

“Advancing Traffic Efficiency and Safety  
through Software Technology”

## *Variability Modeling of the EAST-ADL2*

*Open Workshop March 2008*

*R. Tavakoli Kolagari, M.-O. Reiser*



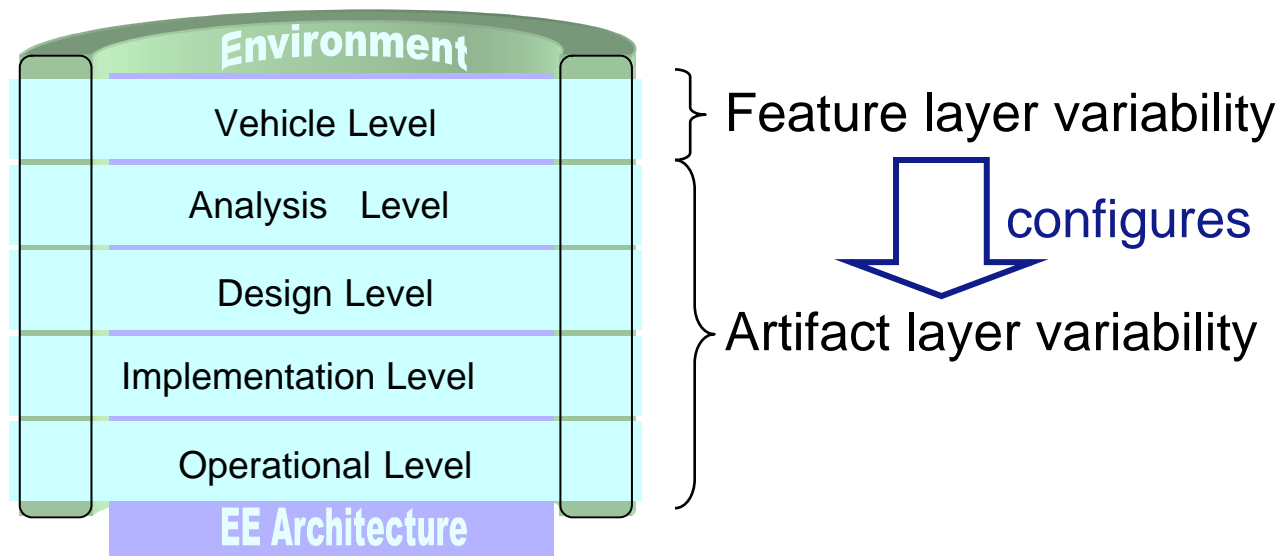
# Overview

1. Variability Overview
2. Basic Concepts
3. Advanced Concepts

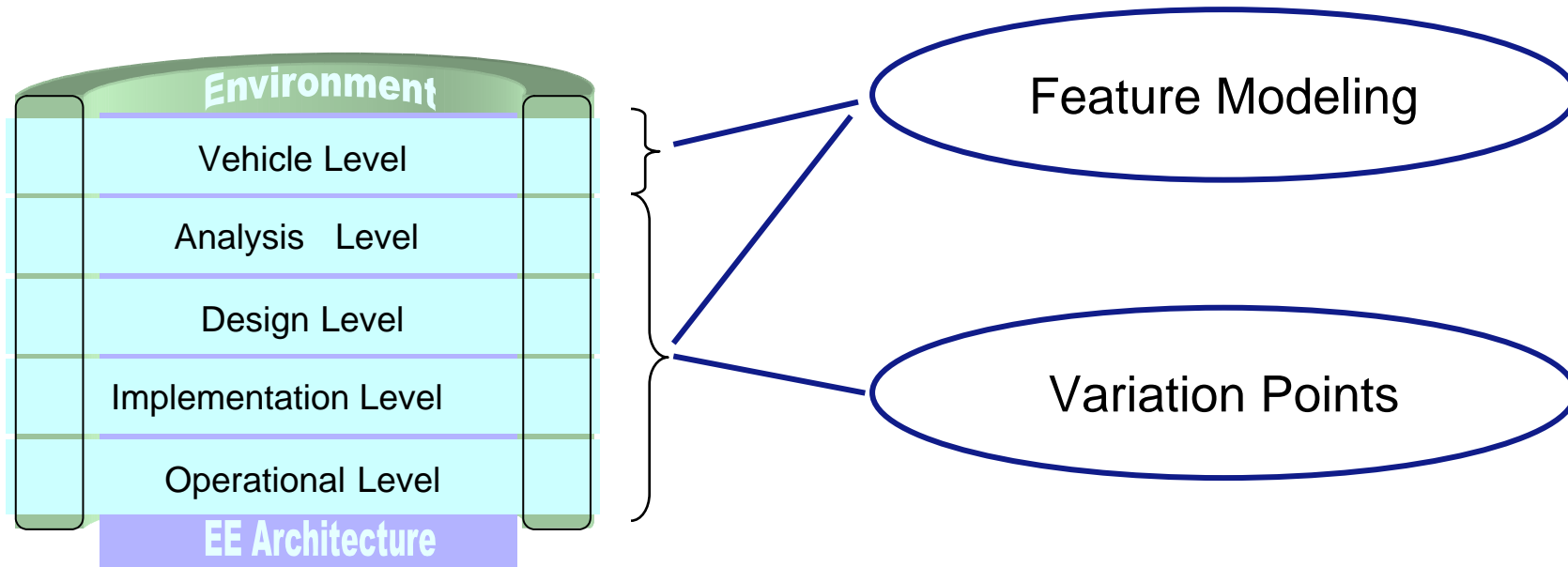
# Variability – Overview

Variability is modeled essentially on two different abstraction layers:

1. Feature layer variability (being the primary source for variant/product configuration)
2. Artifact layer variability (comprising all artifact elements, e.g. requirements, FAA, FDA...)

