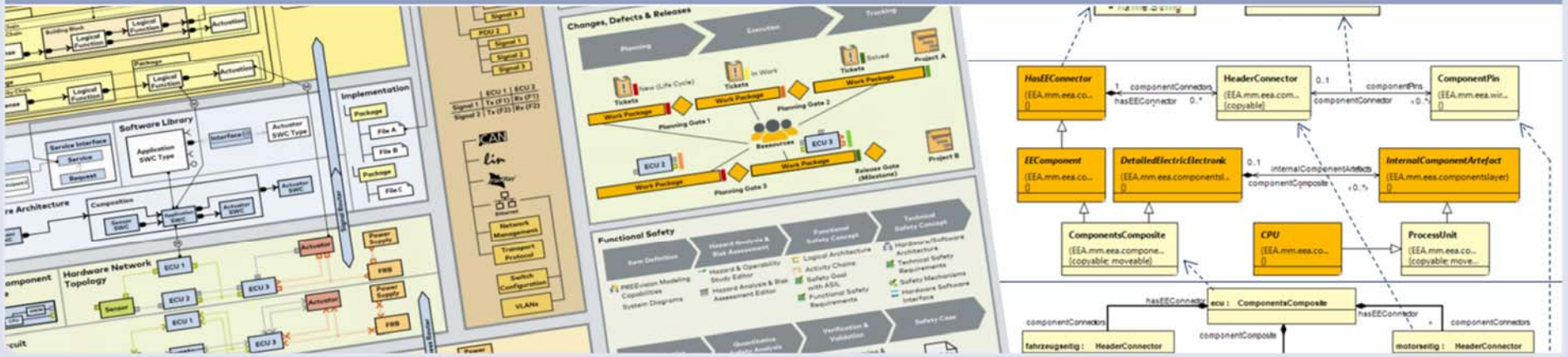
Karlsruher Institut für Technologie

Vorlesung Software Engineering (SE)

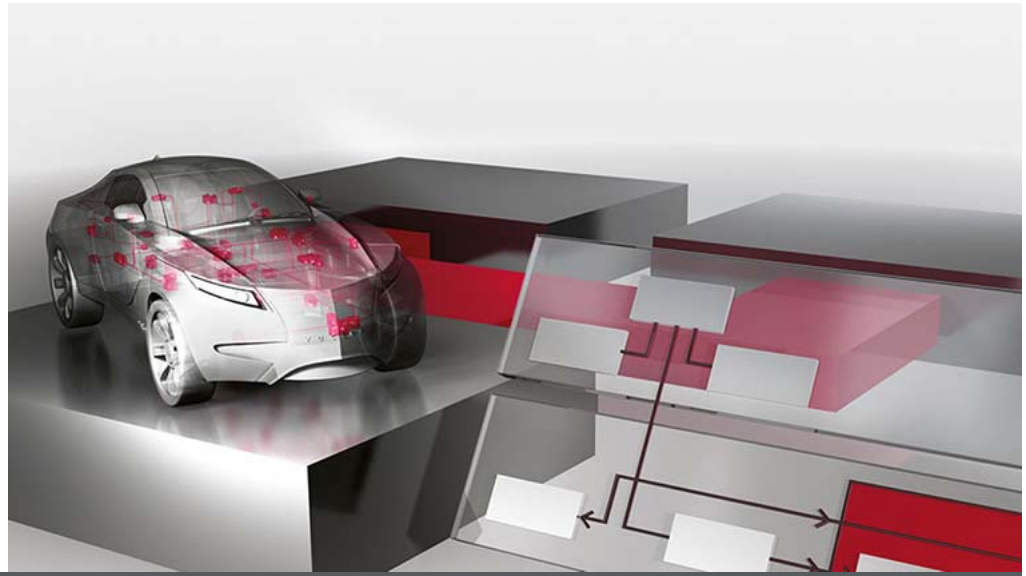
Wintersemester 2017/2018

Kapitel 3.5 – Anforderungsmanagement: E/E-Konzeptbewertung



Dr.-Ing. Clemens Reichmann
Clemens.Reichmann@vector.com,
Tel. 0721 / 91430-200

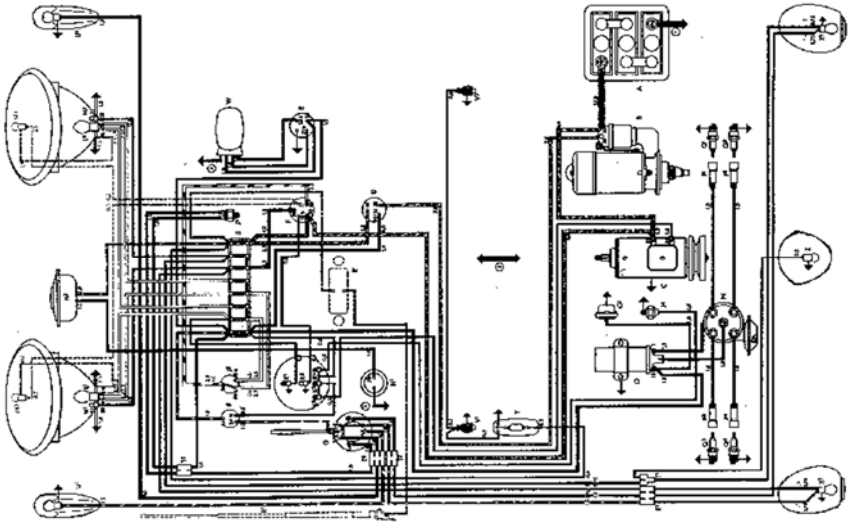
Institut für Technik der Informationsverarbeitung
Fakultät für Elektrotechnik & Informationstechnik
Universität Karlsruhe (KIT)



3.5. Elektrik/Elektronik-System-Modellierung u. Konzeptbewertung

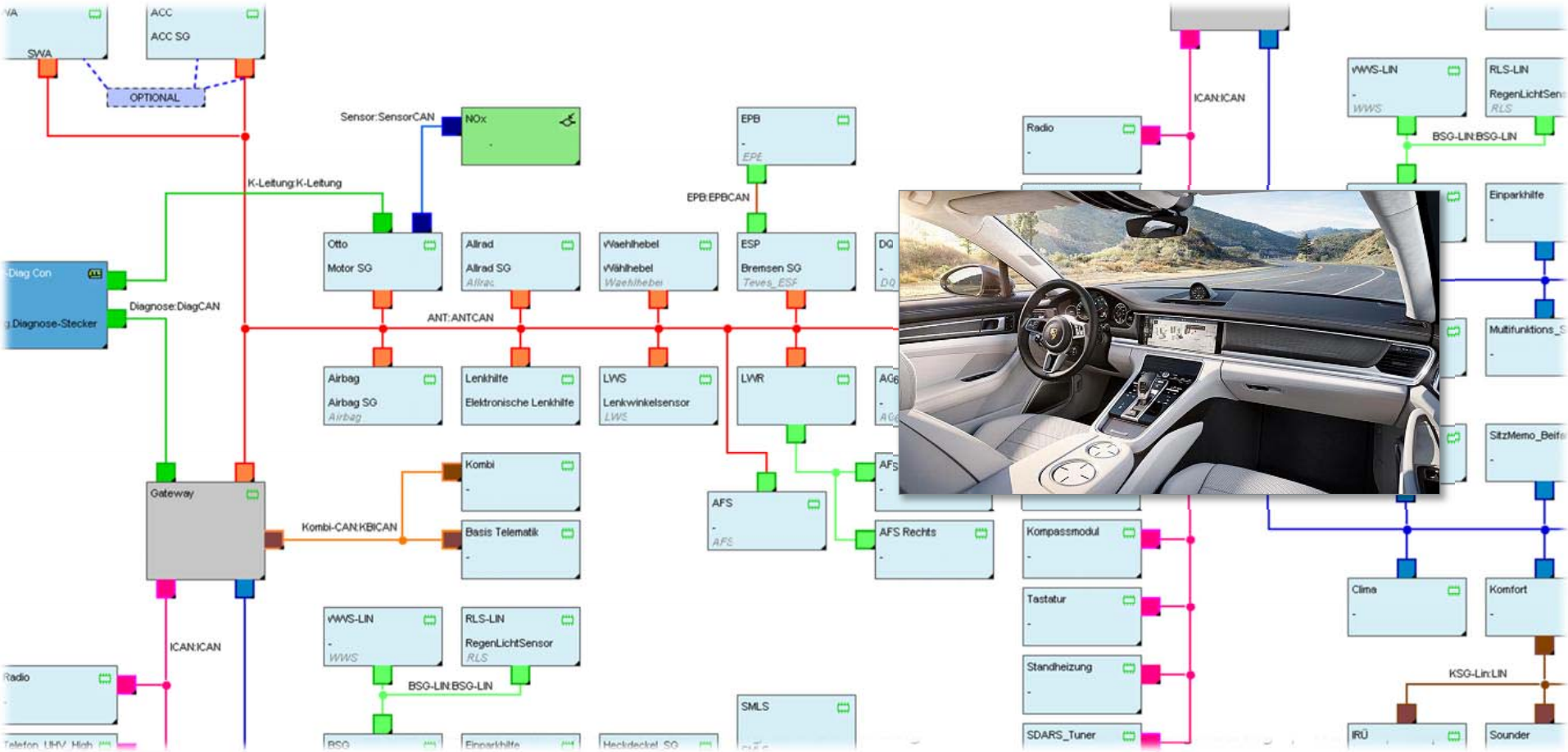
Overall E/E Complexity Increasing over Years now...

Past



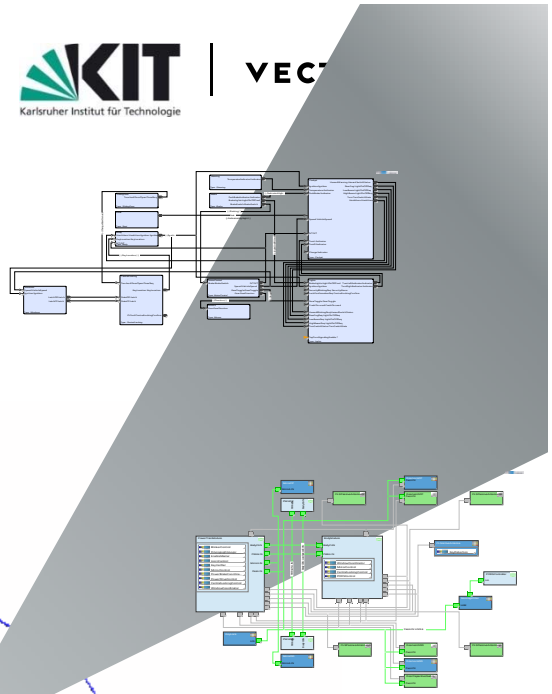
Overall E/E Complexity Increasing over Years now...

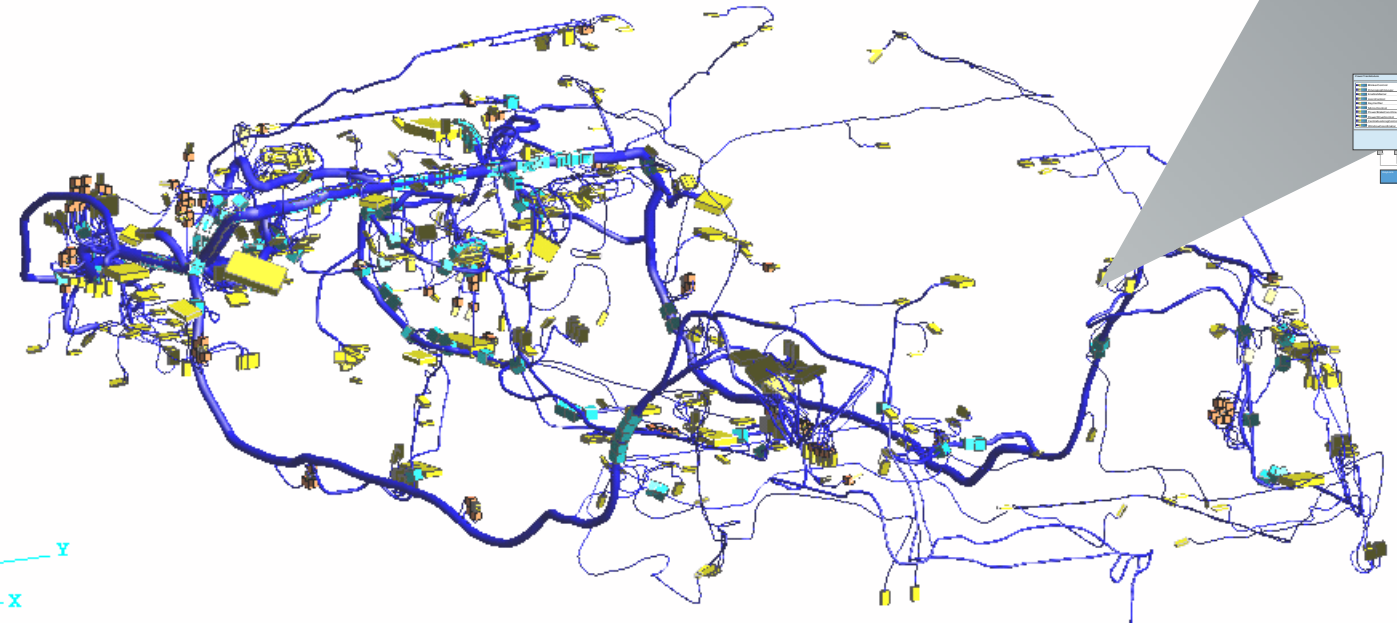
Present



Overall E/E Complexity Increasing over Years now...

- Active Management of System Complexity is necessary
  - Economical **cost** targets
  - Technical targets: weight, **fuel** consumption, …
  - Installation **space** is restricted, deeper integration
  - Upcoming technologies have to be considered and integrated
  - …

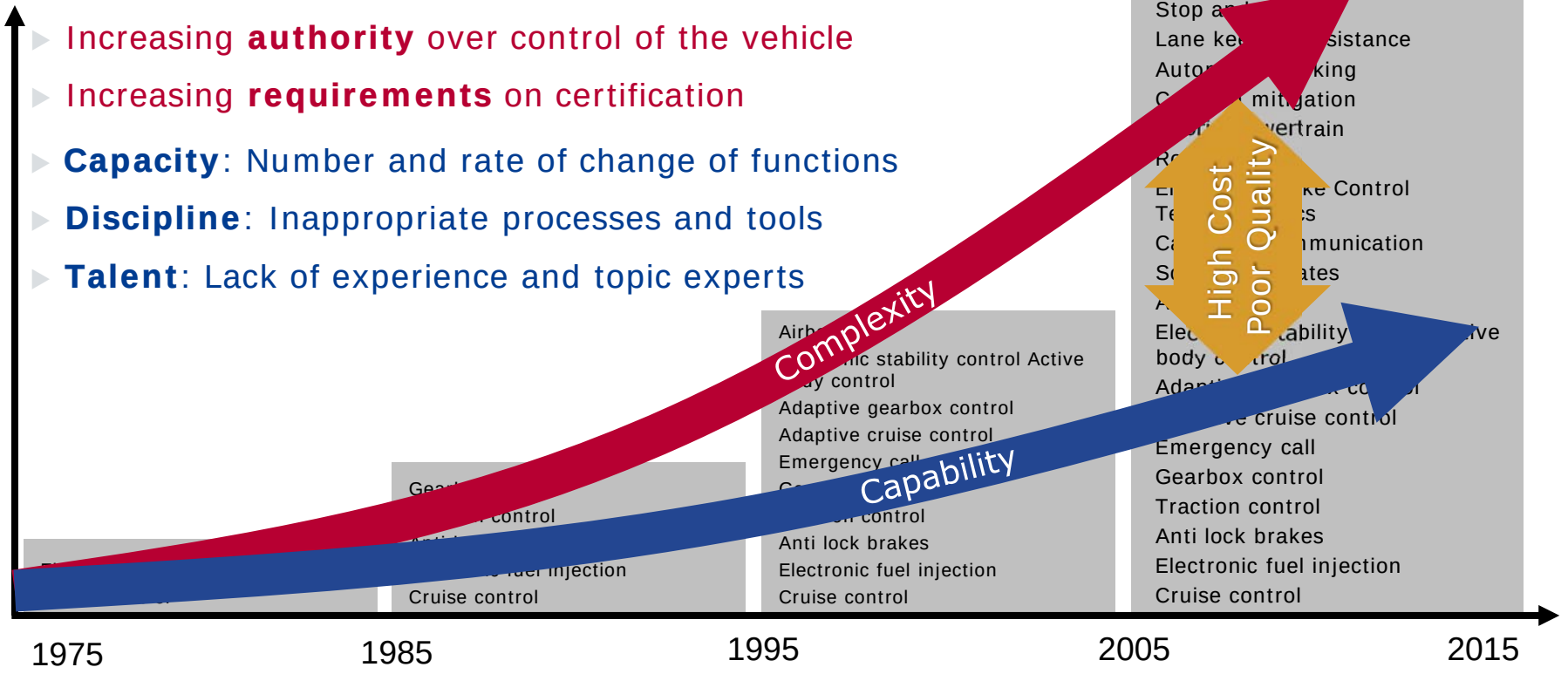




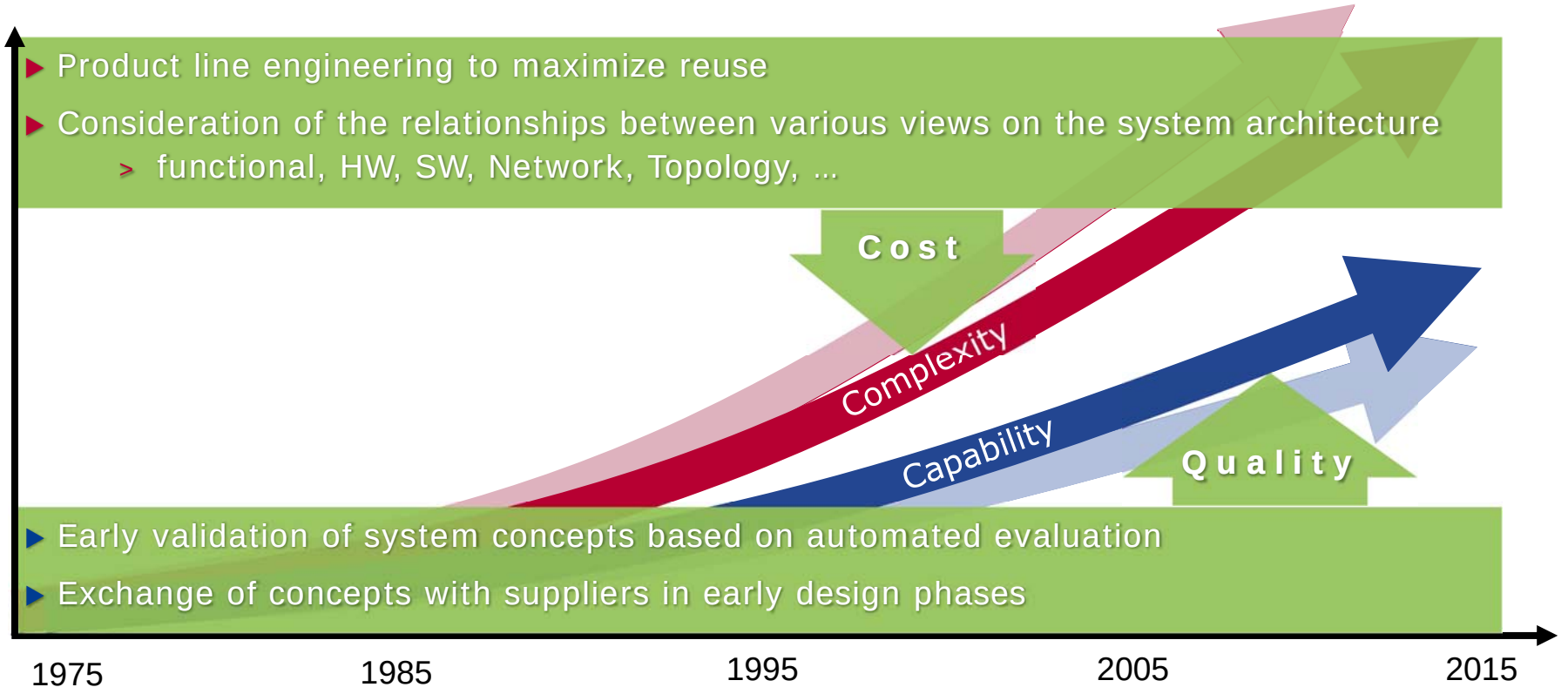
Challenges in E/E Development

The Complexity/Capability Gap

- Number, complexity and interaction of **functions**
- Increasing **authority** over control of the vehicle
- Increasing **requirements** on certification
- **Capacity**: Number and rate of change of functions
- **Discipline**: Inappropriate processes and tools
- **Talent**: Lack of experience and topic experts

