

EndToEndWrite



Toolchain and Workflow

- AUTOSAR Services(DEM, FIM, NVM, etc.)
- Simulink and Stateflow for fault detection
- Qualified ISO26262 tools
- Incremental testing strategy

AUTOSAR Architecture

- Segment like ASIL level components
- Software component as a unit boundary
- Software address method usage
- Rearchitect existing code base
- Data management strategy

AUTOSAR Data Transfer

- RTE for safety critical data
- End to End protection
- Implicit data transfer for high integrity signals
- Port error status usage

- Paper
  - Contact mboyle@mathworks.com