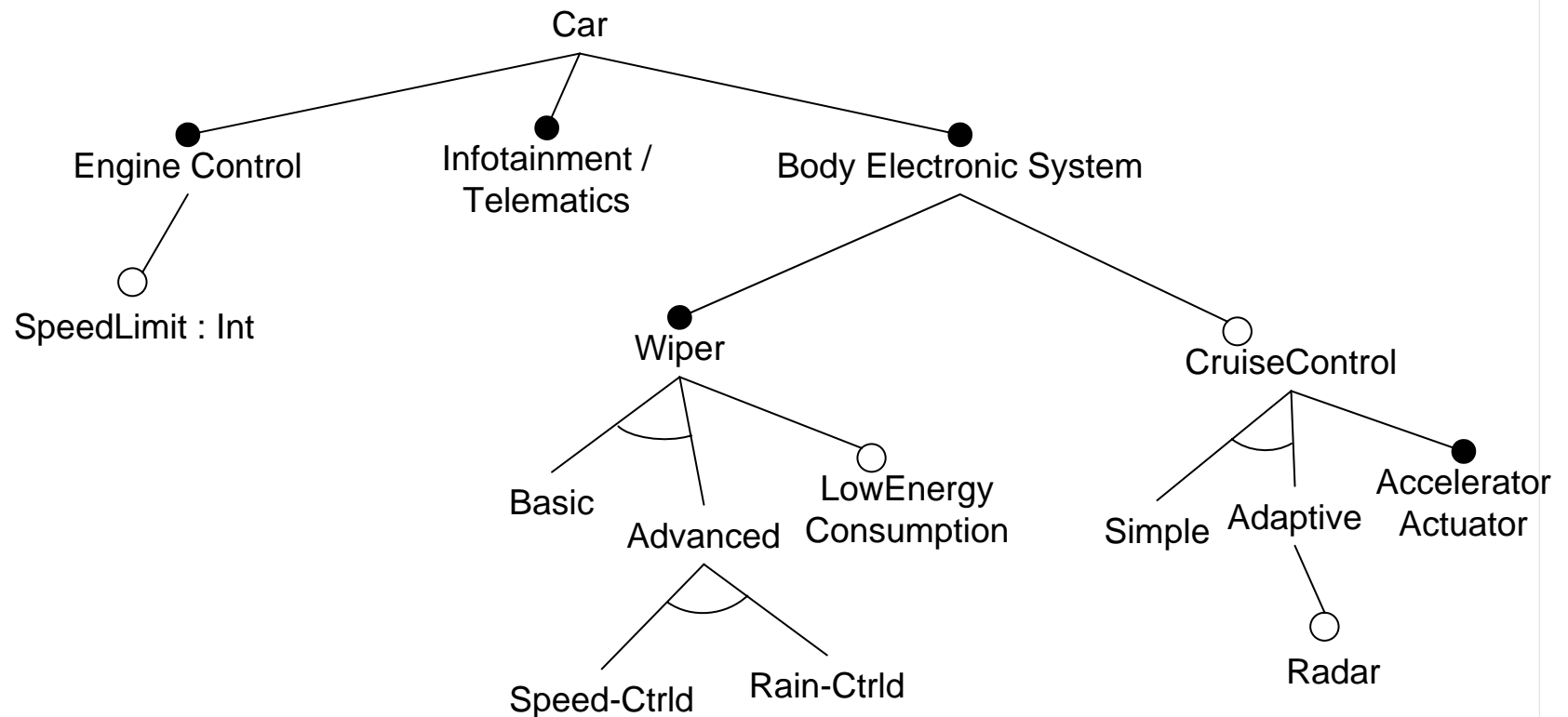


# Basic Concepts



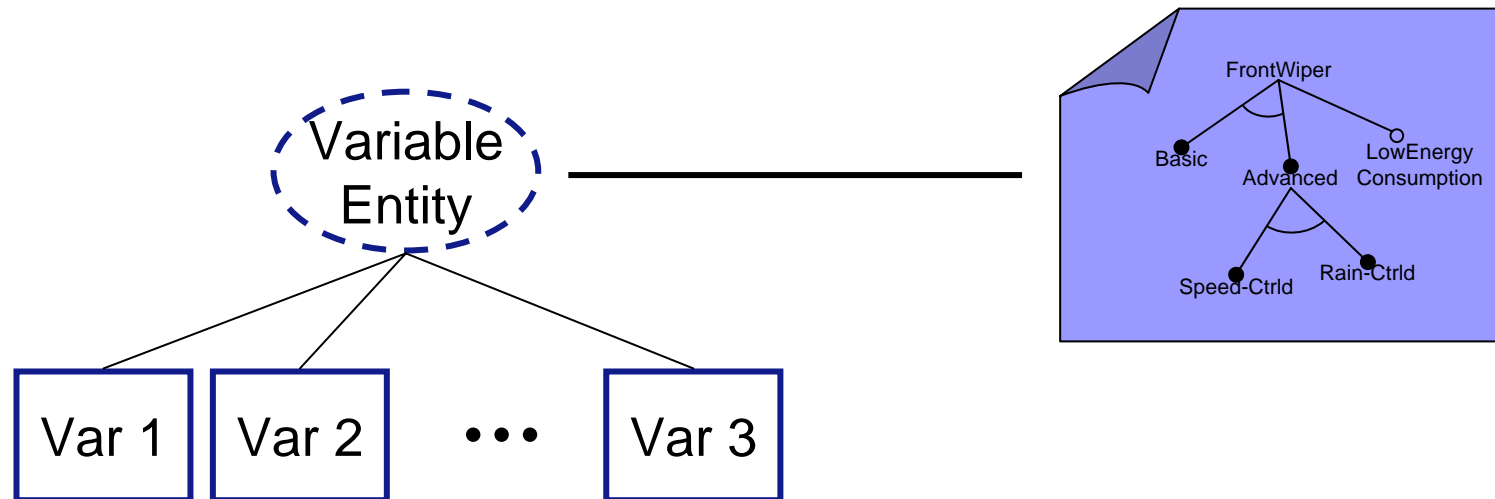
# Basic Concepts – Feature Modeling

- as introduced by Kang et al. in 1990

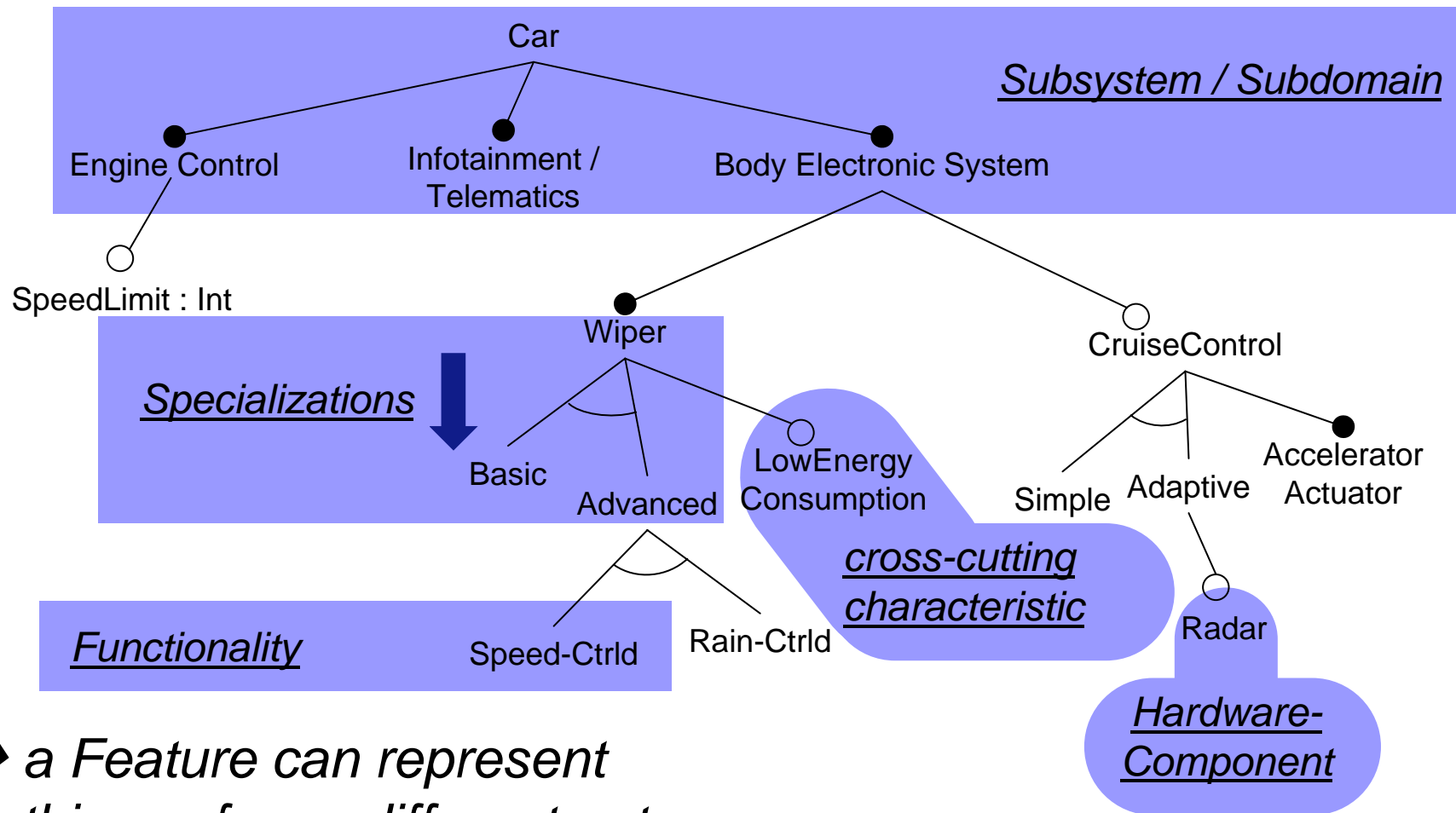


