



Target-Independent Component-Based Design for Automated Driving Systems

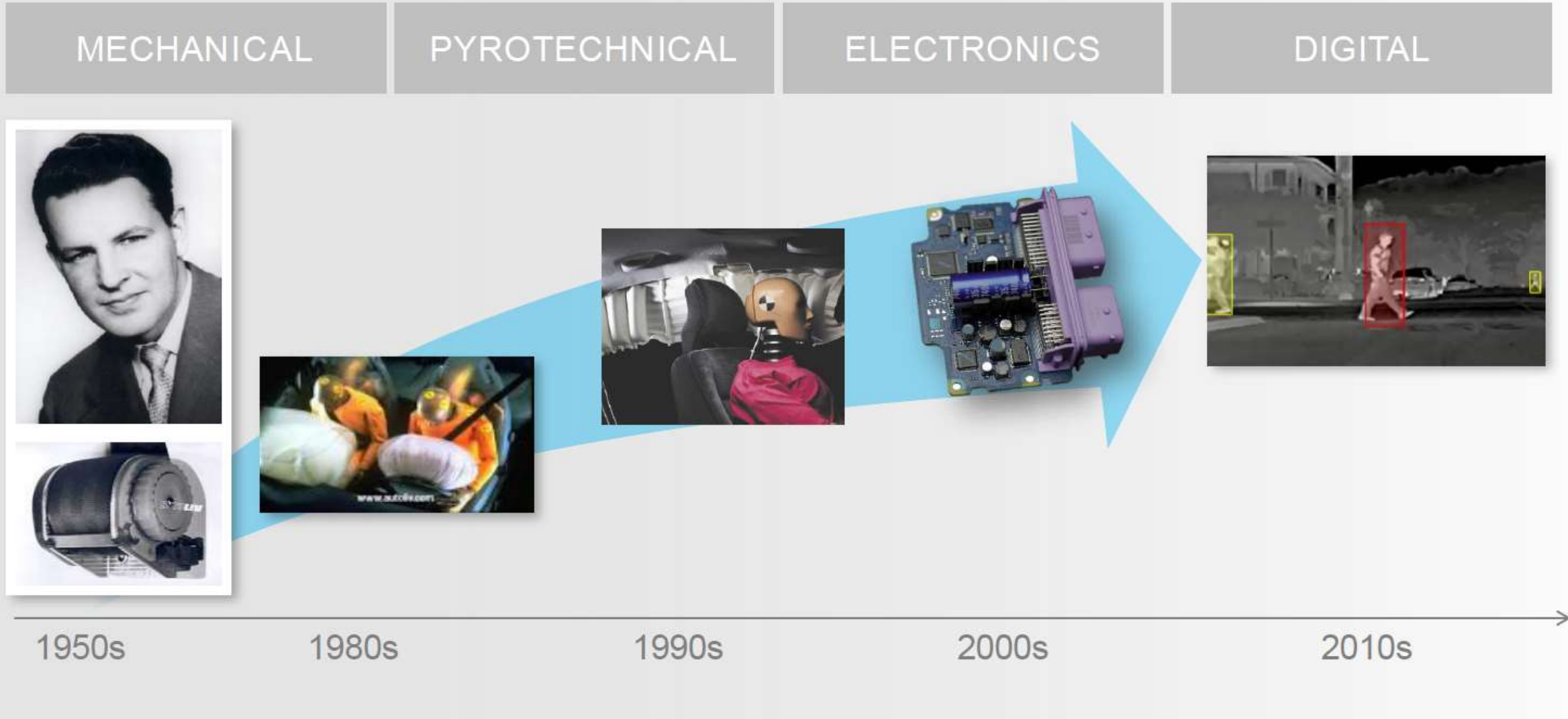
Siddharth D'Silva & Eugene Kagan

