SEVENTH FRAMEWORK PROGRAMME

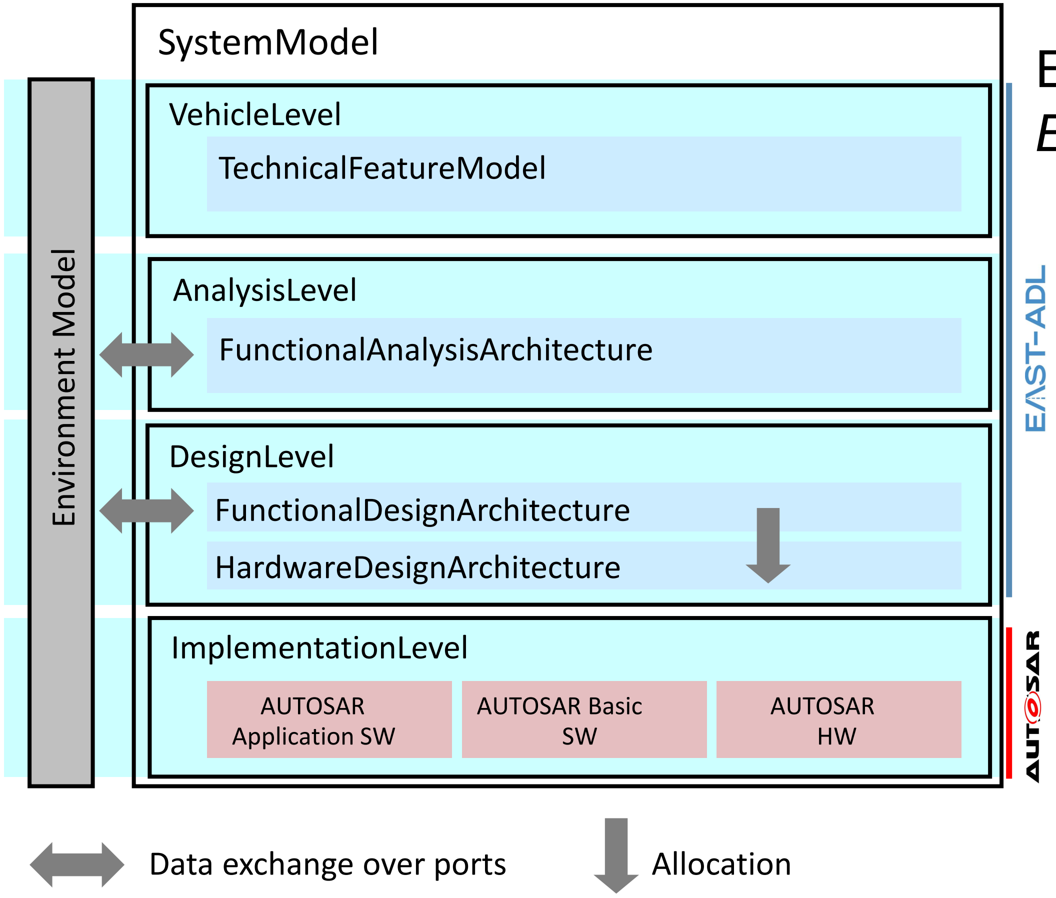
EUCAR
EUROPEAN COUNCIL FOR AUTOMOTIVE R&D

MAENAD

EAST-ADL Introduction

Support for ISO26262

EAST-ADL Overview

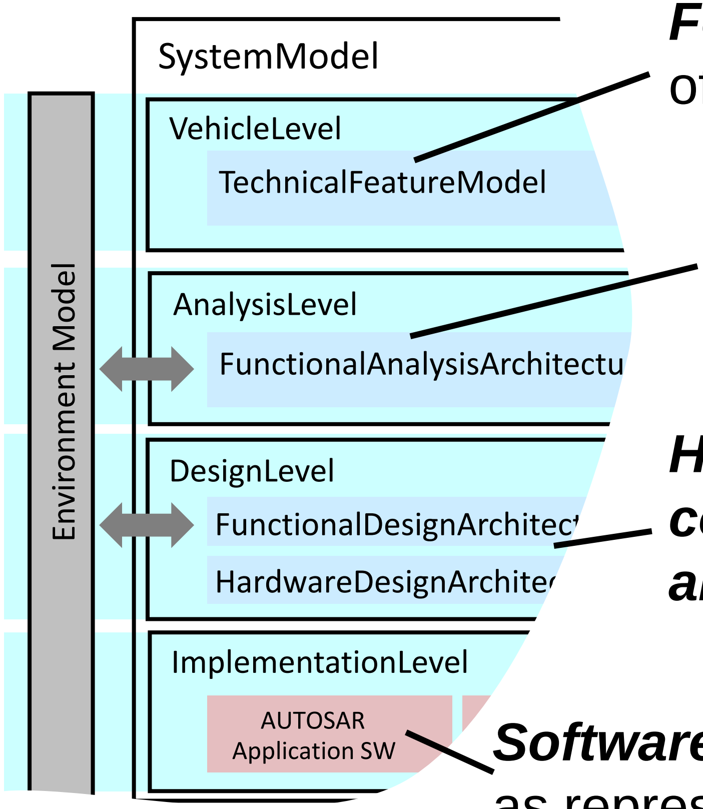


EAST-ADL defines an *Engineering information structure*

- Feature content
- Functional content
- Software architecture

- *Requirements*
- *Variability*
- *Safety information*
- *V&V Information*
- *Behavior*