# What is a Feature ?

*A Feature is a characteristic or trait that the variants of a variable entity may or may not have.*



# What is a Feature ?



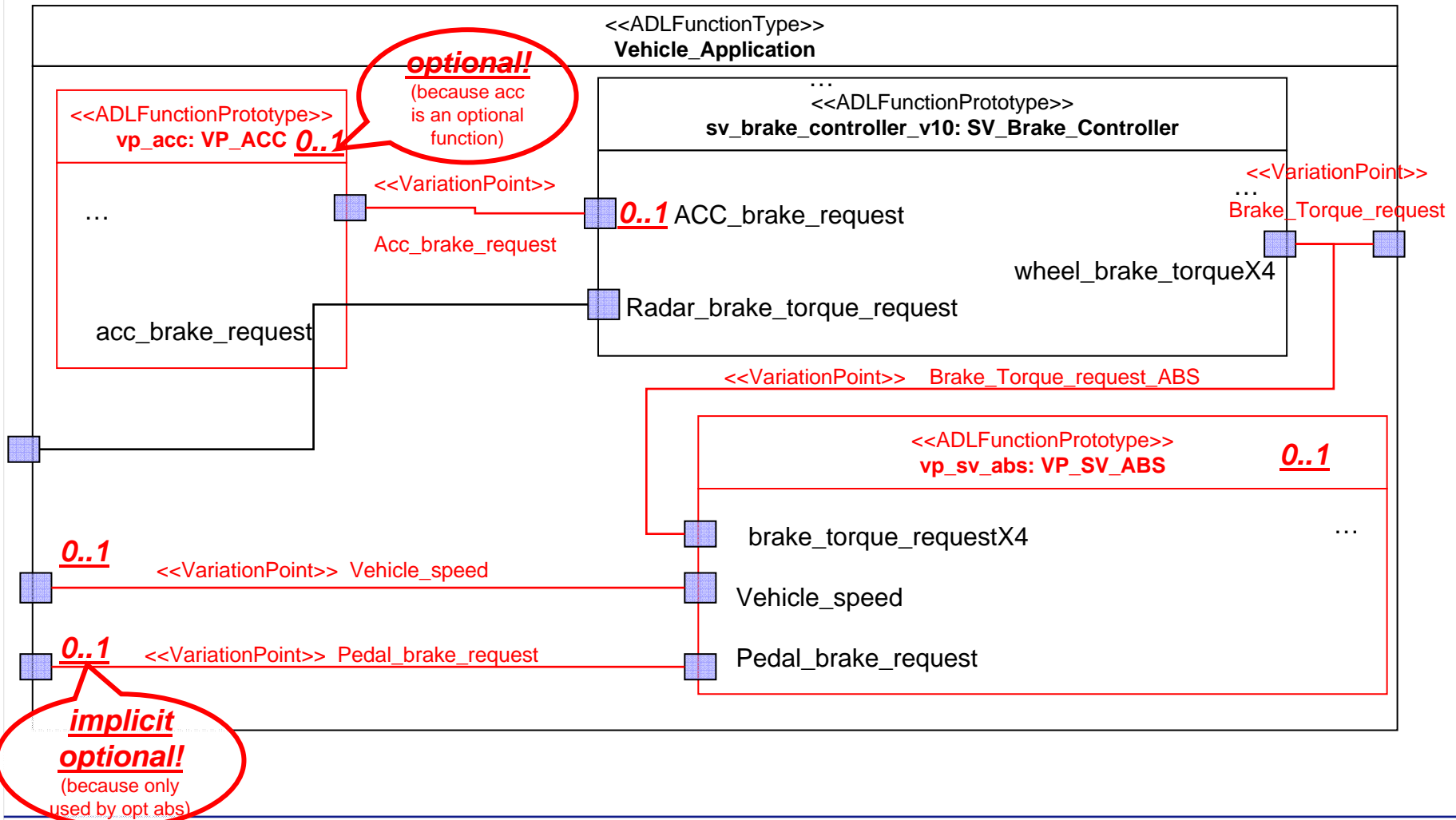
→ a Feature can represent things of very different nature

