

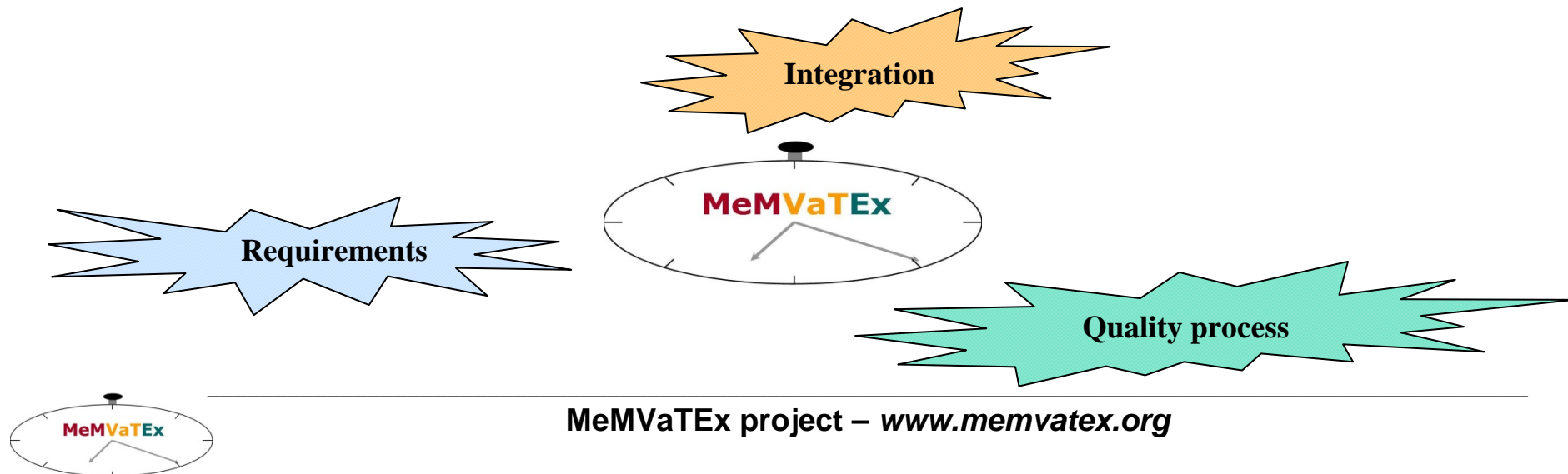
# MeMVaTEx

**Méthode de Modélisation**  
pour la **Validation** et la **Traçabilité** des **Exigences**  
*Modeling method for validation &  
requirements traceability*



# Context and objectives

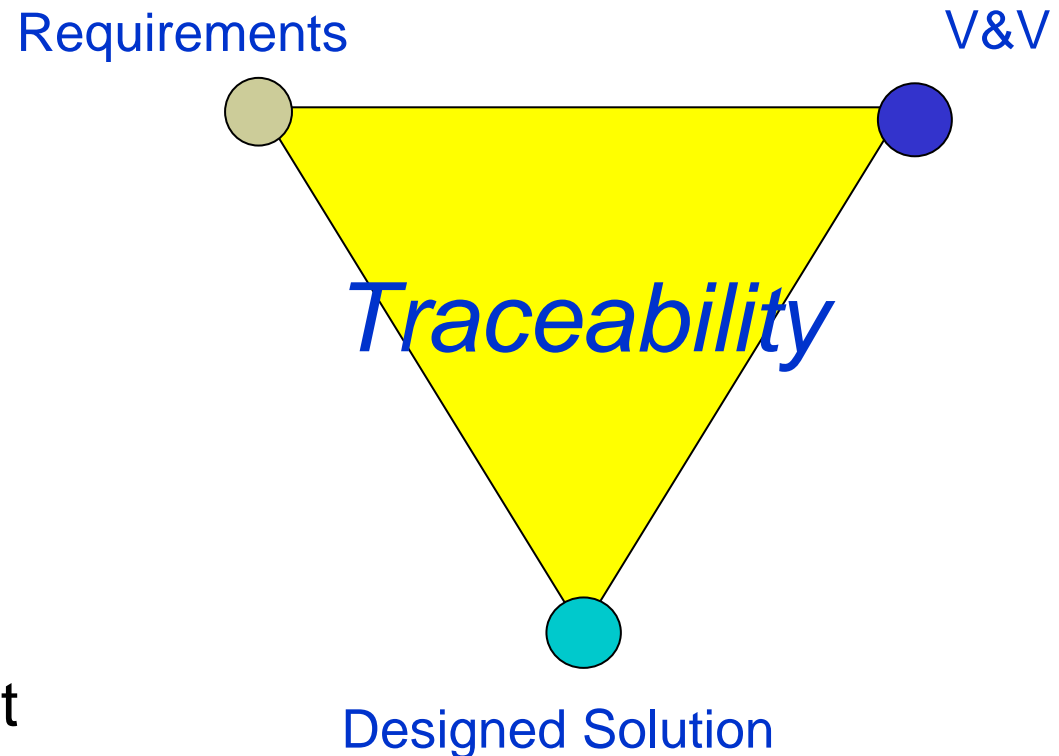
- **Software design method**
  - *Support* for requirement expression and classification
  - *Increases* requirements traceability
  - *Supports* heterogeneous models and tools
  - *Fills gap between* architecture views
  - *Consideration* of Real-time impact



