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 Functional Safety Measures

Overview

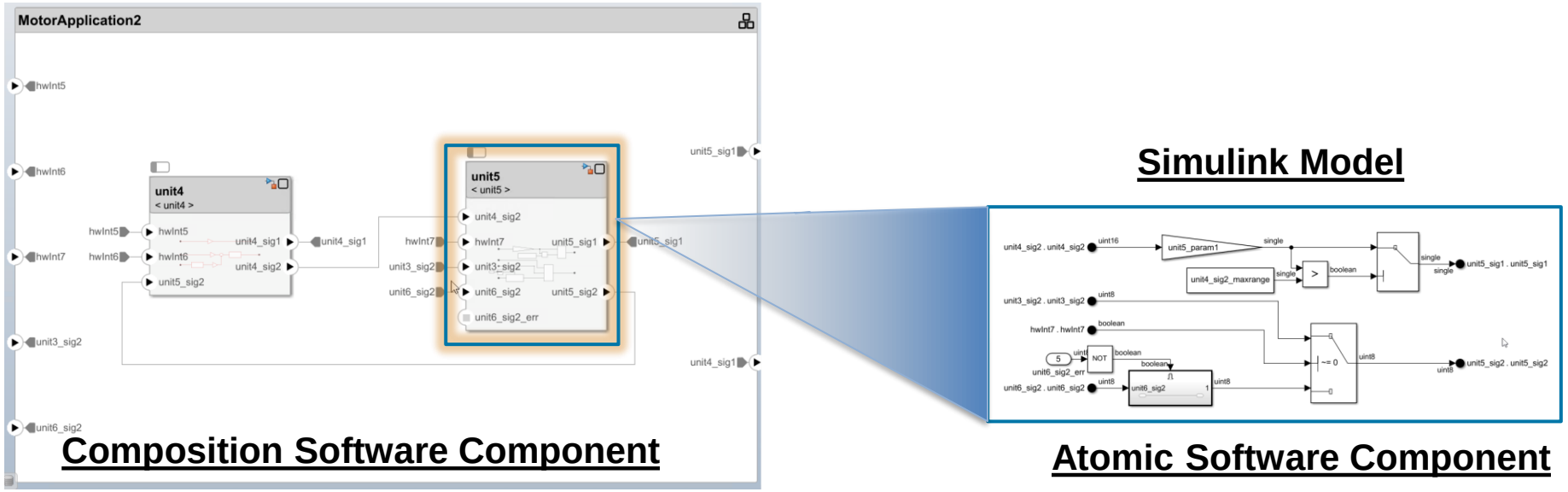
- AUTOSAR with ISO 26262 using Simulink
- Best Practices based on Consulting Engagements
- Focus areas for today:
  - AUTOSAR Architecture
  - AUTOSAR Data Transfer