# Basic Concepts – Variation Points

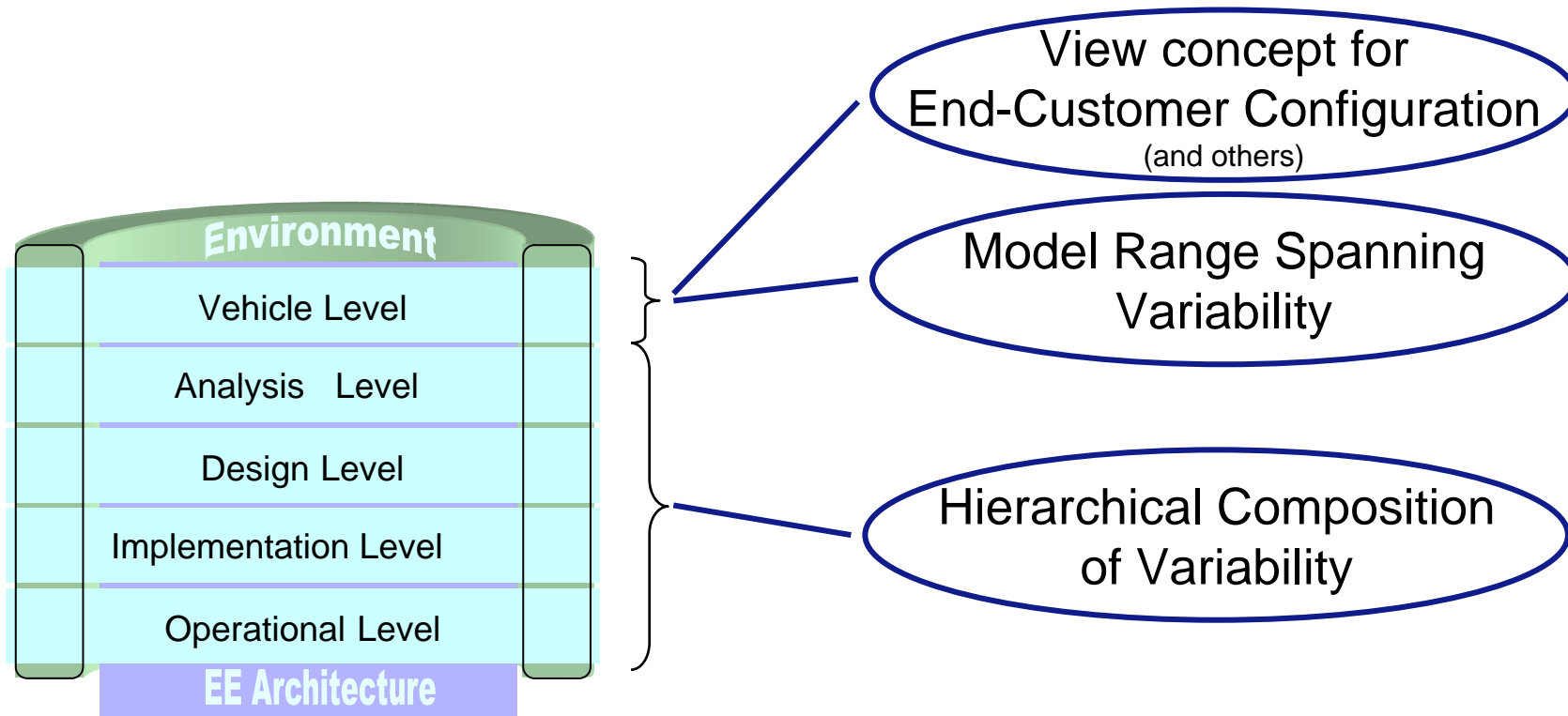
Variability is modeled for the artifacts based on the notion of variation points and variation groups:

- all ADLEntities can be variable
- variable ADLEntities are marked a variation point (and then serve as a placeholder)
- the variation point is linked to the entities that represent the available variants
- variation groups can describe various constraints between arbitrary variable entities

# Variation Point – Example



# Advanced Concepts



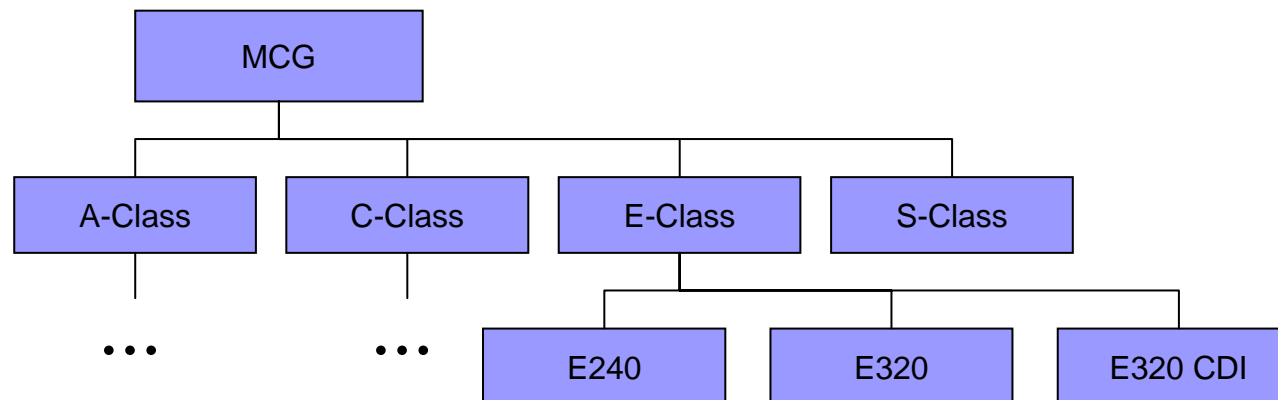
# End-Customer Configuration

- vehicle level contains the core feature model
  - showing variability of the complete system
  - high complexity
  - technical viewpoint  
(terminology, customer-invisible variability, diverse life-cycle)
  - not appropriate for end-customer configuration
- vehicle level supports to define end-customer configuration
  - provides „view“ on core feature model
  - allows for orthogonal „packaging“ of variability
  - supports orthogonal configuration considerations
  - can be used for separation of other concerns

# Model Range Spanning Variability

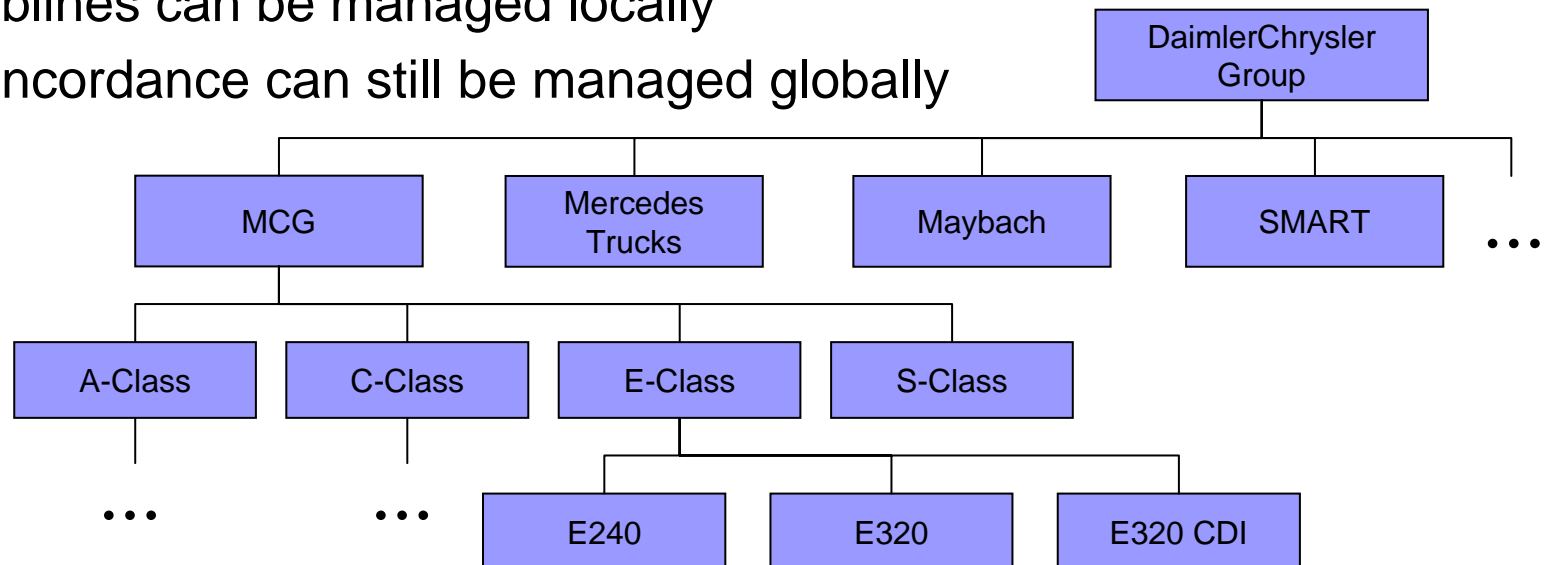
two kinds of variability:

- model range specific variability
- model range spanning variability

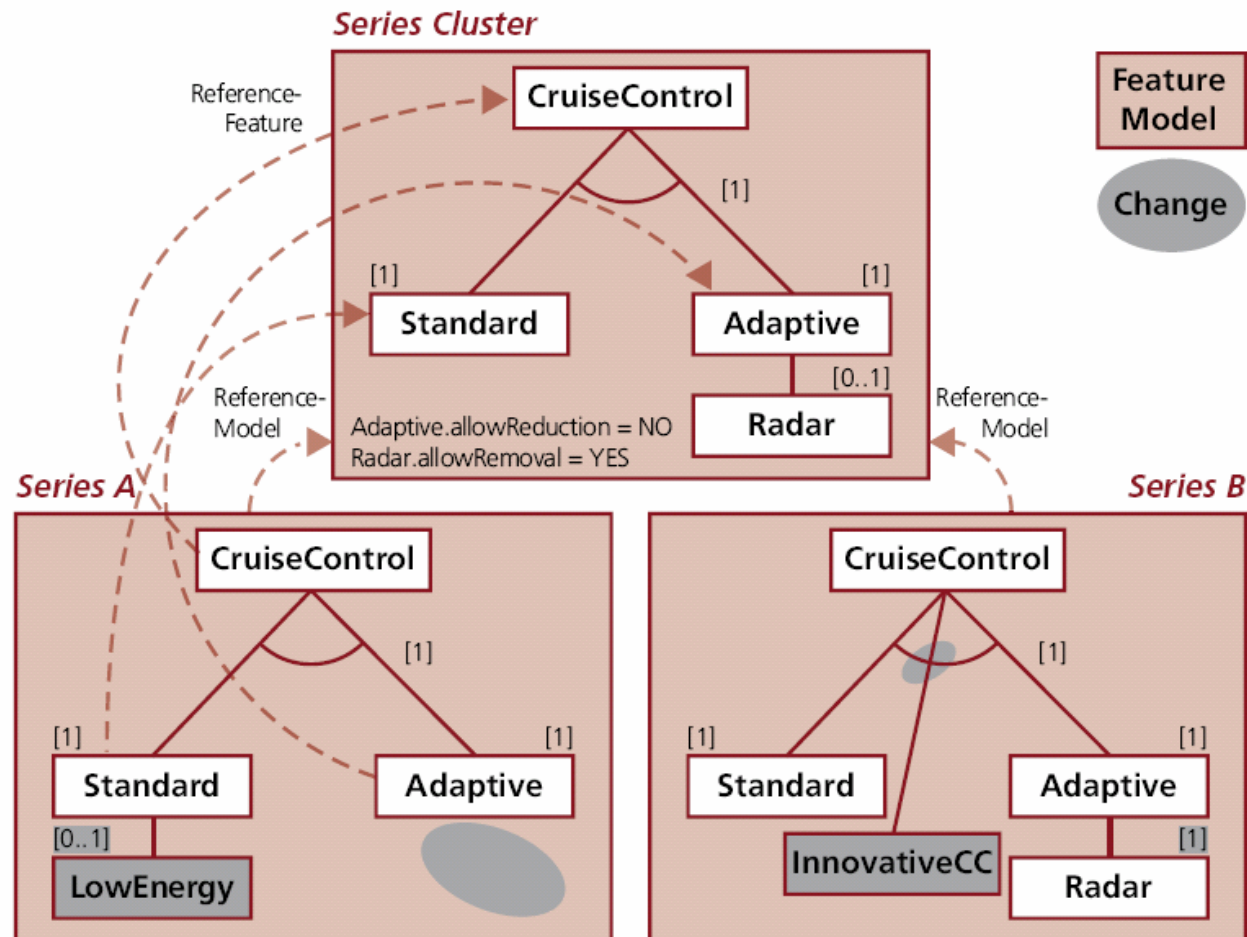


# Model Range Spanning Variability

- traditional solution: either one large software product line or several independent ones
- multi-level concept: compromise between the two
  - sublimes can be managed locally
  - concordance can still be managed globally

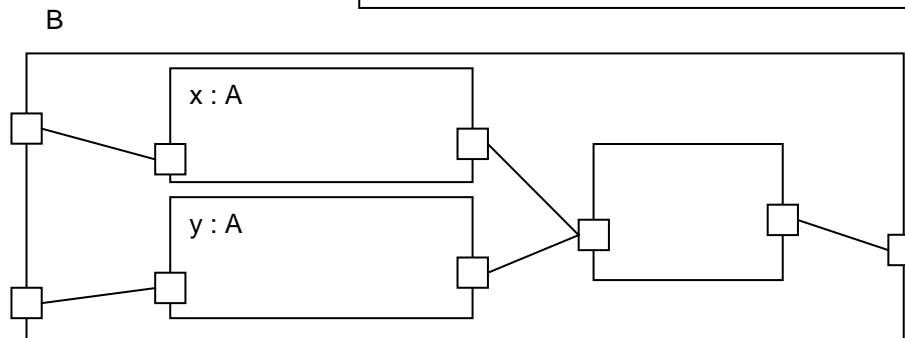
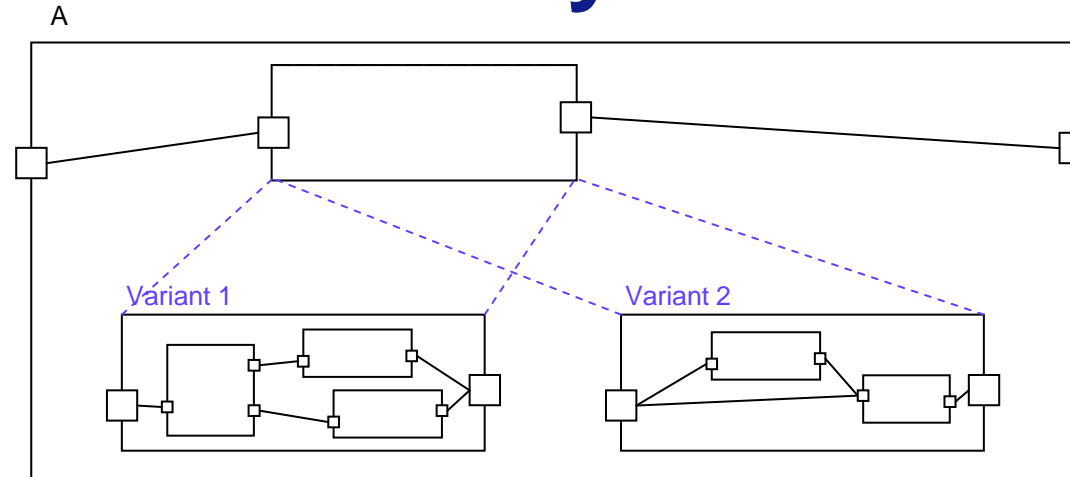


# Multi-Level Feature Model – Example



# Composite Variability – Motivation

A has  
variability  
inside.