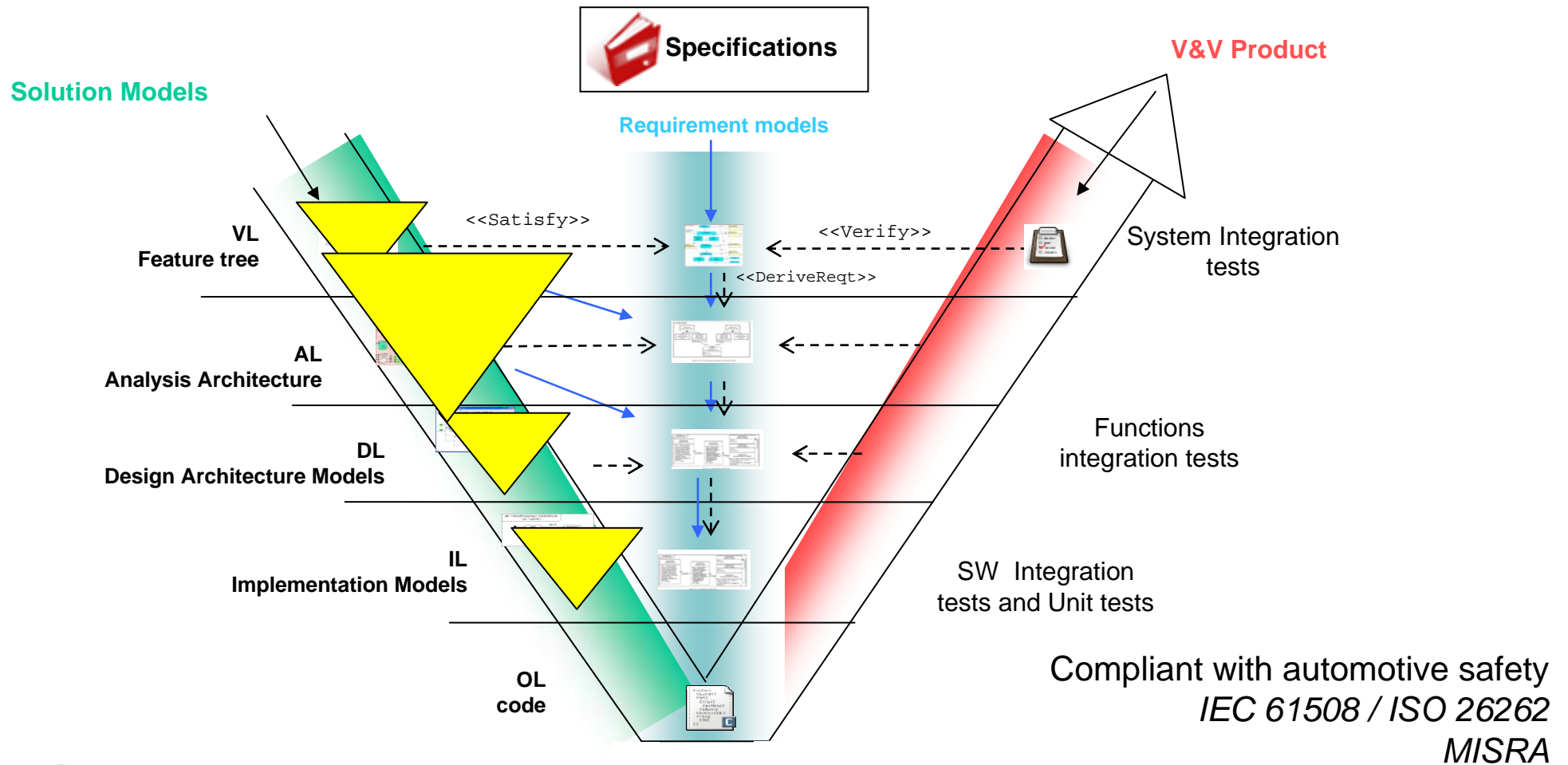
# Overview of the method [Albinet et al., ERTS2008]