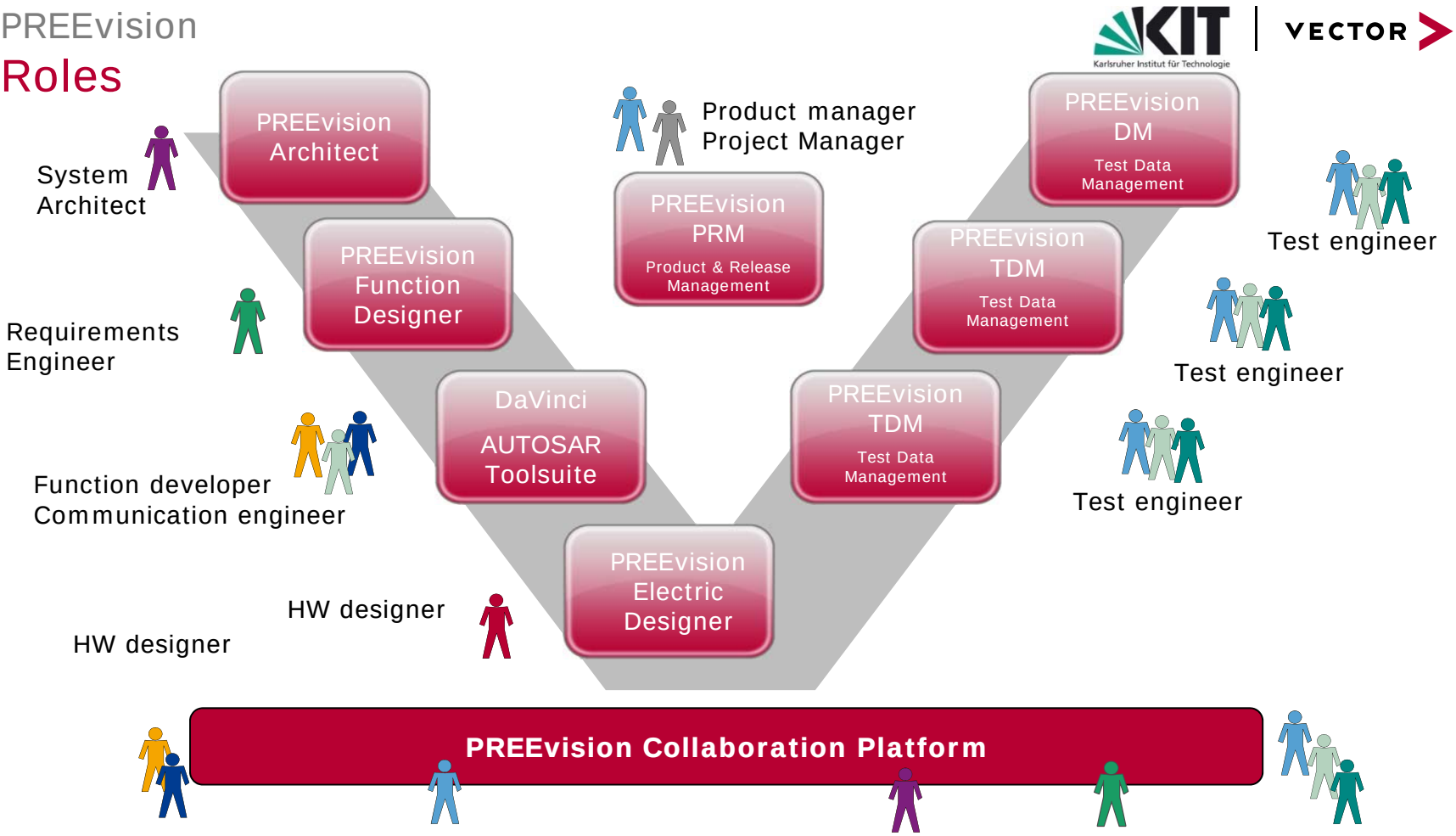


The Complexity/Capability Gap

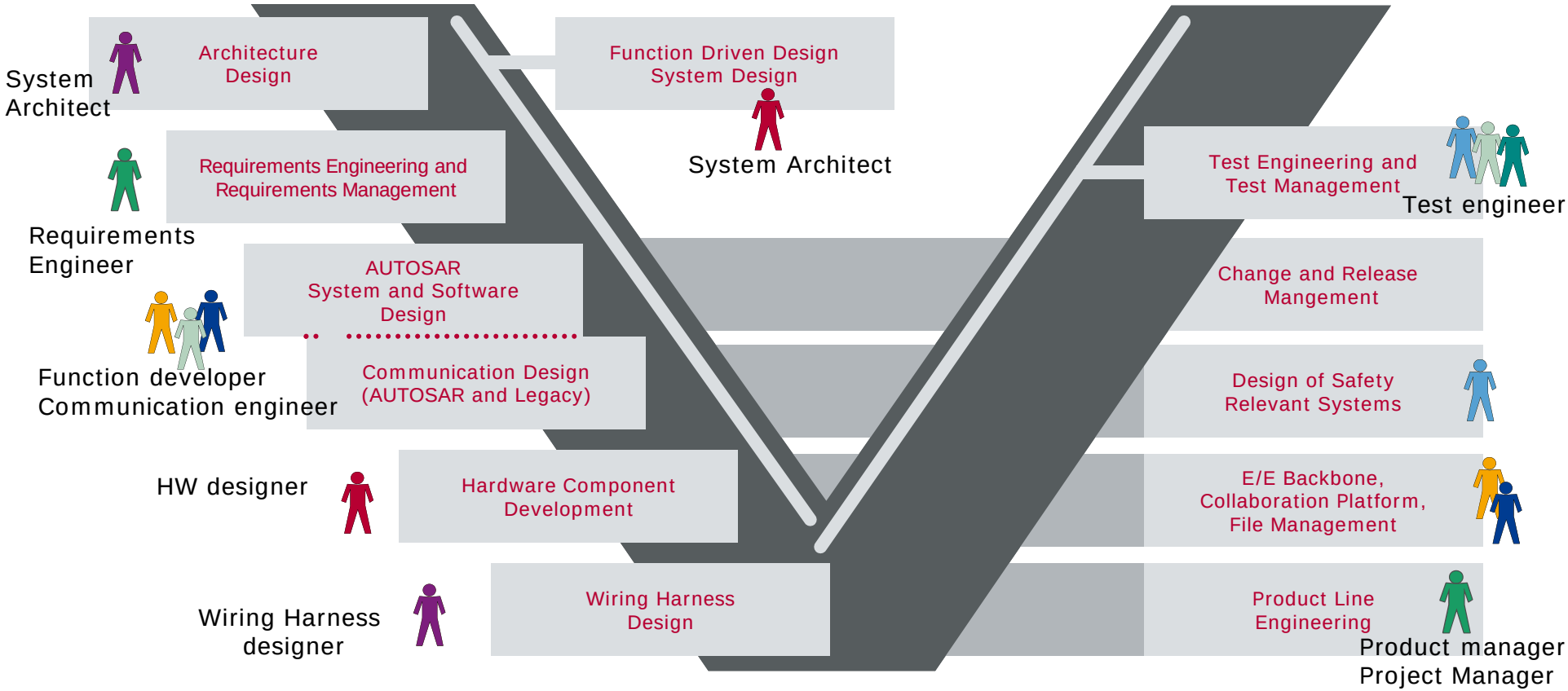


PREEvision

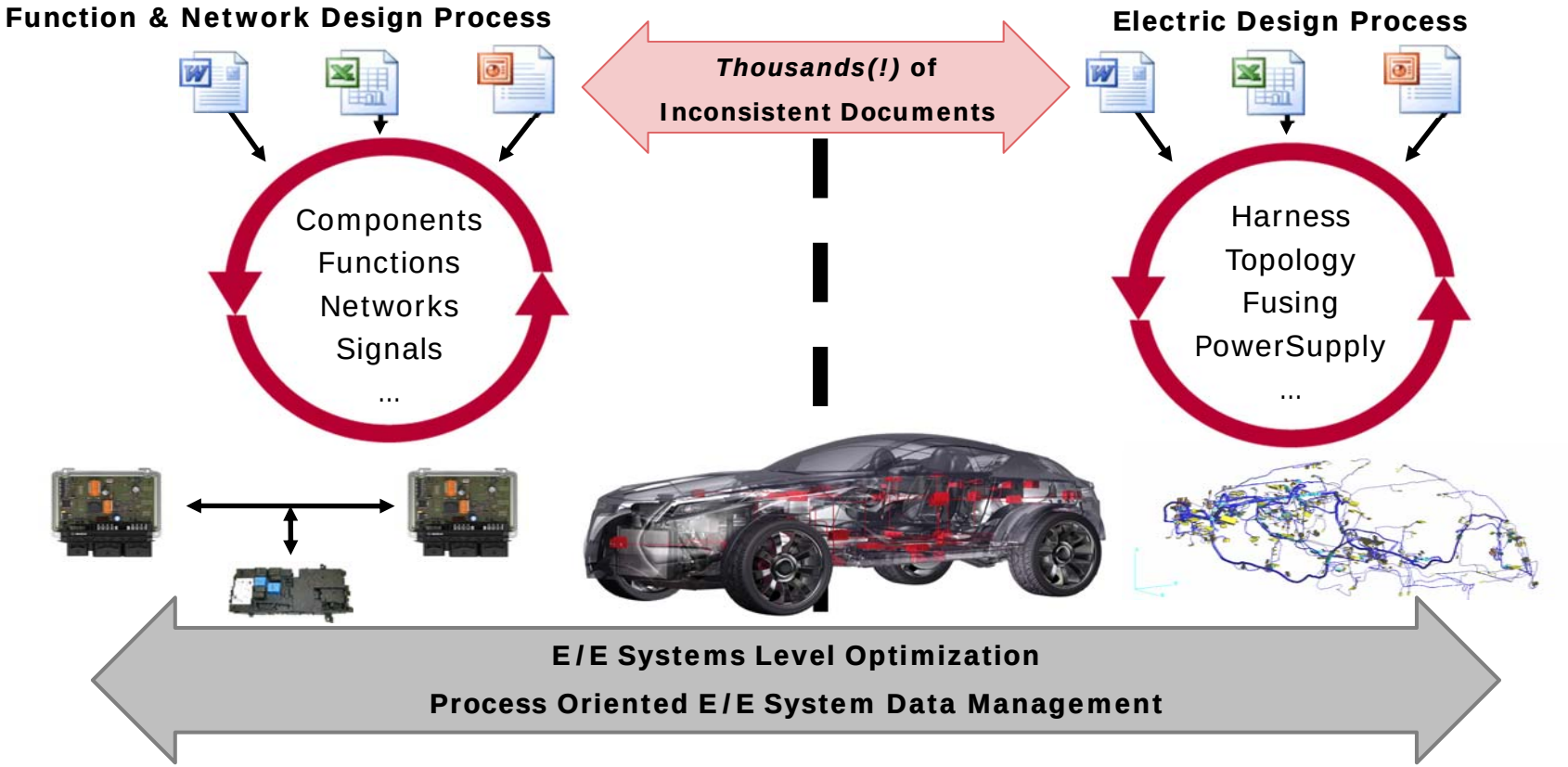
Roles



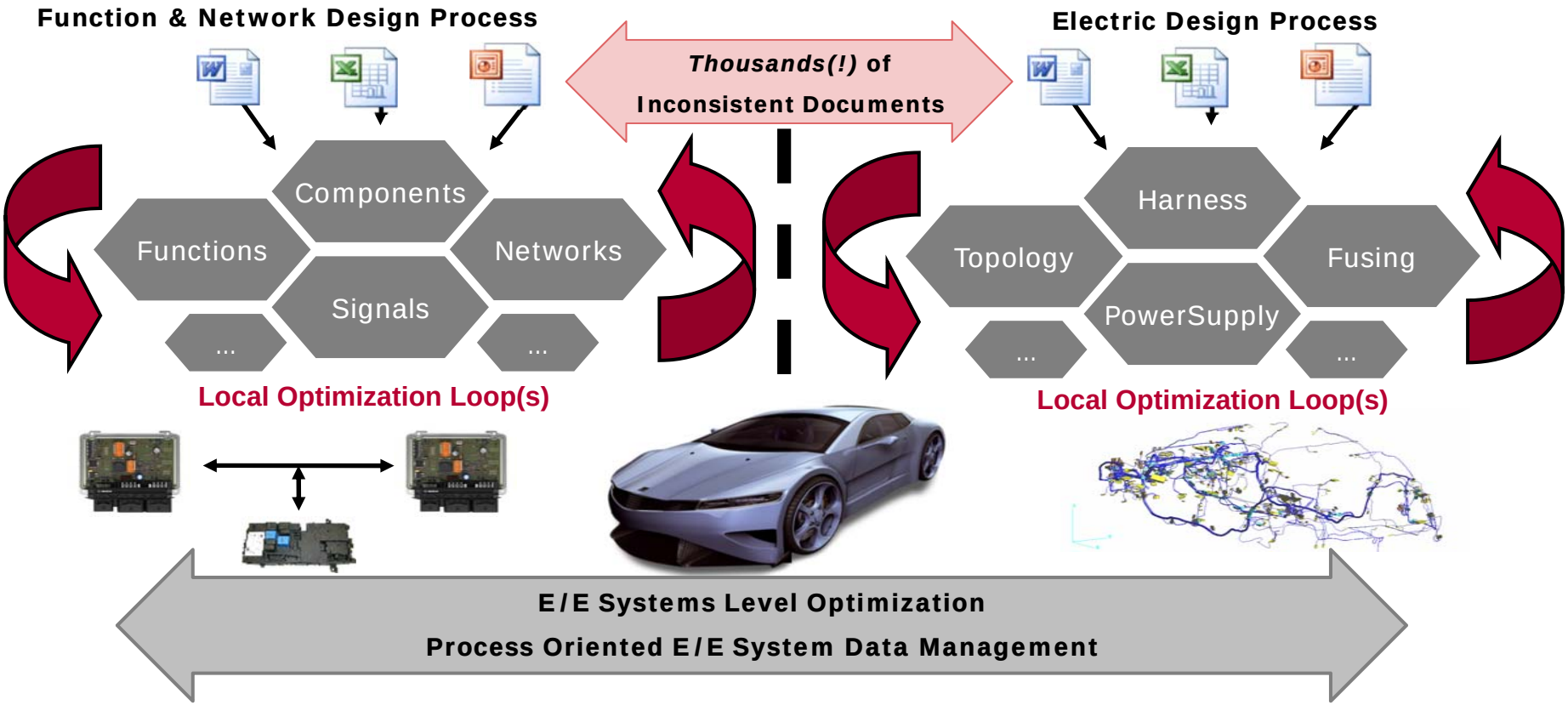
Supported E/E Engineering Use Cases and Roles



Current Situation – Document Based Development Process

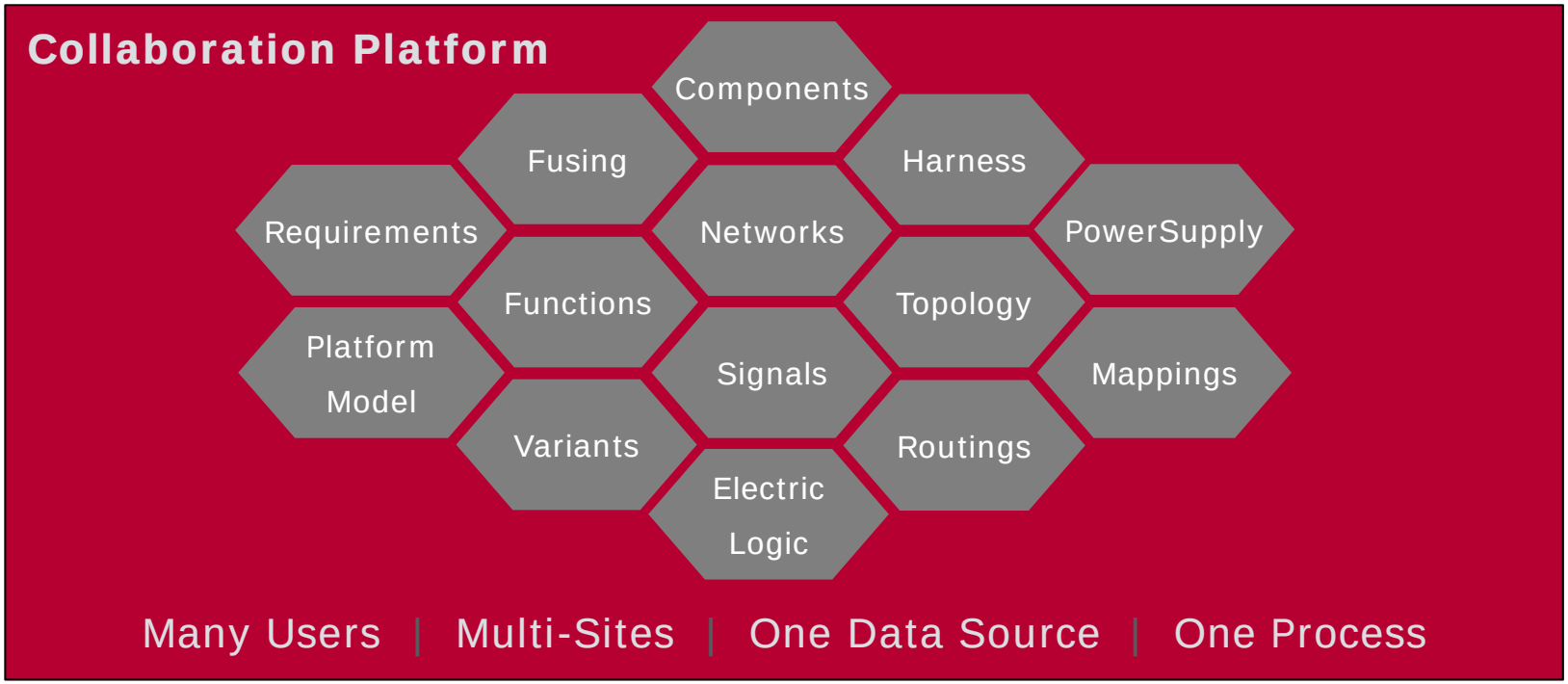


E/E Backbone – Central Team Server



Data-Oriented E/E Development Process

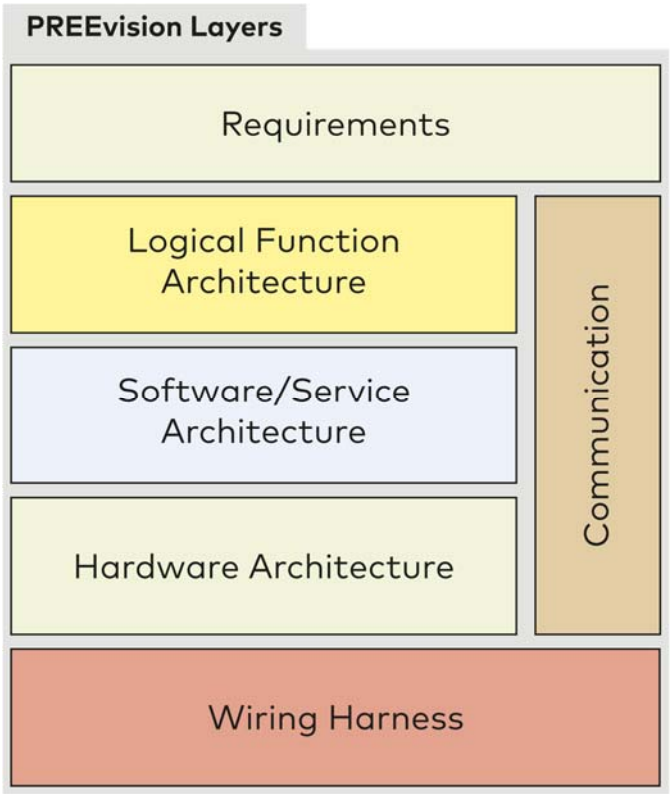
PREEVISION® One Data Model | One GUI | Full Traceability



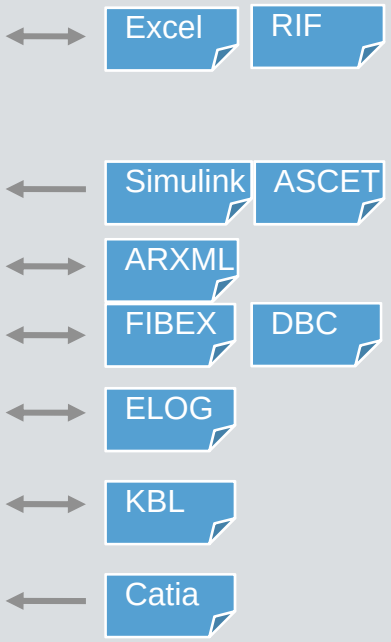
Architecture Layers

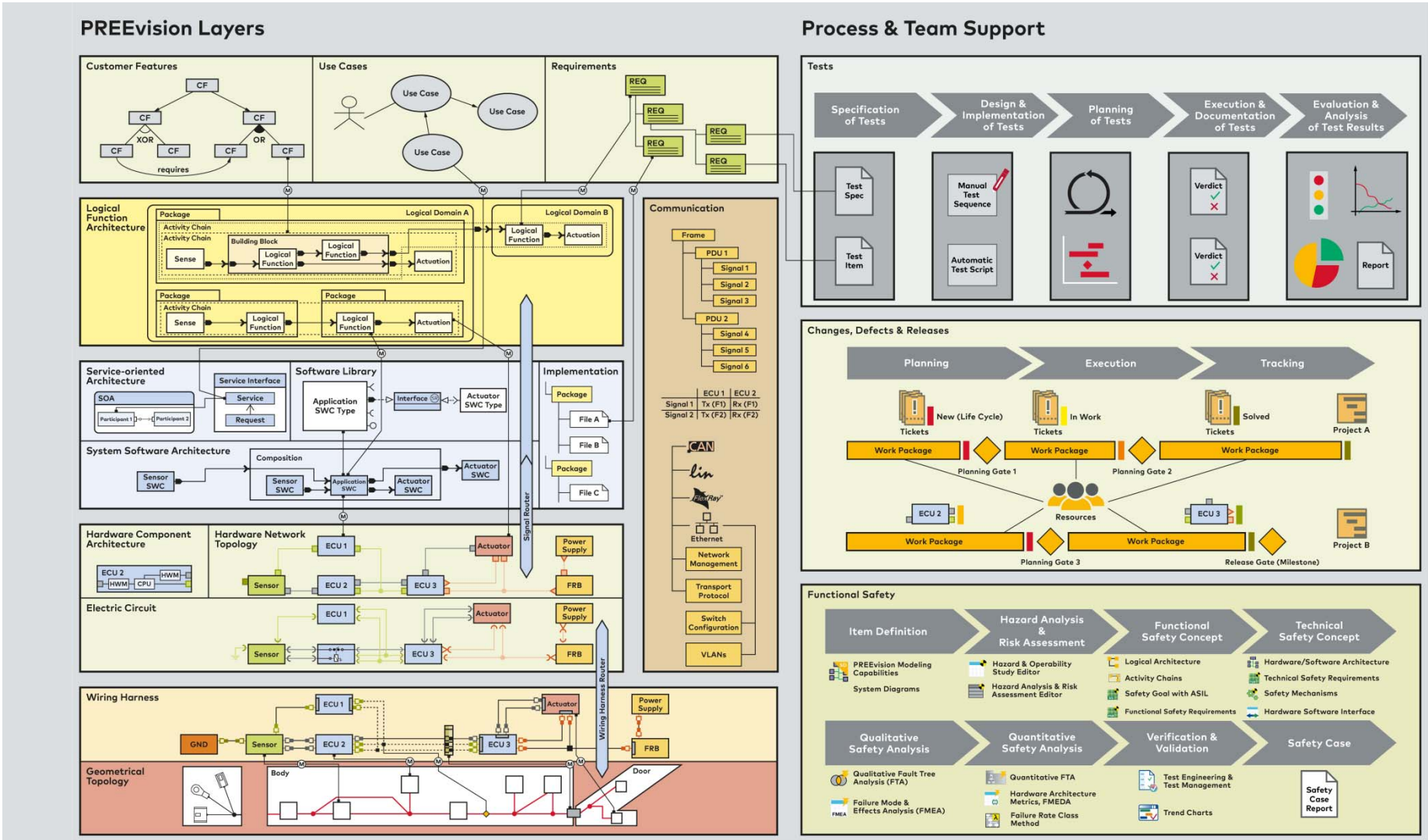
Engineering Aid

- Model Queries
- Consistency checks
- Metrics
- Report generation
- Synthesis
- Routing



Interfaces




PREEvision Layers
Customer Features
CF
XOR
OR
requires
Use Cases
Use Case
Requirements
REQ
Logical Function Architecture
Package
Logical Domain A
Logical Domain B
Activity Chain
Building Block
Sense
Logical Function
Actuation
Communication
Frame
PDU 1
Signal 1
Signal 2
Signal 3
PDU 2
Signal 4
Signal 5
Signal 6
ECU 1
ECU 2
Tx (F1)
Rx (F1)
Tx (F2)
Rx (F2)
CAN
lin
FlexRay
Ethernet
Network Management
Transport Protocol
Switch Configuration
VLANs
Service-oriented Architecture
SOA
Service Interface
Service
Request
Software Library
Application SWC Type
Interface
Actuator SWC Type
Implementation
File A
File B
File C
System Software Architecture
Composition
Sensor SWC
Application SWC
Actuator SWC
Signal Router
Hardware Component Architecture
HWM
CPU
Hardware Network Topology
Electric Circuit
Sensor
Actuator
Power Supply
FRB
Wiring Harness
GND
Wiring Harness Router
Geometrical Topology
Body
Door
Process & Team Support
Tests
Specification of Tests
Design & Implementation of Tests
Planning of Tests
Execution & Documentation of Tests
Evaluation & Analysis of Test Results
Test Spec
Test Item
Manual Test Sequence
Automatic Test Script
Verdict
Report
Changes, Defects & Releases
Planning
Execution
Tracking
Tickets
New (Life Cycle)
In Work
Solved
Project A
Project B
Work Package
Planning Gate 1
Planning Gate 2
Planning Gate 3
Release Gate (Milestone)
Resources
Functional Safety
Item Definition
Hazard Analysis & Risk Assessment
Functional Safety Concept
Technical Safety Concept
PREEvision Modeling Capabilities
System Diagrams
Hazard & Operability Study Editor
Hazard Analysis & Risk Assessment Editor
Logical Architecture
Activity Chains
Safety Goal with ASIL
Functional Safety Requirements
Hardware/Software Architecture
Technical Safety Requirements
Safety Mechanisms
Hardware Software Interface
Qualitative Safety Analysis
Quantitative Safety Analysis
Verification & Validation
Safety Case
Qualitative Fault Tree Analysis (FTA)
Failure Mode & Effects Analysis (FMEA)
Quantitative FTA
Hardware Architecture Metrics, FMEDA
Failure Rate Class Method
Test Engineering & Test Management
Trend Charts
Safety Case Report

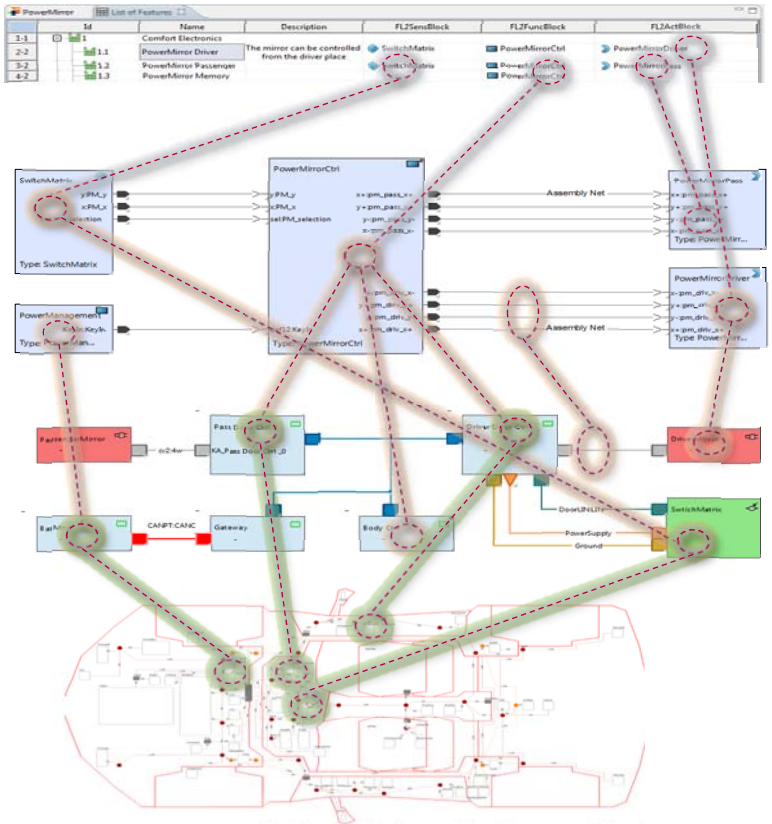
Model-Based Systems Engineering

Require-ments

Logical/SW Architecture

Network/HW Architecture

Wiring/ Geometry



- **Domain specific** language and data model.
- **Single source model** across all development levels and disciplines
- **Collaboration** Platform
- Support for **reuse** and **product line engineering**.
- Automated **report generation** and **consistency checks**.
- **Metrics** for Benchmarking and **Automated algorithms** for scheduling, signal routing, etc.
- **Import and export** (e.g. AUTOSAR, KBL, FIBEX...)

Managing Complexity

Variants / Reuse

Productline Support

Extended Vehicle-Variant-Management (brands, construction types)

- Definition of ProductLine structure
- Reuse of ReuseUnits
- Adaption of ReuseUnits
- Integration of ReuseUnits

E/E Domain Modeling

Seamless E/E Data Model

- Integrated Requirements and Test Management
- E/E & SW/Impl.
  - E/E-Modeling
  - Data-/Filemanagement
  - Metadata of Impl. Artifacts
  - Traceability

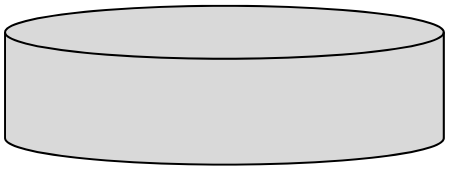
Feasibility & Efficiency

- Cost, Optimizations
- Synthesis
- Evaluation / Benchmarking

Many People

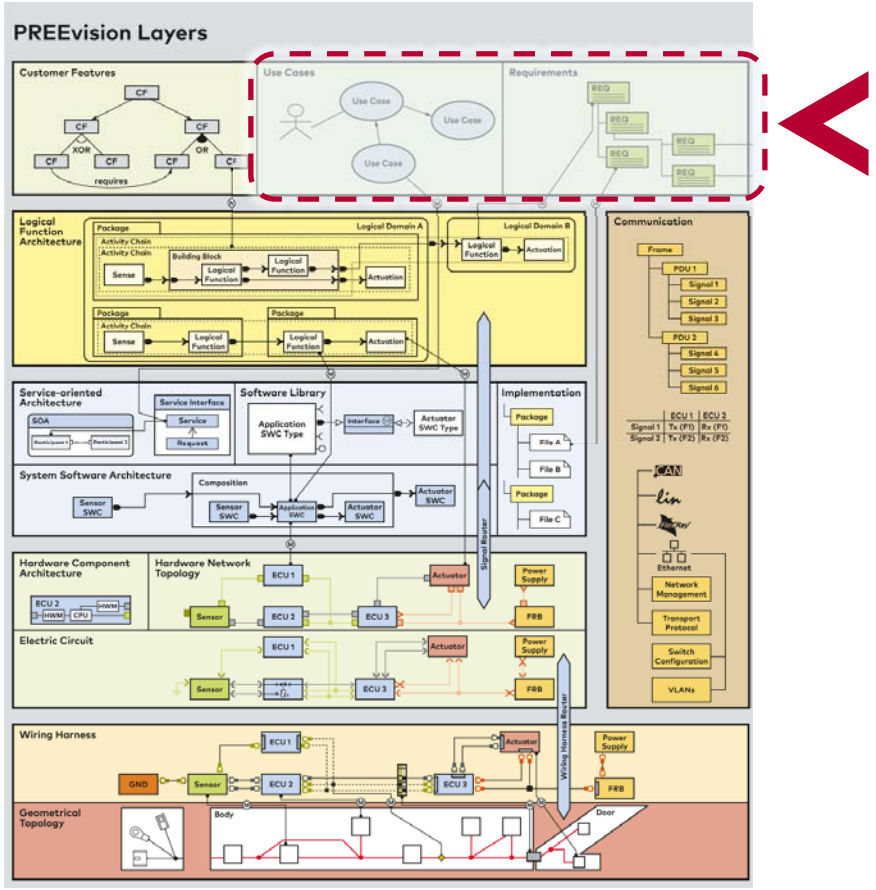
Multiuser Environment

- Team Collaboration
- E/E Data Backbone
- Process Integration

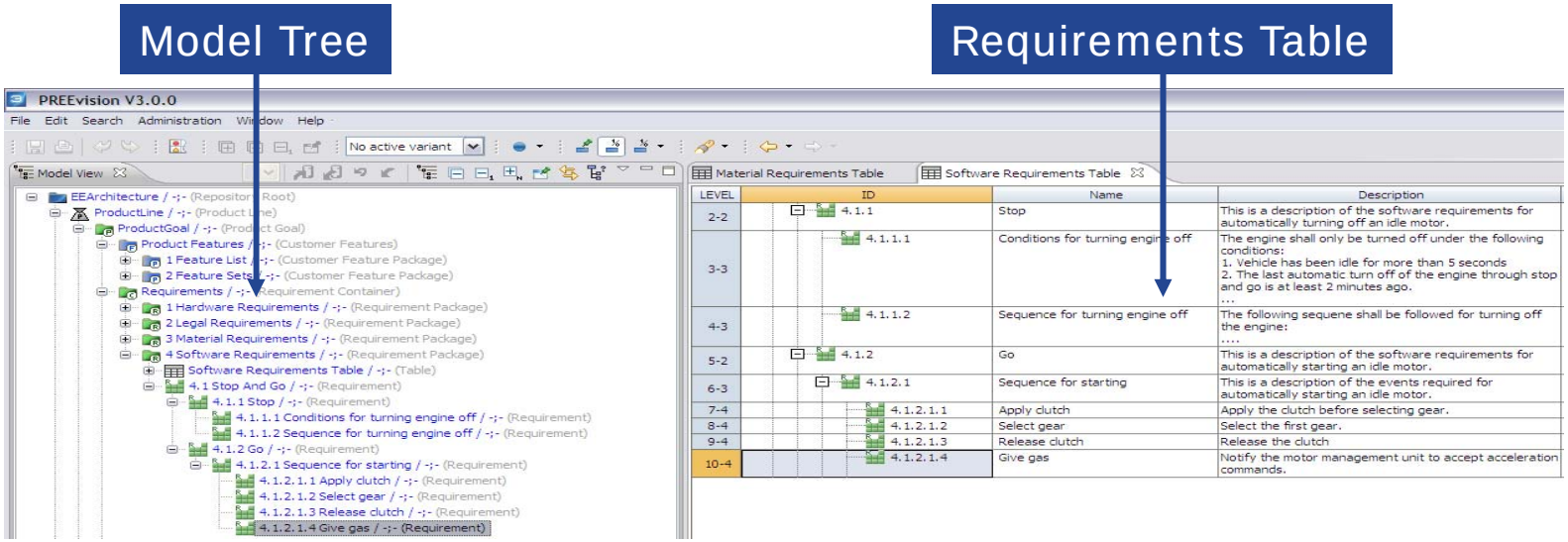


PREEvision Collaboration Platform (I)