EAST-ADL+AUTOSAR Representation

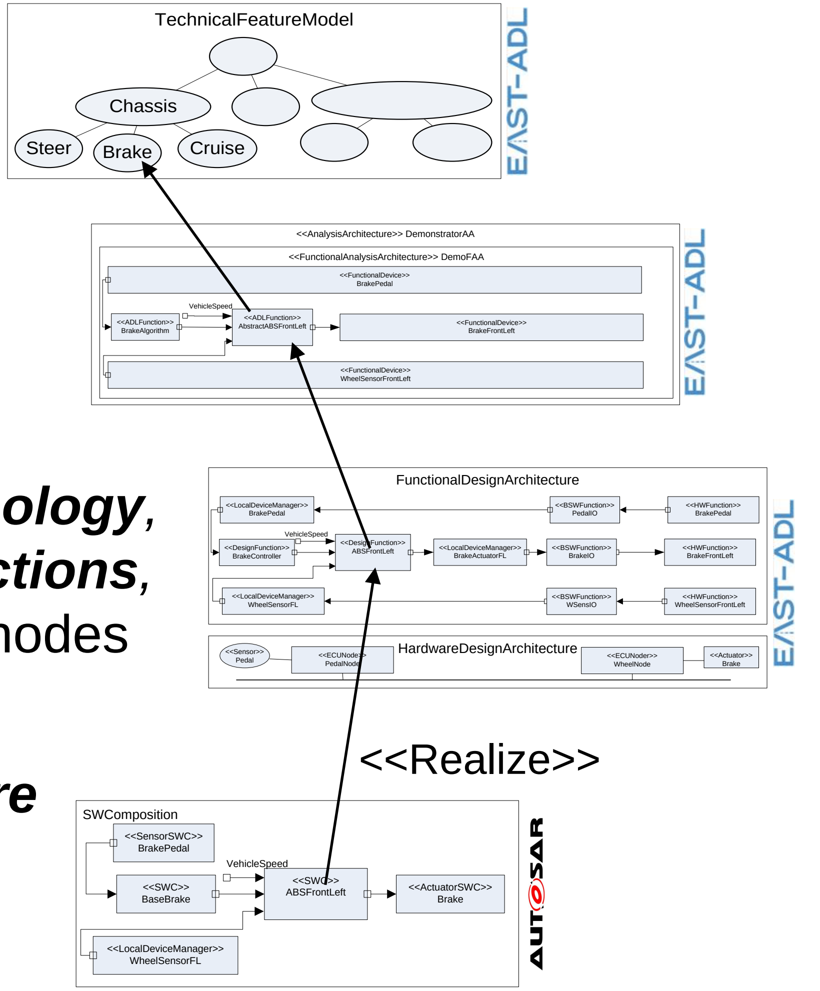


Features of the vehicle

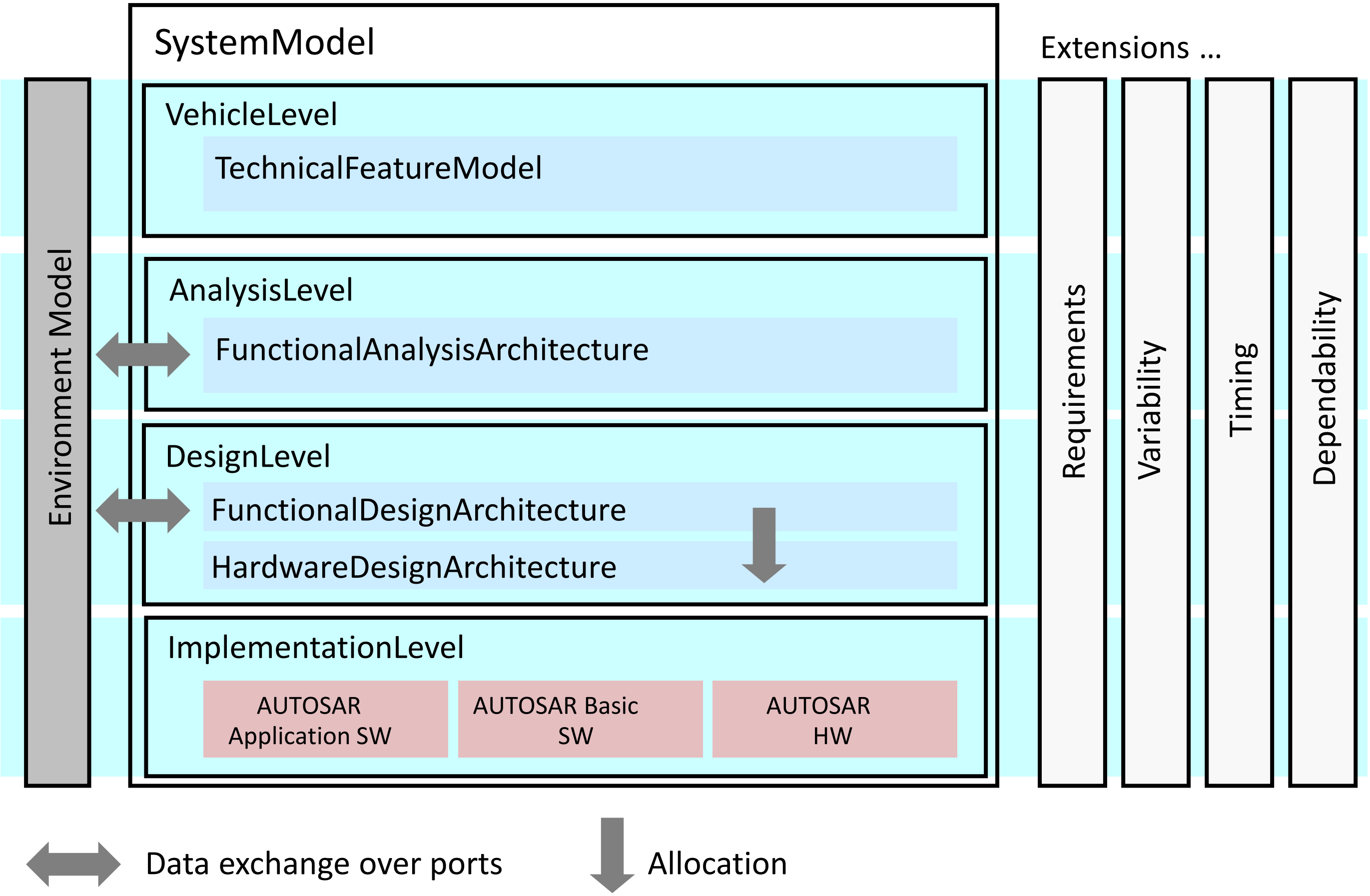
Abstract functions

Hardware topology, ***concrete functions***, ***allocation*** to nodes

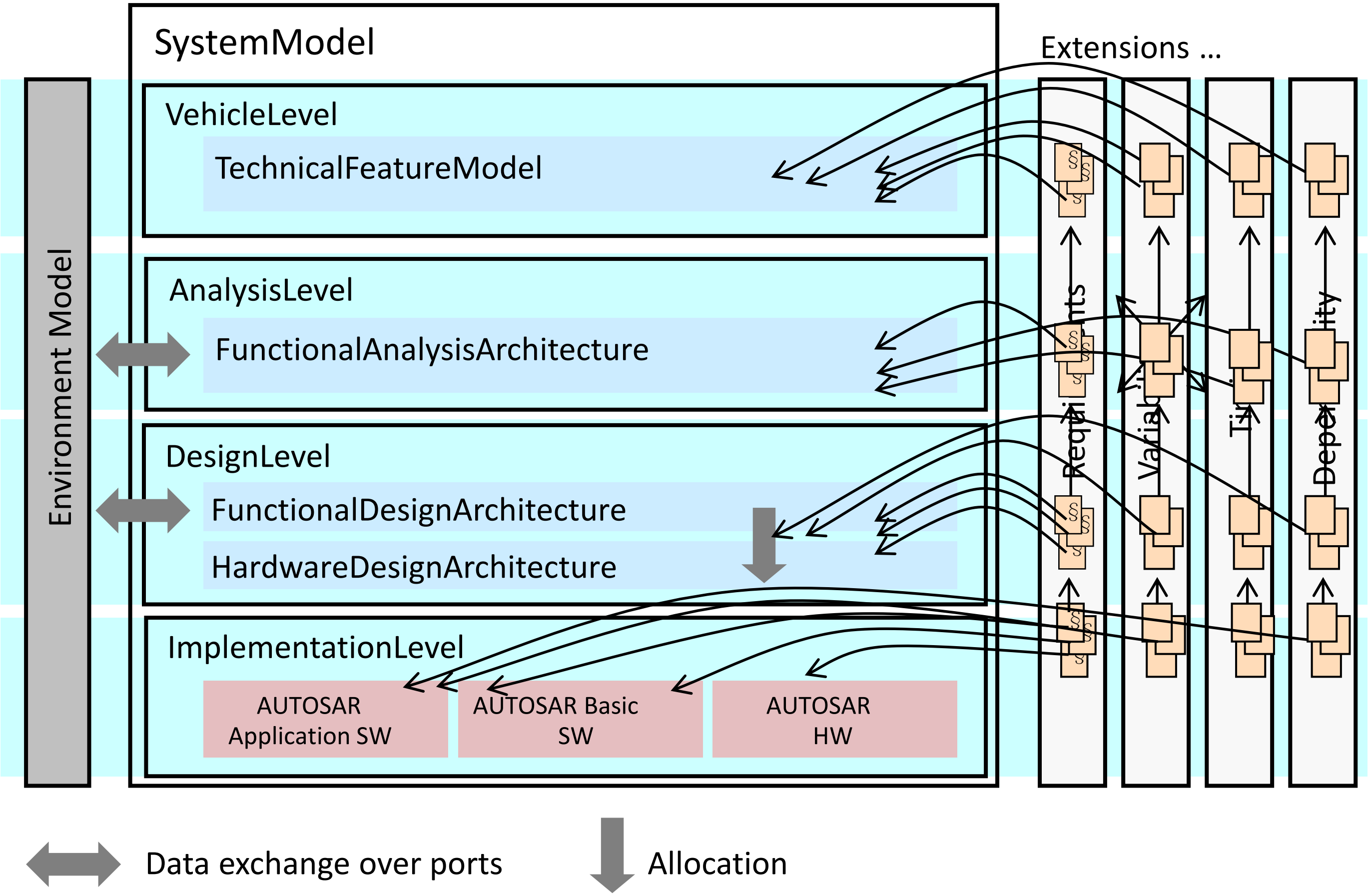
Software Architecture as represented by AUTOSAR



EAST-ADL Extensions



EAST-ADL Extensions



EAST-ADL vs AUTOSAR

EAST-ADL

For Features, Functional Architecture and Topology

AUTOSAR

For Software Architecture and Execution Platform

EAST-ADL vs AUTOSAR

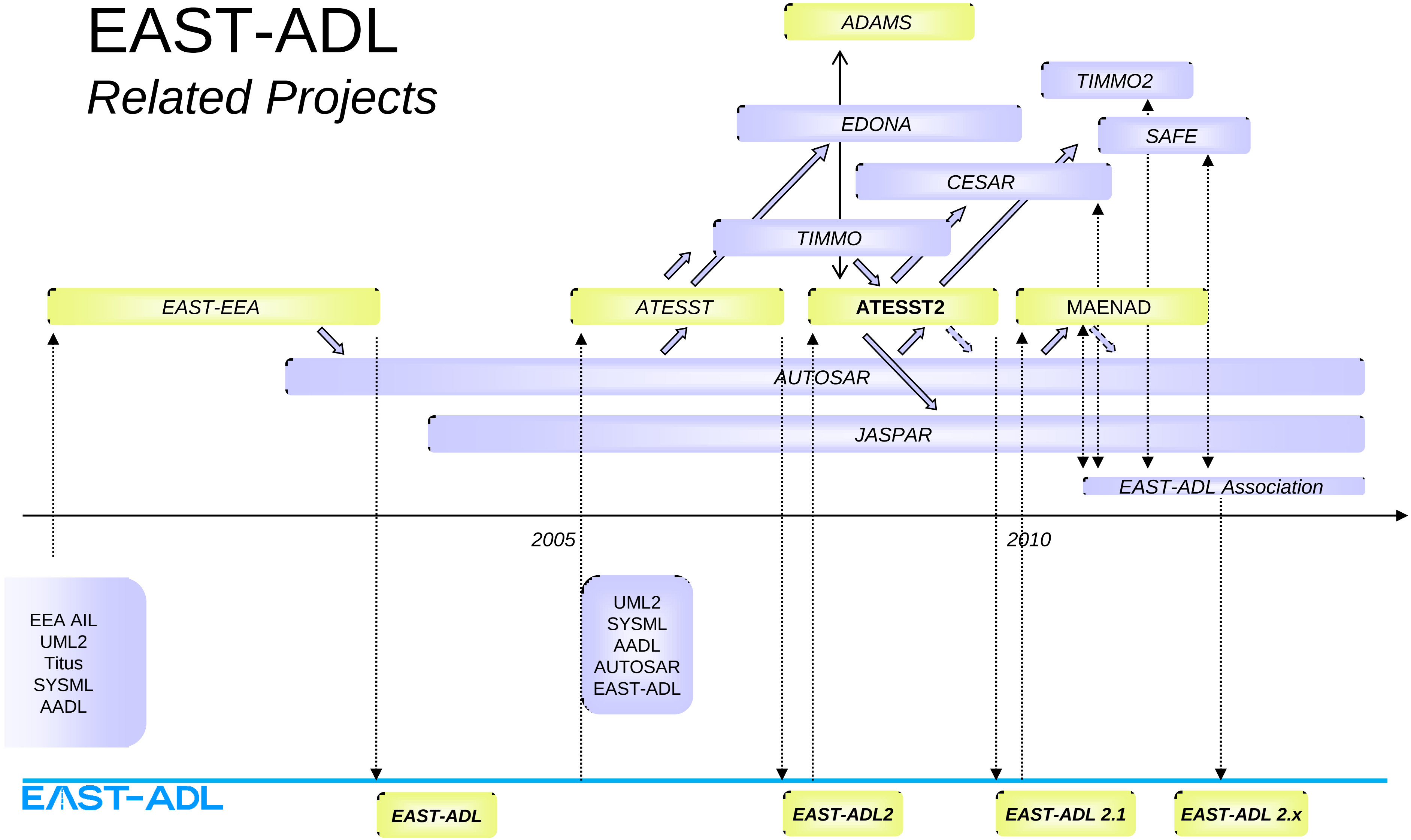
- Different Abstraction Levels:
  - EAST-ADL complements AUTOSAR with “early phase” information
- Different Engineering Information Scope:
  - EAST-ADL complements AUTOSAR
    - Requirements Engineering
    - Variant Management
    - Behaviour (nominal/error)
    - Timing
    - Safety

Scope in AUTOSAR depending on version

- Same Meta-Metamodel
  - Enterprise Architect model used for both
  - Same file exchange ARXML-EAXML
  - Same tool infrastructure possible ARTOP-EATOP

EAST-ADL

Related Projects



ADAMS

TIMMO2

EDONA

SAFE

CESAR

TIMMO

EAST-EEA

ATESST

ATESST2

MAENAD

AUTOSAR

JASPAR

EAST-ADL Association

2005

2010

EEA AIL
UML2
Titus
SYSML
AADL

UML2
SYSML
AADL
AUTOSAR
EAST-ADL

EAST-ADL

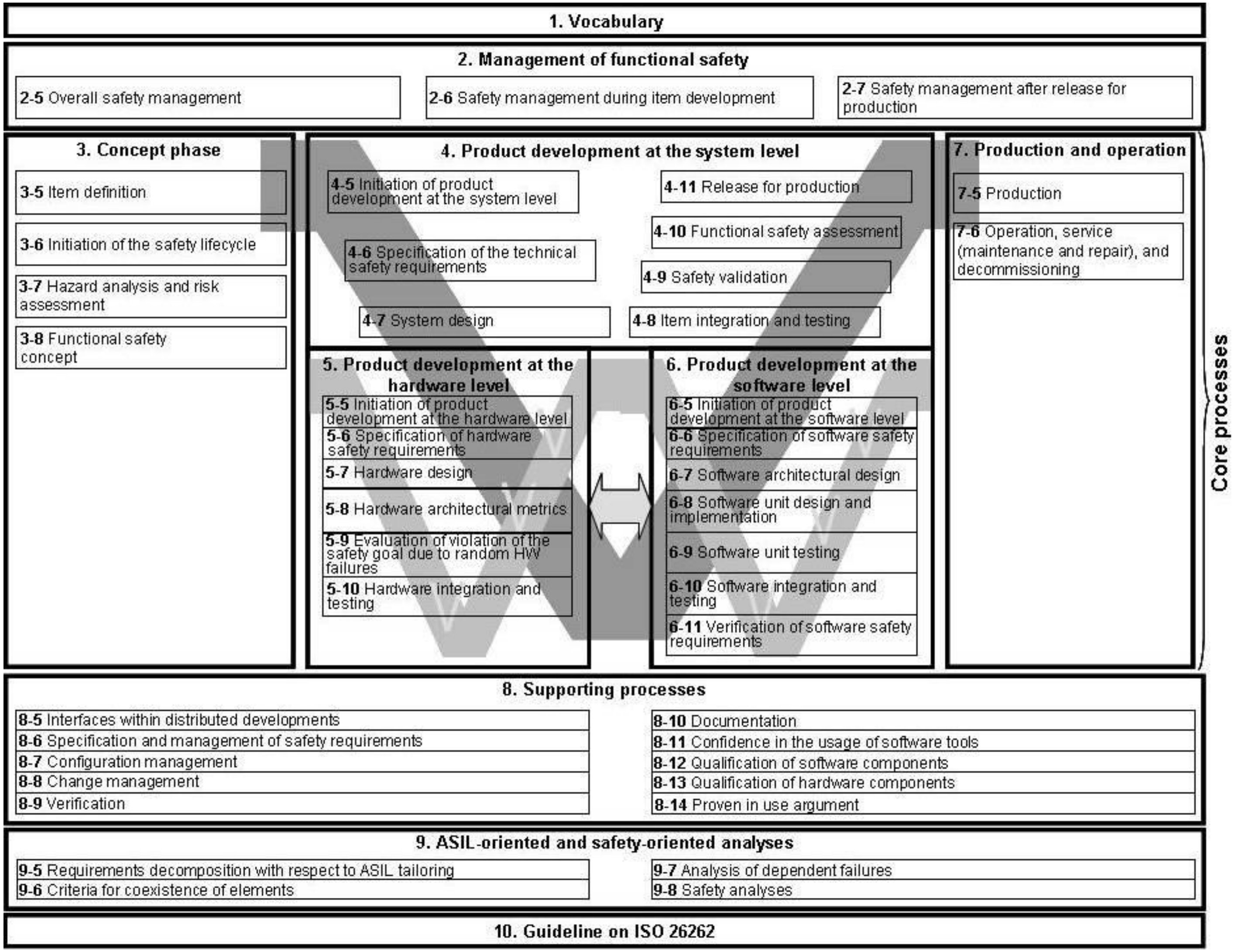
EAST-ADL

EAST-ADL2

EAST-ADL 2.1

EAST-ADL 2.x

ISO 26262 reference life cycle



1. Vocabulary

2. Management of functional safety

- 2-5 Overall safety management
- 2-6 Safety management during item development
- 2-7 Safety management after release for production