- 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



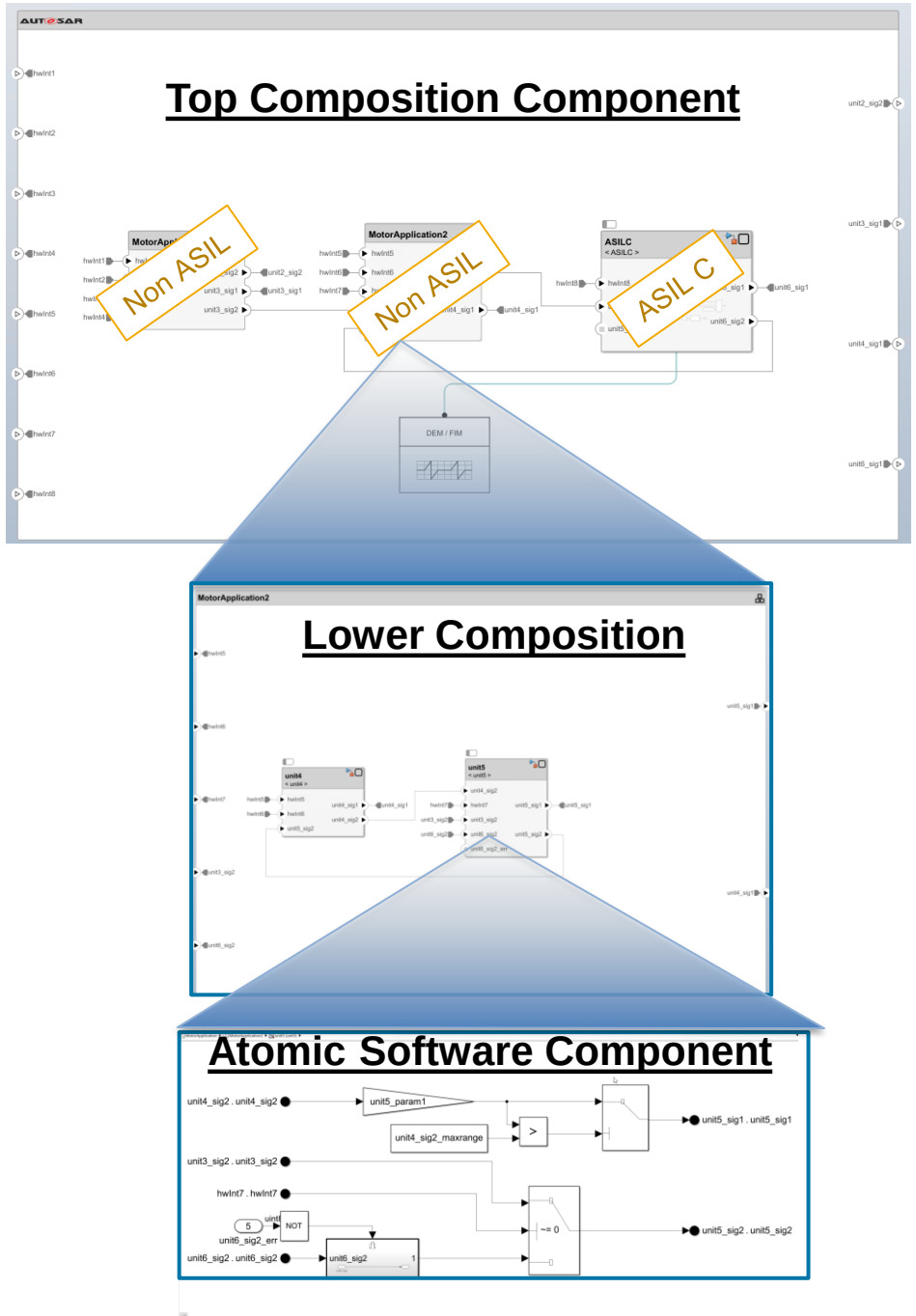
Composition Software Component

Simulink Model

Atomic Software Component

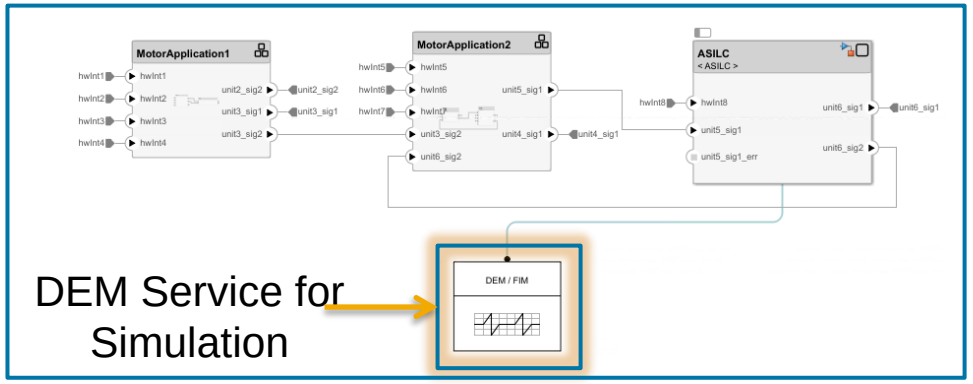
Segment like ASIL Components

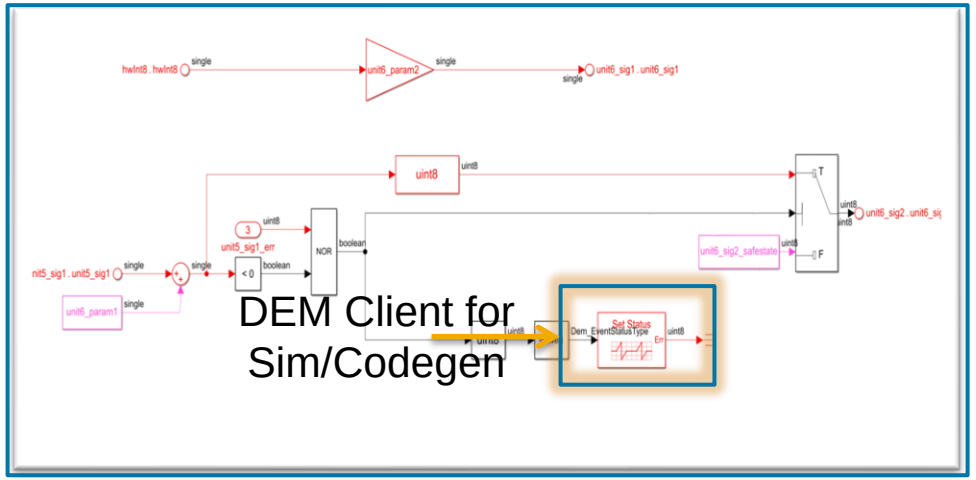
- Issues:
  - How should I architect my AUTOSAR software for ISO 26262?
- Best Practice
  - Split non ASIL and ASIL sections into separate AUTOSAR compositions
    - Eases managing freedom from interference concerns
  - Add composition hierarchy to split features in each ASIL section