Requirements Management

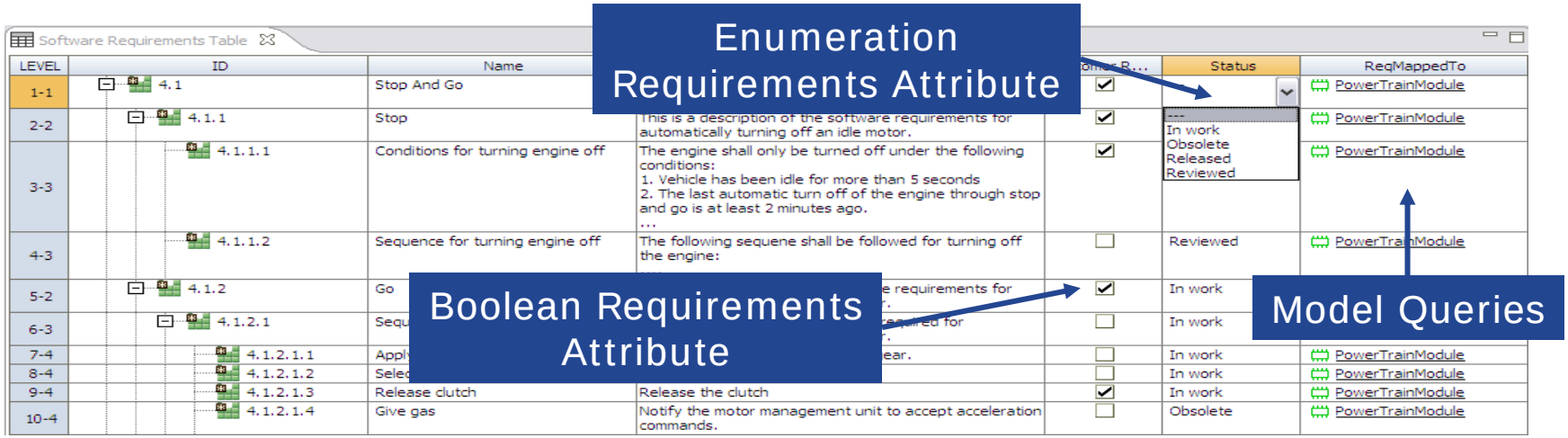


Structuring and Editing



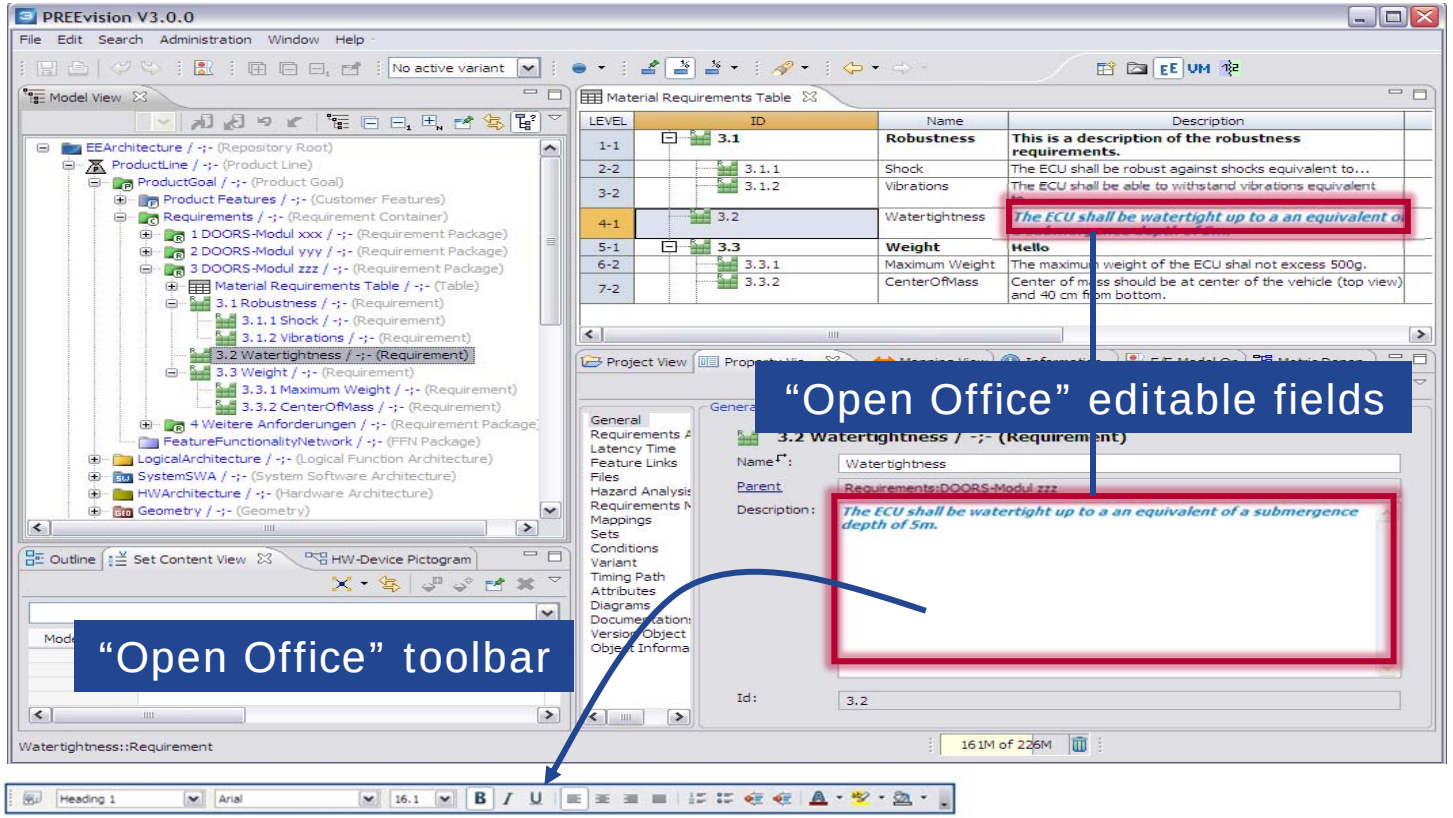
- Requirements can be grouped into requirements packages and structured hierarchically.
- Tables provide an efficient overview and editing capability.

Attributes and tables

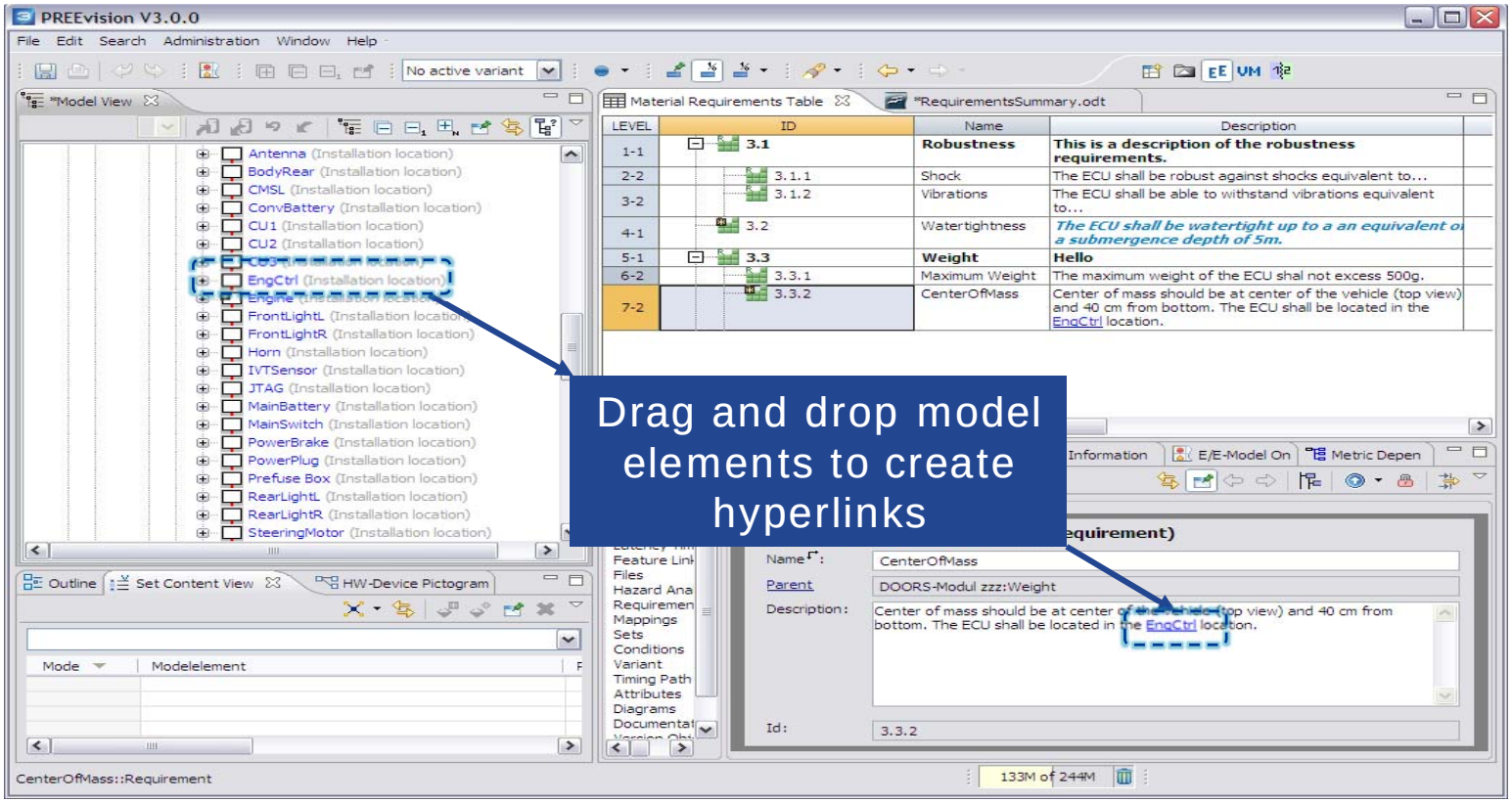


- Requirements can be extended with user-defined attributes that can be directly edited in the requirements tables.
- Attributes can be typed, e.g. Boolean, Enumeration, Integers with valid value ranges,..., and are handled accordingly in tables.
- Requirements tables can display the results of model queries.
  (E.g. ECUs to which the requirements are mapped.)

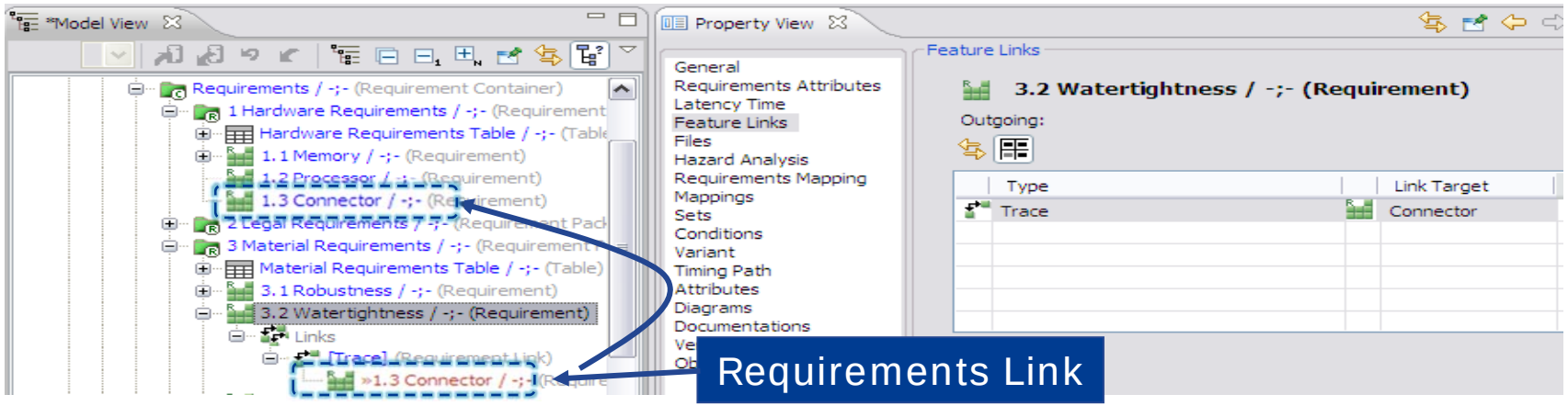
Editable Fields



Hyperlinks

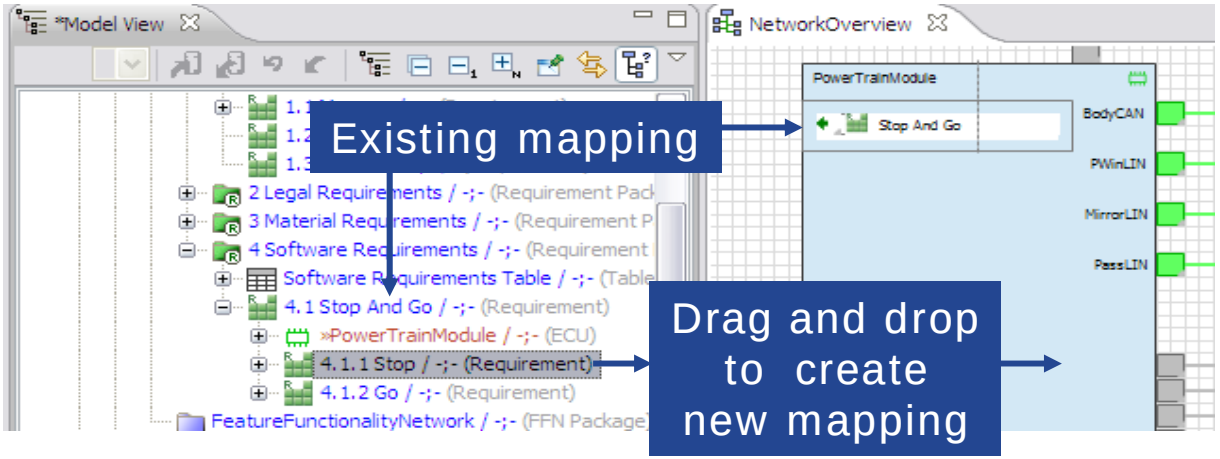

PREEvision V3.0.0
File Edit Search Administration Window Help
No active variant
*Model View
Material Requirements Table
*RequirementsSummary.odt
LEVEL
ID
Name
Description
1-1 3.1 Robustness This is a description of the robustness requirements.
2-2 3.1.1 Shock The ECU shall be robust against shocks equivalent to...
3-2 3.1.2 Vibrations The ECU shall be able to withstand vibrations equivalent to...
4-1 3.2 Watertightness The ECU shall be watertight up to a an equivalent of a submergence depth of 5m.
5-1 3.3 Weight Hello
6-2 3.3.1 Maximum Weight The maximum weight of the ECU shal not excess 500g.
7-2 3.3.2 CenterOfMass Center of mass should be at center of the vehicle (top view) and 40 cm from bottom. The ECU shall be located in the EngCtrl location.
Antenna (Installation location)
BodyRear (Installation location)
CMSL (Installation location)
ConvBattery (Installation location)
CU1 (Installation location)
CU2 (Installation location)
EngCtrl (Installation location)
FrontLightL (Installation location)
FrontLightR (Installation location)
Horn (Installation location)
IVTSensor (Installation location)
JTAG (Installation location)
MainBattery (Installation location)
MainSwitch (Installation location)
PowerBrake (Installation location)
PowerPlug (Installation location)
Prefuse Box (Installation location)
RearLightL (Installation location)
RearLightR (Installation location)
SteeringMotor (Installation location)
Drag and drop model elements to create hyperlinks
Outline
Set Content View
HW-Device Pictogram
Mode
Modelelement
Feature Link
Files
Hazard Ana
Requiremen
Mappings
Sets
Conditions
Variant
Timing Path
Attributes
Diagrams
Documentat
Information
E/E-Model On
Metric Depen
Name:
CenterOfMass
Parent
DOORS-Modul zzz:Weight
Description:
Center of mass should be at center of the vehicle (top view) and 40 cm from bottom. The ECU shall be located in the EngCtrl location.
Id:
3.3.2
CenterOfMass::Requirement
133M of 244M

Linking



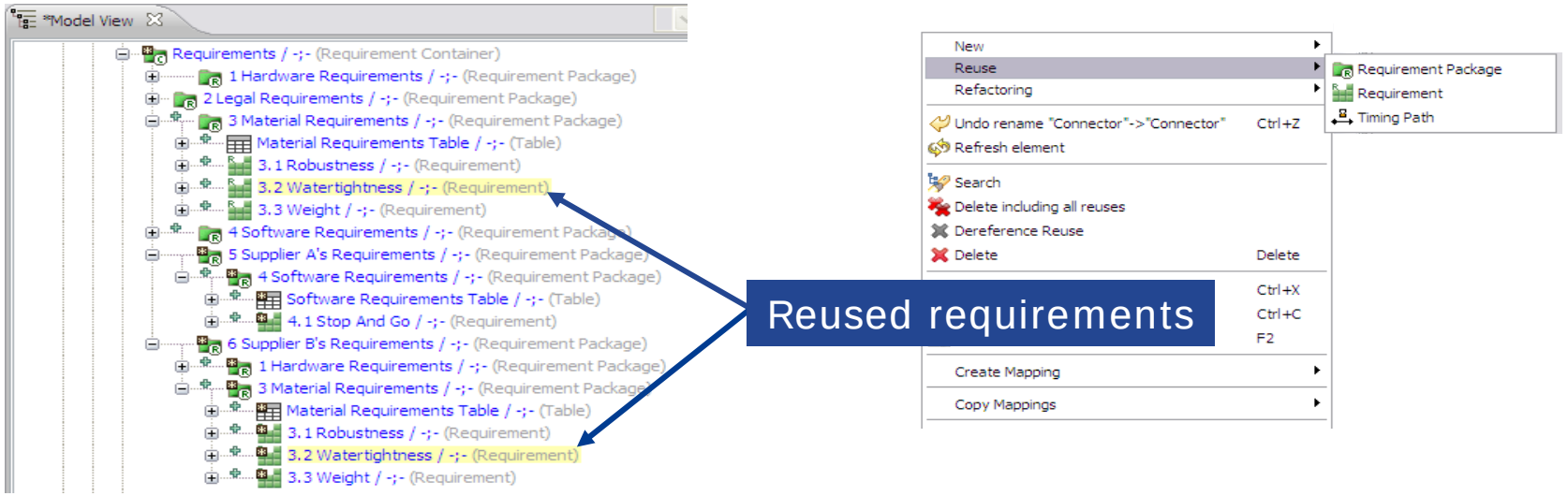
- Requirements can be linked to other requirements to maintain traceability.
- Linked requirements are shown in the model tree and property view.
- User can directly navigate to linked requirements by pressing the space bar.

Mapping



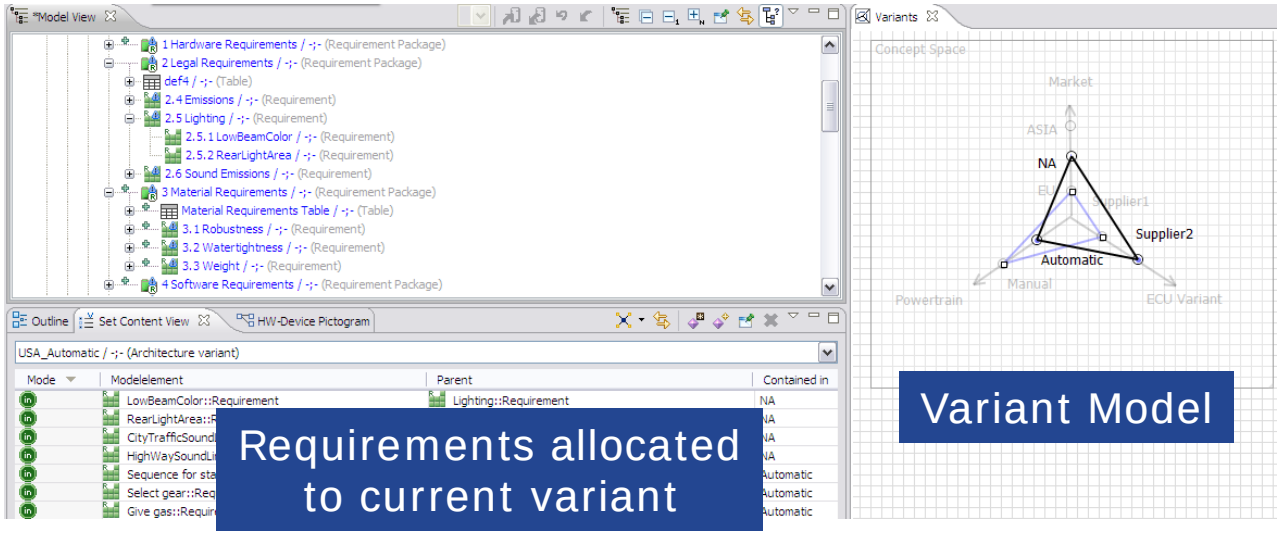
- Requirements can be directly mapped to architecture artefacts
  E.g. Logical functions, SW Functions, ECUs, Hardware modules,…
- Mappings can be created via drag and drop between requirements in the model tree and graphical elements.
- The results of the mappings can be displayed in the diagrams and model tree.

Reuse of Requirements



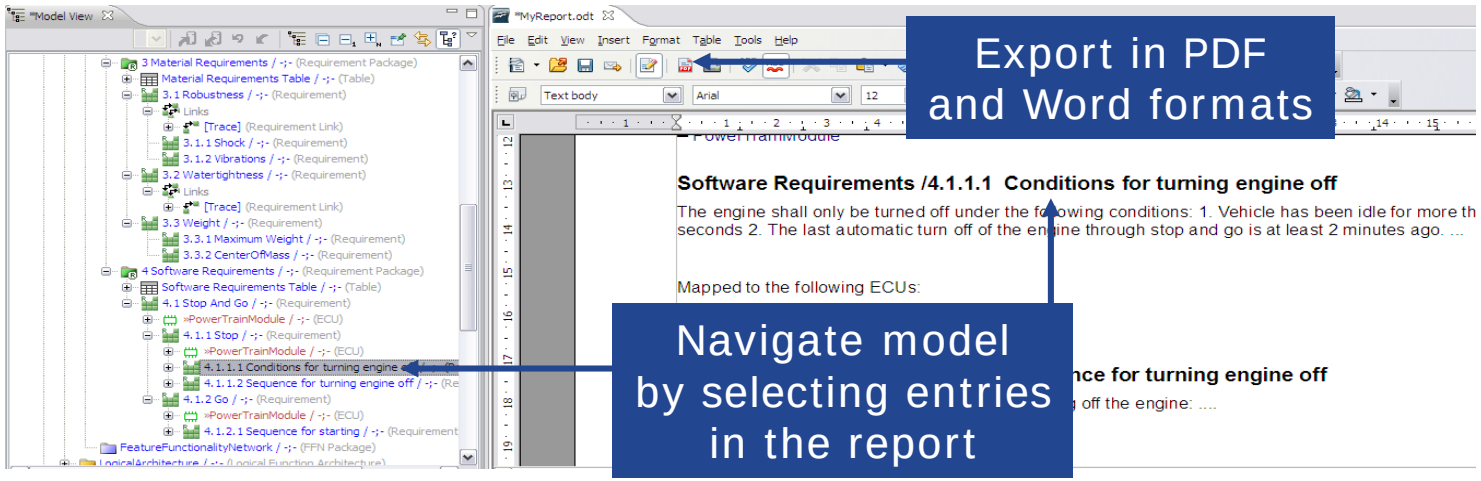
- Requirements, Requirements Packages and Timings can be reused in a number of contexts.
- Changes in the description are copied across all reuses.
- Reuse can be used to efficiently manage different groupings of the same requirements, e.g. for different suppliers.

Variant Management



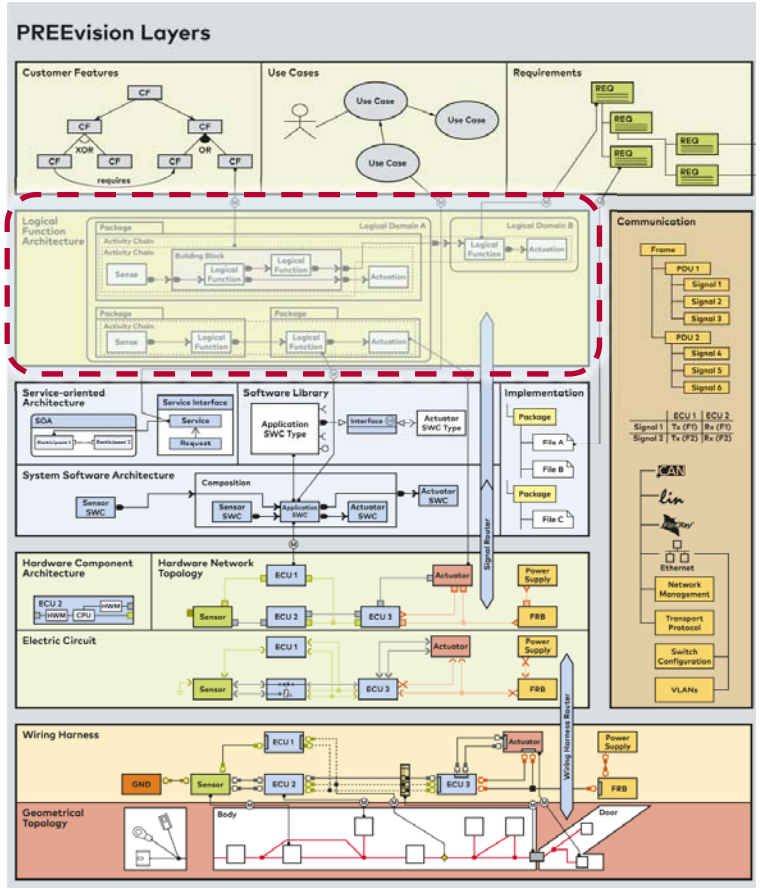
- Requirements can be allocated to sets and alternatives defined in the variant model.
- Requirements not allocated to the current selected variant are indicated in the model tree ( )
- Can be used to manage variant-specific requirements.

Report Generation



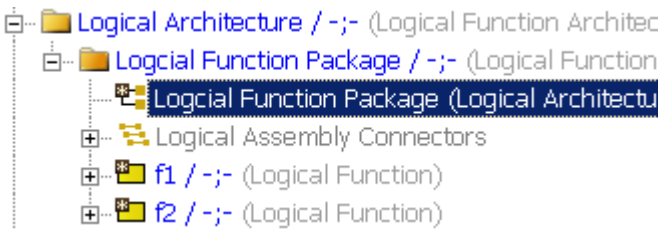
- Reports can be generated based on user defined templates.
- High level of flexibility in the report format including the use of tables, diagrams and complex model queries.
- Generation of variant-specific reports possible (e.g. to create supplier specific specifications in PDF).