When A is used twice in B,  
then it must be configured  
separately for each  
occurrence.

→ Thus, definition of A is not a suitable context for configuration !!

# Composite Variability

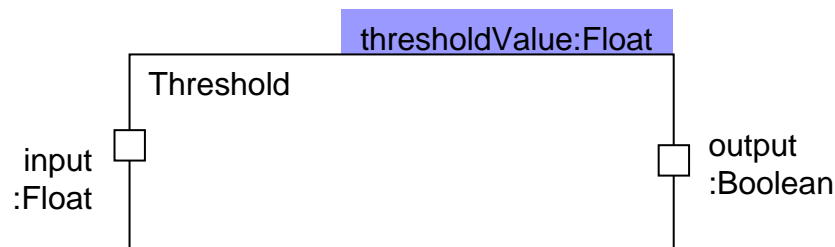
- variability is managed in a hierarchical manner
- key elements:
  - Step 1: ADLFunctions get a „public“ feature model
  - Step 2: internal structure of composite ADLFunctions can be variable (by way of variation points)
  - Step 2: for each composite ADLFunction a mapping is defined from its public feature model to ...
    - (a) the public feature models of its contained ADLFunctions and
    - (b) the variants of the contained variation points

(in other words: it is defined how the configuration of the contained ADLFunctions' feature models can be derived from a given configuration of the public feature model of the container ADLFunction)

## Step 1: ADLFunctions get „public“ FM

- as part of their public interface
- for elementary ADLFunctions, the configuration of the public feature model will be made available for reference within the behavioral description of the ADLFunction
  - details depend on type of behavioral description
  - e.g. for C-Source-Code, parameters can be realized as pre-processor macros

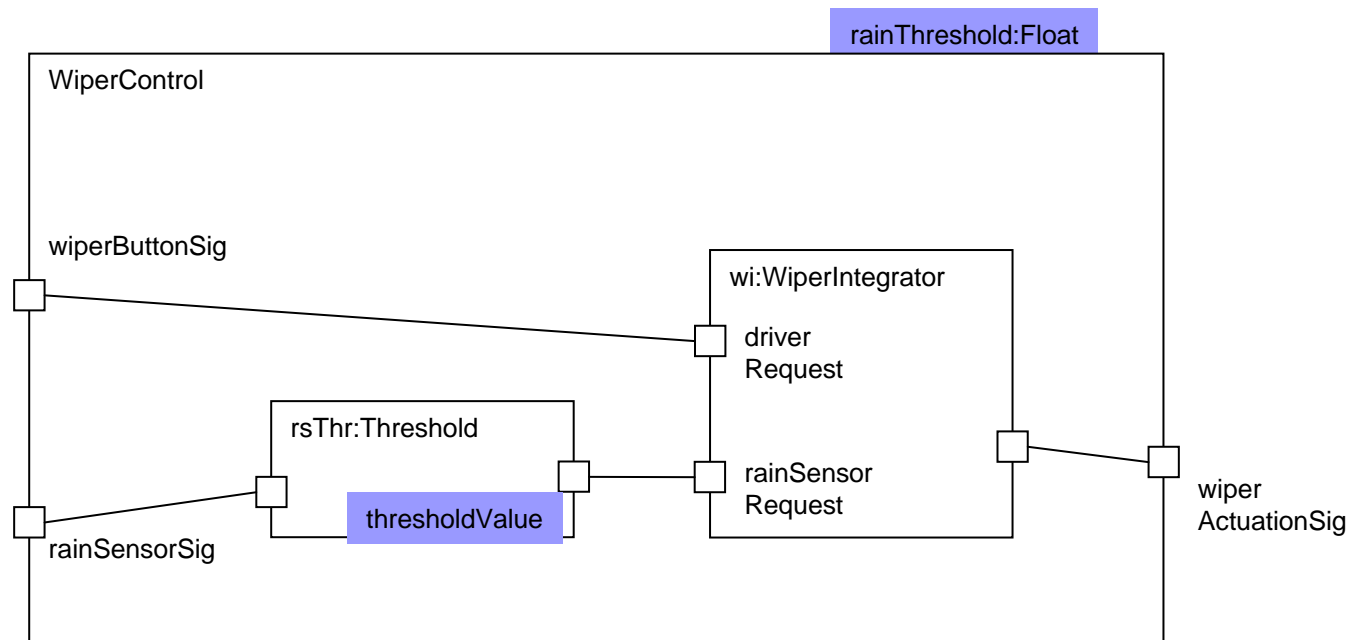
EXAMPLE: ElemSWComp that realizes a „Threshold“



if input  $\geq$  thresholdValue  
then output = True  
else output = False.