3. Concept phase

- 3-5 Item definition
- 3-6 Initiation of the safety lifecycle
- 3-7 Hazard analysis and risk assessment
- 3-8 Functional safety concept

4. Product development at the system level

- 4-5 Initiation of product development at the system level
- 4-6 Specification of the technical safety requirements
- 4-7 System design
- 4-8 Item integration and testing
- 4-9 Safety validation
- 4-10 Functional safety assessment
- 4-11 Release for production

5. Product development at the hardware level

- 5-5 Initiation of product development at the hardware level
- 5-6 Specification of hardware safety requirements
- 5-7 Hardware design
- 5-8 Hardware architectural metrics
- 5-9 Evaluation of violation of the safety goal due to random HW failures
- 5-10 Hardware integration and testing

6. Product development at the software level

- 6-5 Initiation of product development at the software level
- 6-6 Specification of software safety requirements
- 6-7 Software architectural design
- 6-8 Software unit design and implementation
- 6-9 Software unit testing
- 6-10 Software integration and testing
- 6-11 Verification of software safety requirements

7. Production and operation

- 7-5 Production
- 7-6 Operation, service (maintenance and repair), and decommissioning

Core processes

8. Supporting processes

- 8-5 Interfaces within distributed developments
- 8-6 Specification and management of safety requirements
- 8-7 Configuration management
- 8-8 Change management
- 8-9 Verification
- 8-10 Documentation
- 8-11 Confidence in the usage of software tools
- 8-12 Qualification of software components
- 8-13 Qualification of hardware components
- 8-14 Proven in use argument

9. ASIL-oriented and safety-oriented analyses

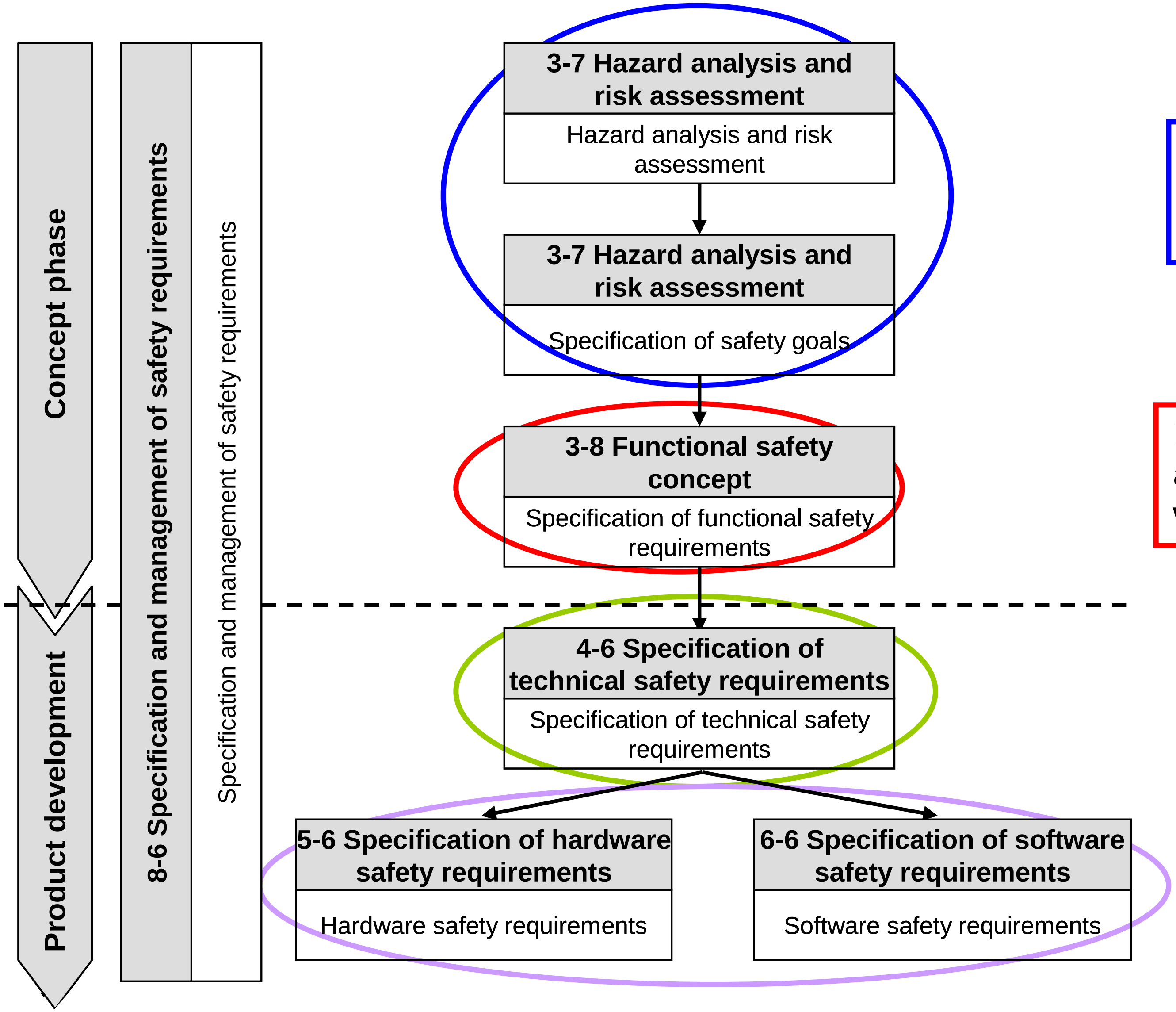
- 9-5 Requirements decomposition with respect to ASIL tailoring
- 9-6 Criteria for coexistence of elements
- 9-7 Analysis of dependent failures
- 9-8 Safety analyses

10. Guideline on ISO 26262

Six ISO26262 Concerns

1. Concept Phase – Safety Goals
   - Risk assessment
2. Concept Phase – Functional Safety Concept
   - Topology-independent Solution
3. Product Development – Technical Safety Concept
   - Preliminary System solution
4. Product Development – Hardware and Software
   - Detailed hardware and software architecture
5. Safety Element out of Context
   - Matching ASIL with ASIL
6. Supplier-OEM Exchange
   - Matching ASIL with ASIL

ISO 26262 - What to handle for each phase



Concept phase
Product development
8-6 Specification and management of safety requirements
Specification and management of safety requirements
3-7 Hazard analysis and risk assessment
Hazard analysis and risk assessment
3-7 Hazard analysis and risk assessment
Specification of safety goals
3-8 Functional safety concept
Specification of functional safety requirements
4-6 Specification of technical safety requirements
Specification of technical safety requirements
5-6 Specification of hardware safety requirements
Hardware safety requirements
6-6 Specification of software safety requirements
Software safety requirements

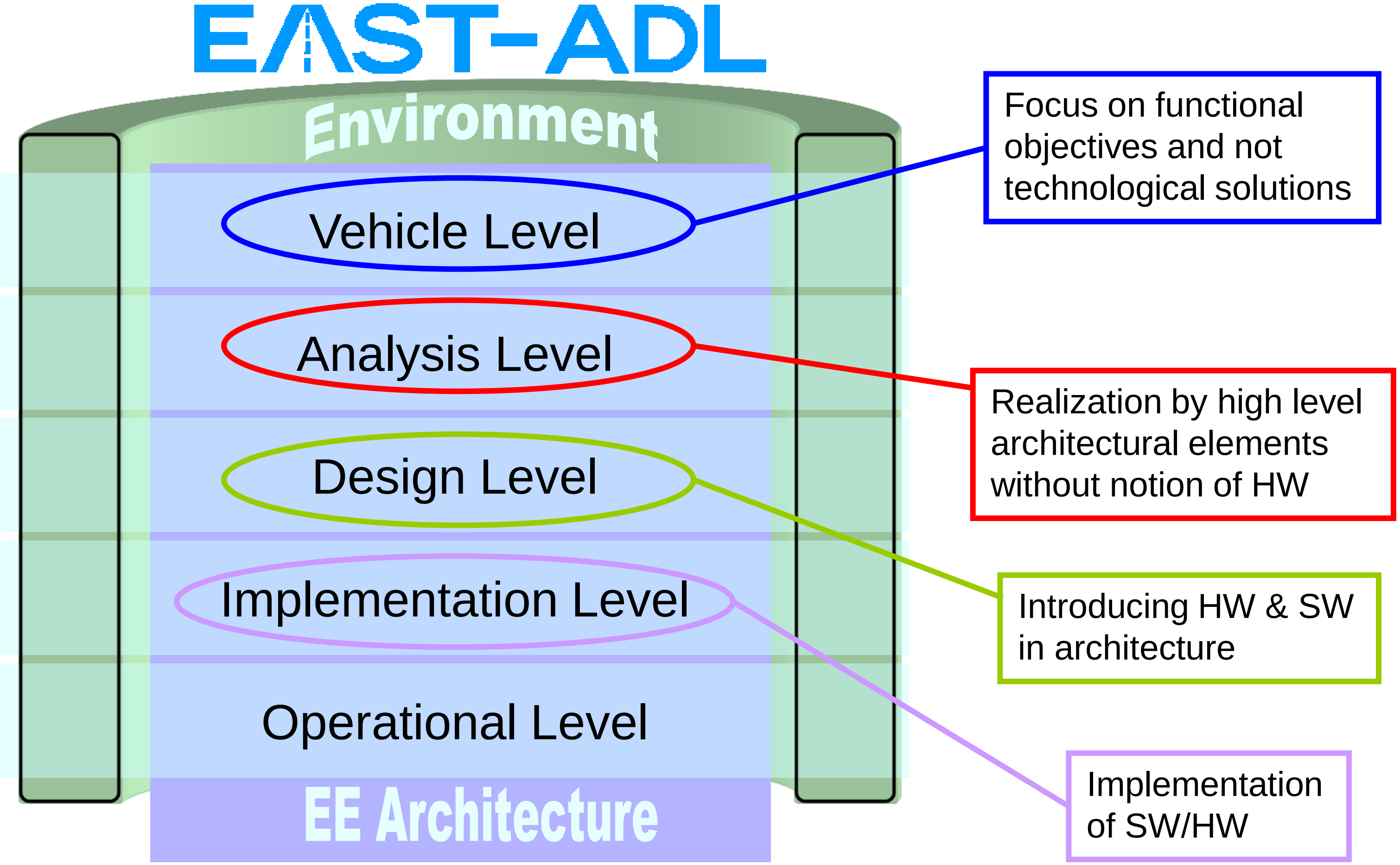
Focus on functional objectives and not technological solutions