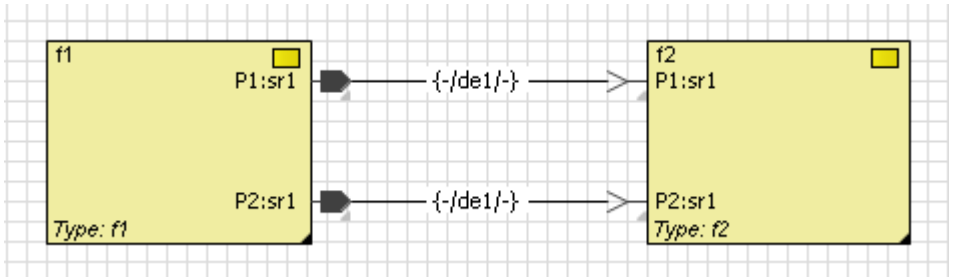
Logische Architektur



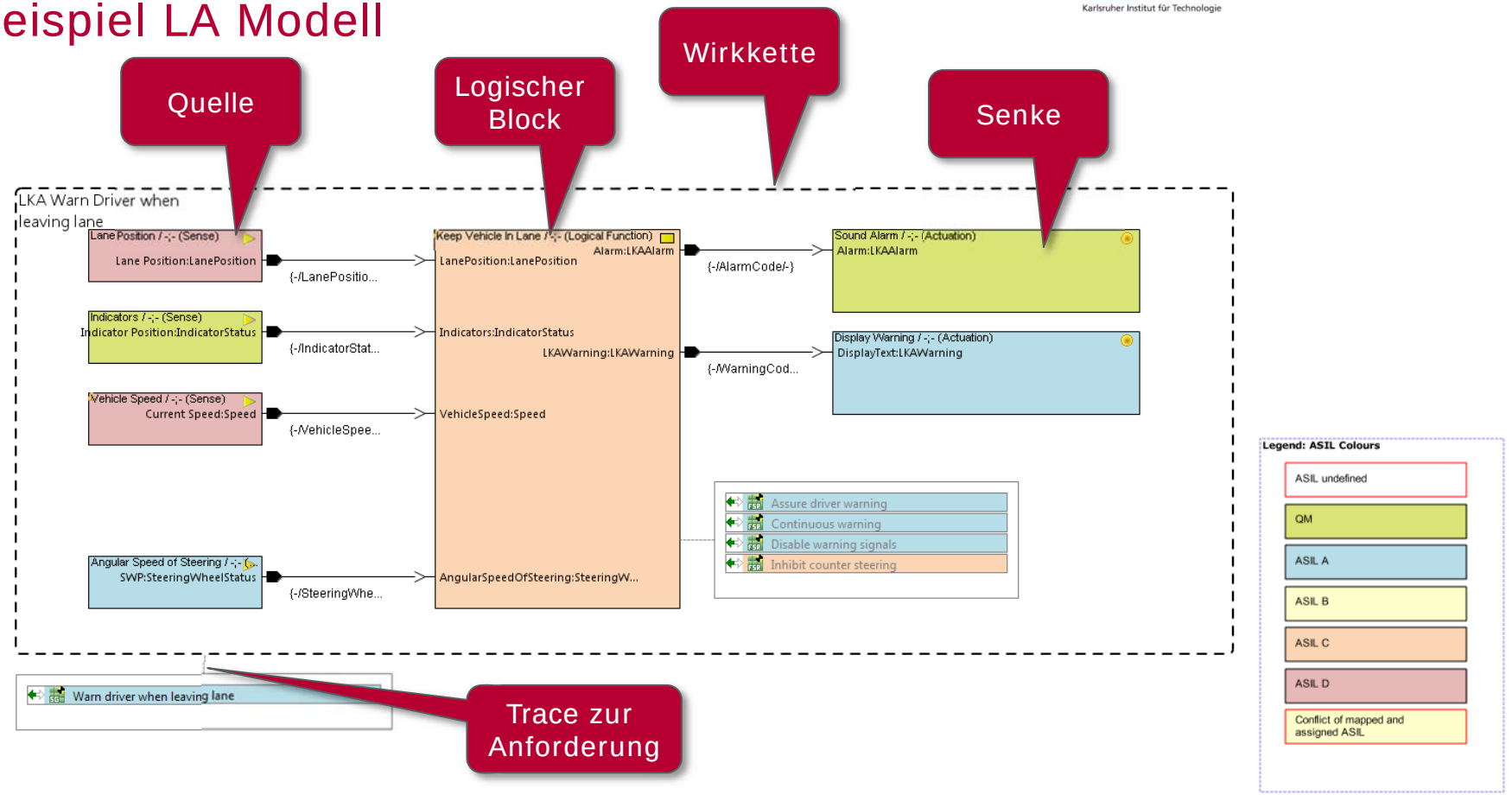
Logical Architecture

- The network communication specification for distributed systems is typically driven by the Logical Architecture and its mapping to the Hardware Architecture
- The Logical Architecture specification is supported by graphical block diagrams
- The communication between logical functions in the Logical Architecture is specified by ports and connections (in a similar way as AUTOSAR)

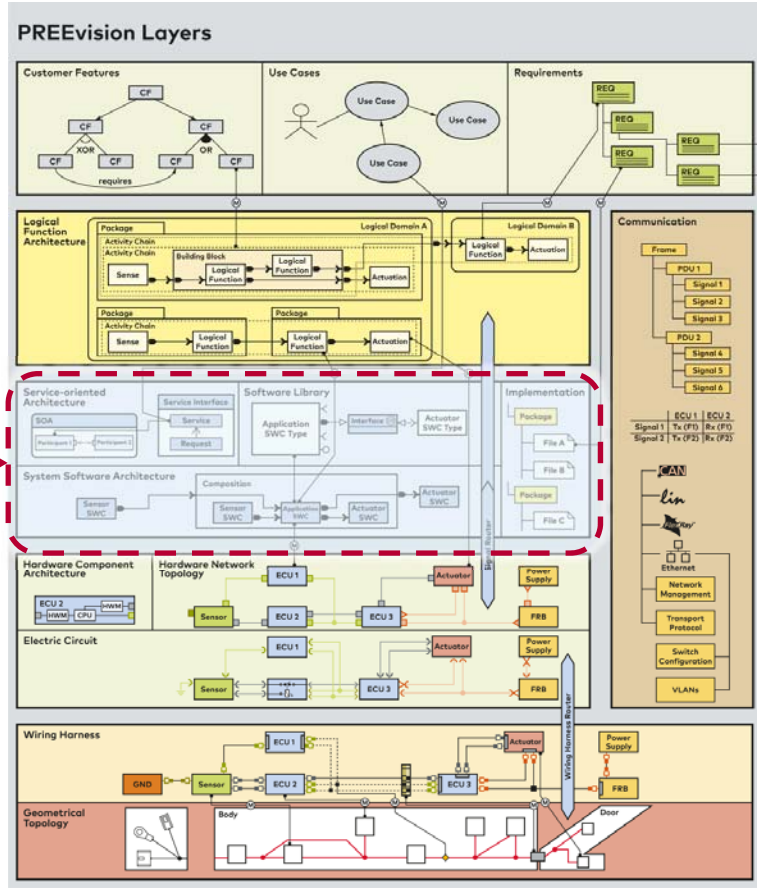




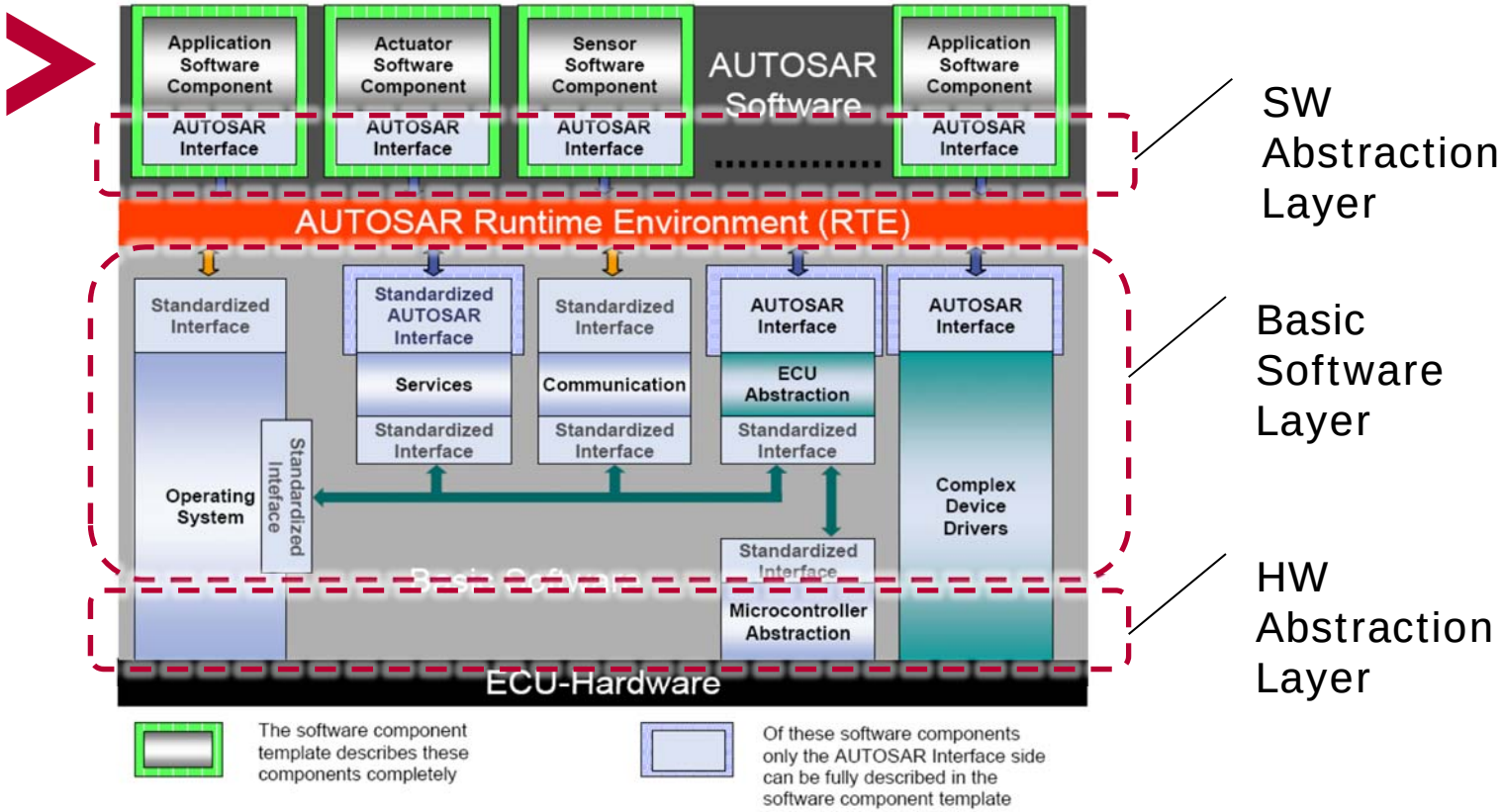
Beispiel LA Modell



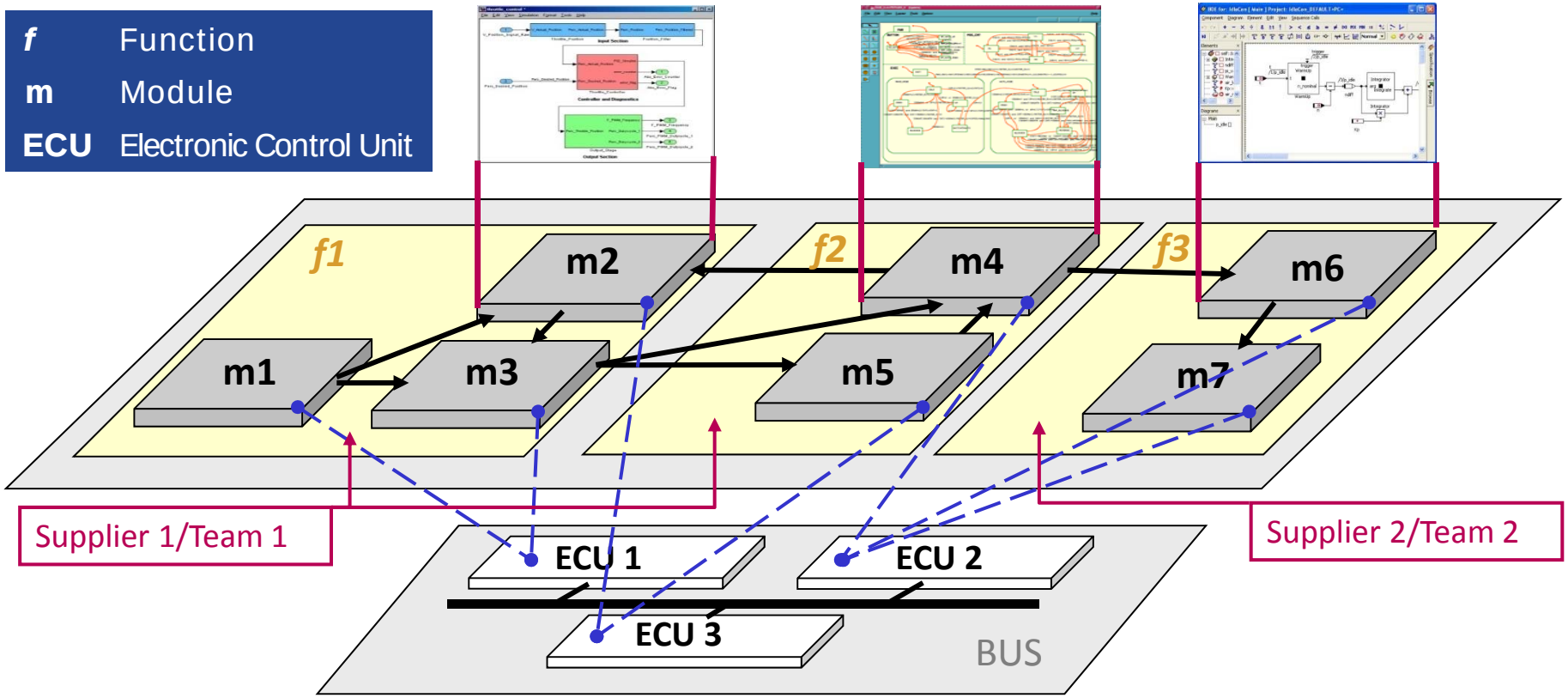
SW Architektur



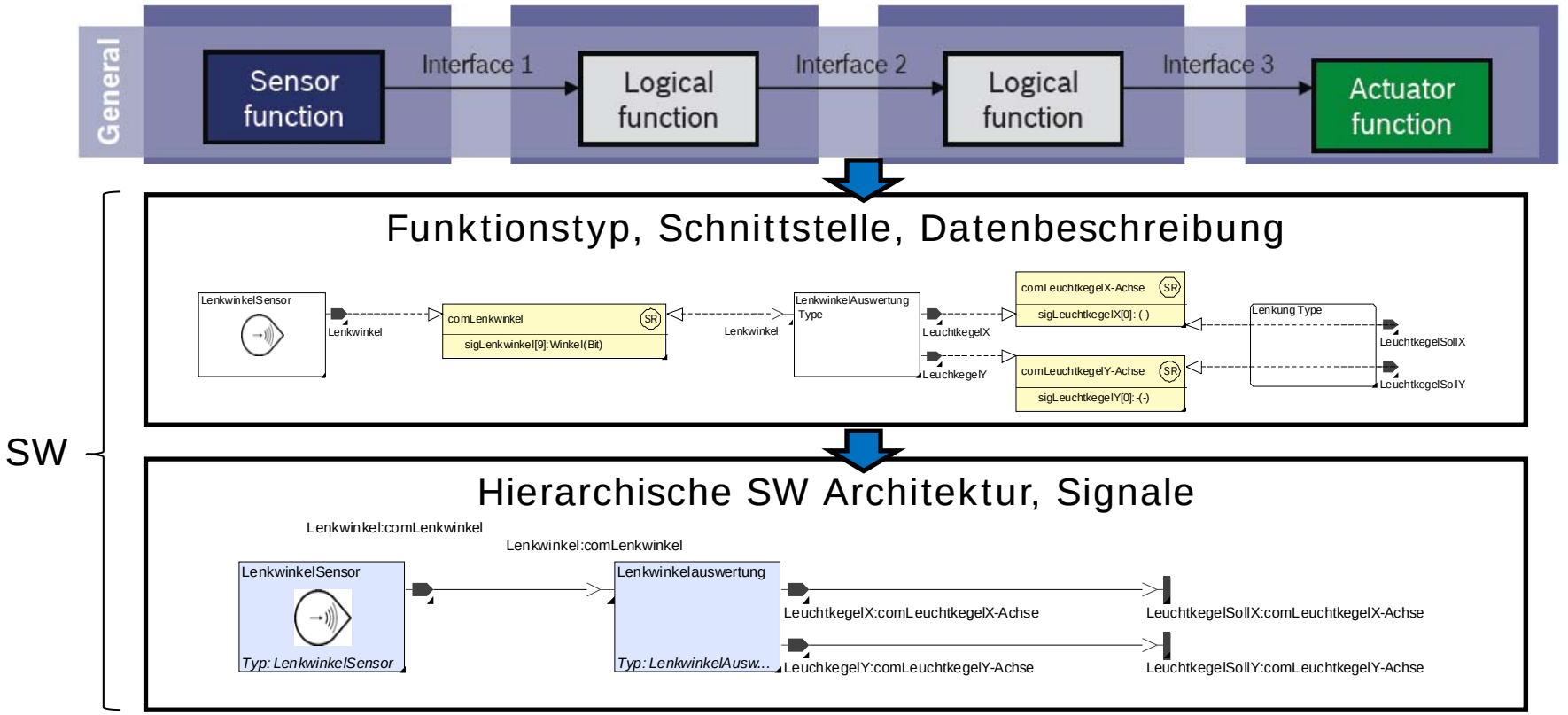
Scope in the ECU SW Architecture



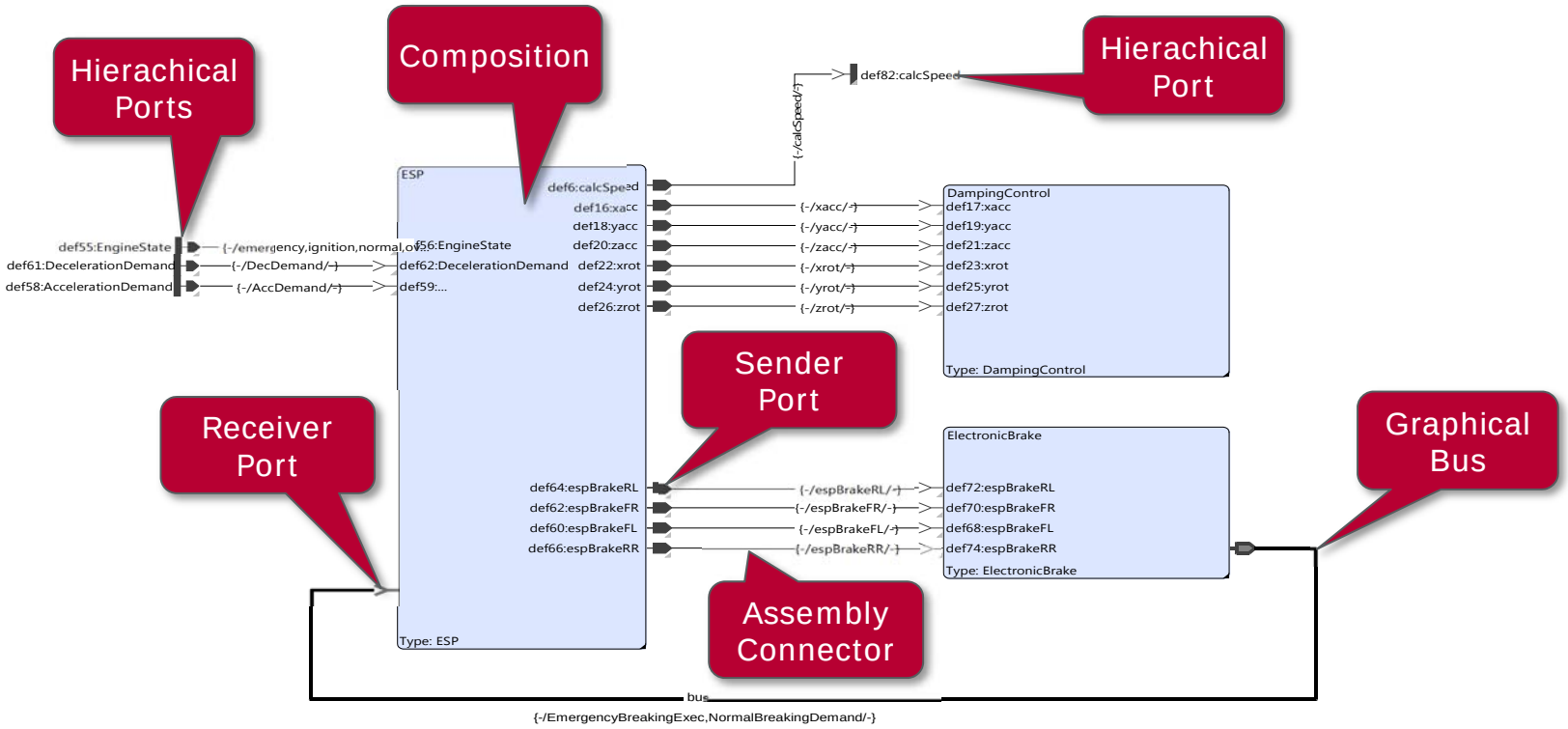
Function Oriented System Development



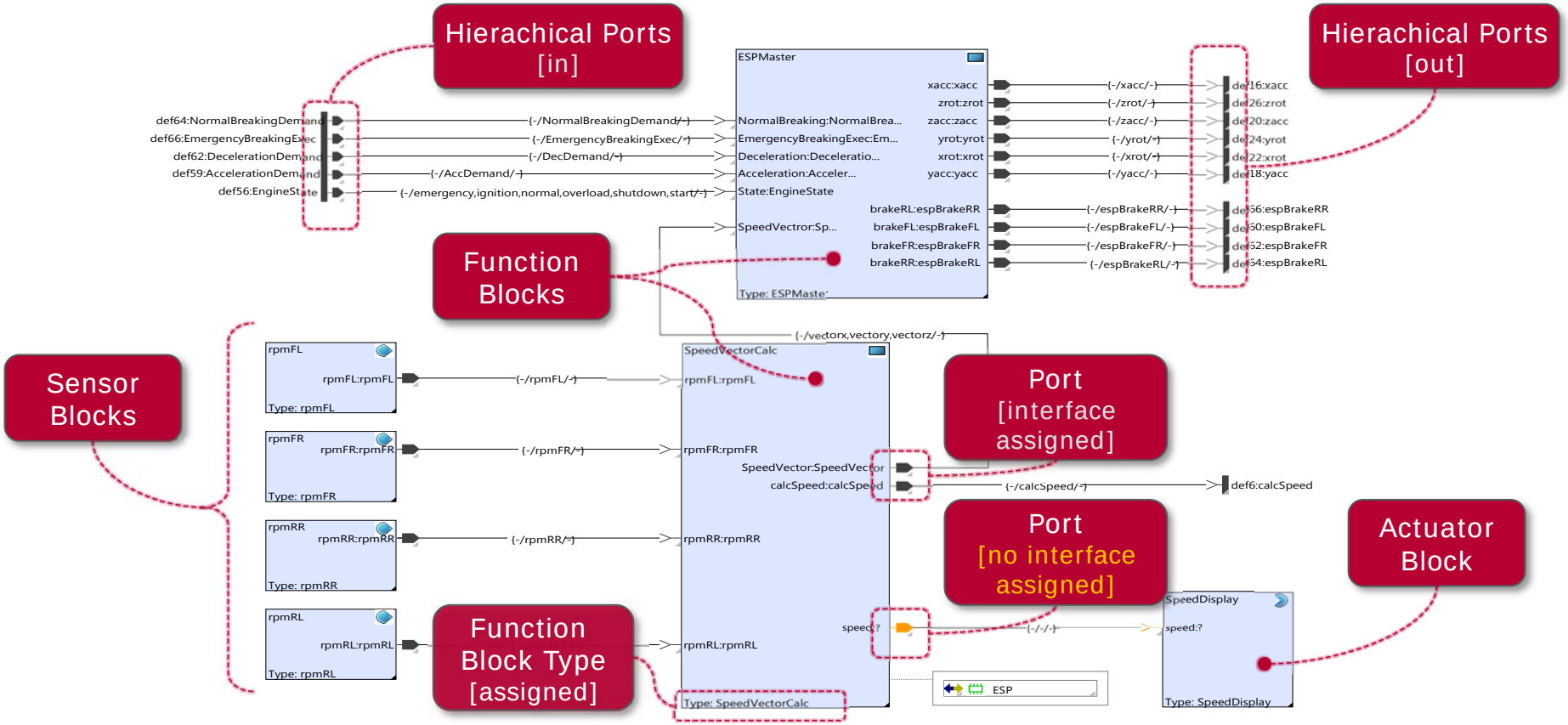
SW Architektur



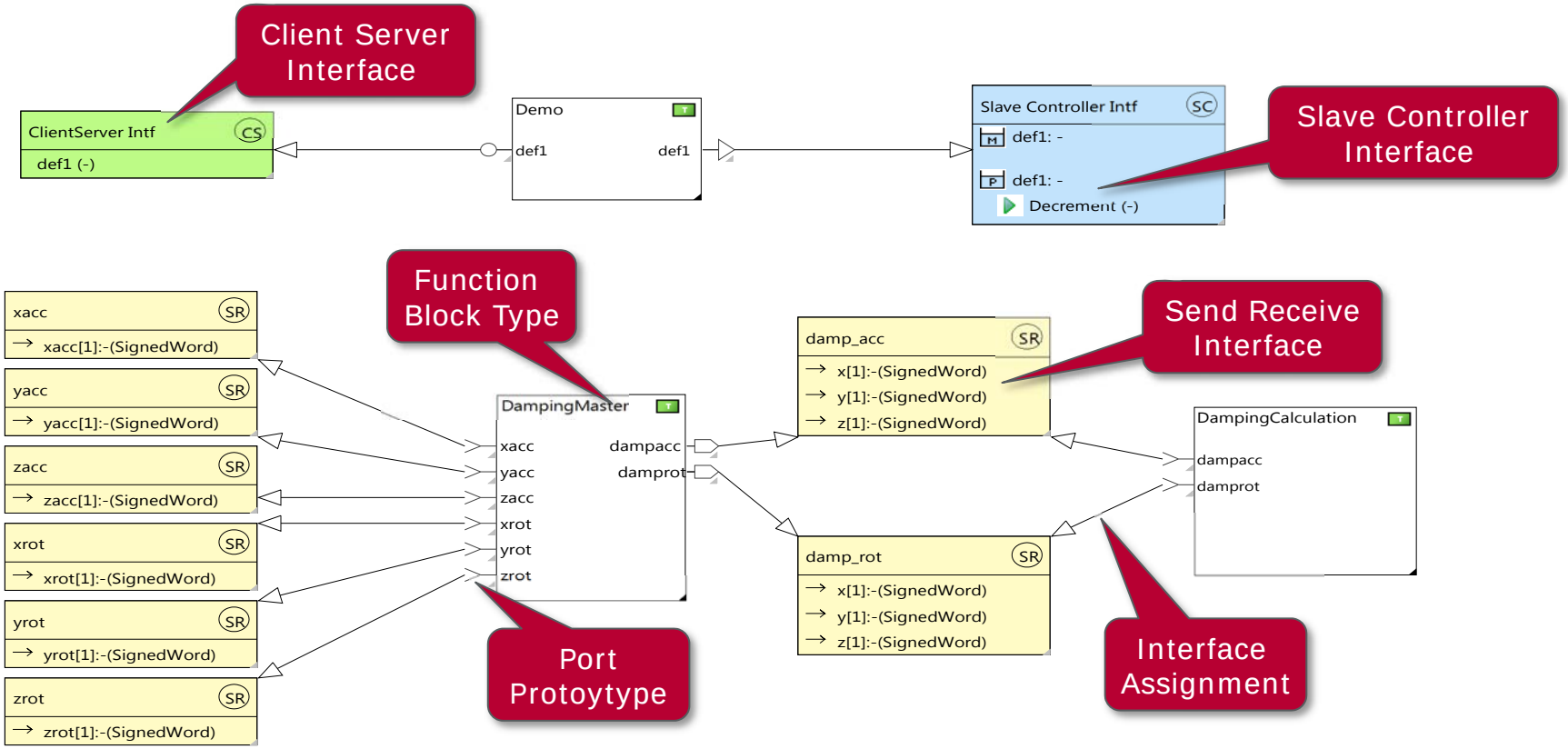
Ports and Connectors



Ports and Blocks



Interfaces


Client Server Interface
ClientServer Intf
CS
def1 (-)
Demo
def1
def1
Slave Controller Intf
SC
def1: -
def1: -
Decrement (-)
Slave Controller Interface
Function Block Type
xacc
SR
xacc[1]:-(SignedWord)
yacc
yacc[1]:-(SignedWord)
zacc
zacc[1]:-(SignedWord)
xrot
xrot[1]:-(SignedWord)
yrot
yrot[1]:-(SignedWord)
zrot
zrot[1]:-(SignedWord)
DampingMaster
xacc
yacc
zacc
xrot
yrot
zrot
dampacc
damprot
damp_acc
x[1]:-(SignedWord)
y[1]:-(SignedWord)
z[1]:-(SignedWord)
Send Receive Interface
damp_rot
x[1]:-(SignedWord)
y[1]:-(SignedWord)
z[1]:-(SignedWord)
DampingCalculation
dampacc
damprot
Port Protoytype
Interface Assignment