MathWorks Automotive Conference

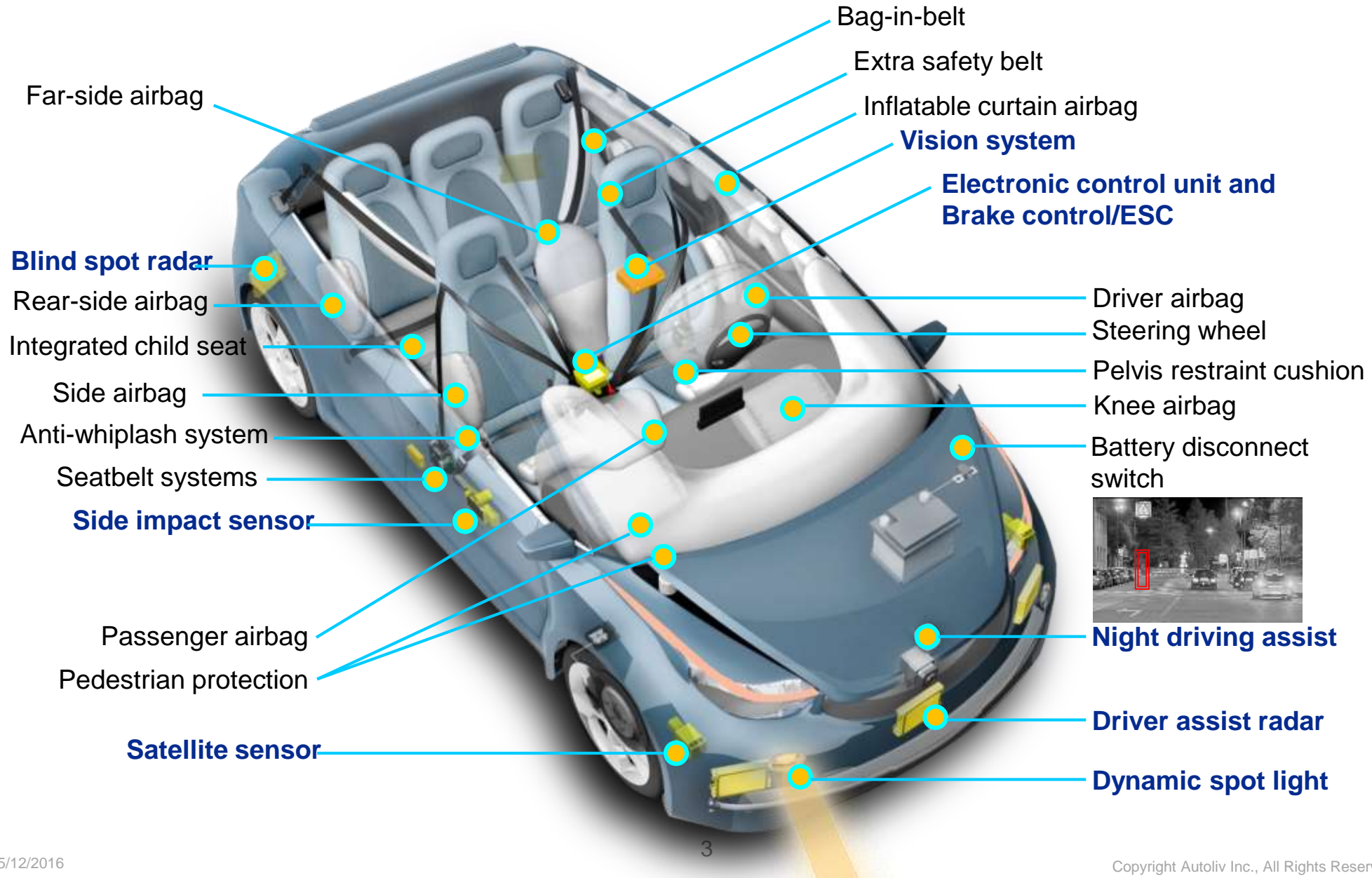
May 12, 2016



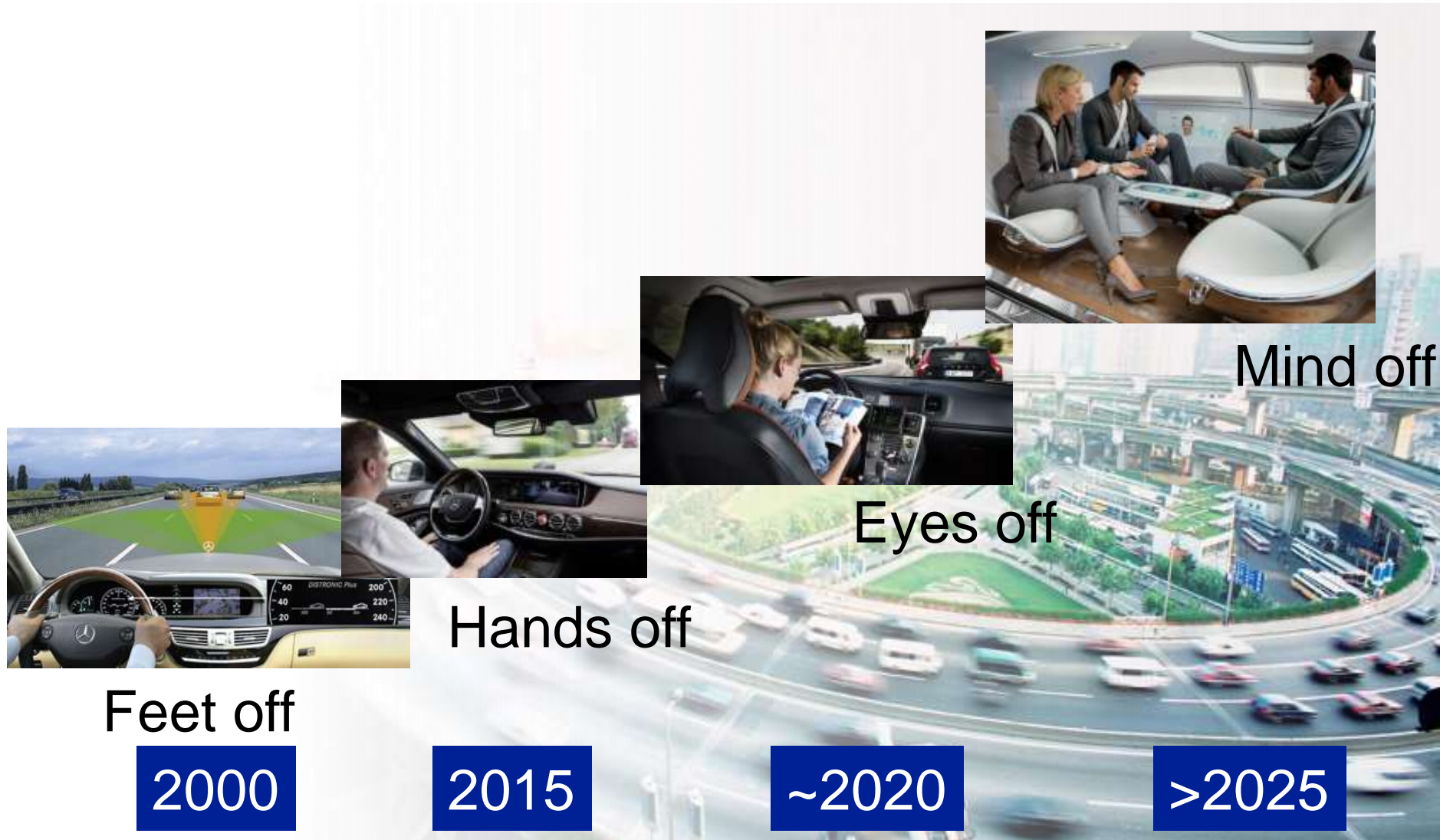
Autoliv – An Industry Pioneer for 60+ Years in Automotive Safety



Autoliv – A Complete System Safety Supplier



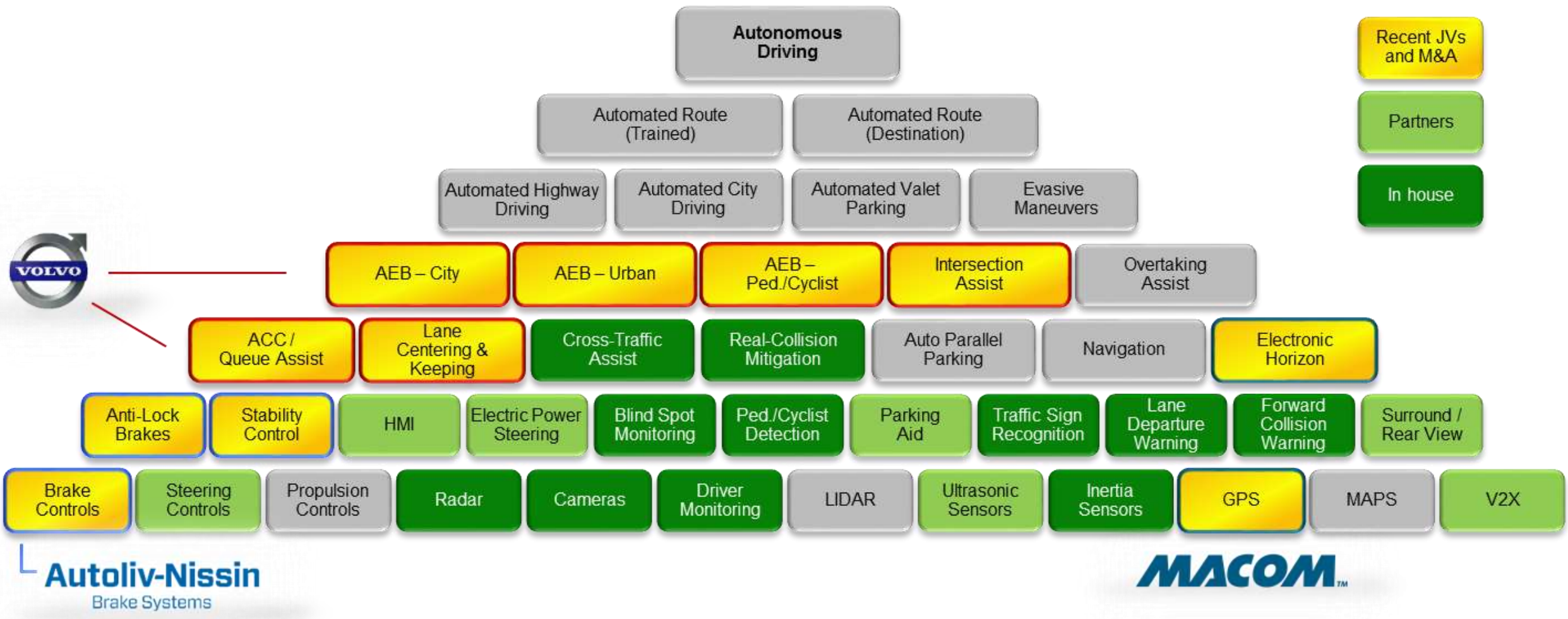
The Automated Driving System Team Roadmap