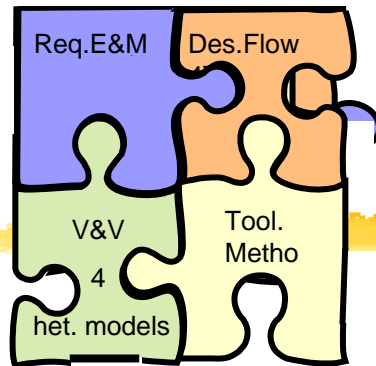
A **Method** for ensuring system modeling:

- ✓ Requirements
  - ✓ Designed solution
  - ✓ V&V considerations
- 
- ✓ Traceability management



# Overview of the method

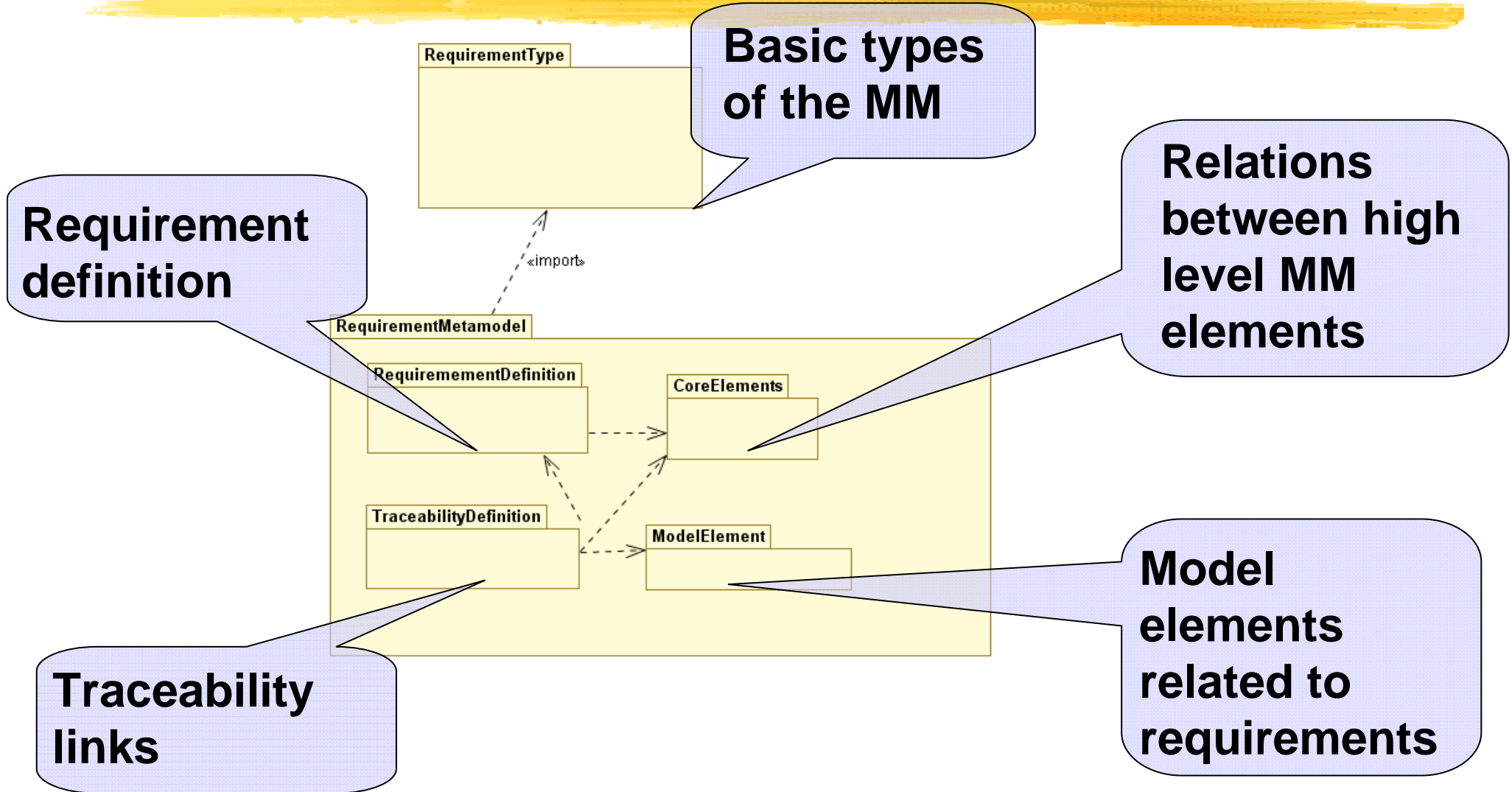


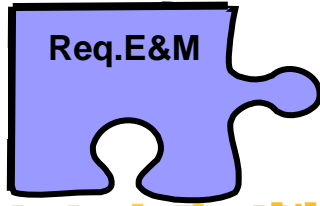