Requirement to Function

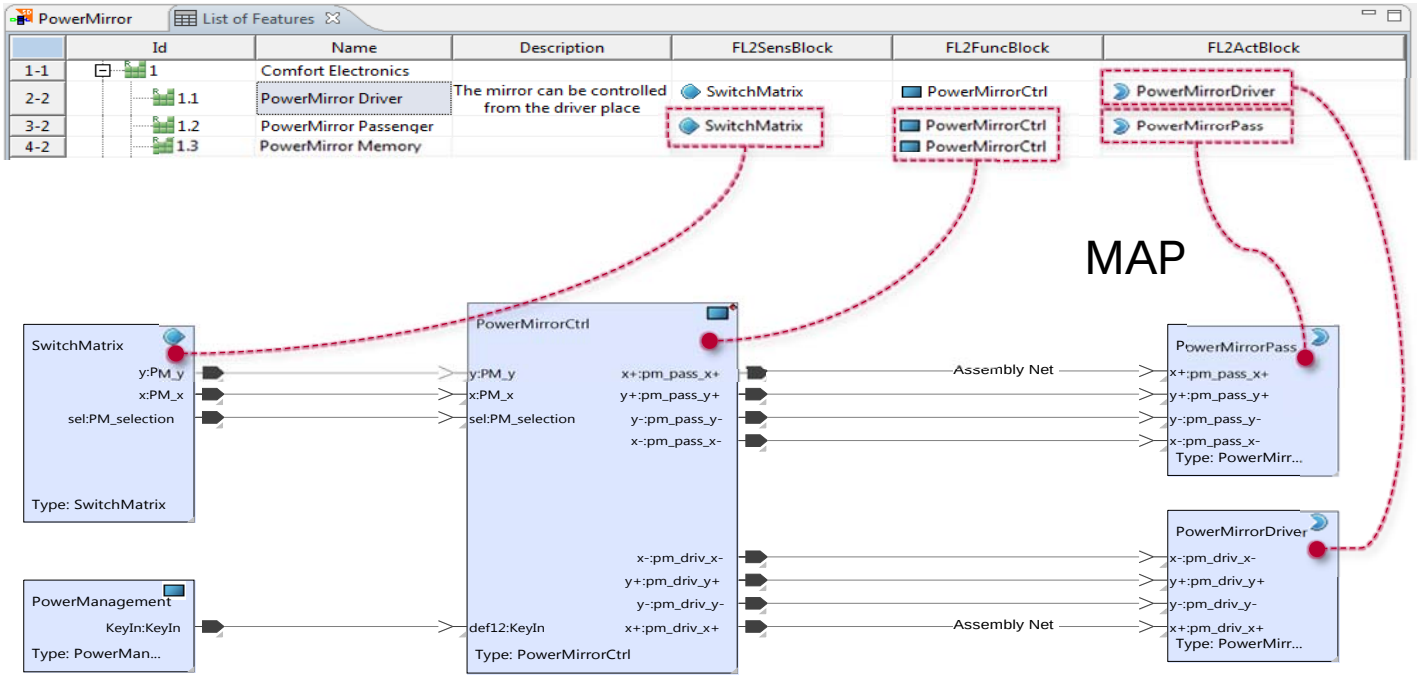
Requirements

Feature List

MAP

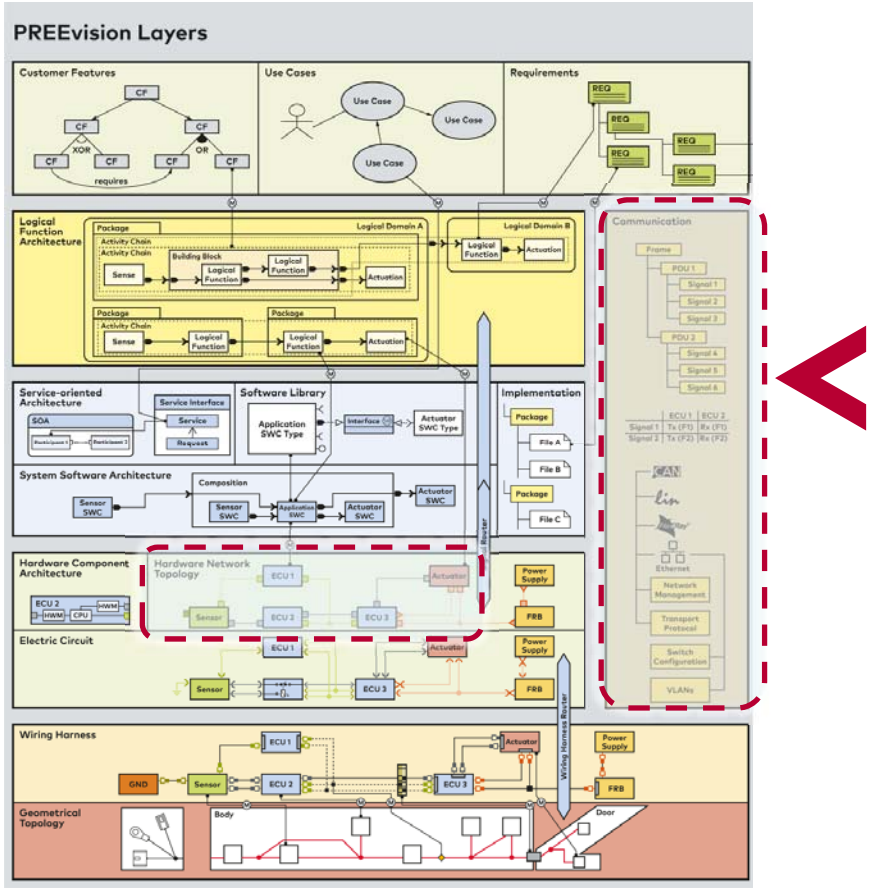
Function

Network



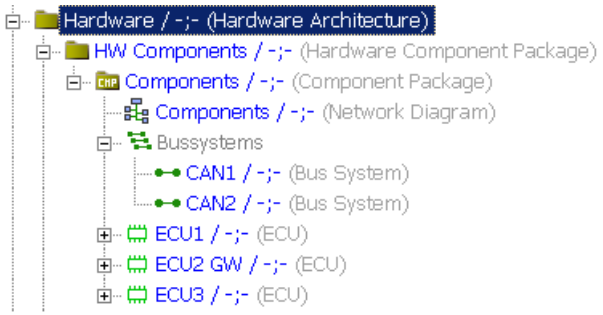
PREEvision Collaboration Platform (III)

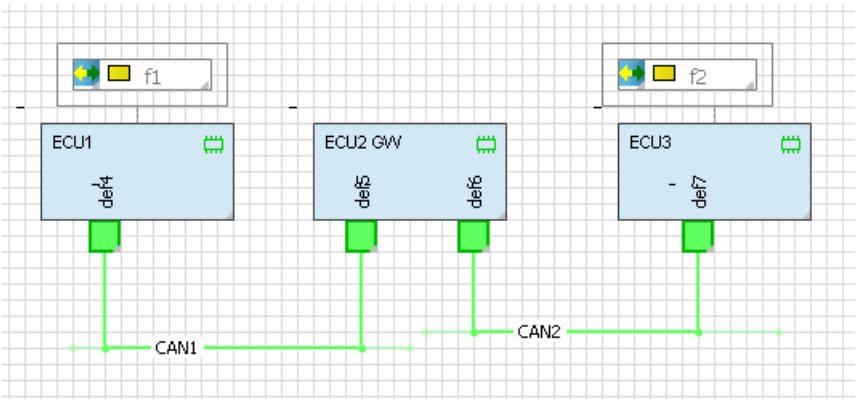
Vernetzungs-Architektur



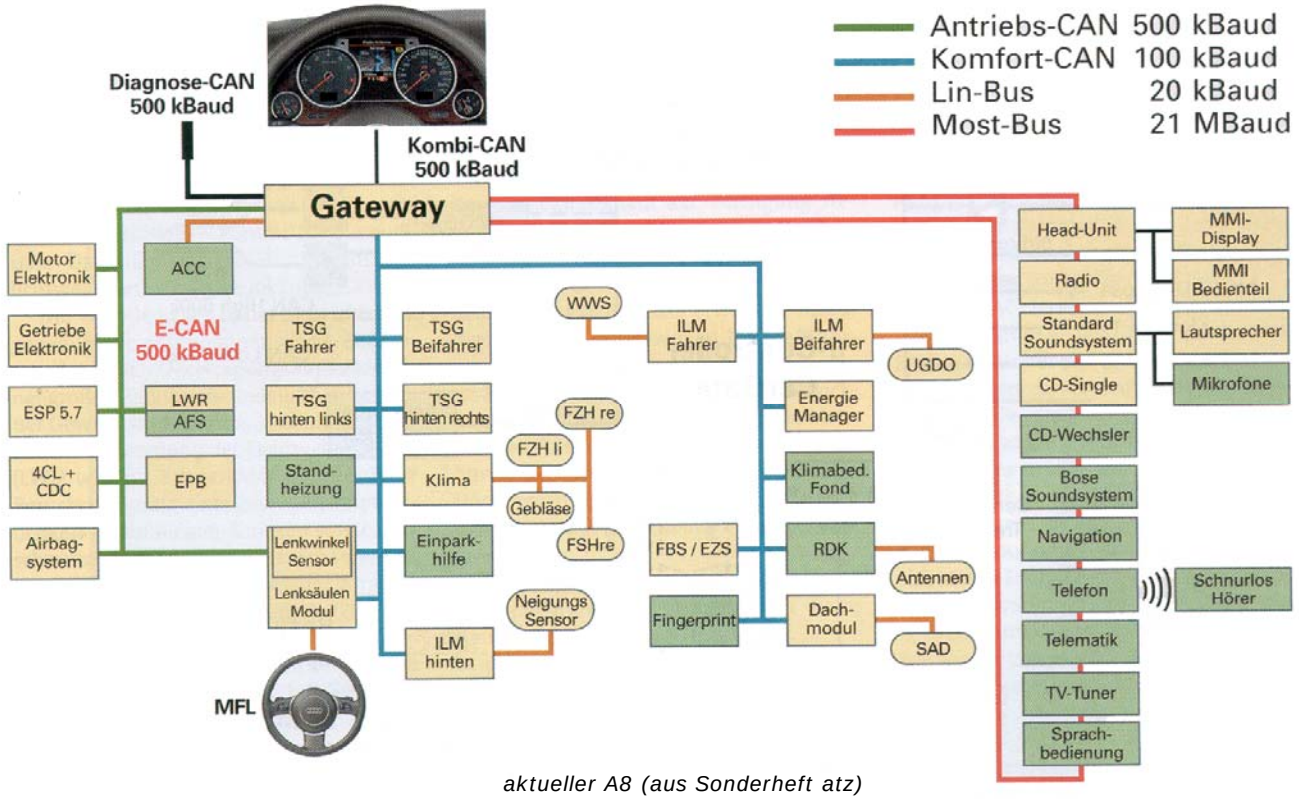
Hardware Architecture

- The Hardware Architecture specification is also supported by graphical block diagrams.
- The mapping of logical functions can be displayed directly in the graphics, shown over the ECU block



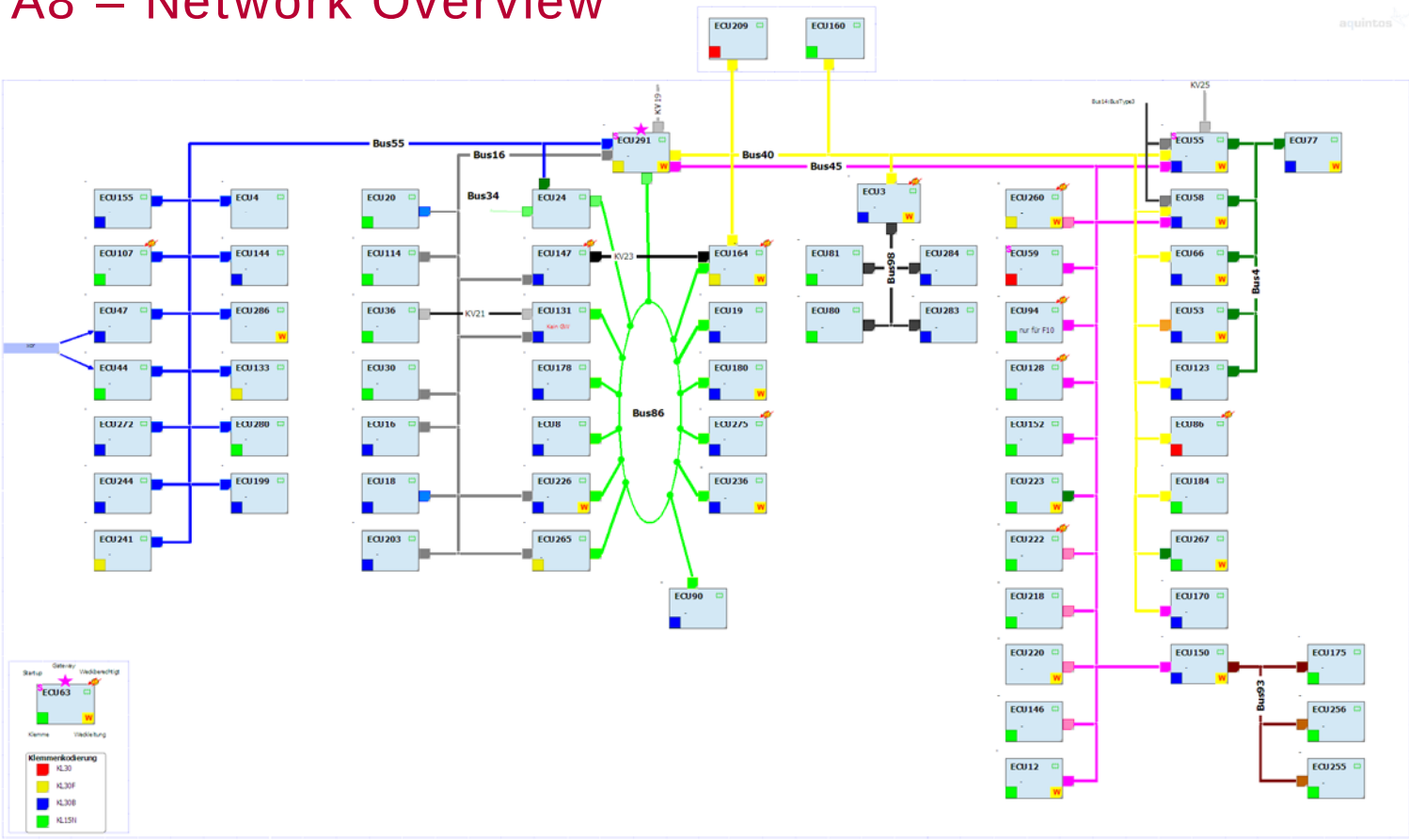


Audi A8

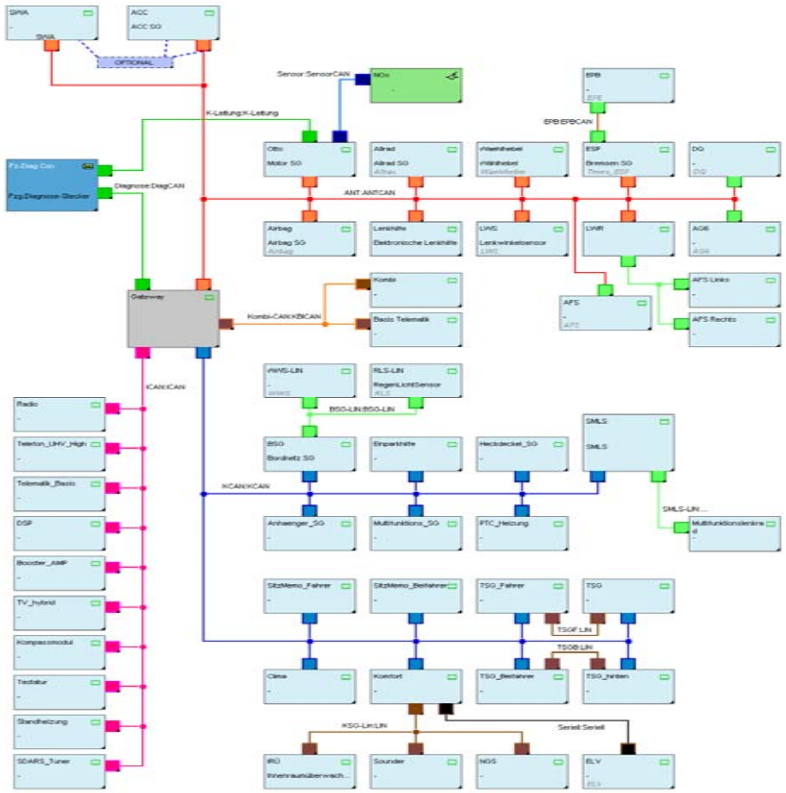


aktueller A8 (aus Sonderheft atz)

Audi A8 – Network Overview



Domänen



E/E-Architekturen im KFZ werden meistens in die folgenden Domänen unterteilt:

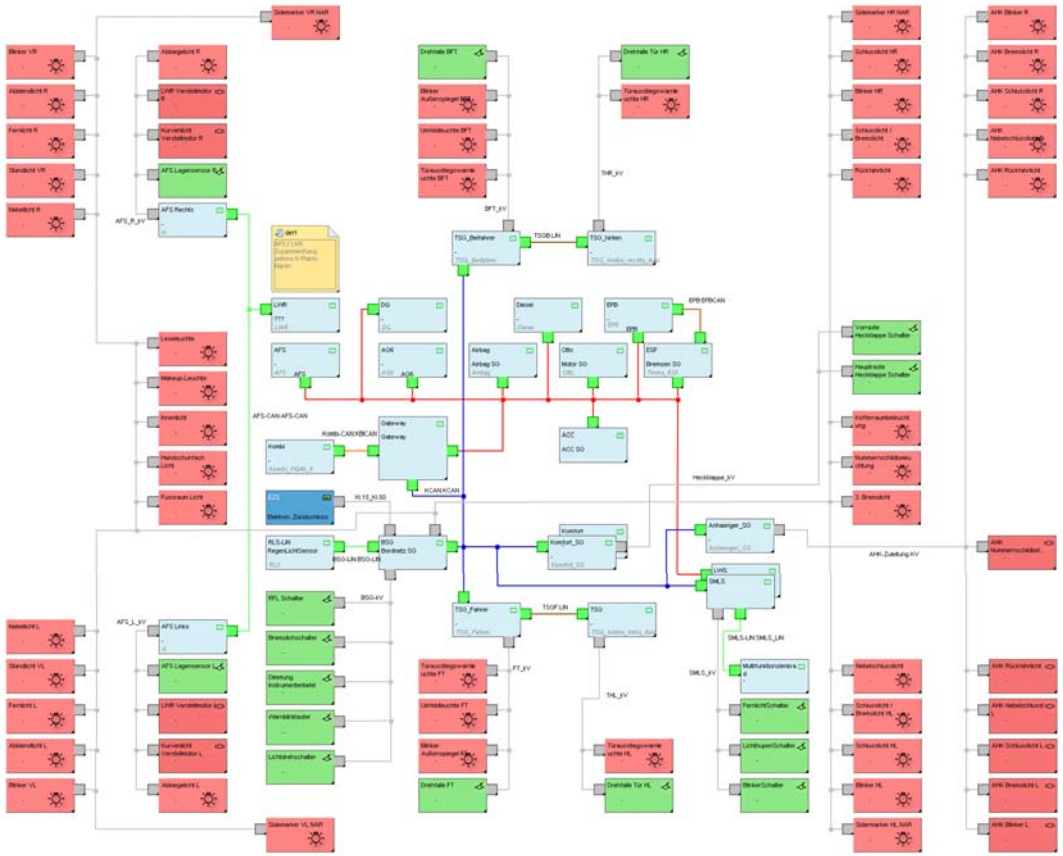
- Antriebsstrang
- Komfort (Innenraum)
- Chassis
- Telematik oder Infotainment

Das E/E-System eines Fahrzeugs ist stark verteilt. Die Kommunikation der Systeme erfolgt über standardisierte Bussysteme (CAN, LIN, MOST, FlexRay).

Der modellbasierte Architekturentwurf unterstützt den Entwurfsprozess stark verteilter Elektroniksysteme und erlaubt eine Optimierung z.B. nach Kosten, Gewicht, Bauraumbedarf etc.