    - Improves design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



Utilize AUTOSAR Standard Services

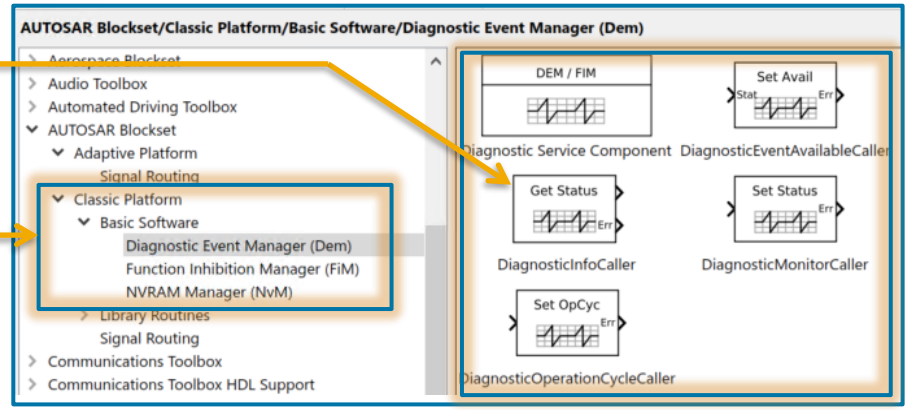
- Issues:
  - How can I reduce my development and verification effort for my AUTOSAR software?
- Best Practice
  - Use AUTOSAR Standardized Interfaces to leverage Basic Software Services
    - AUTOSAR Blockset provides service components for simulation
    - Enables reuse and exchange of application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