Realization by high level architectural elements without notion of HW

Introducing HW & SW in architecture

Implementation of SW/HW

What to handle on each abstraction level



Focus on functional objectives and not technological solutions

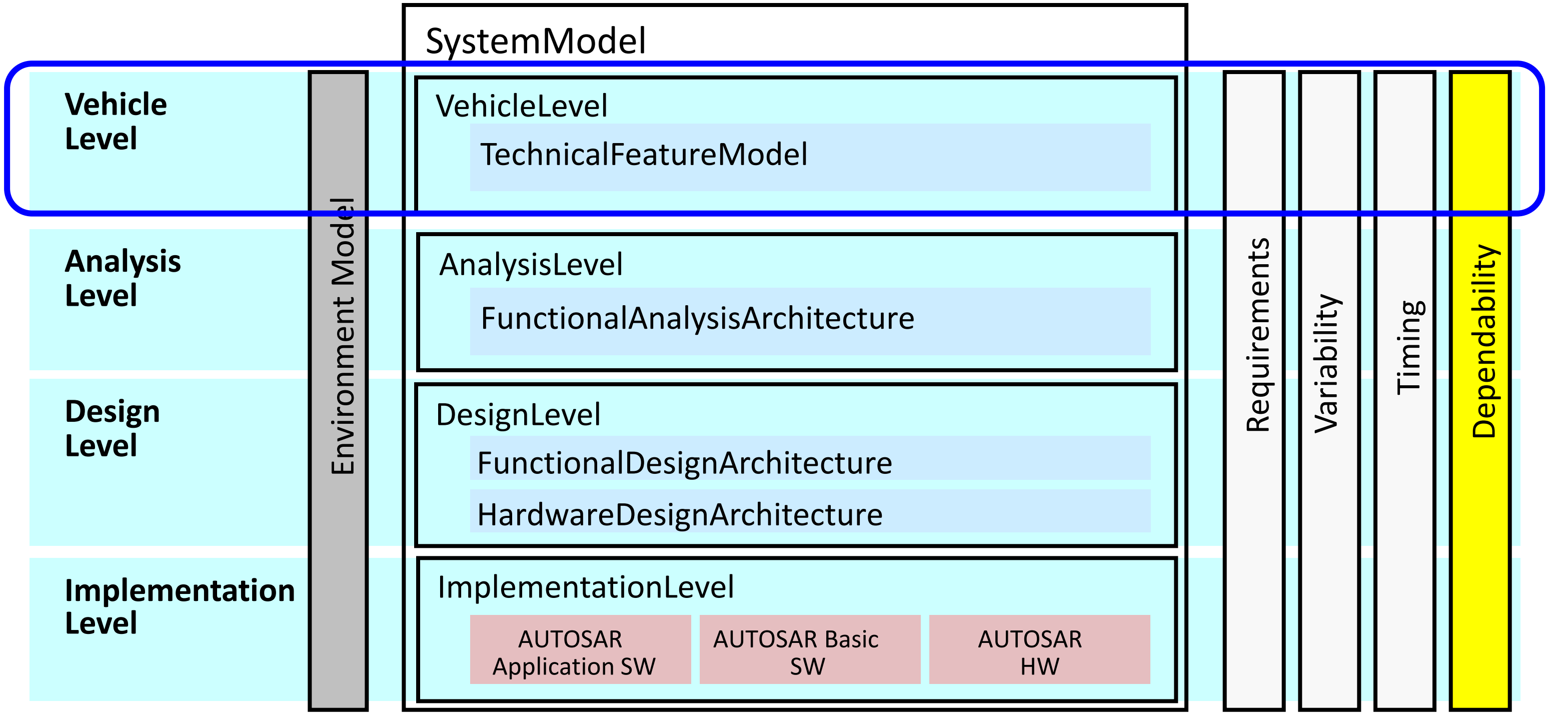
Realization by high level architectural elements without notion of HW