Typisches Vernetzungskonzept PKW

Aufbau eines Lichtsystems



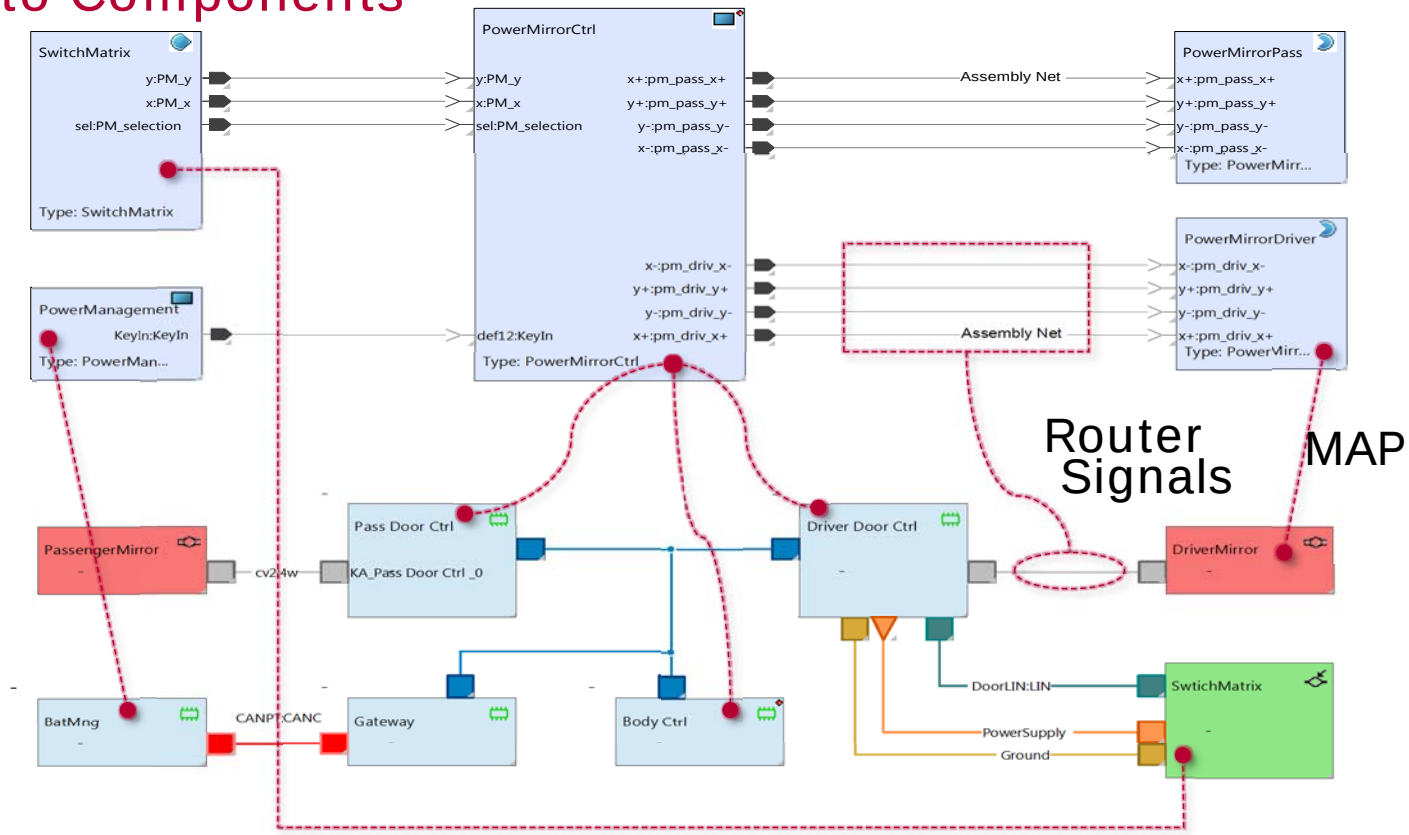
Function Net to Components

Function Network

ROUTE

MAP

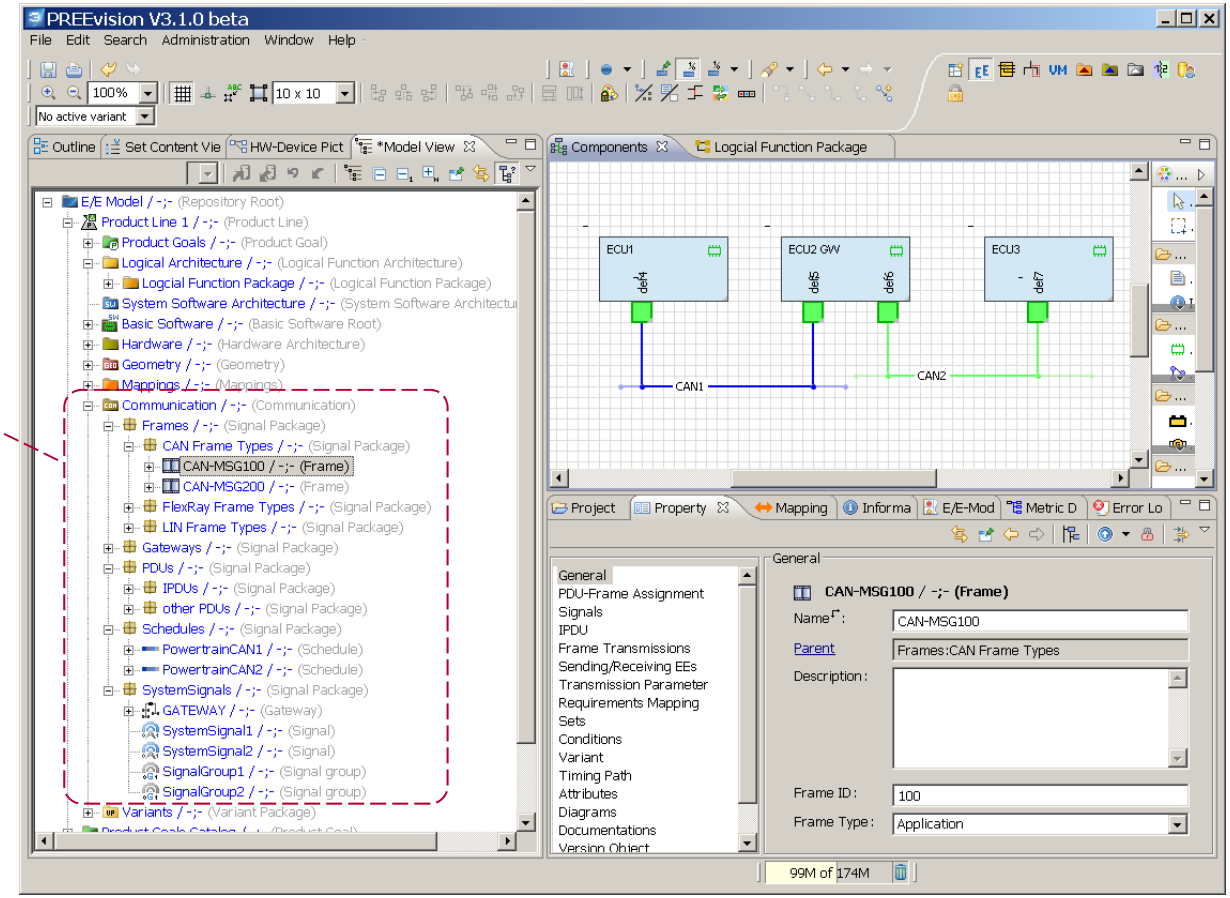
Component Network



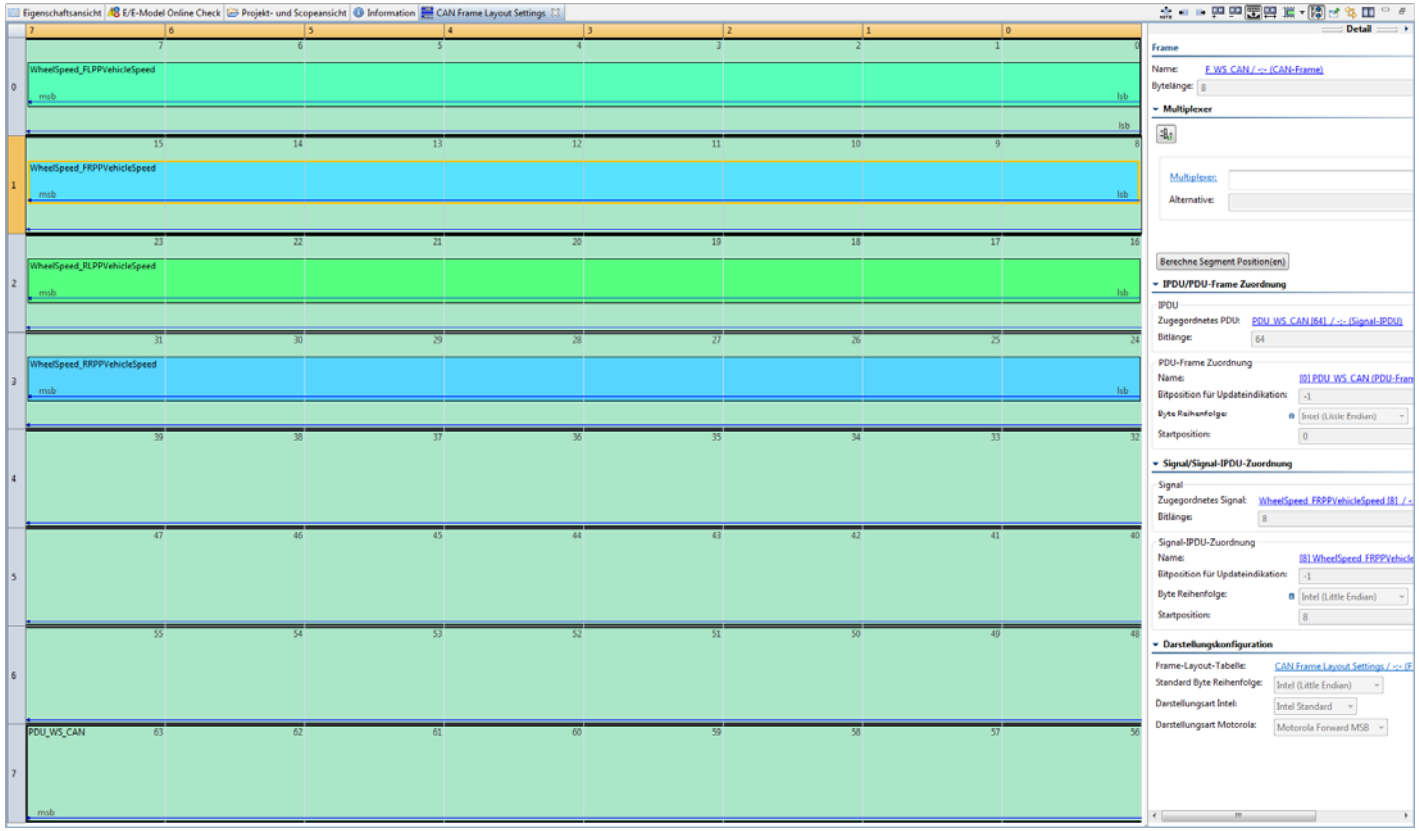
Communication Layer in PREEvision

Communication Layer to specify

- Signals
- PDUs
- Frames
- Schedules

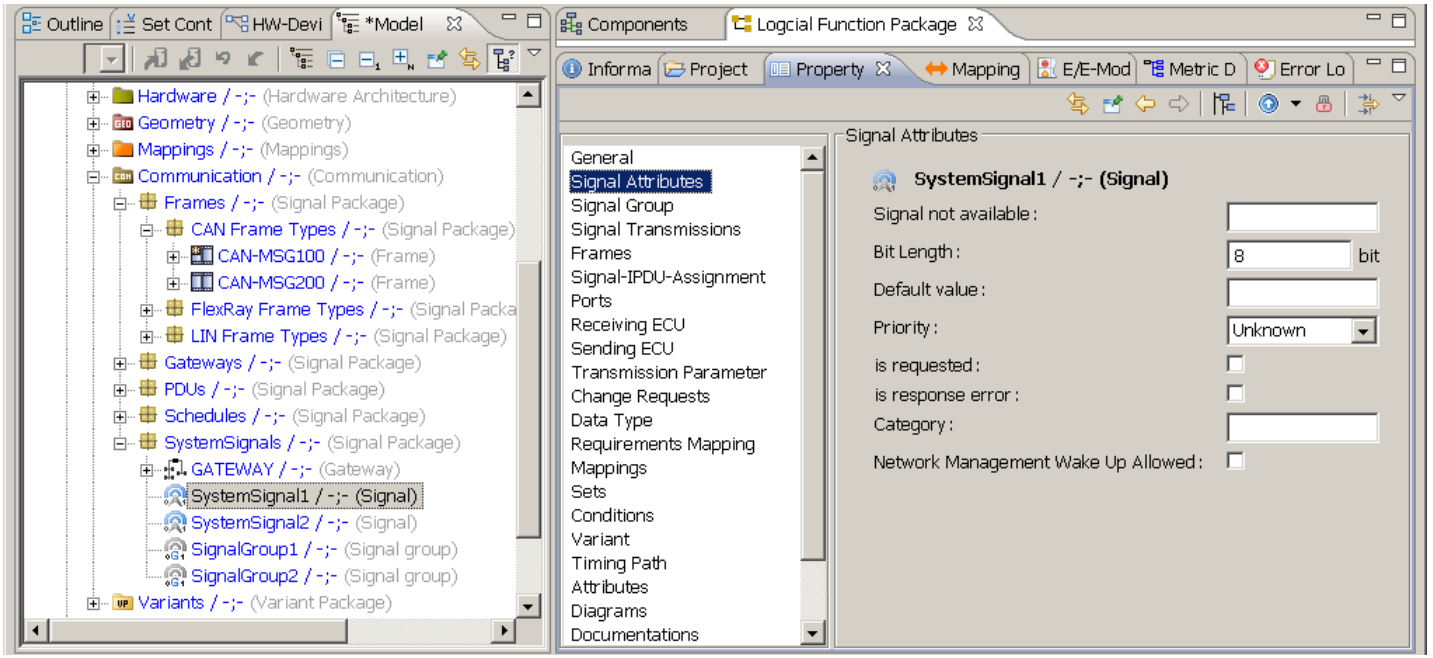


Frame Layout Design



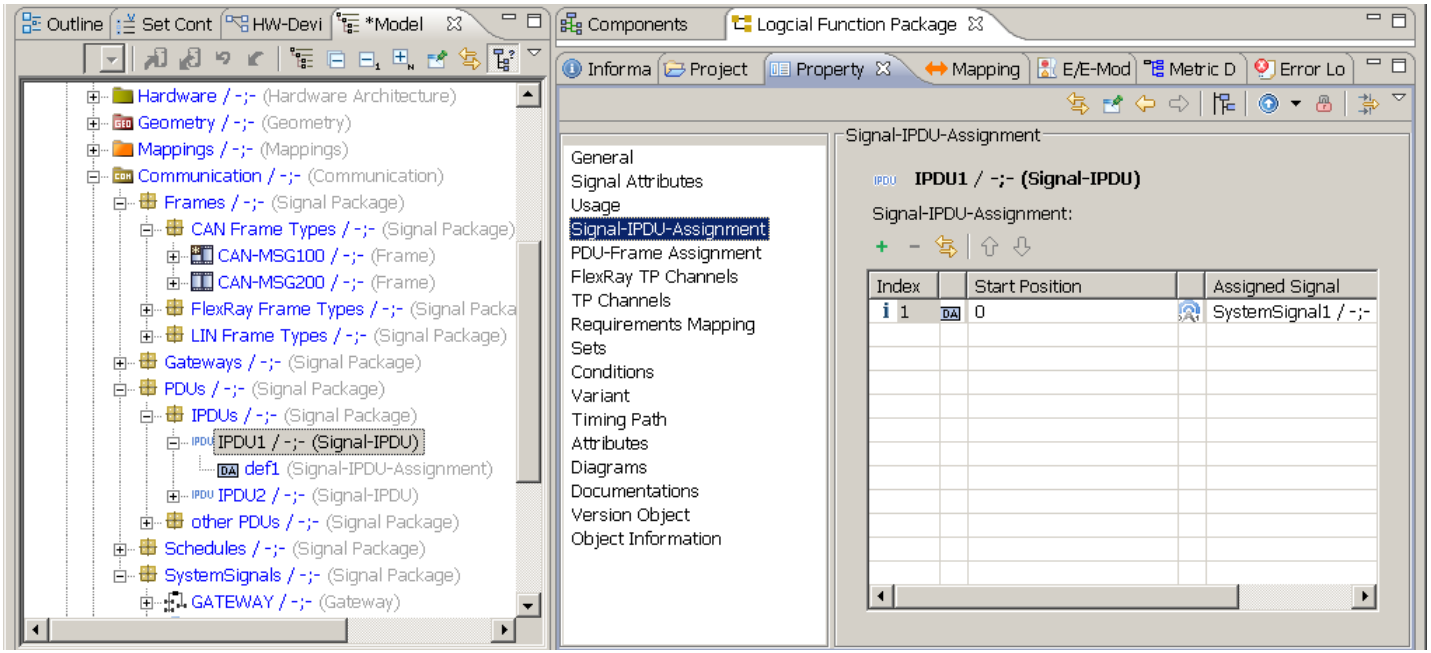
System Signal Specification

The specification of System Signals can be done by the Property Editor...



PDU Specification

PDUs are specified interactively based on System Signals

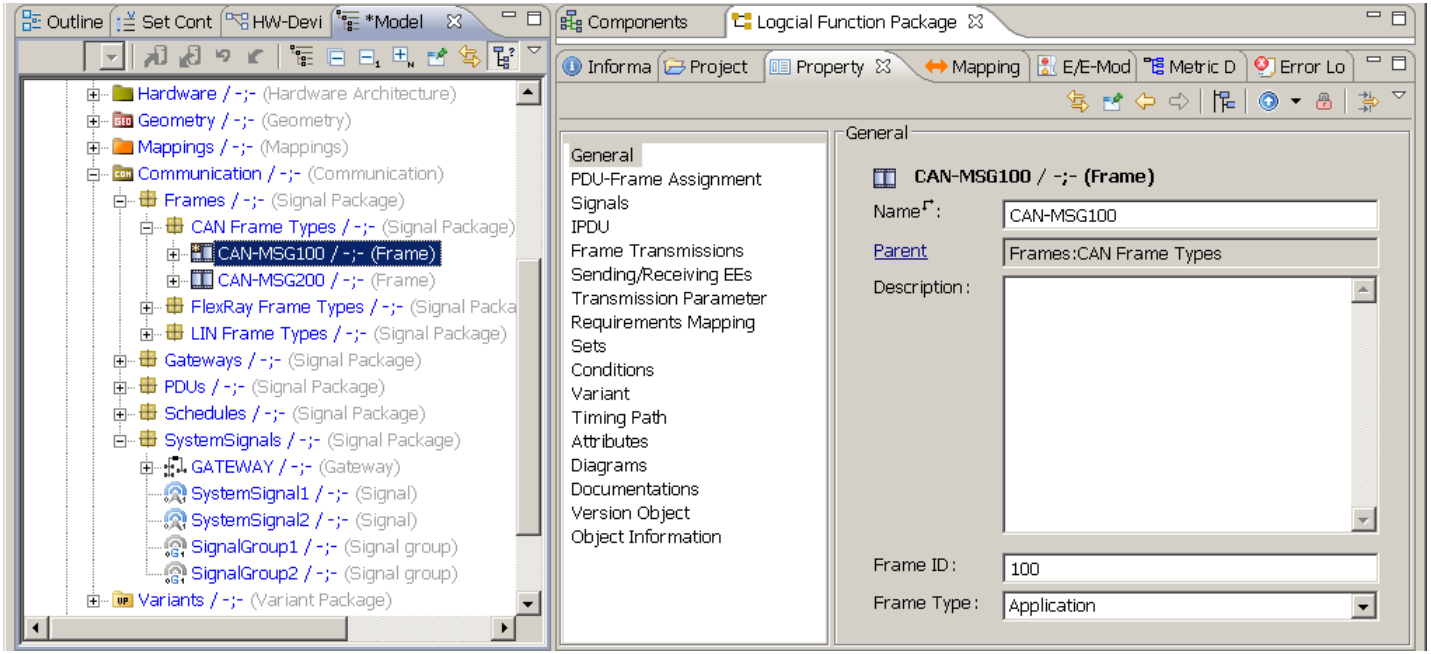


PDU: Protocol-Data-Unit

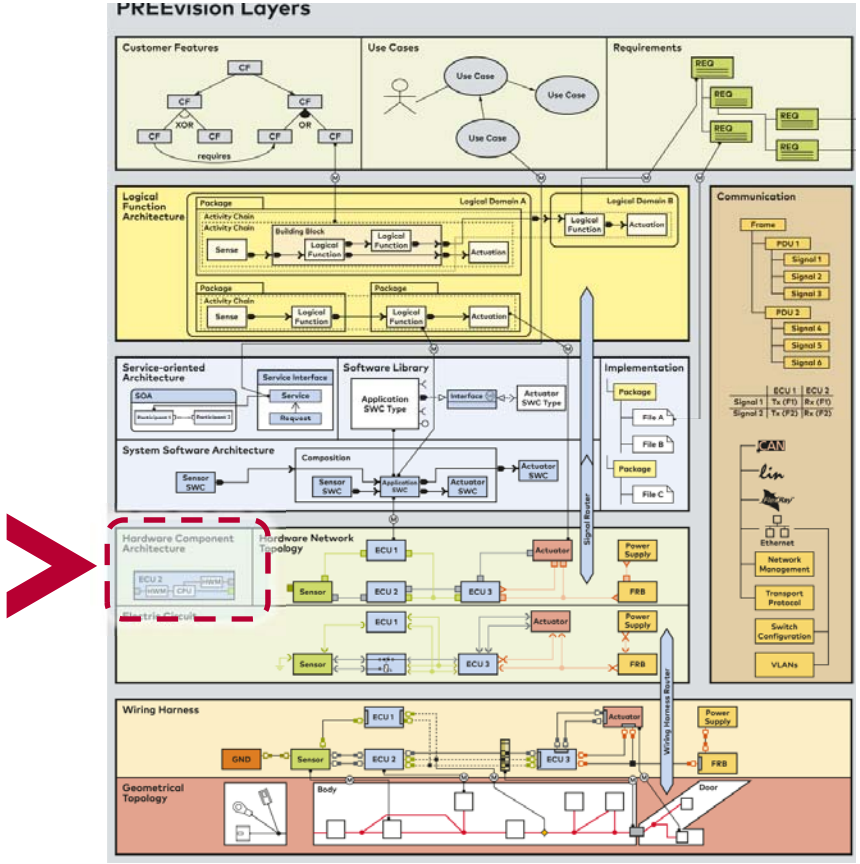
Frame Specification

Frames are specified interactively based on PDUs and are available for

- CAN
- LIN
- FlexRay



Bauteil/Steuergerät



Aufbau einer ECU

- CPU, FPGA, RAM, etc.
- Gateway-Struktur

Kommunikation zwischen Busanbindung und CPU

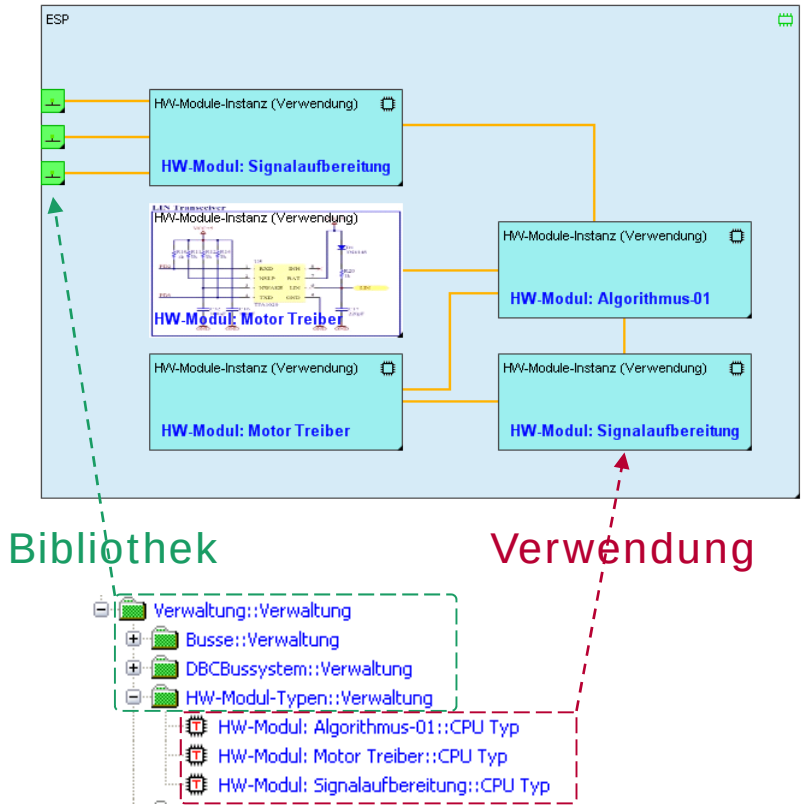
ECU: Electronic Control Unit

Aus Zulieferer-Sicht

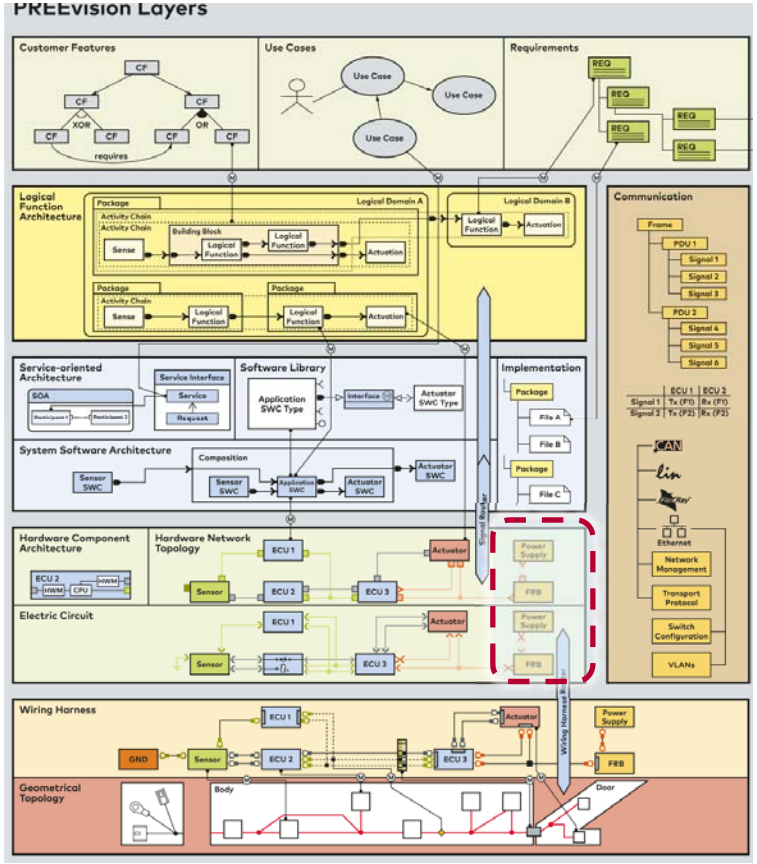
(a) zur Verwendung | (b) als Bibliothek

Attribute eines HW Moduls

- Stückliste
  - Bauteilbezeichner
  - Teilenummer
  - Gehäuse
  - Fläche
  - Anzahl
  - Einzelkosten (aus DB) Import
  - Bild
- Kumulierte Attribute
  - Kosten
  - Fläche (aus Bauform, Overhead Anteil, Offset)
  - Gewicht



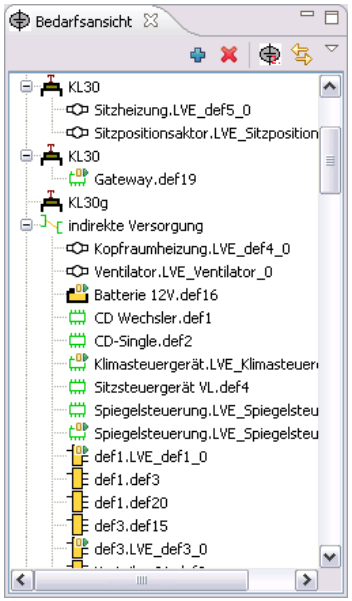
Leitstungsversorgungs-Architektur

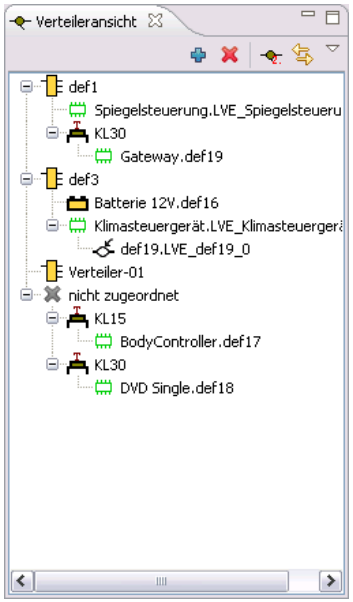


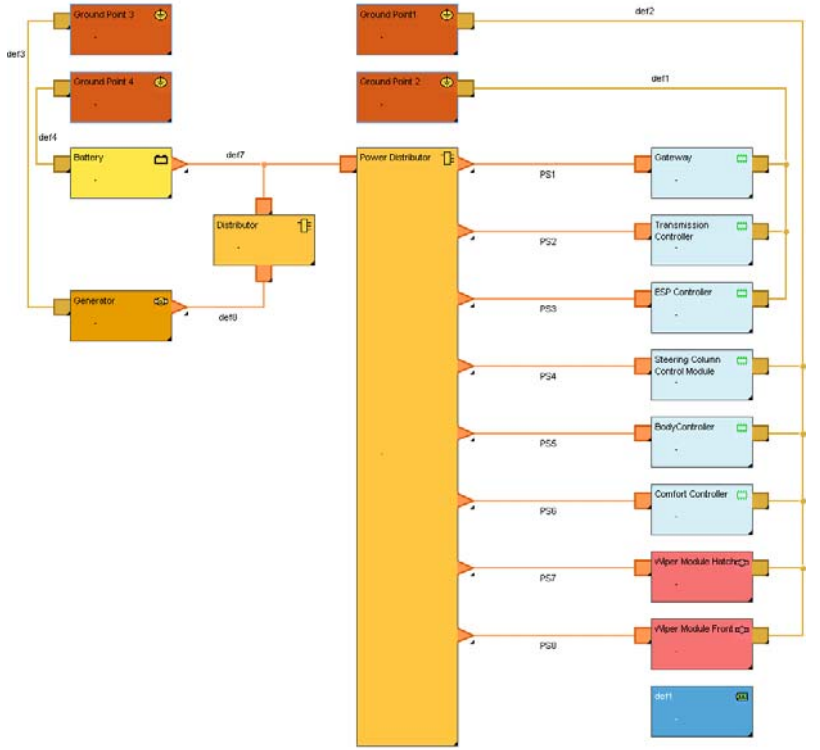
Stromverteilungsmodellierung

Schnelle Modellerzeugung durch zwei Ansichten:

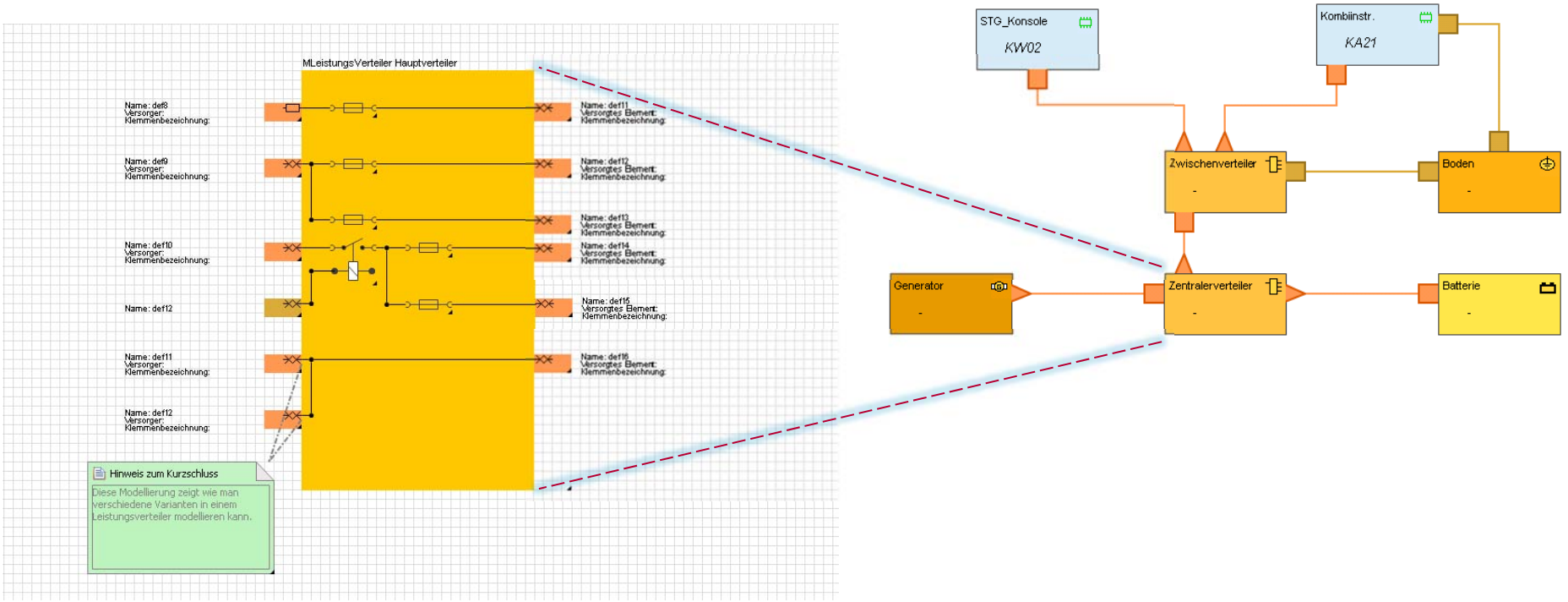
- Bedarfsansicht (Klemmenbedarf)
- Verteilungsansicht