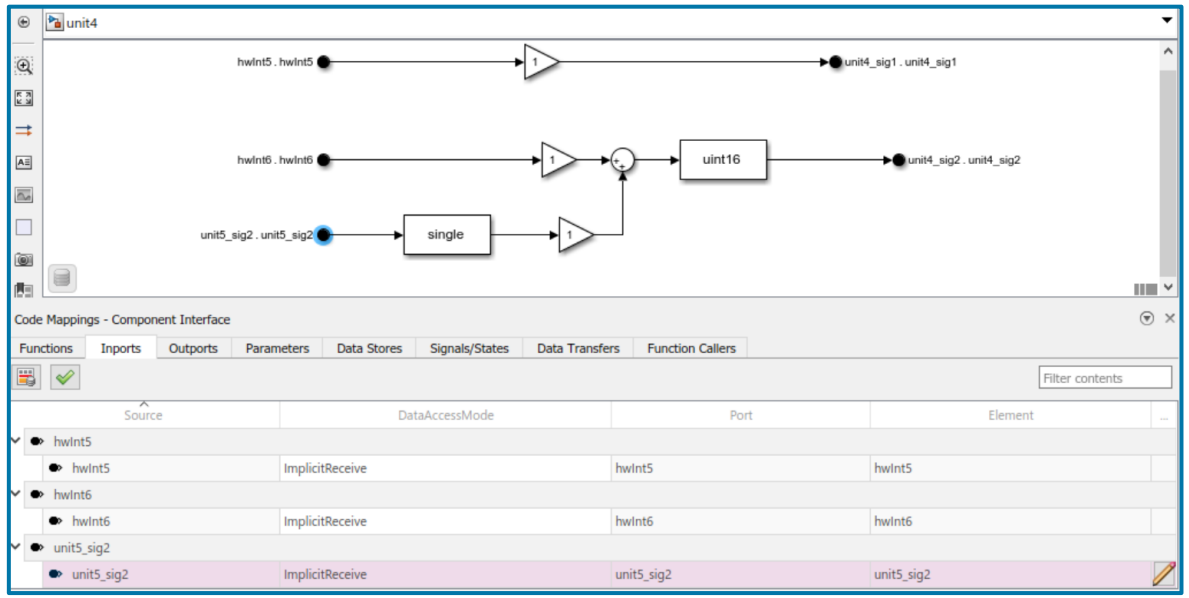
Block Available

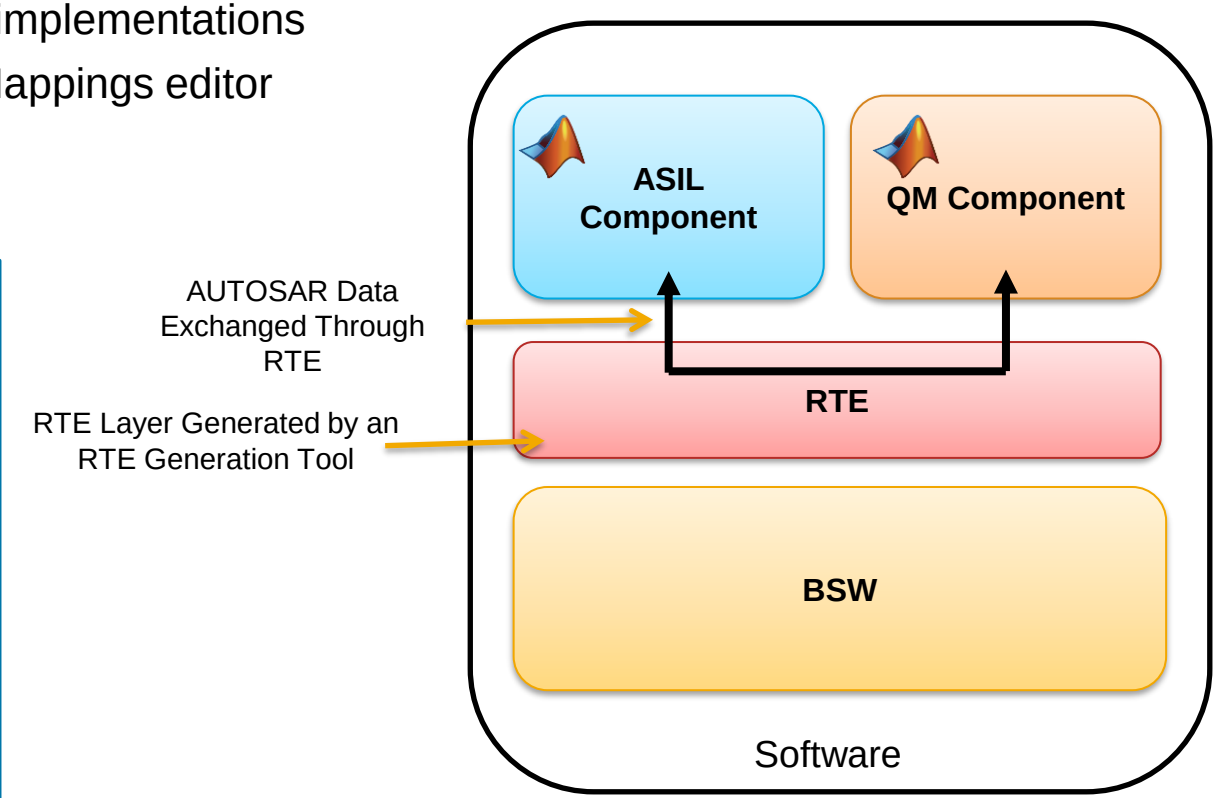
Services Supported



RTE for Safety Critical Data

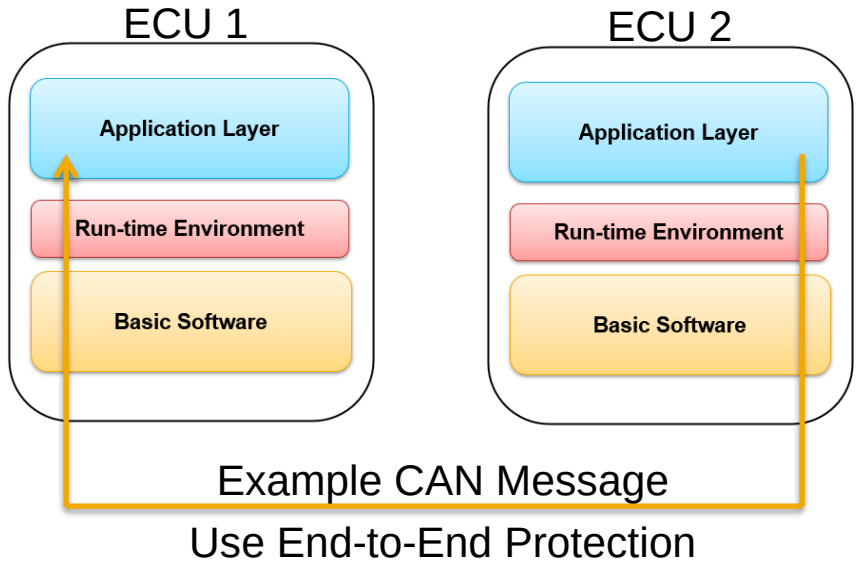
- Issues:
  - How do I manage data transfers throughout the AUTOSAR software?
- Best Practice
  - Safety critical data should go through the RTE
    - Supports encapsulation of software component implementations
    - Easy to configure for model elements in Code Mappings editor