Introducing HW & SW in architecture

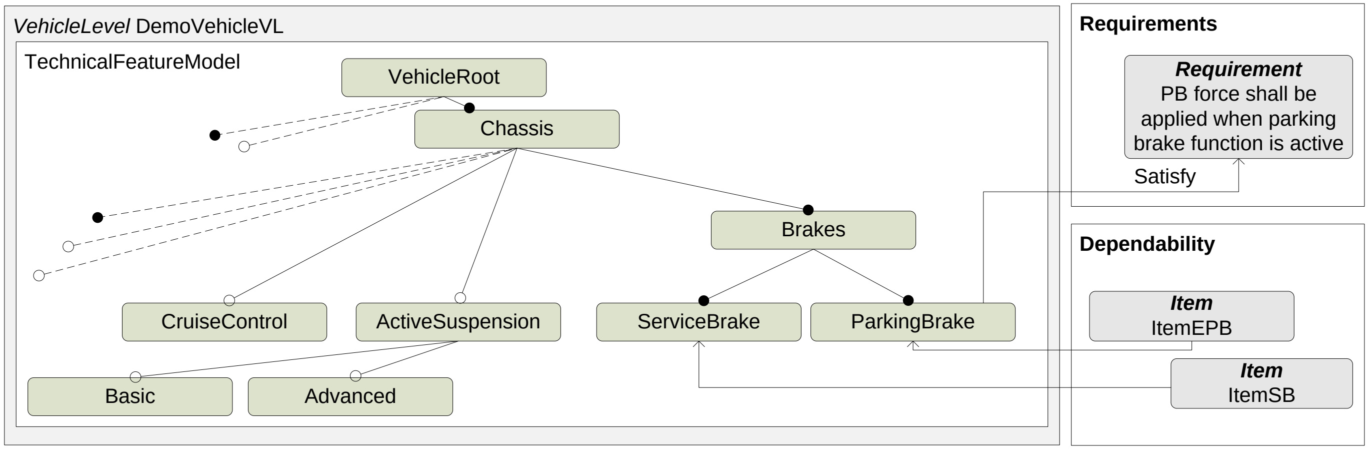
Implementation of SW/HW

1. Safety Goals: Vehicle Level

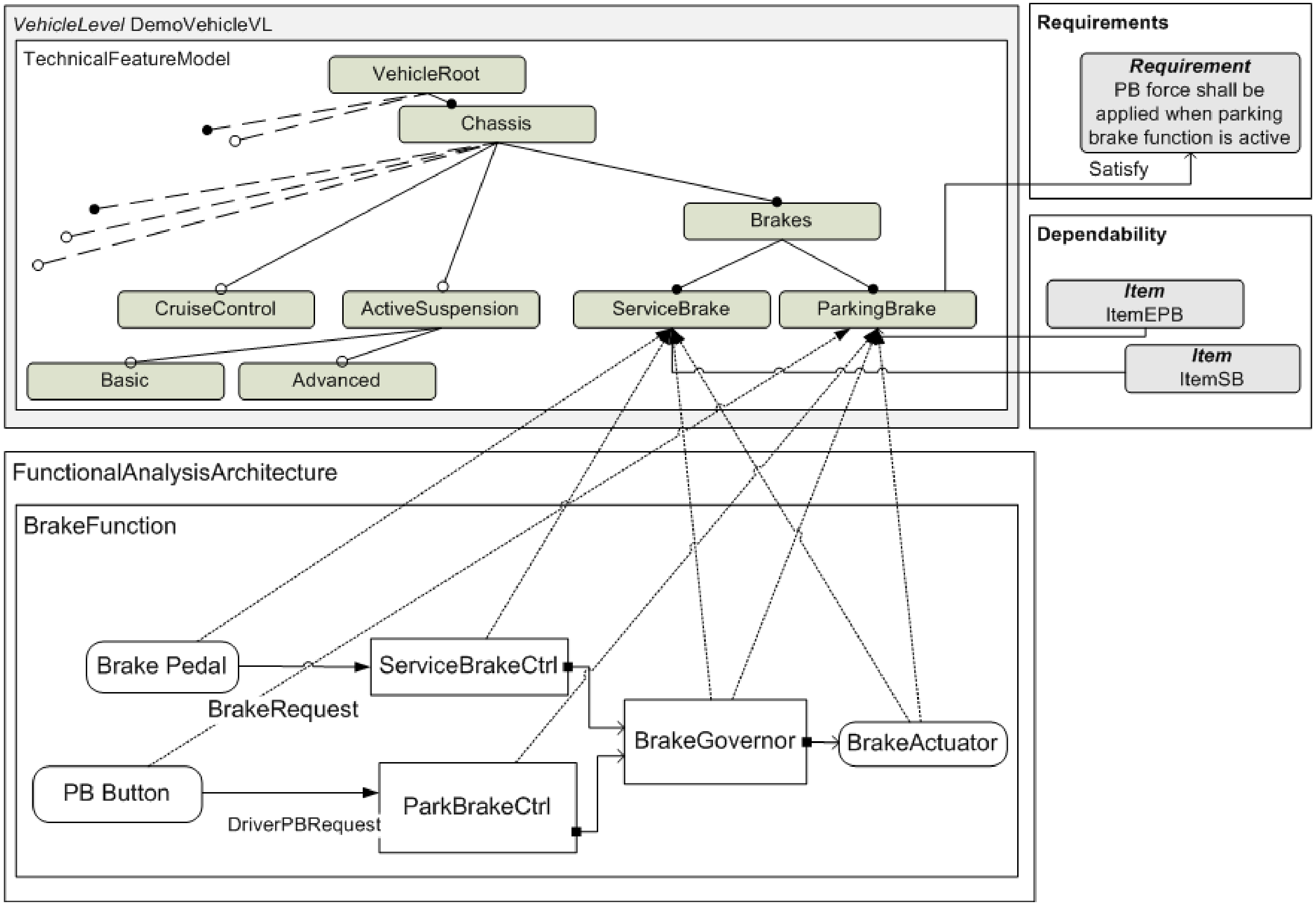
- Part 3.7 artifacts in EAST-ADL



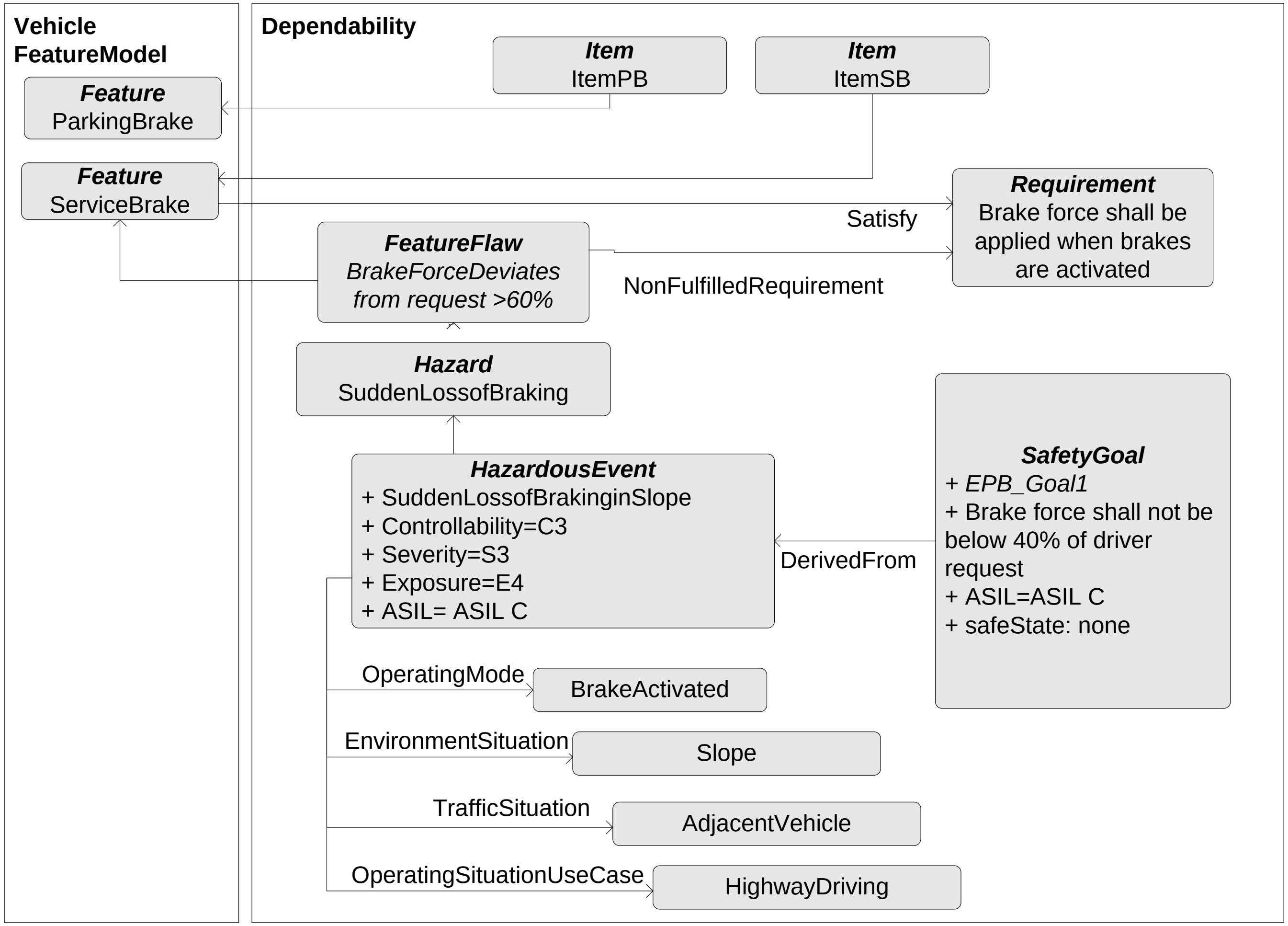
Item Definition



Item Definition

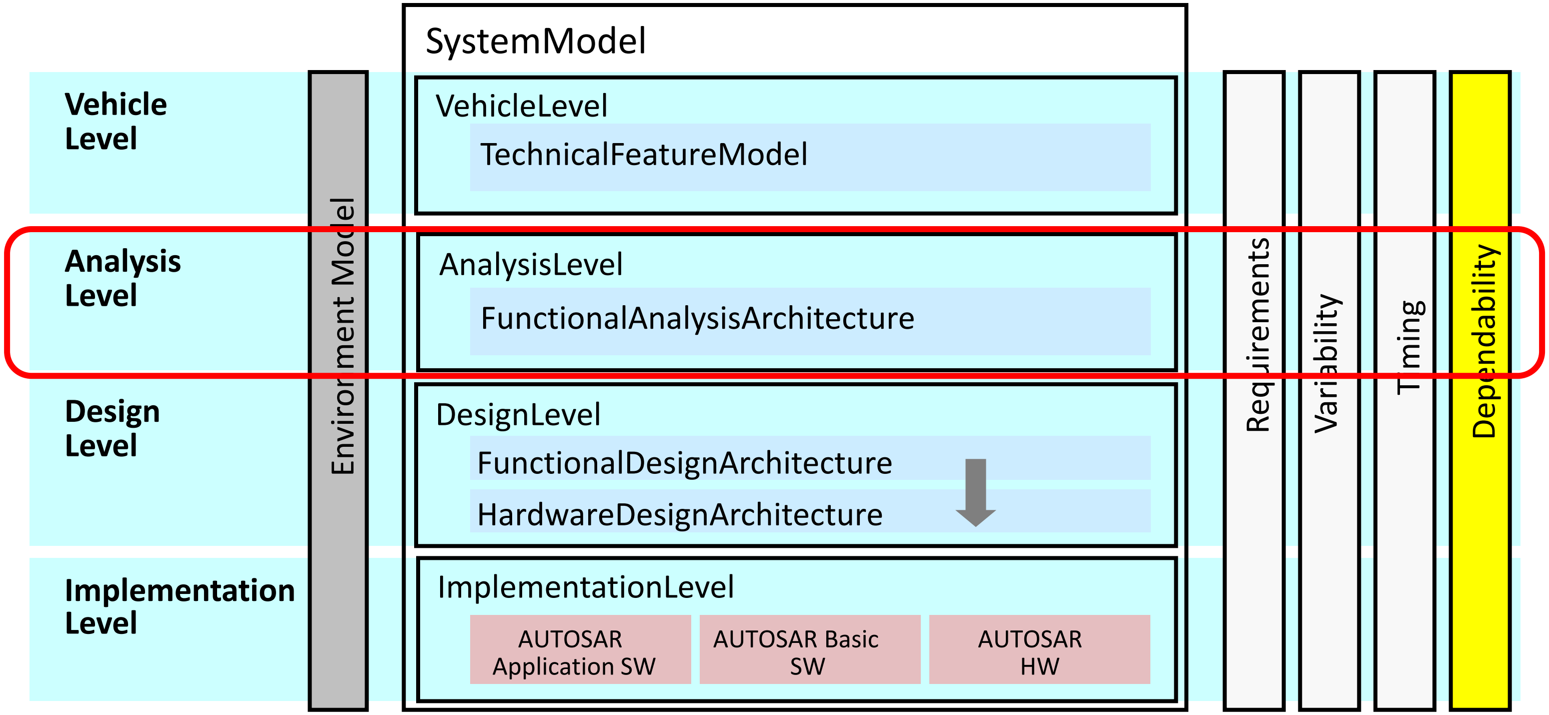


Preliminary Hazard Analysis