## Technical enhancements

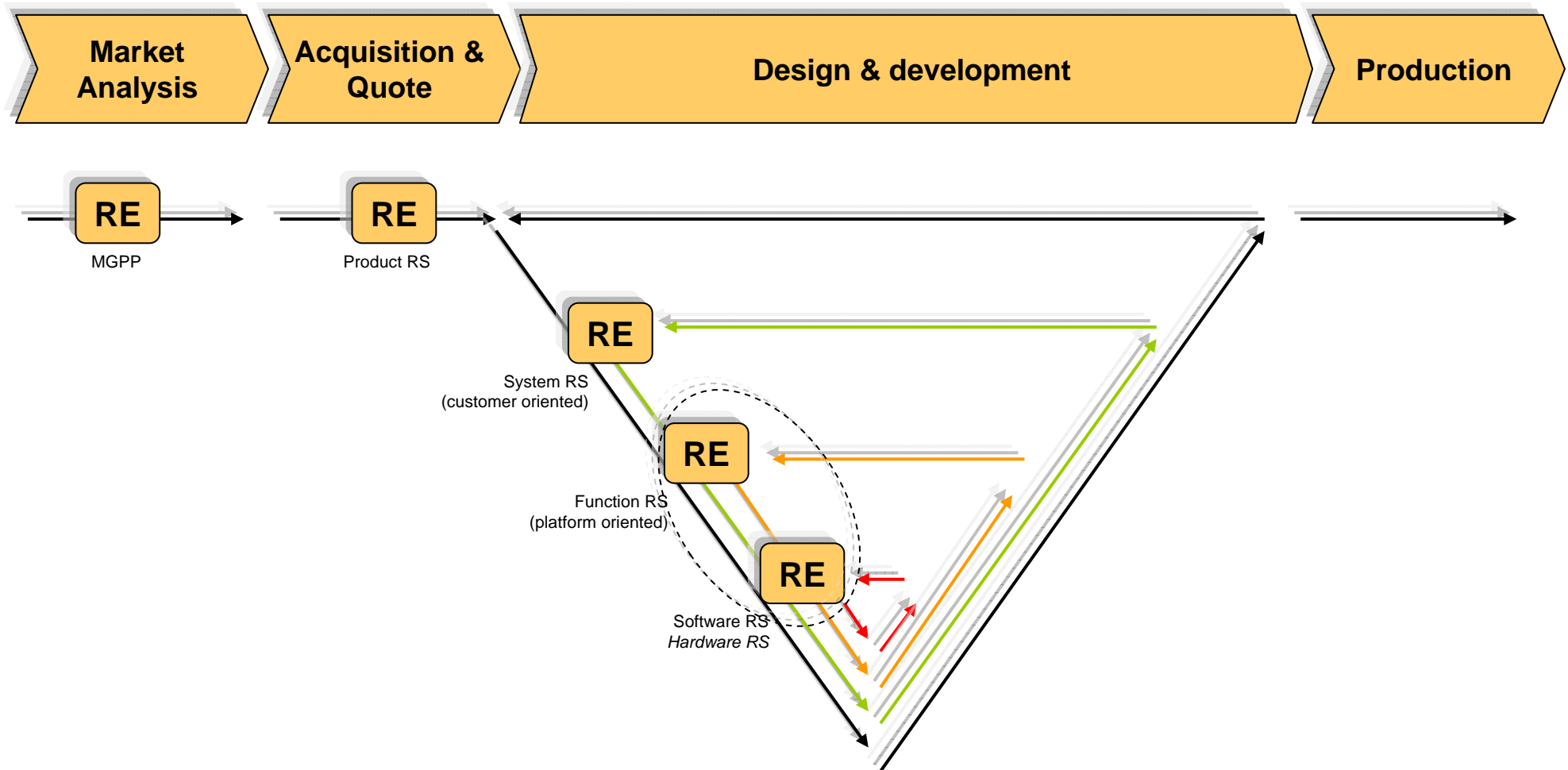
- **Expression and modeling of requirements (CEA LIST)**
- **Design flow for timing requirements analysis (INRIA)**
- **V&V for heterogeneous models (UTC)**
- **Tools adaptation for MeMVaTEx method (Sherpa)**
- **Demonstrator (Continental)**