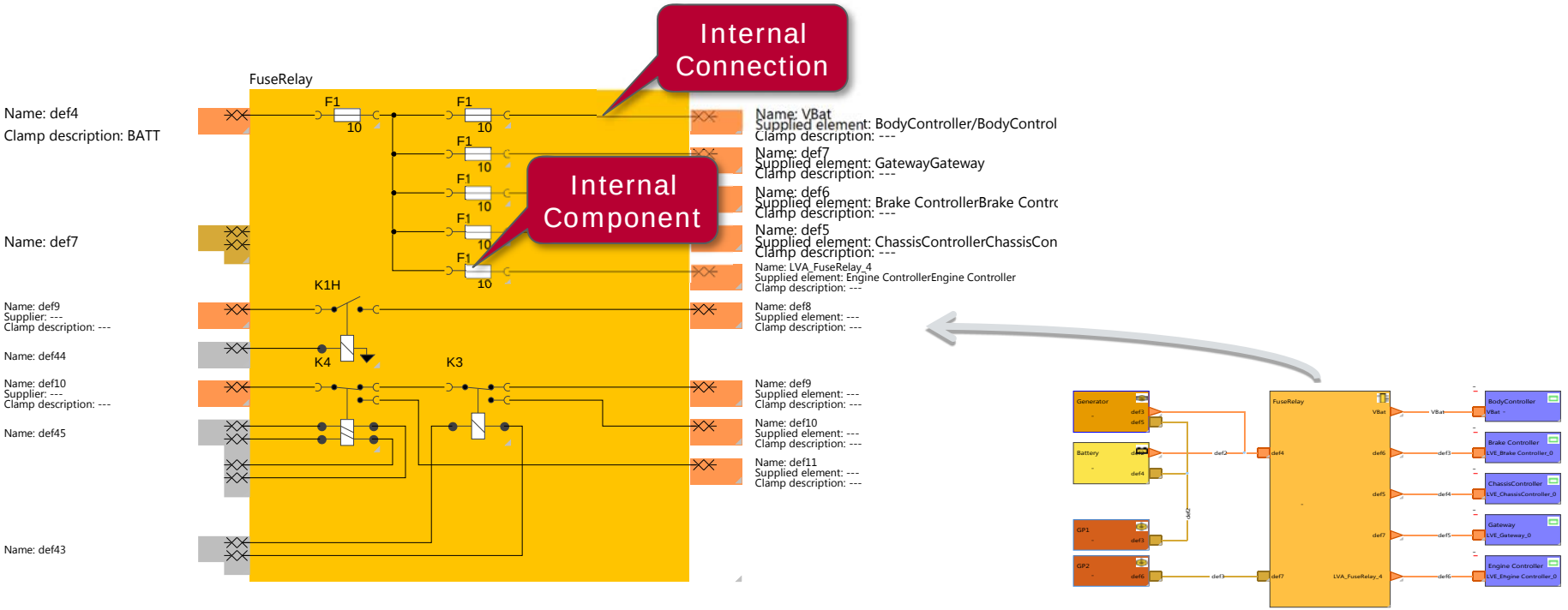


Spezifikation von Sicherungen und Relais

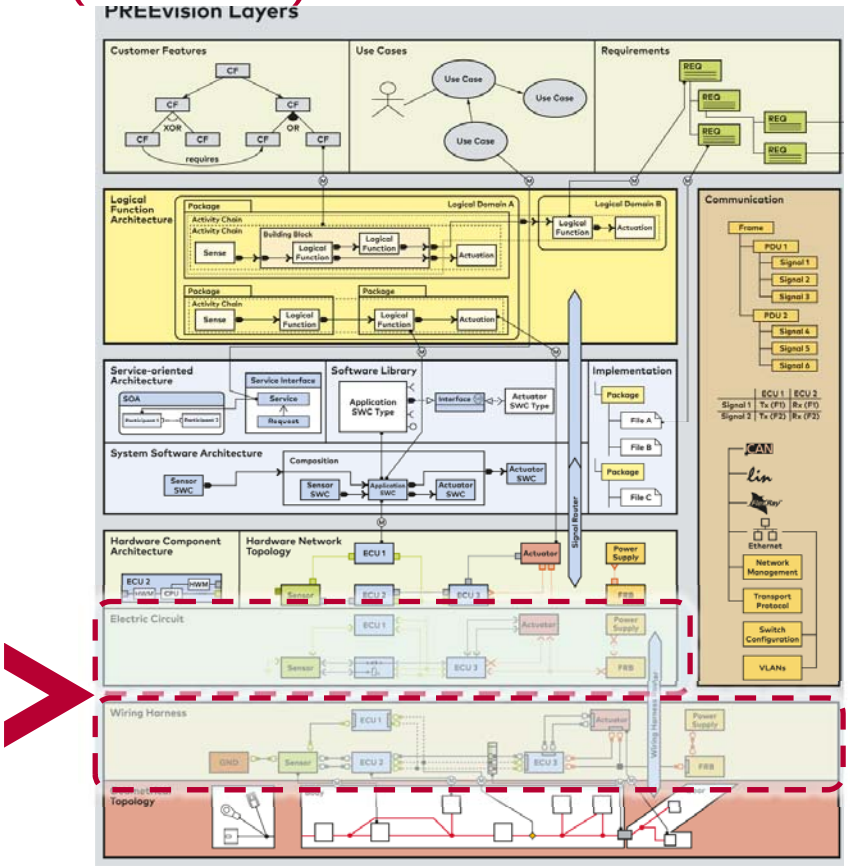


Internal Schematic Diagram is auto-layouted

Quick Conceptualization of e.g. Fuse-Relay Systems



Stromlaufplan Leitungssatz (Elektrik)

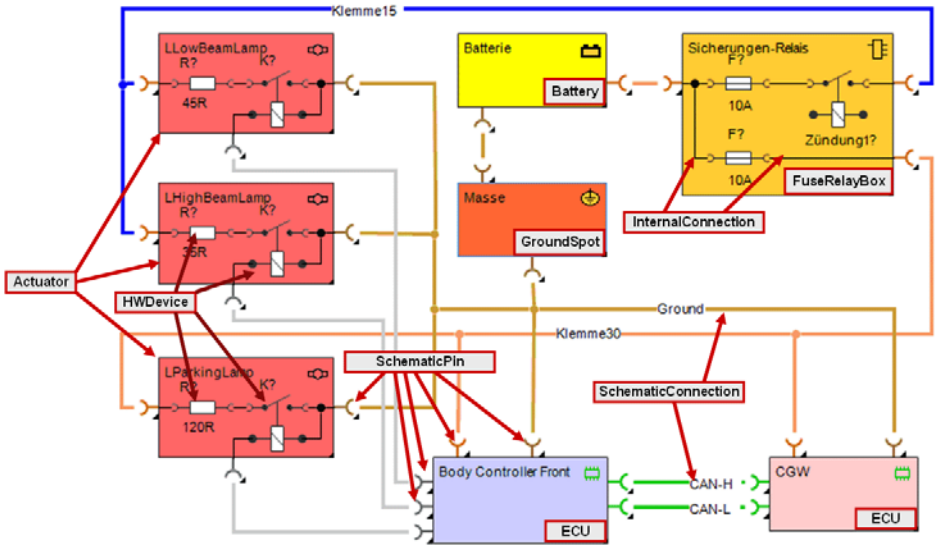


Darstellung der elektrischen Verbindungen zwischen Komponenten

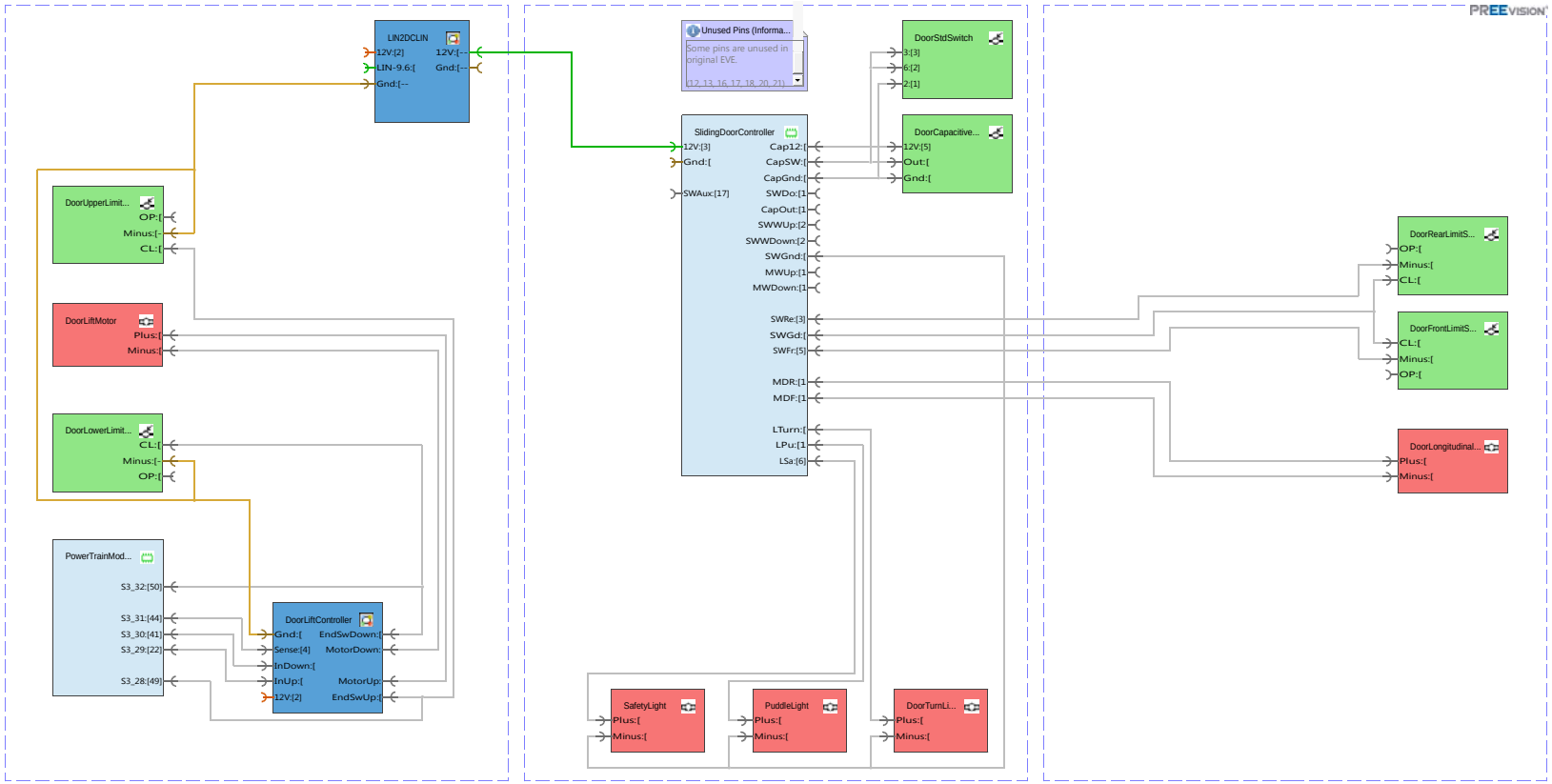
- Keine Trennstellen
- Keine Splices
- Keine Stecker
- Vereinfachte Darstellung von Pins

Komponenten können mit „Ersatzschaltbild“ hinterlegt werden

- ✓ Leichteres Verständnis der elektrischen Zusammenhänge
- ✓ Automatisierte Generierung von Werkstattmaterial



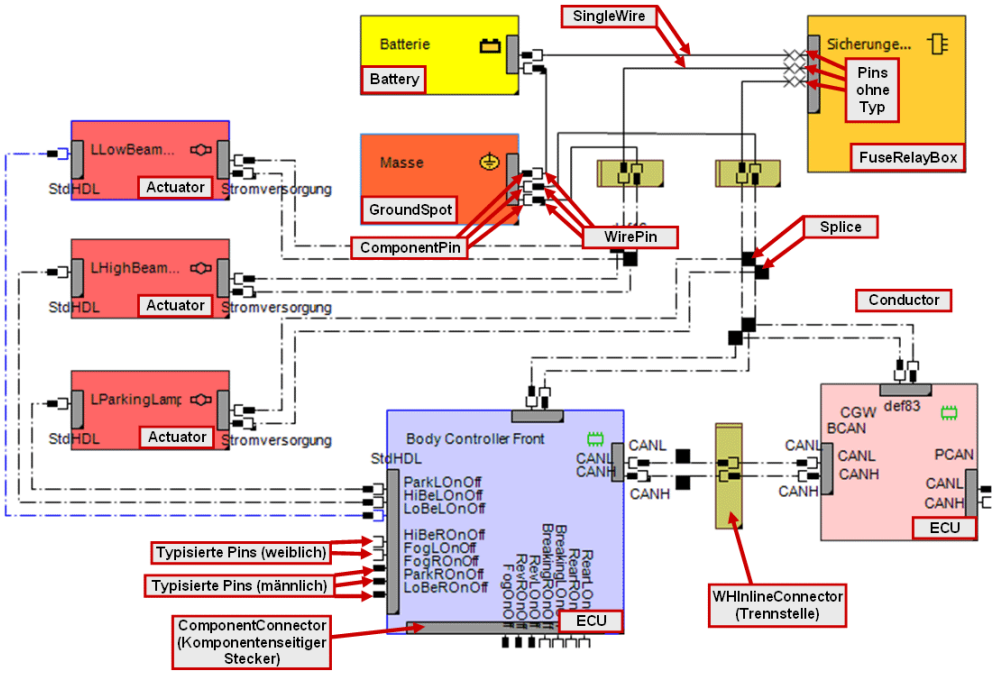
Beispiel Stromlaufplan



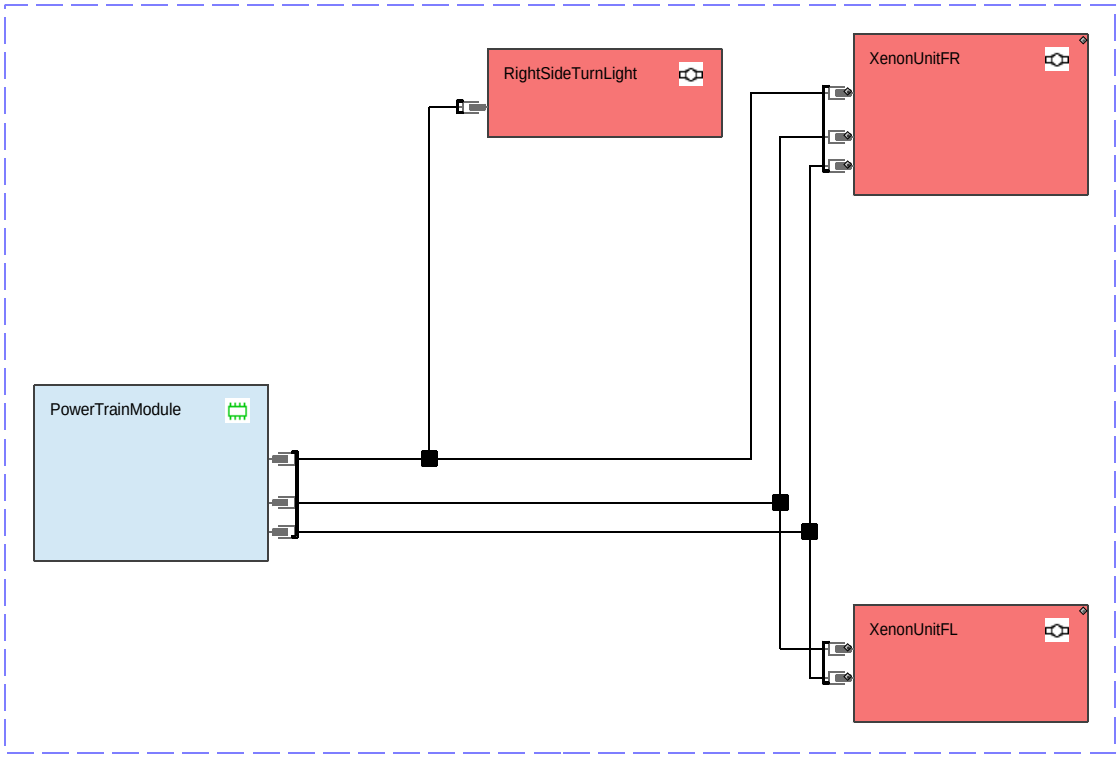
Darstellung aller Leitungssatzelemente

- Trennstellen
- Splices
- Detaillierte Pins
- Stecker-Partitionierung

✓ Dokumentation des Leitungssatzes



Beispiel Leitungssatzdiagramm

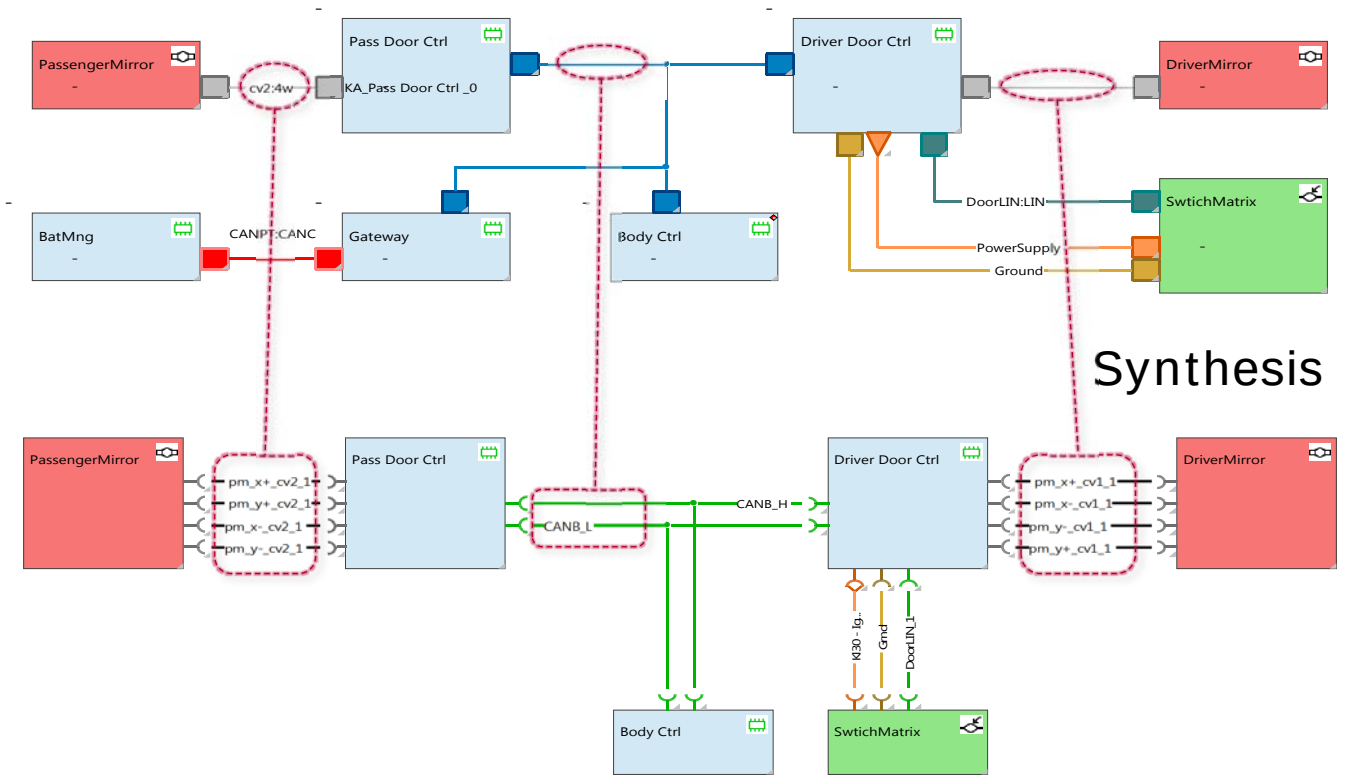


Components & Circuits

Component Network

Synthesis

Circuits

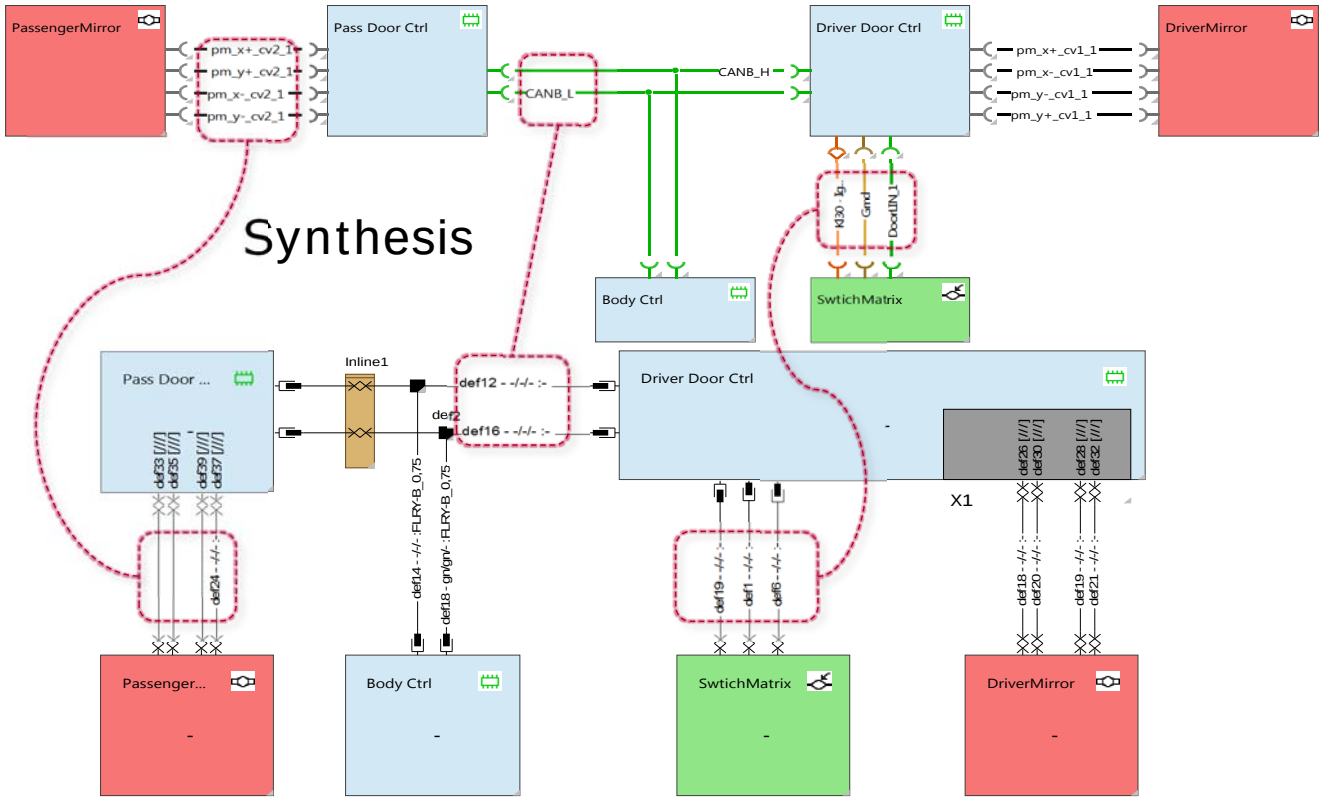


Components & Wiring Harness

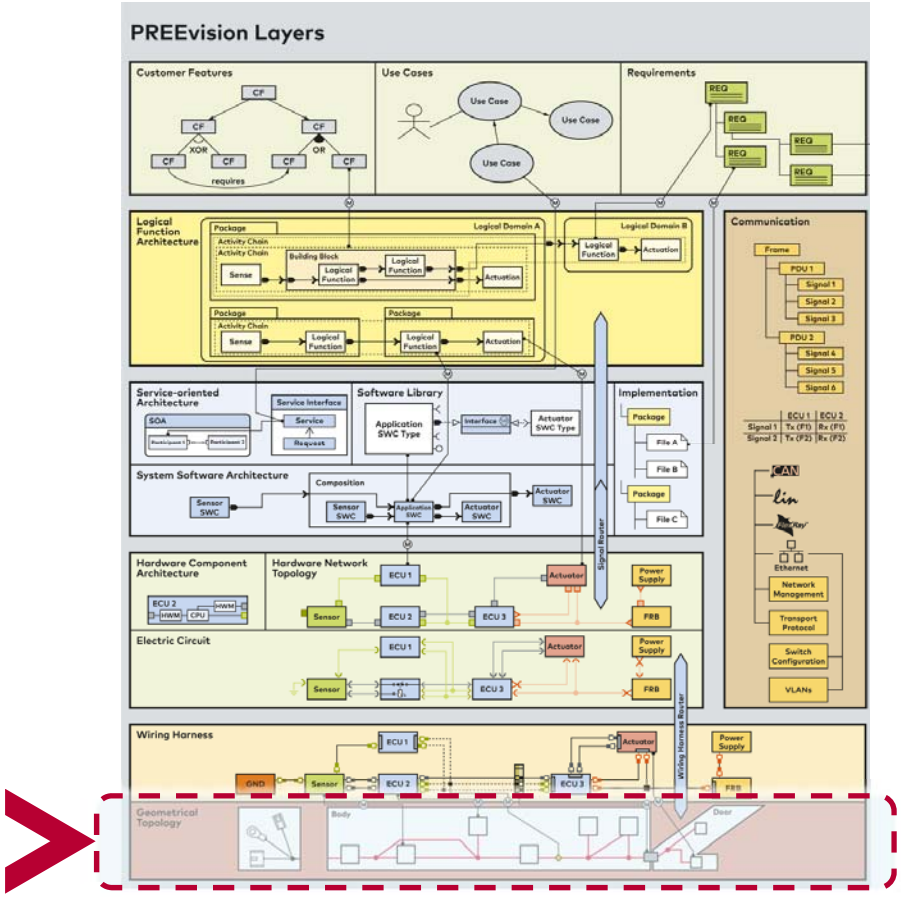
Circuits

Synthesis

Wiring Harness



Geometrie

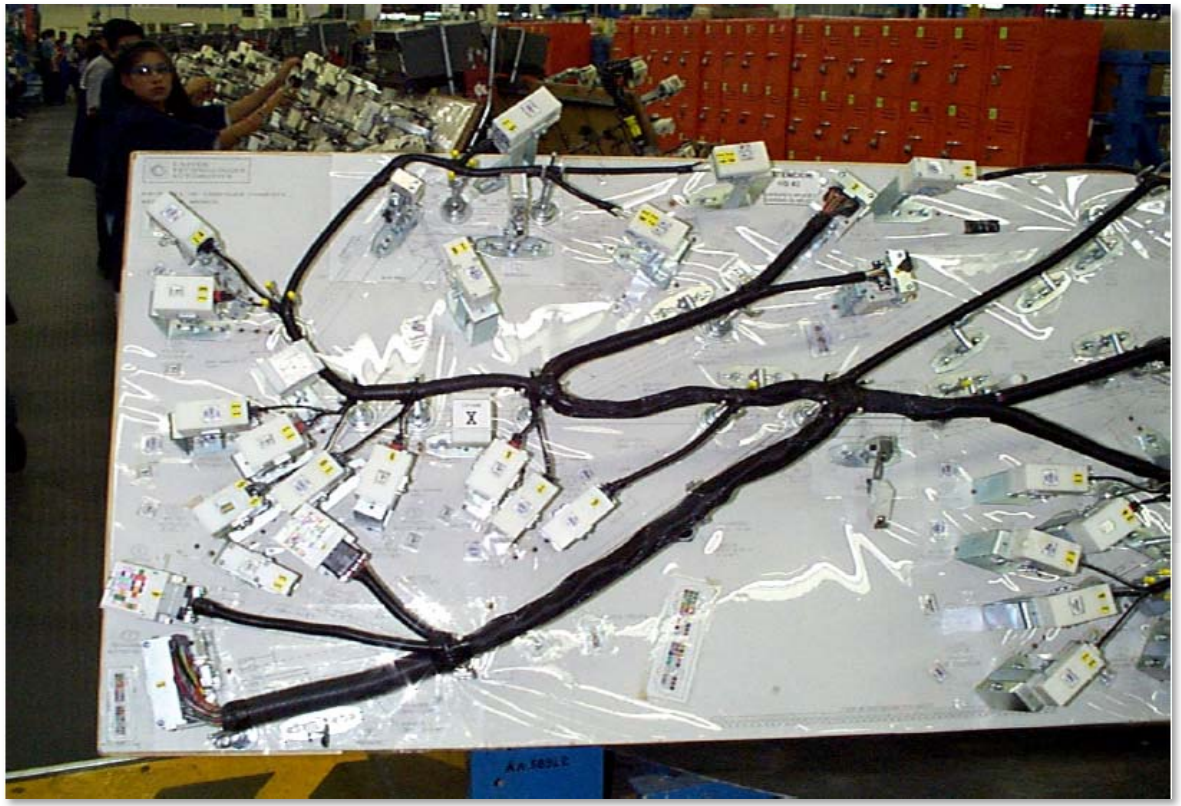

PREEvision Layers
Customer Features
Use Cases
Requirements
Logical Function Architecture
Communication
Service-oriented Architecture
Software Library
Implementation
System Software Architecture
Hardware Component Architecture
Hardware Network Topology
Electric Circuit
Wiring Harness
Geometrical Topology