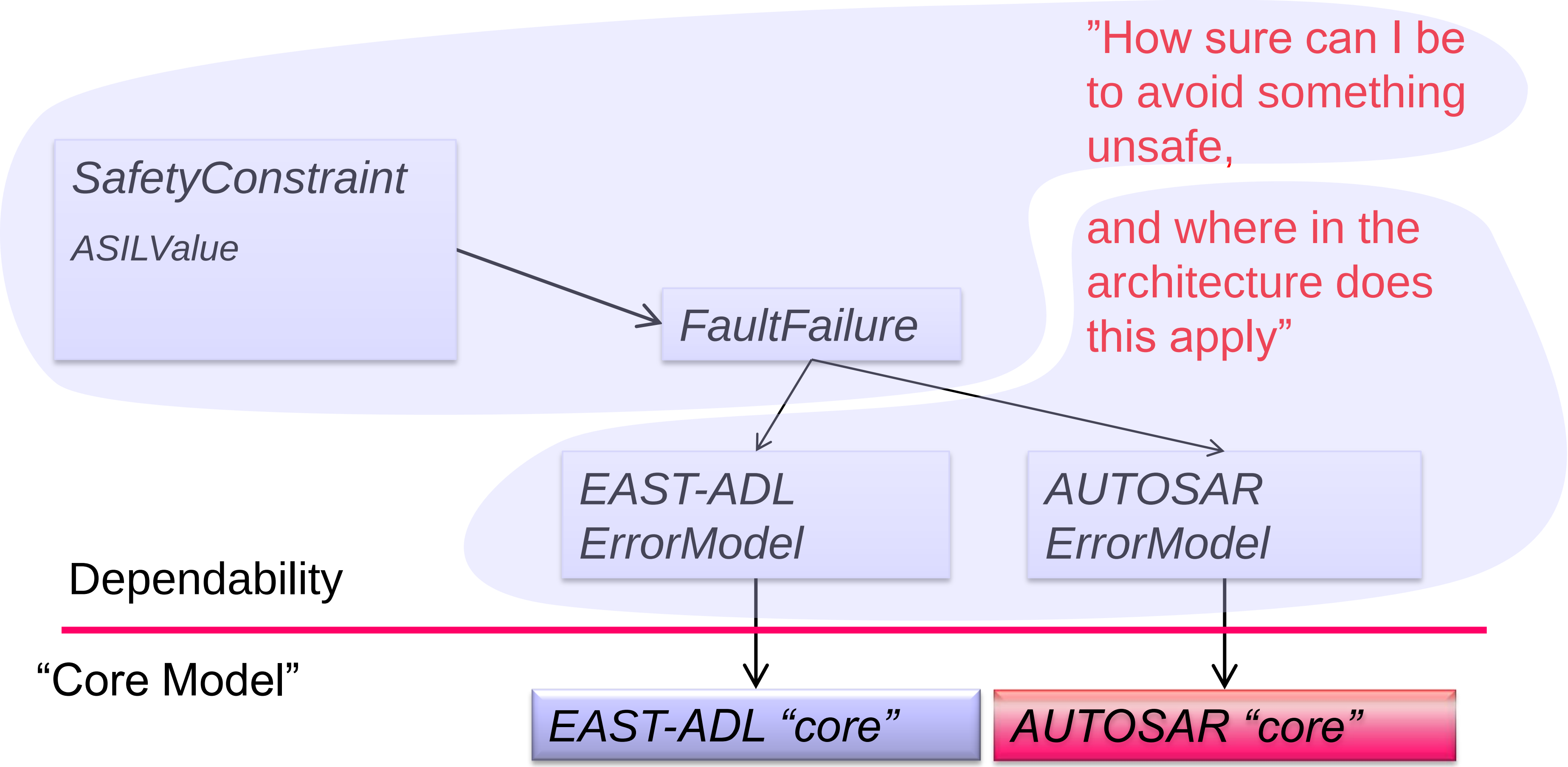


2. Functional Safety Concept: Analysis Level

- Part 3.8 artifacts in EAST-ADL



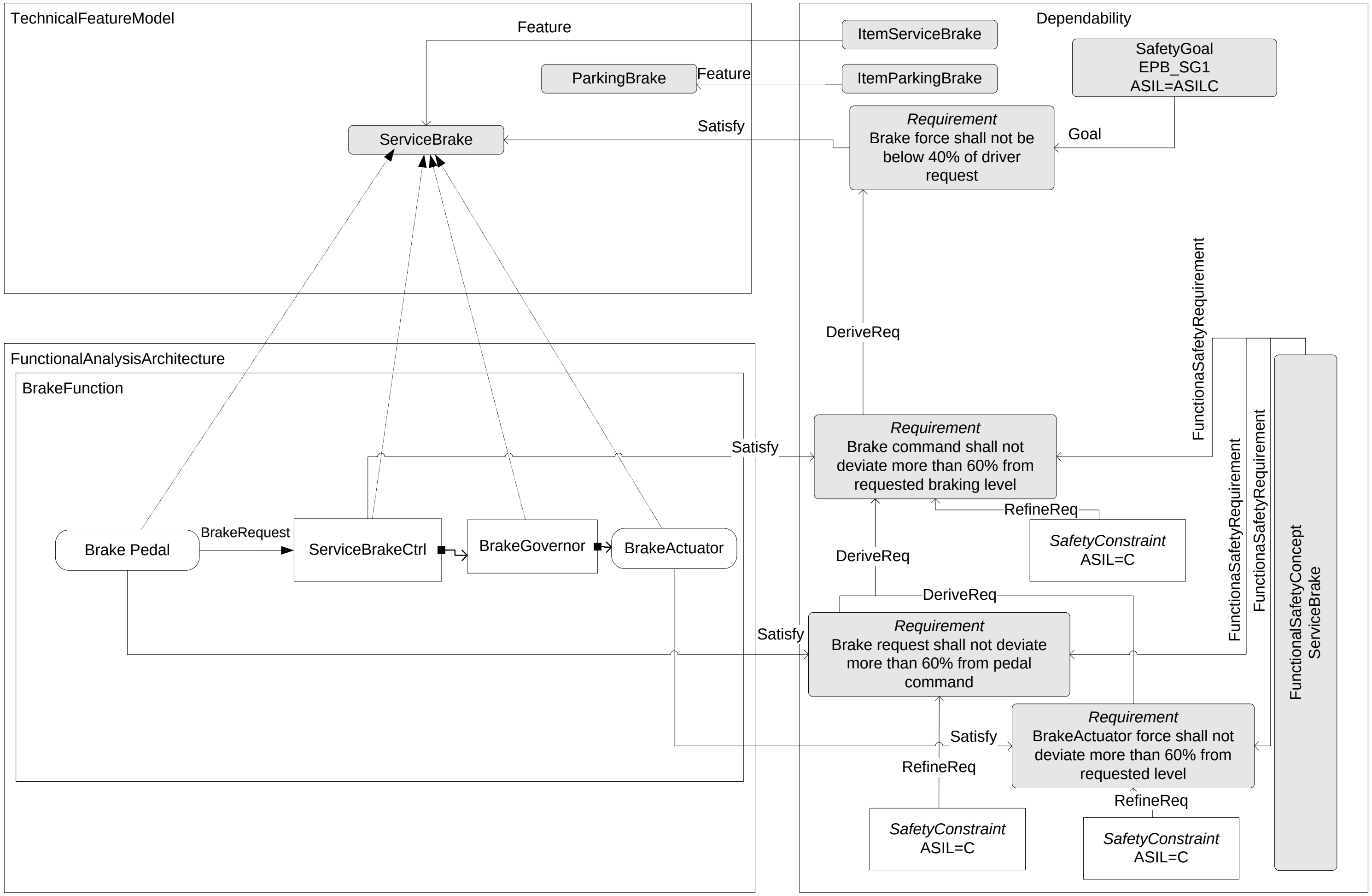
Safety Modelling – Basic Concept



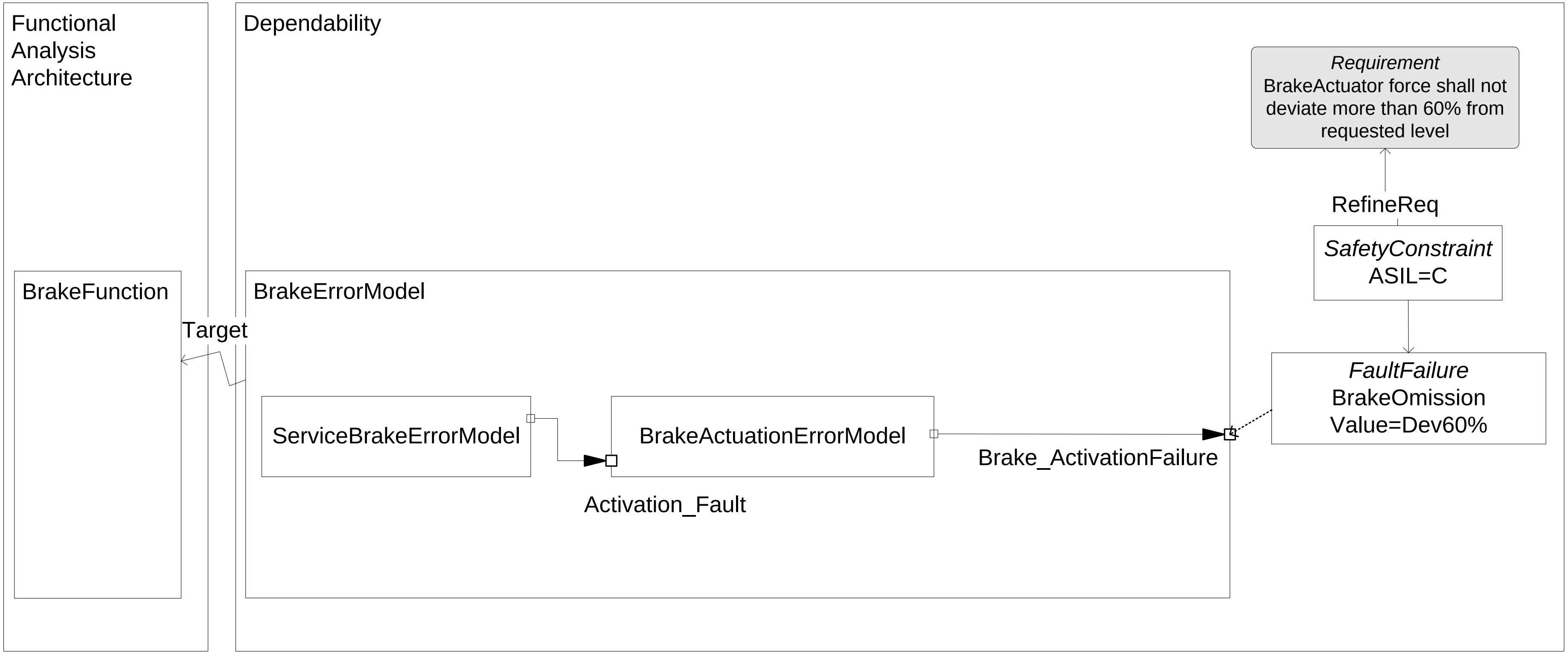
”How sure can I be to avoid something unsafe,

and where in the architecture does this apply”