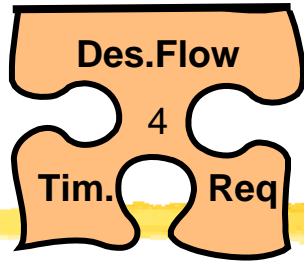
# Metamodel for requirements





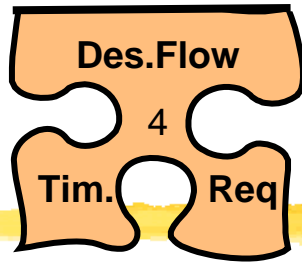
# Integration in the MeMVaTEEx method





## Phase 1: From timing requirements to solution models

- ✓ Formalization of Non-Functional MeMVaTEx requirements [ISORC09]
  - ✓ use of EAST-ADL2 Timing Requirements
  - ✓ use of MARTE time concepts (clocks)
- ✓ Consistency of temporal features within solution model
  - ✓ EAST-ADL2 Timing Requirements, EAST-ADL2 structural parts, Native/External Behavioral parts
  - ✓ All temporal features are based on MARTE time model
- ✓ Links with traceability features (*satisfy*, *verify*)



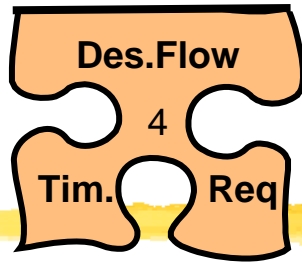
## Phase 2: From design models to real-time implementation

### Focus on Design level (EAST-ADL2) and Implementation level (AUTOSAR)

- ✓ Transition between levels
- ✓ Consistency between levels

### 3 steps approach

- ✓ Extraction of timing features (MARTE, etc.) from solution models
- ✓ Transition from function (ADLFunction) to software components (SWC) and allocation onto hardware solution model
- ✓ Automatic implementation with SynDEx



## Phase 3: From solution models to validation

### ■ Two kinds of validation

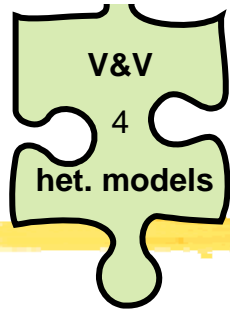
- Translation of Timing requirements into CCSL for **simulation** in the **TimeSquare** environment
- Extraction of information from solution models for **scheduling analysis** with **SynDEX**

### ■ Validation at Analysis and Design Levels

- Timing simulation with timing analysis tool: TimeSquare and SynDEX,
- End-to-end timing test, verdict: ok/not\_ok
  - Link to corresponding requirement

### ■ Validation at Implementation Level (AUTOSAR)

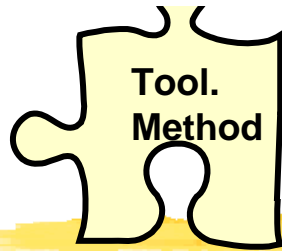
- Real-time code execution with code generation tool: SynDEX
- End-to-end timing test, verdict: ok/not\_ok
  - Link to corresponding requirement



# Requirement verification and validation

## Project main contributions :

- Integration of V&V concepts and processes during design modeling.
  - **V&V/model**: verification of the requirements satisfaction by the **models solution** through the different abstraction levels.
    - Inspection, conformance analysis, test, ....
  - **V&V/product**: verification of the requirement satisfaction by the **product**.
    - Test (unit test, integration test, system test)
- Integration of V&V concepts in MeMVaTEx profile



# Demonstrator

## System Level

- System Model

## Vehicle Level

- Vehicle Feature Model

## Analysis Level

- Functional Architecture Analysis

## Design Level

- Functional Design architecture
- MW Abstraction
- Hardware Design Architecture

## Implementation Level

- AUTOSAR

## V&V

- T<sup>2</sup>
- Simulink
- SynDEX

## V&V

- SynDEX

Requirements

## Conclusion

- **A unified method for requirement expression, modeling traceability and validation during system development (guideline)**
  - **A metamodel and a profile for requirements engineering**
  - **A design flow for timing requirement to real-time execution**
  - **A dedicated toolset prototype for the demonstrator**
- 
- **Demonstrator (EMS) with automotive case study**