The Road to Autonomous Driving



Autoliv's Current Footprint Within the Automated Driving Pyramid



Autoliv-Nissin
Brake Systems

MACOMTM

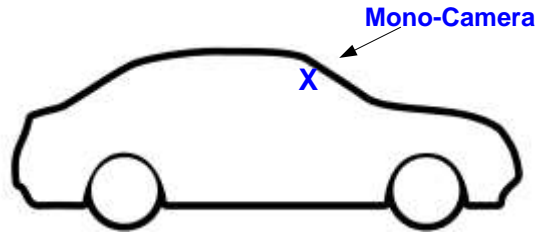
How does an OEM view us in the Domain of Automated Driving?

- Are we a radar sensor supplier?
- Are we a camera sensor supplier?
- Are we an ECU supplier?
- Are we an active safety feature supplier?
- Are we a system software supplier?
- Are we software integrators?
- Are we a full active safety system supplier?
- Are we collaborators on future system designs?

The answer is Yes to all

Example Real-Life Customer Pursuits

OEM A



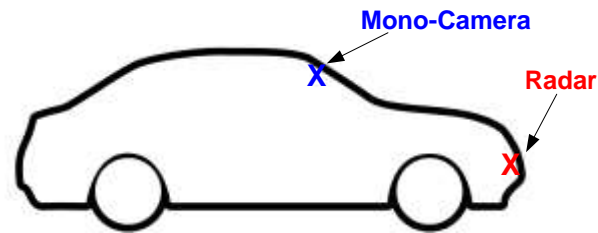
Camera: Supplier A

Feature Set: Supplier A/OEM

Integration ECU: Camera

Feature Integrator: Supplier A

OEM B



Camera: Supplier A

Radar: Supplier B

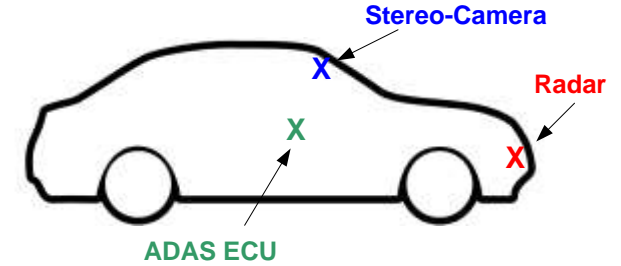
Fusion: Supplier B

Feature Set: OEM

Integration ECU: Radar

Feature Integrator: Supplier B

OEM C



Camera: Supplier A

Radar: Supplier B