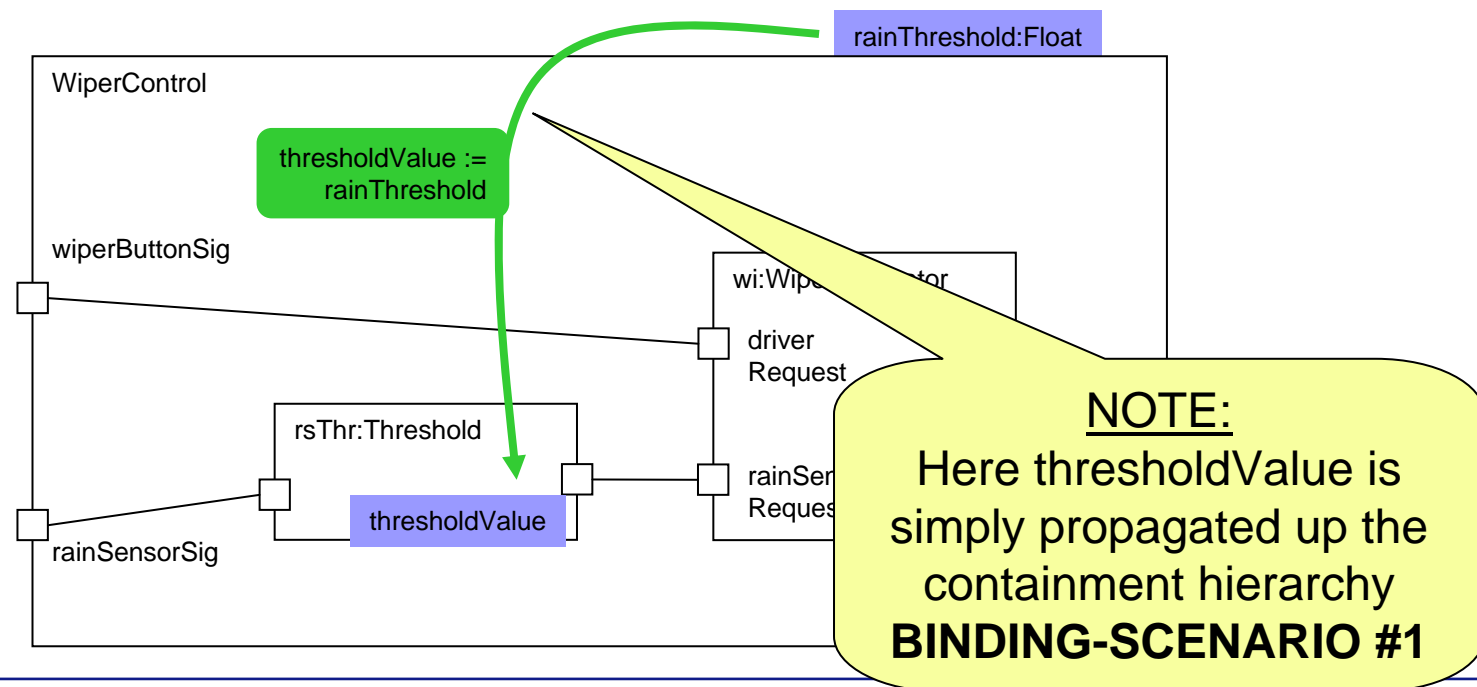
# Step 1: cont'd

EXAMPLE: CompSWFunc that controls a wiper depending on driver input and rain sensor

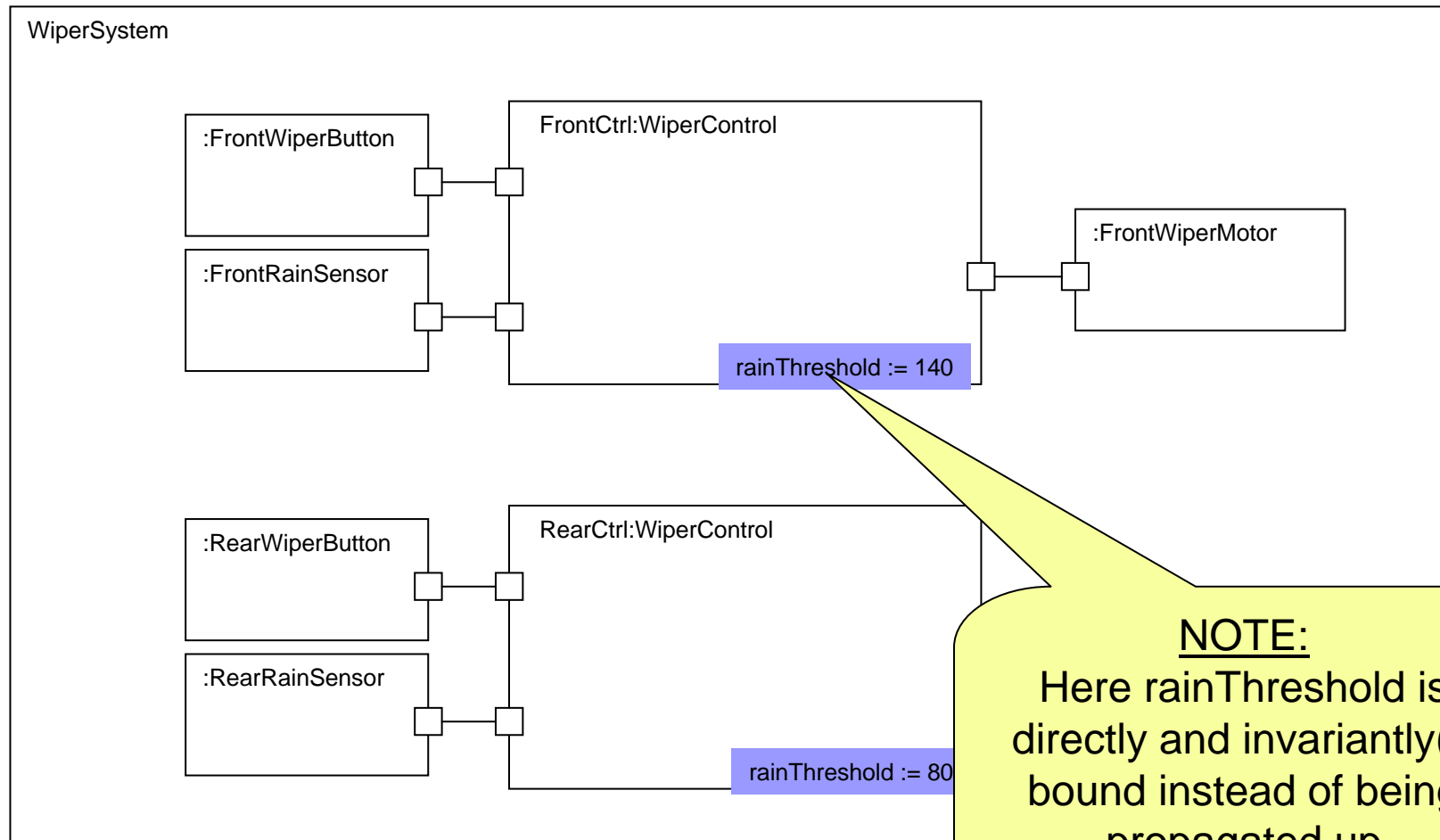


## Step 3: Mapping

- mapping from the public feature model of a composite ADLFunction to
  - (a) the public feature models of its contained ADLFunctions and
  - (b) the variants of the contained variation points
- in this case: only (a) used