Functional Safety Concept

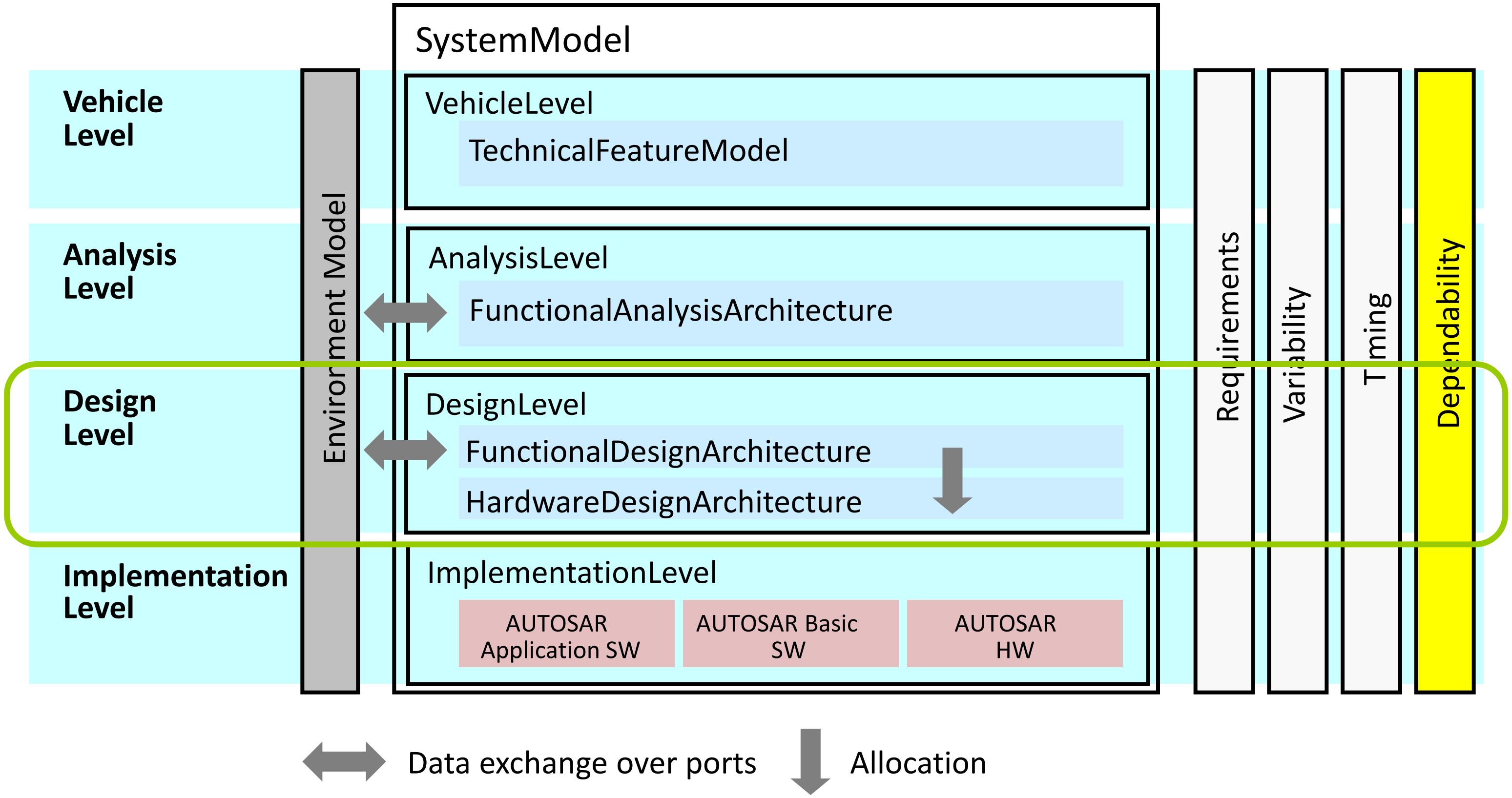


Functional Safety Requirement

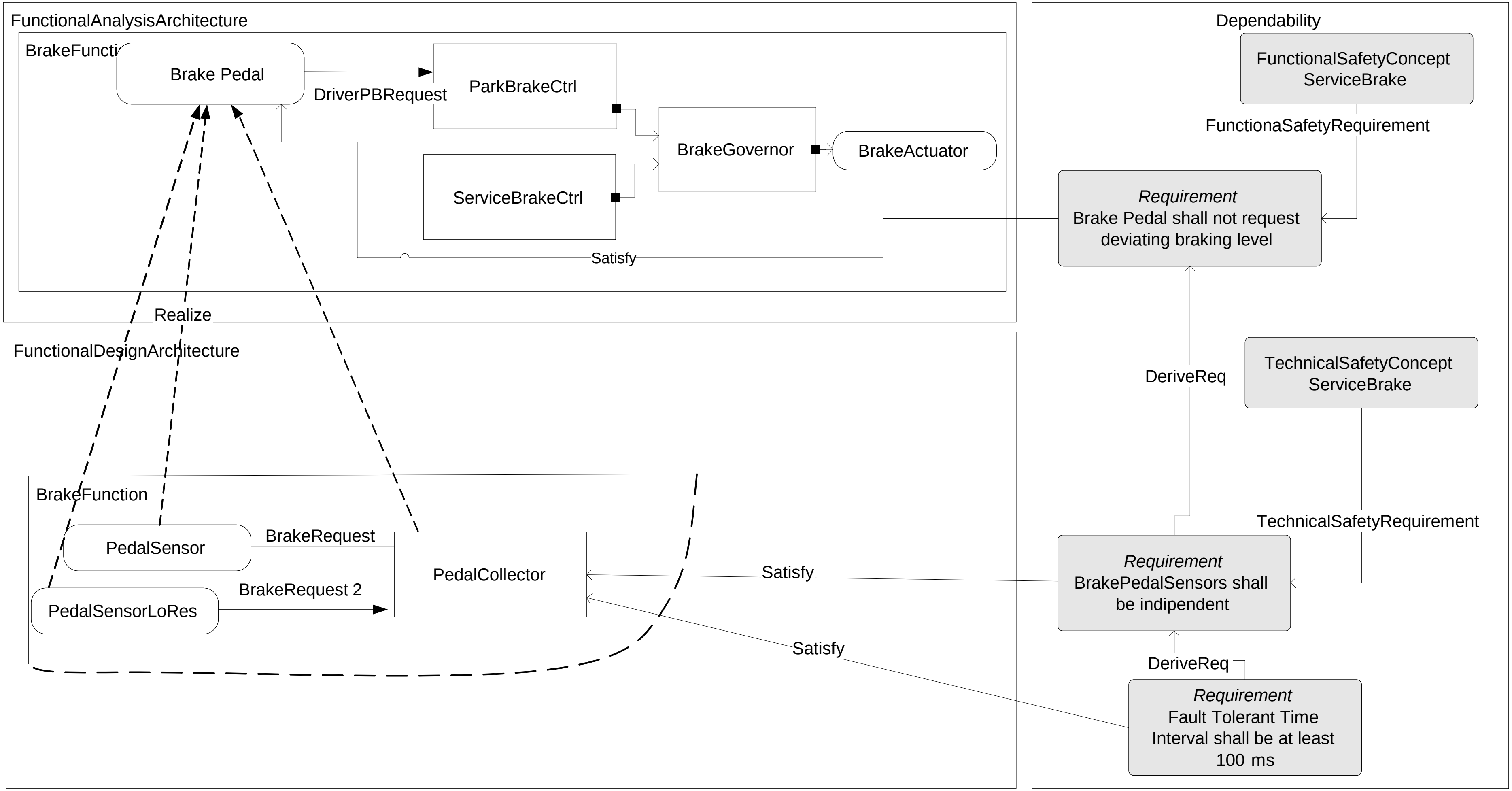


3. Technical Safety Concept: Design Level

- Part 4 artifacts in EAST-ADL

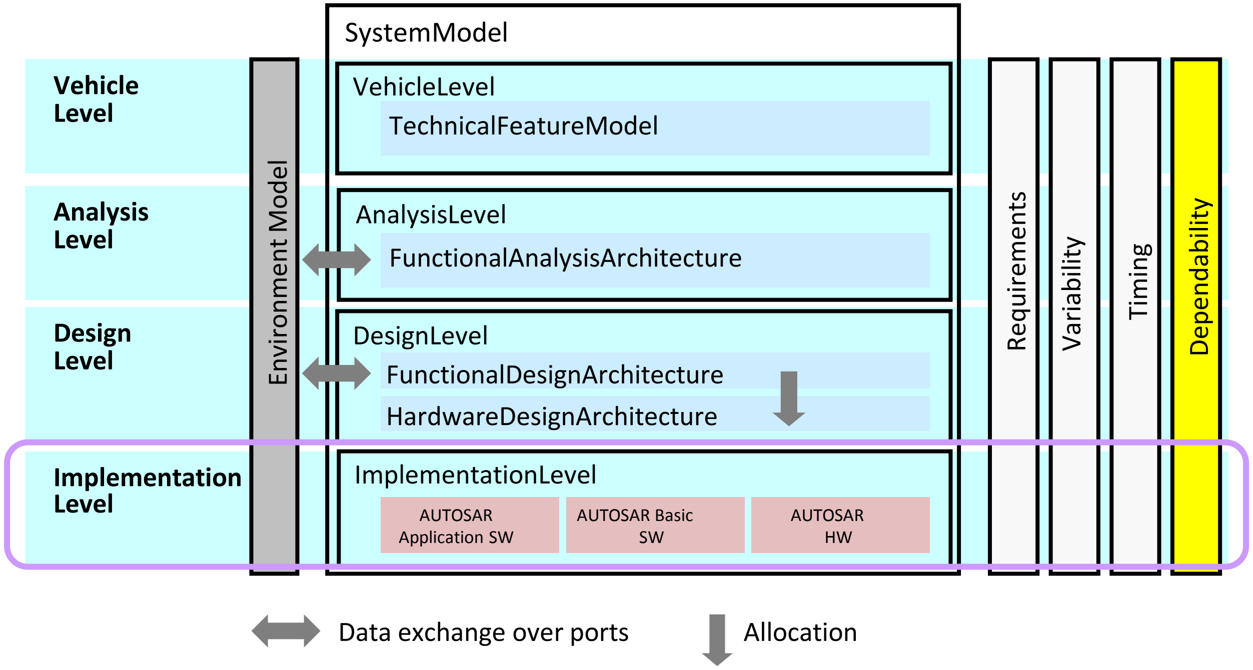


Technical Safety Concept

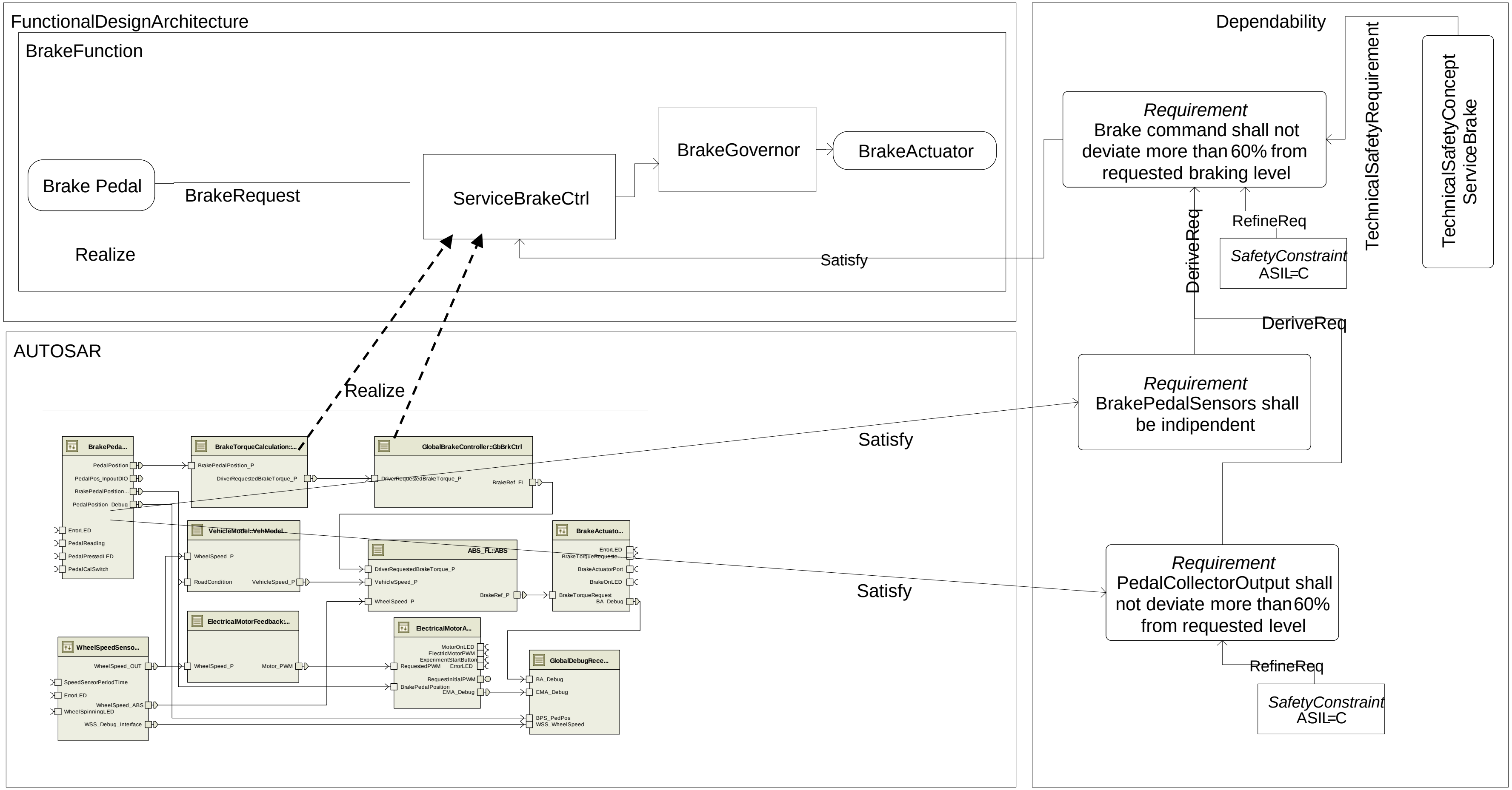

FunctionalAnalysisArchitecture
BrakeFuncti
Brake Pedal
DriverPBRequest
ParkBrakeCtrl
BrakeGovernor
BrakeActuator
ServiceBrakeCtrl
Satisfy
Realize
FunctionalDesignArchitecture
BrakeFunction
PedalSensor
BrakeRequest
PedalCollector
BrakeRequest 2
PedalSensorLoRes
Satisfy
Satisfy
Dependability
FunctionalSafetyConcept ServiceBrake
FunctionaSafetyRequirement
Requirement
Brake Pedal shall not request deviating braking level
DeriveReq
TechnicalSafetyConcept ServiceBrake
TechnicalSafetyRequirement
Requirement
BrakePedalSensors shall be indipendent
DeriveReq
Requirement
Fault Tolerant Time Interval shall be at least 100 ms

4. HW & SW Requirements: Implementation Level

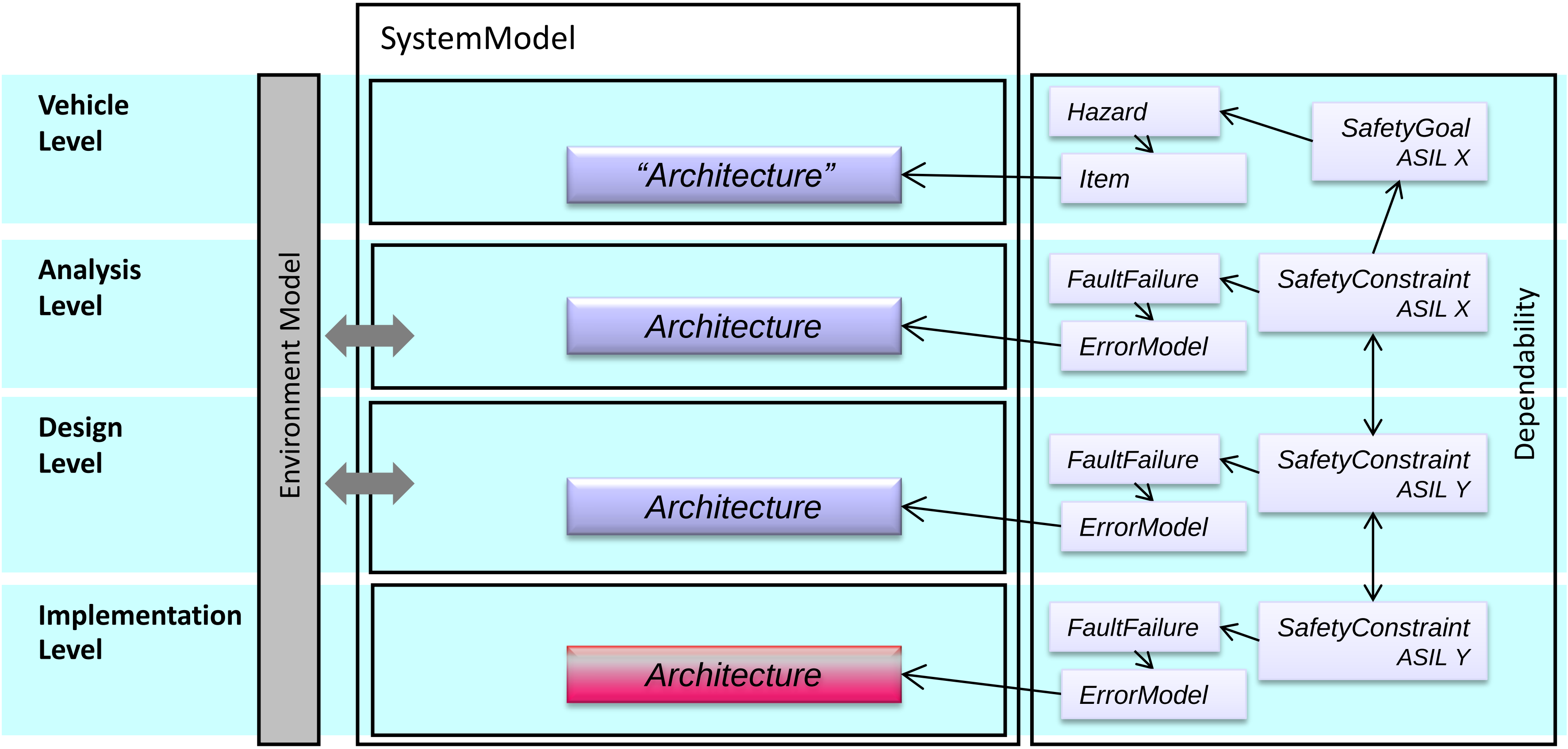
- Part 5 artifacts in AUTOSAR (and IP-XACT)
- Part 6 artifacts in AUTOSAR