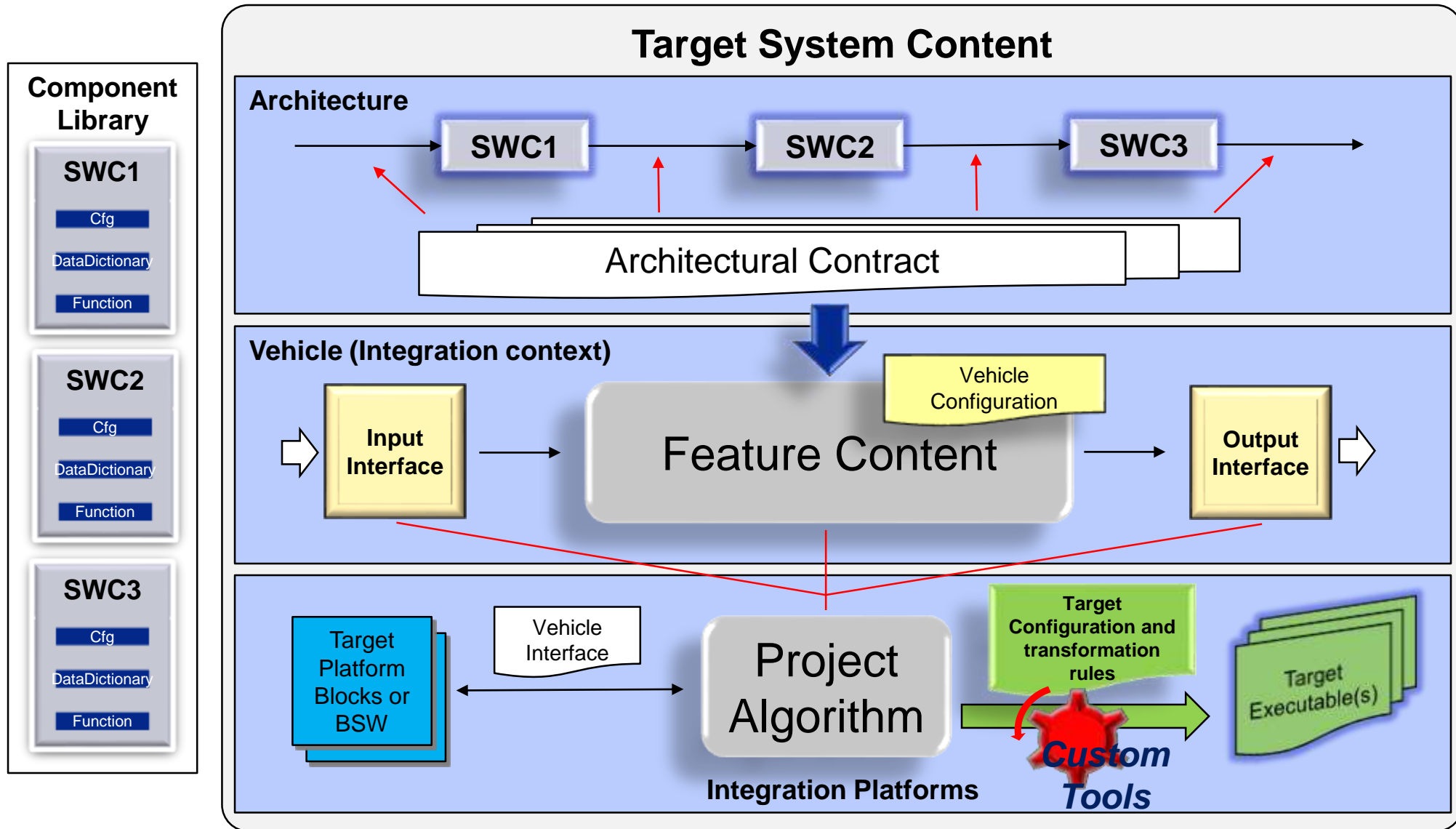
Fusion: Supplier C

Feature Set: OEM/Supplier C

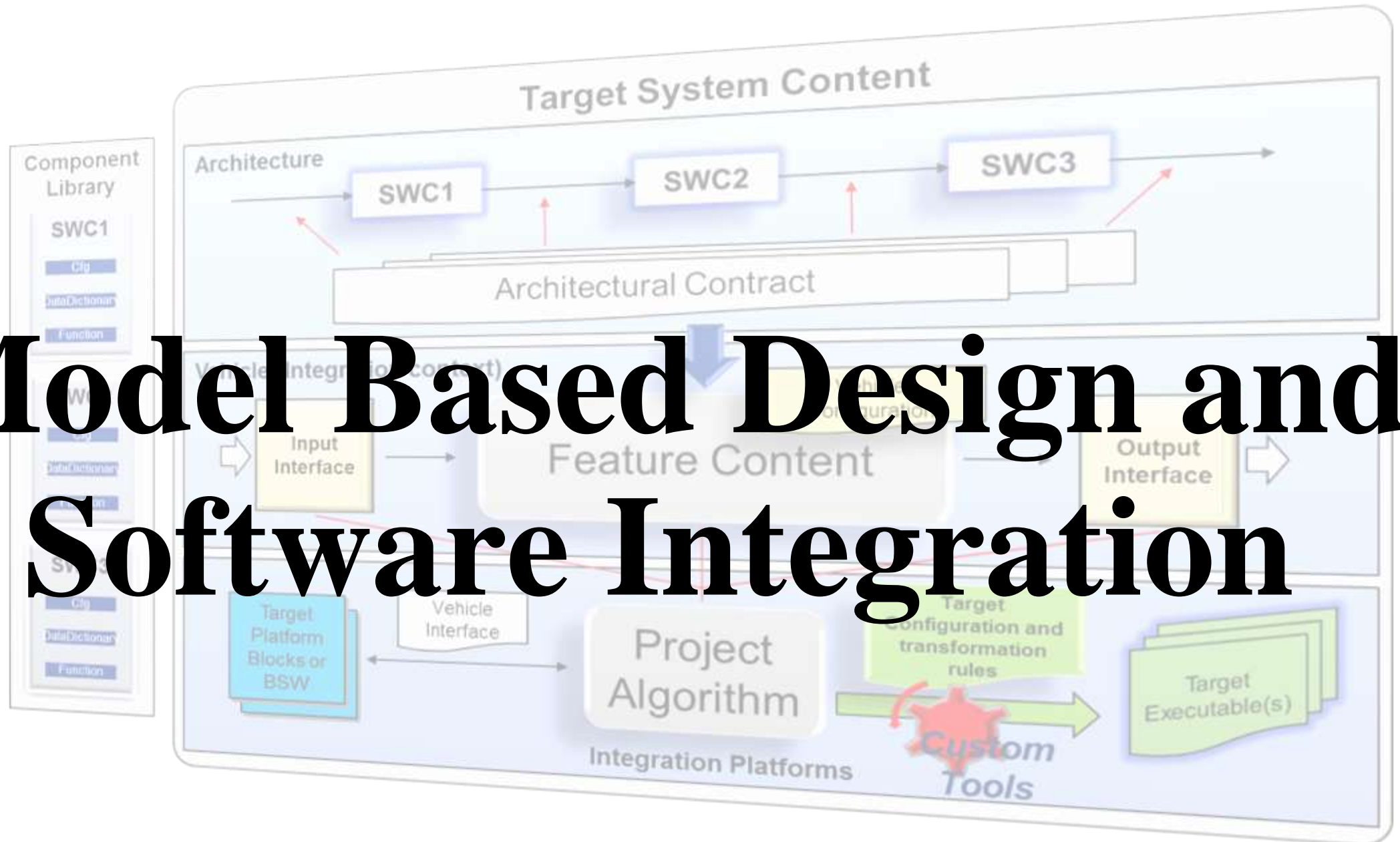
Integration ECU: ADAS ECU

Feature Integrator: Supplier C

The Autoliv Software Integration Workflow



Model Based Design and Software Integration



Challenges of Model Based Design and Software Integration

- **Multiple internal development sites across the world**
 - Local constraints on access to tools
- **Autoliv is participating in several co-development activities involving multiple external parties**
 - Bi-directional exchange of models
 - Incompatible development environments
- **A single project may see multiple integration platforms**
 - E.g. PC Simulation and replay, 3rd party simulation environments
 - E.g. Real-time platforms: RCP, production target ECU
- **Variety of Component formats for integration**
 - Simulink Models: white box and IP protected
 - C source files
 - Object files
- **Subject matter expert challenge**
 - Subject matter expertize versus “know it all”