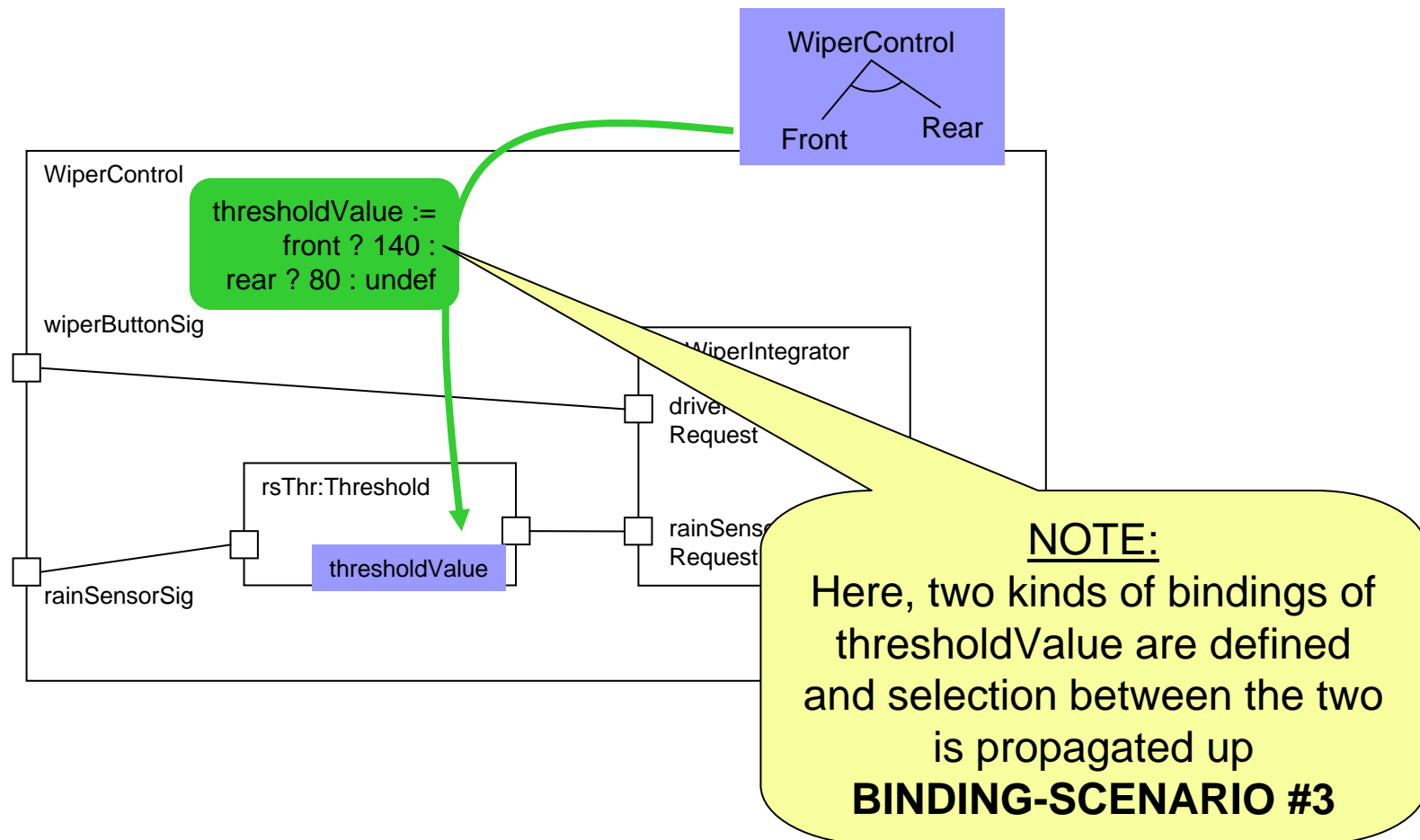


# Applying WiperControl

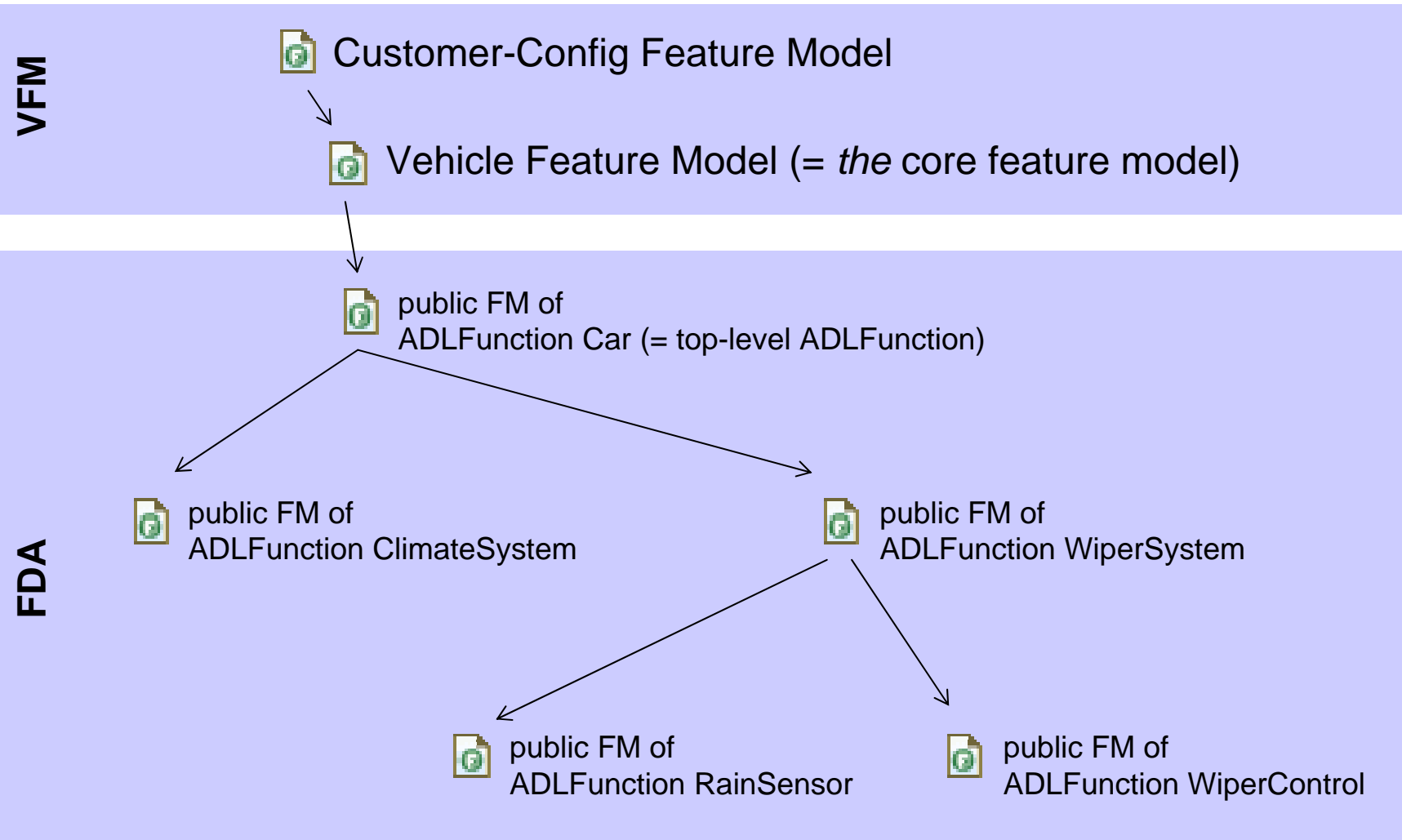


**NOTE:**  
Here rainThreshold is directly and invariantly(!) bound instead of being propagated up  
**BINDING-SCENARIO #2**

# Alternative



# Feature Models in ATESST



# Summary

- Variant handling based on feature modeling
- Supports end-customer configuration
- Model range spanning variability
- Hierarchical composition of variable entities
- Variability modeling within the architecture (variation points)

# Backup