AUTOSAR Elements



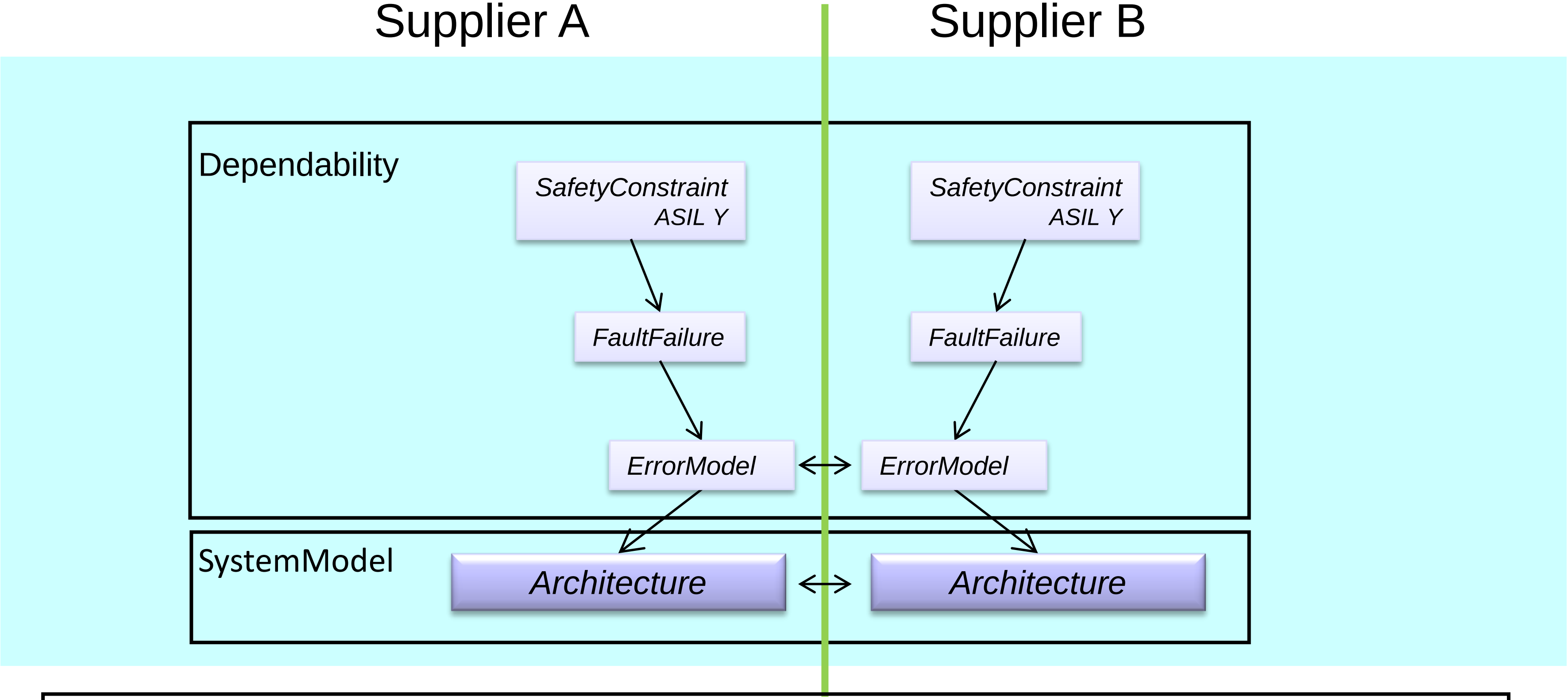
5. Safety Element out of Context



E.g. Technical Safety Concept without Functional Safety Concept:

Allocated Safety Constraints can play the role of Technical Safety Requirements when Functional Safety Concept is available

6. Supplier-OEM interaction: A/D/I Level



Nominal aspects: Interfaces match between subsystems

Dependability aspects: Safety Constraints Match between subsystems

EAST-ADL vs. Safety Bench Marking

- Safety is about avoiding Failures that may cause Hazards
- ISO26262 defines a systematic approach:
  1. Identify Safety Goal
  2. Create a safe architecture with safety requirements that meet safety Goal

ISO26262 element	Purpose
Safety Goal	Avoid Hazard / FeatureFlaw
Functional Safety Concept	Avoid Failure (of abstract Function)
Technical Safety Concept	Avoid Failure (of Function on HW)
HW and SW requirements	Avoid Failure (of SW Component on HW)

Trace

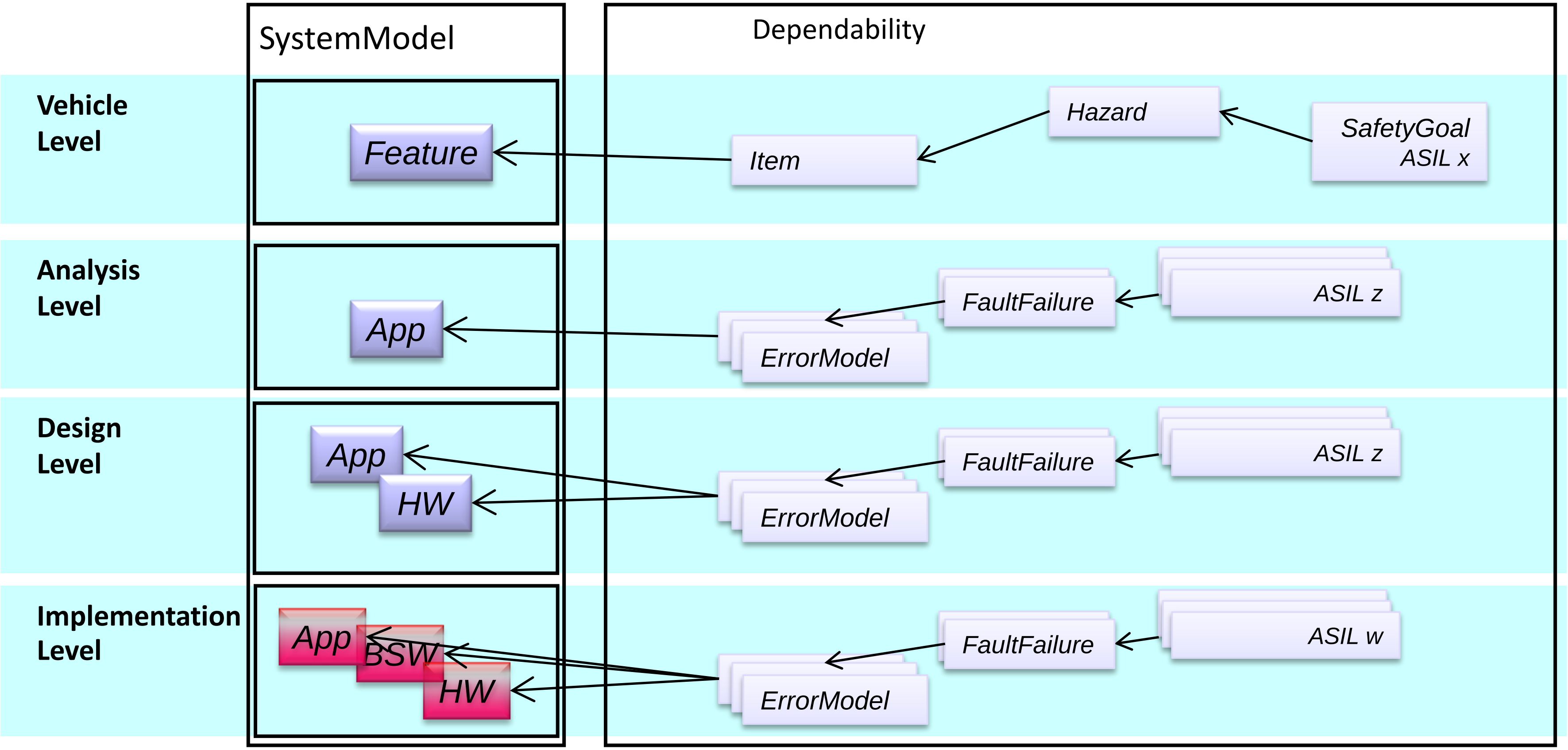
EAST-ADL vs. Safety Bench Marking

- Safety Benchmarking is about assessing how well a system/subsystem/component/mechanism/… fulfills requirements
  - In-context
  - Out-of-context
- Assessing Ability to Meet ASIL X Safety Goal
  - Conformance to Functional Safety Requirements
  - Conformance to Technical Safety Requirements
  - Conformance to HW and SW Requirements

EAST-ADL vs. Safety Bench Marking

- Benchmarking out-of-context = Conformance to anticipated
  - Functional Safety Requirements
  - Technical Safety Requirements
  - HW and SW Requirements
- To be able to draw conclusions on safety, the assessment of fault tolerance must
  - Address relevant faults
  - Be represented adequately
    =the fault tolerance capability can be related to requirements and safety goal

EAST-ADL vs. Safety Bench Marking



ErrorModel capture Failure propagation logic – can be identified using fault injection

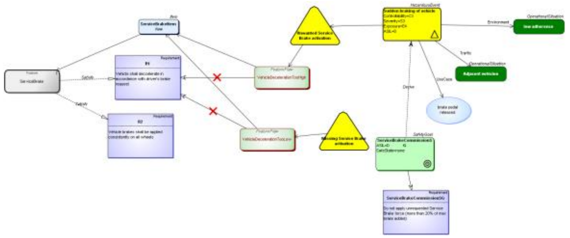
FaultFailure capture faults and failures on ports of ErrorModel

ASIL constraint define expected or established “probability” of the fault or failure

Activities vs. Abstraction Levels

EAST-ADL	Vehicle Level	Define Features and requirements Identify FeatureFlaw and Hazard Identify Scenorios and Hazardous Event Define SafetyGoal
	Analysis Level	Define Functional Architecture Define Functional Safety Requirements and Concept Define ErrorModel and FaultFailure Define SafetyConstraints
	Design Level	Define Concrete Functional and Hardware Architecture Define Technical Safety Requirements and Concept Define ErrorModel and FaultFailure Define SafetyConstraints
AUTOSAR	Implementation Level	Define Software and detailed Hardware Architecture Define Software and Hardware Requirements Define ErrorModel and FaultFailure Define SafetyConstraints

Finally…



- EAST-ADL is a language for Automotive EE engineering information
  - Shared ontology/terminology across companies and domains
  - EAXML exchange format to secure tool interoperability
  - Allows joint efforts on methodology, modelling and tools
- …supports cross-cutting aspects through extensions.
- …is aligned with AUTOSAR elements and modelling infrastrucure
- …provides means to plan, document and utilize safety benchmarking
- EATOP Eclipse platform can foster tool prototyping
- EAST-ADL Association is a structure to coordinate and harmonize language progress
- *Collaborative aspect of EAST-ADL is particularly relevant for ISO26262*