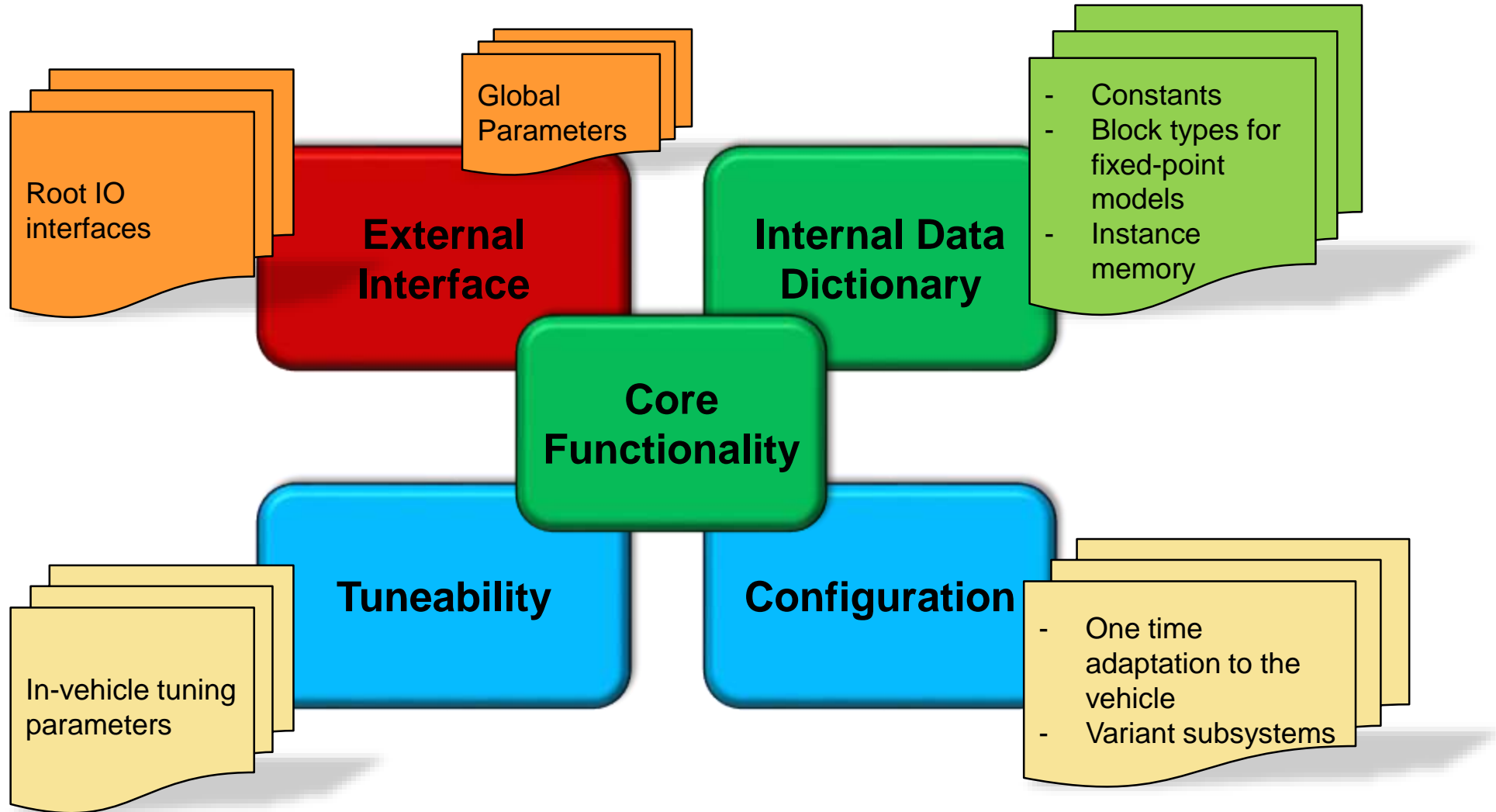
PnP

C?