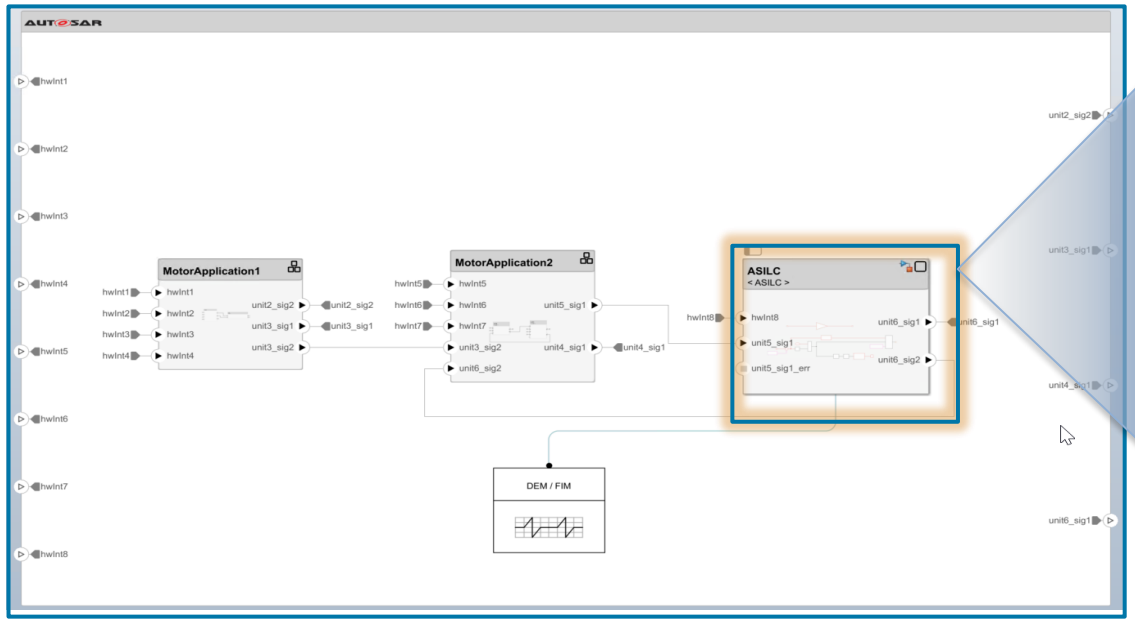


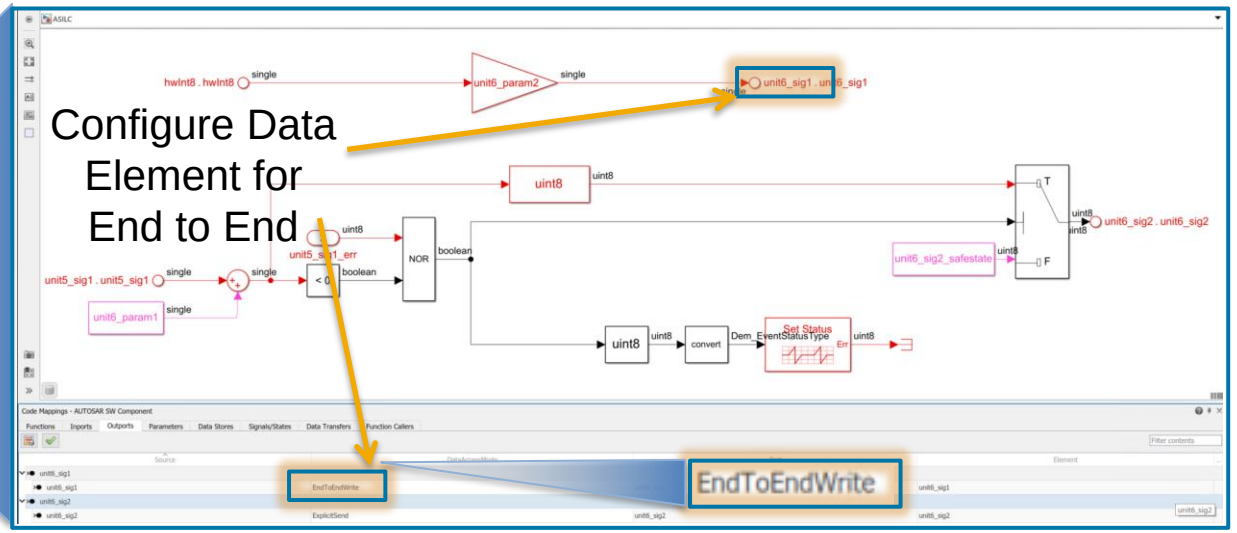


End to End Protection for Critical Signals

- 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







Toolchain and Workflow	AUTOSAR Architecture	AUTOSAR Data Transfer
• AUTOSAR Services(DEM, FIM, NVM, etc.) • Simulink and Stateflow for fault detection • Qualified ISO26262 tools • Incremental testing strategy	• Segment like ASIL level components • Software component as a unit boundary • Software address method usage • Rearchitect existing code base • Data management strategy	• 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