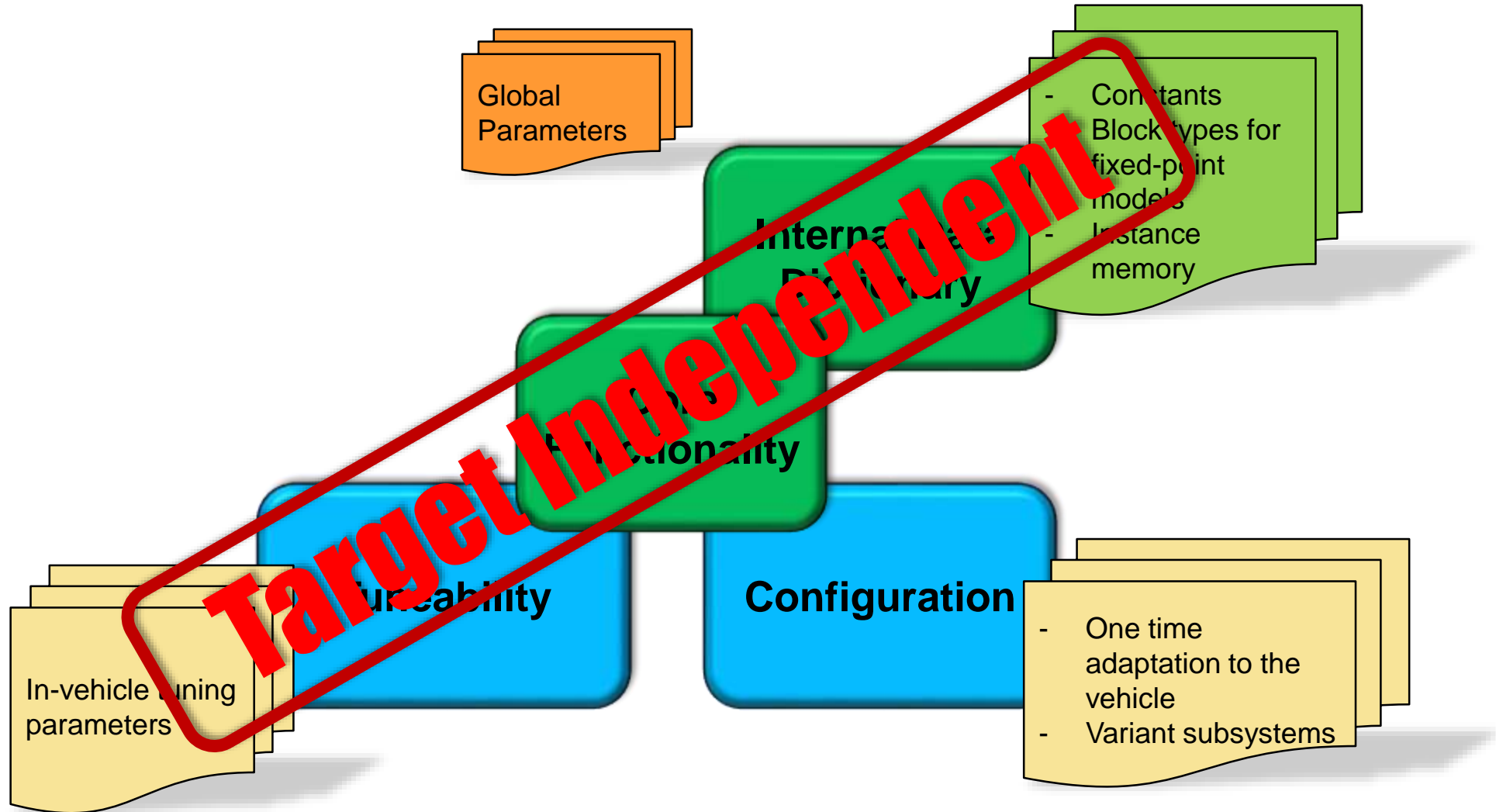
Autoliv's Approach using the MathWorks Suite

- **Packaging Internal Software Components for re-use in multiple projects**
 - Explicit boundary and external dependency
 - Clear separation between the function and the data
- **Establishing a framework for multi-site development of feature content**
 - Uniform MBD project setup with a foundation in common and portable project configuration/build system
 - Scalability: Not every development site will need a full project toolset
- **Supporting multiple integration platforms**
 - Target independence in defining a component functionality and data
 - Custom toolset for mapping component functionality and data onto a target platform
- **Collaborating with external companies**
 - Flexibility in accepting model formats and content packages from external collaborators
 - Provisioning for mapping external deliveries to the selected targets
- **Encouraging subject matter expertise**
 - Let the experts concentrate on what they know and do the best

Software Component Packaging



Software Component Packaging

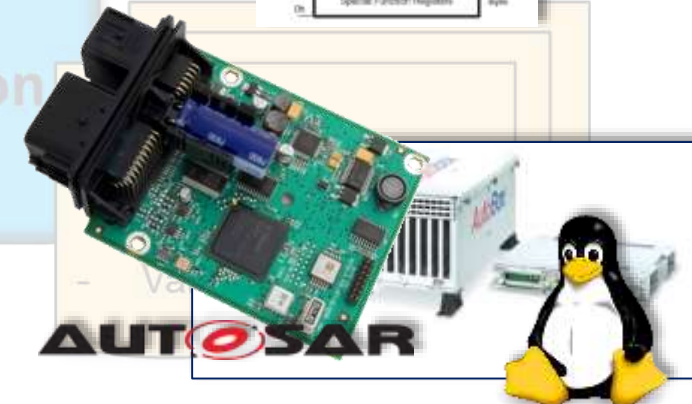
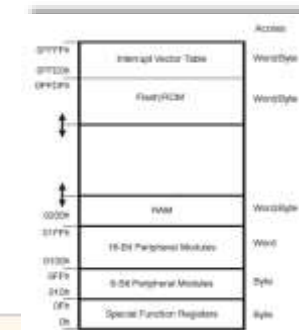
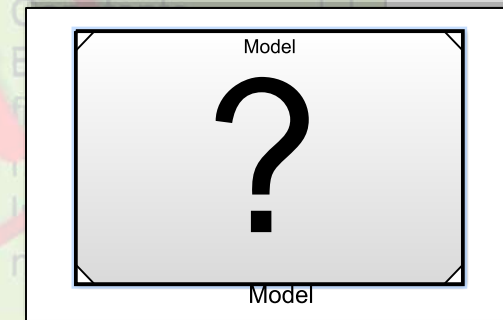


What is Target Independence?

- **The component owner should primarily care about its design & functionality**
 - Proper representation of the execution model: E.g. floating point versus fixed point designs
 - Simulink-based component is delivered without the assumption of an integration environment
- **Enforcing adherence to internal modeling standards**
- **All component relevant data sets are defined in the generic form**
 - E.g. generic Matlab variables (discouraged)
 - E.g. Simulink.Parameter objects without specification of Custom Storage Classes
- **Existence of well established transformation rules**
 - E.g. Mapping the data and functions onto the various targets
 - E.g. Custom code generation with standardized build toolset

Software Integration of Target Independent Models

- **Standardized code-generation toolset should support:**
 - **Adaptation to incompatible external interfaces**
 - Model reference is a good integration unit but...
 - Is it a good re-use unit?
 - **Flexible target memory allocation**
 - E.g. End-Of-Line calibration
 - E.g. Inline or non-inline constant section
 - E.g. Non-volatile memory
 - **Ability to transform models into target platform compatible code**
 - E.g. Real-time RCP targets vs. target ECU
 - E.g. AUTOSAR vs. non-AUTOSAR targets



Example Success Stories



- The presented methodology has been successfully applied to the following Autoliv products:
- Passive Restraint System
- Variants of ESC/ESB systems
- Automated Driving applications
 - Mono-Vision AEB System with internal SW components
 - Forward Looking Radar based ACC with external SW components
 - Best in-class ADAS system with Mono-Vision Camera, Forward Looking Radar and a combination of mixed internal and external SW components

Thank You!



Autoliv

Every year, Autoliv's products save over 30,000 lives

and prevent ten times as many severe injuries