Geometrie

Zentralleitungssatz eines PKWs

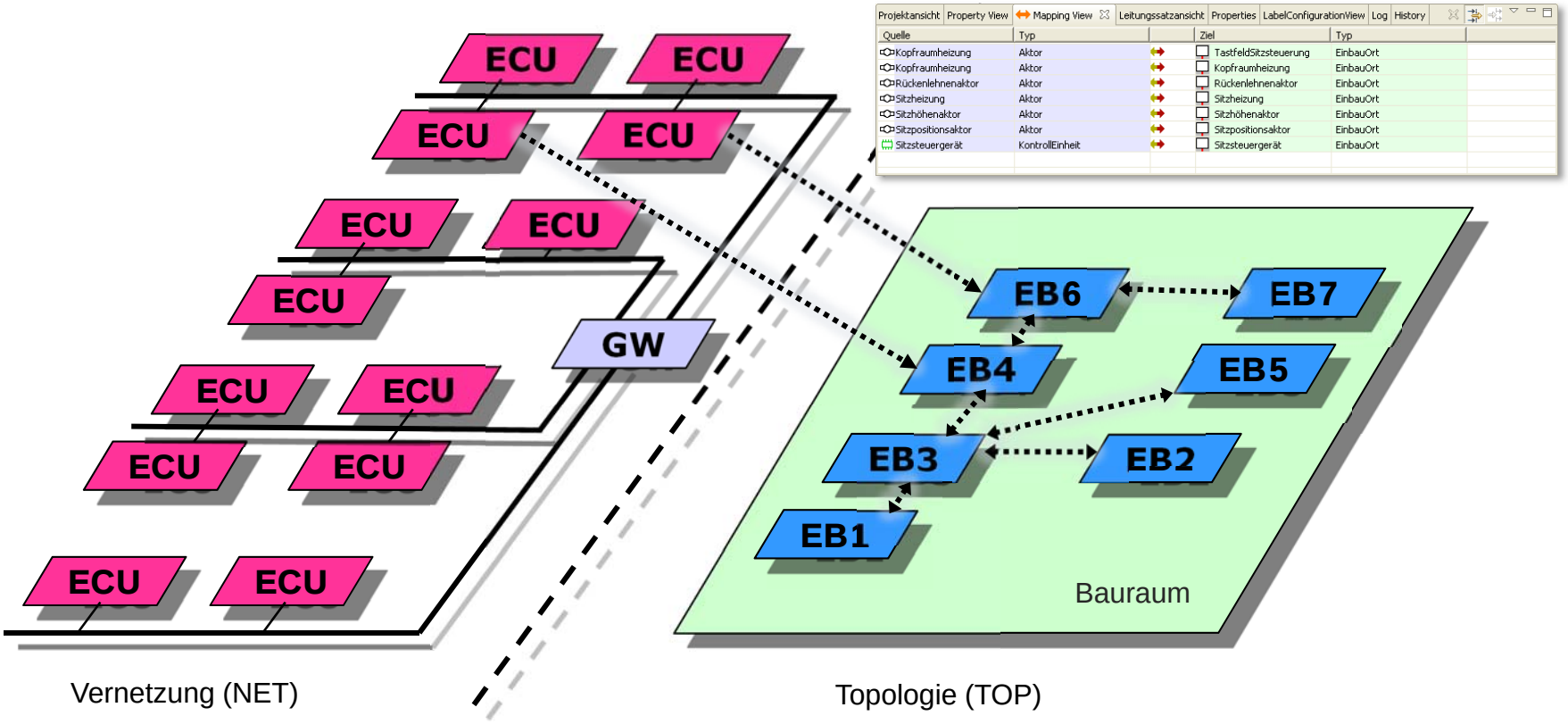
Ca. 2500 verlegte Leitungen mit Querschnitten zwischen 1qmm und 16qmm Kupfer

Beispeil Leitungssatz mit Steuergeräten

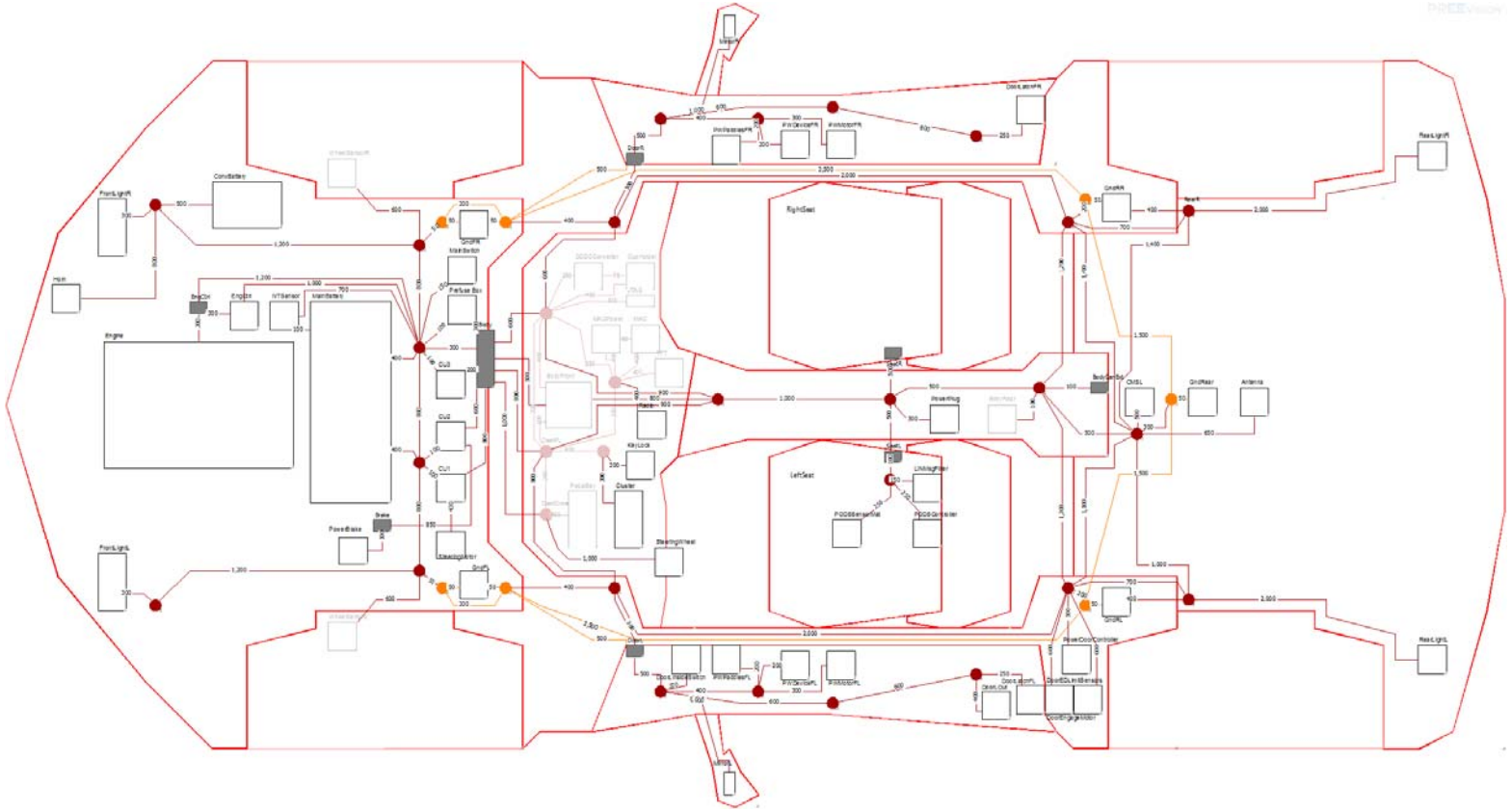


Beispiel Leitungssatz

Mapping der Steuergeräte auf Topologie-Ebene

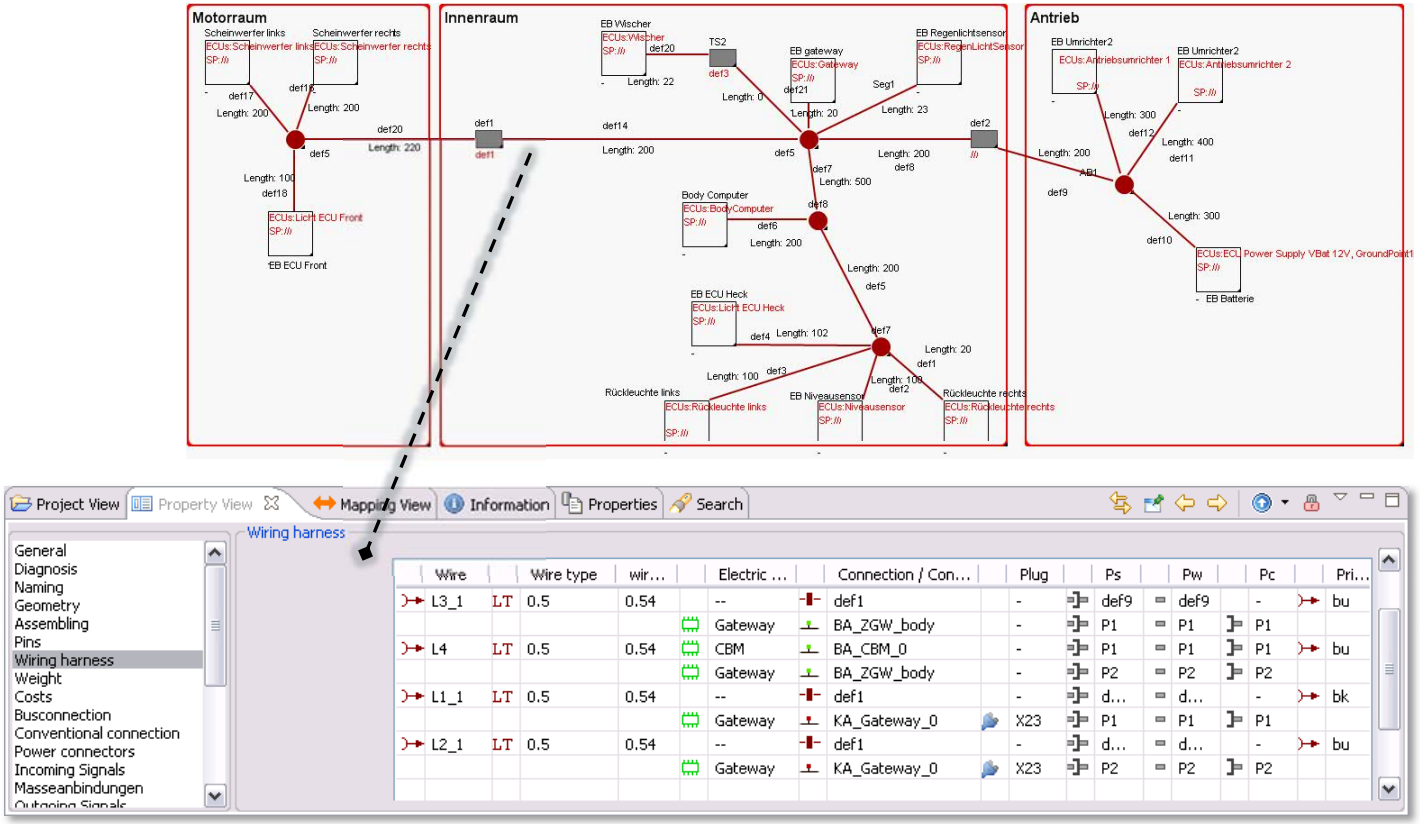


Topologie des Fahrzeugs

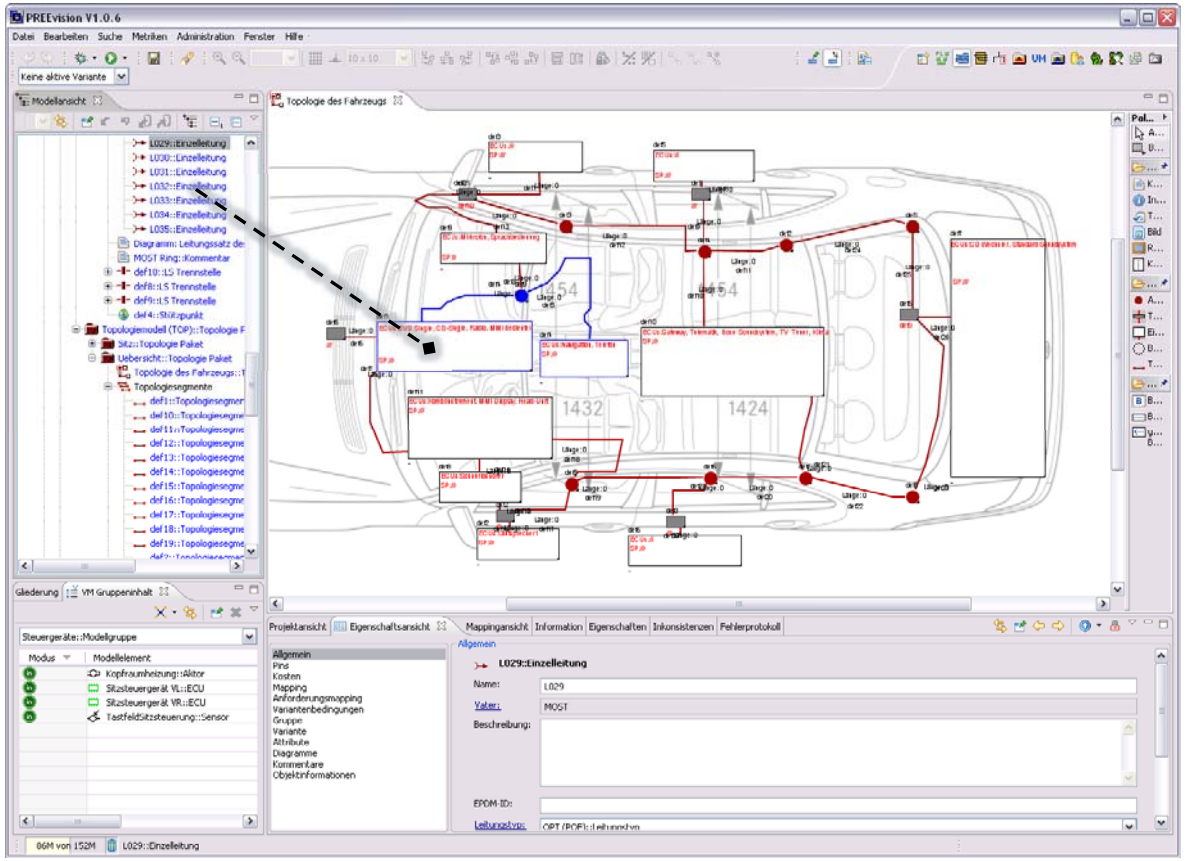


Topologie Sitzmodul

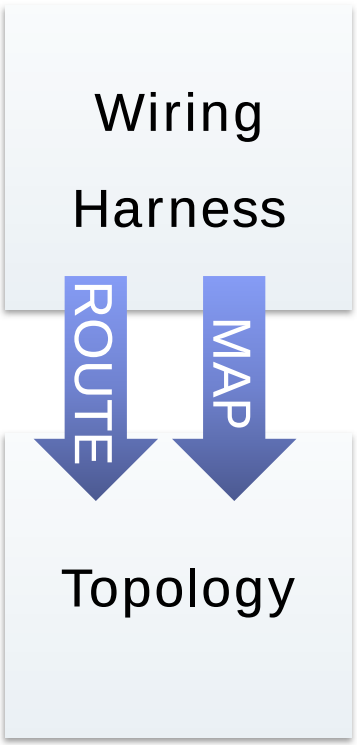
Leitungssatz

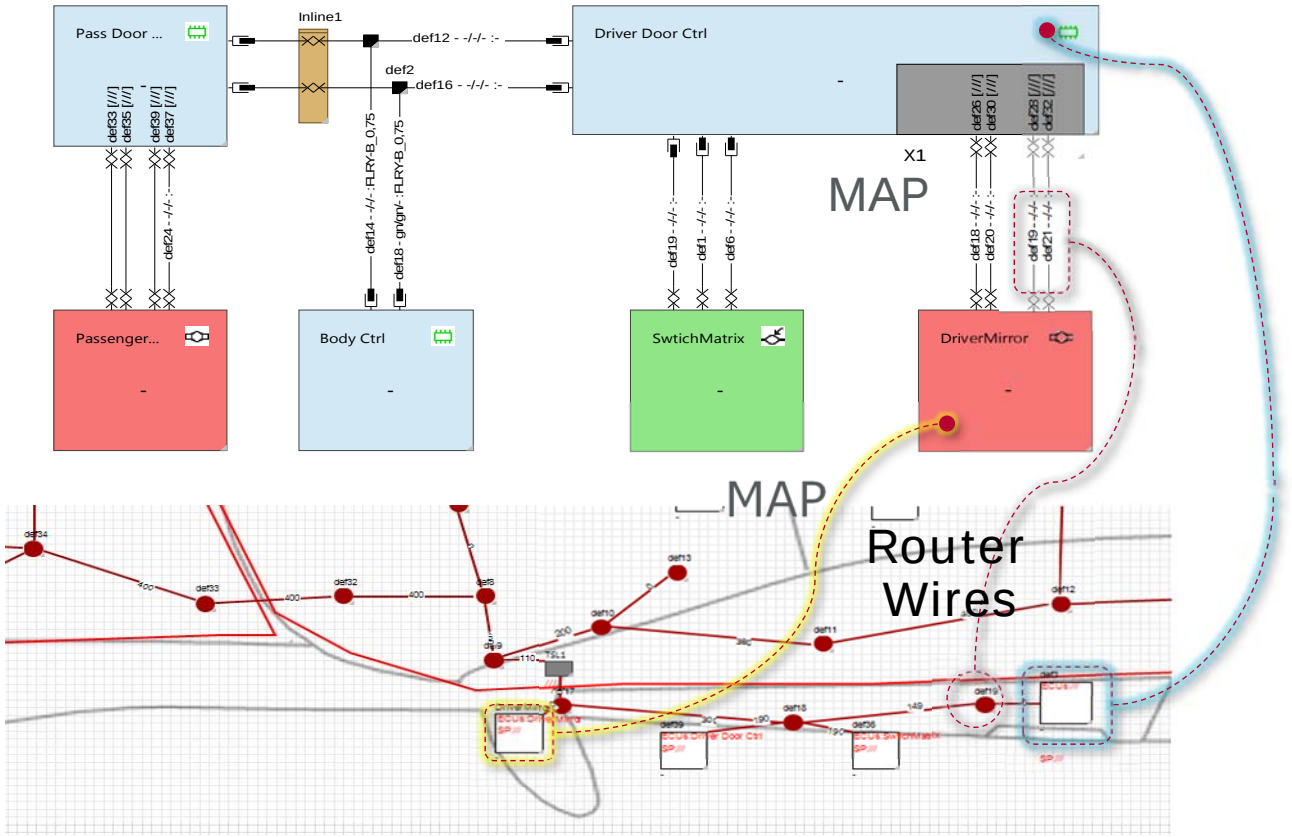


2D Topologie in PREEvision

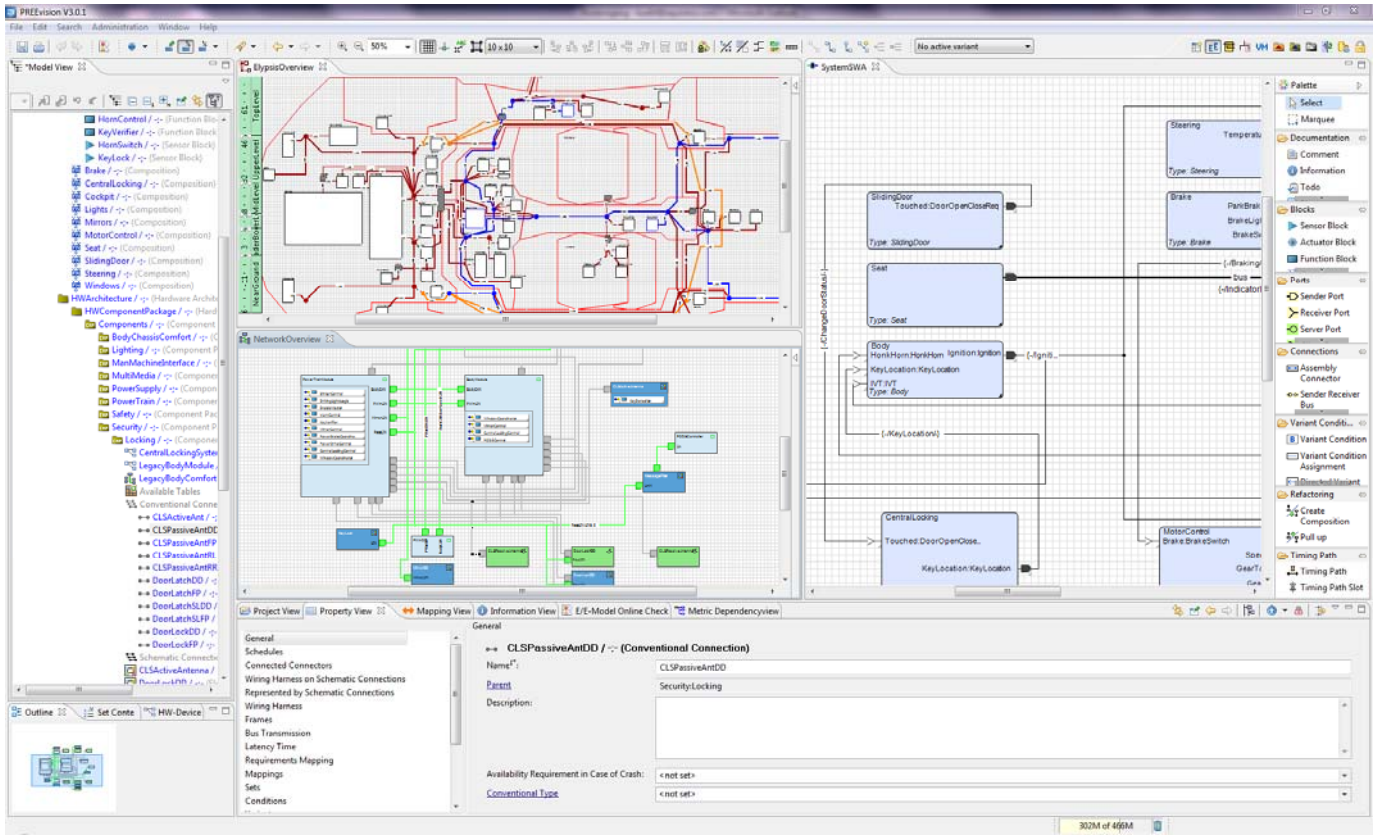


Harness + Components to Topology



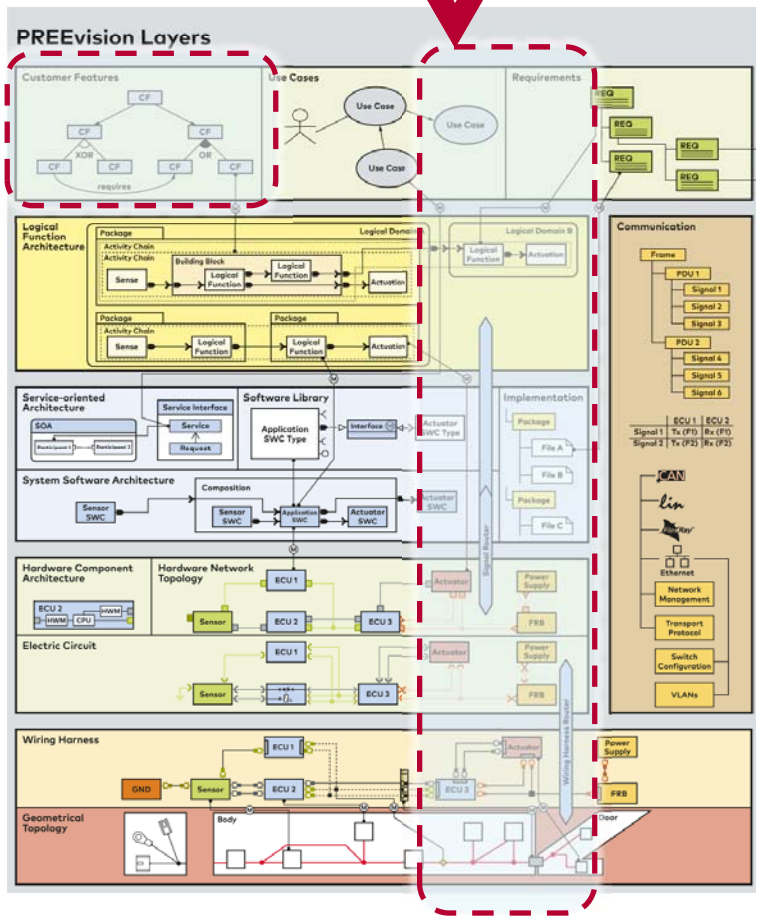


GUI concept



PREEvision Collaboration Platform (VII)

Varianten



Solution Space in Product Lines

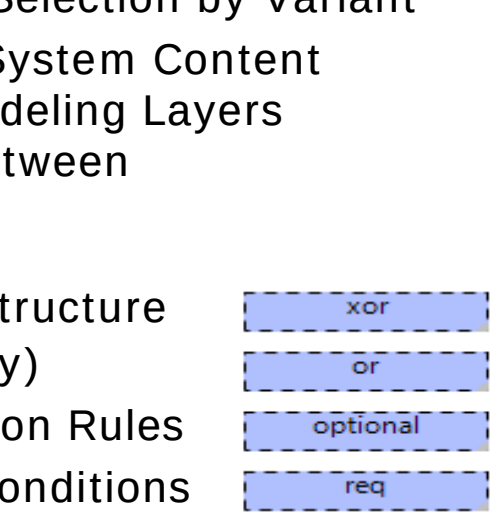
Explicit vs. Implicit Definition

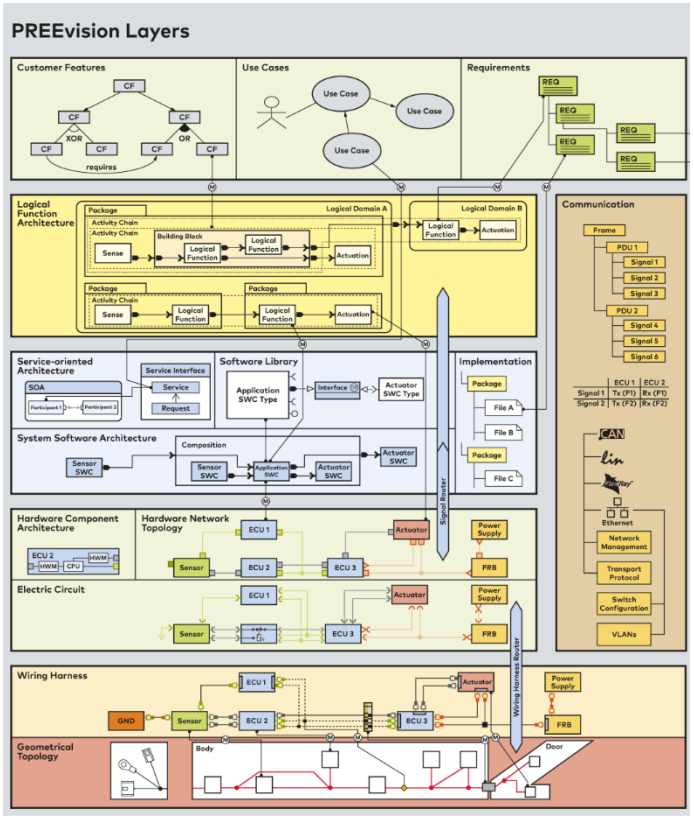
Explicit – Variants as Representatives

- Concept Selection by Variant
- Set and System Content
  On all Modeling Layers
  And in between

Implicit

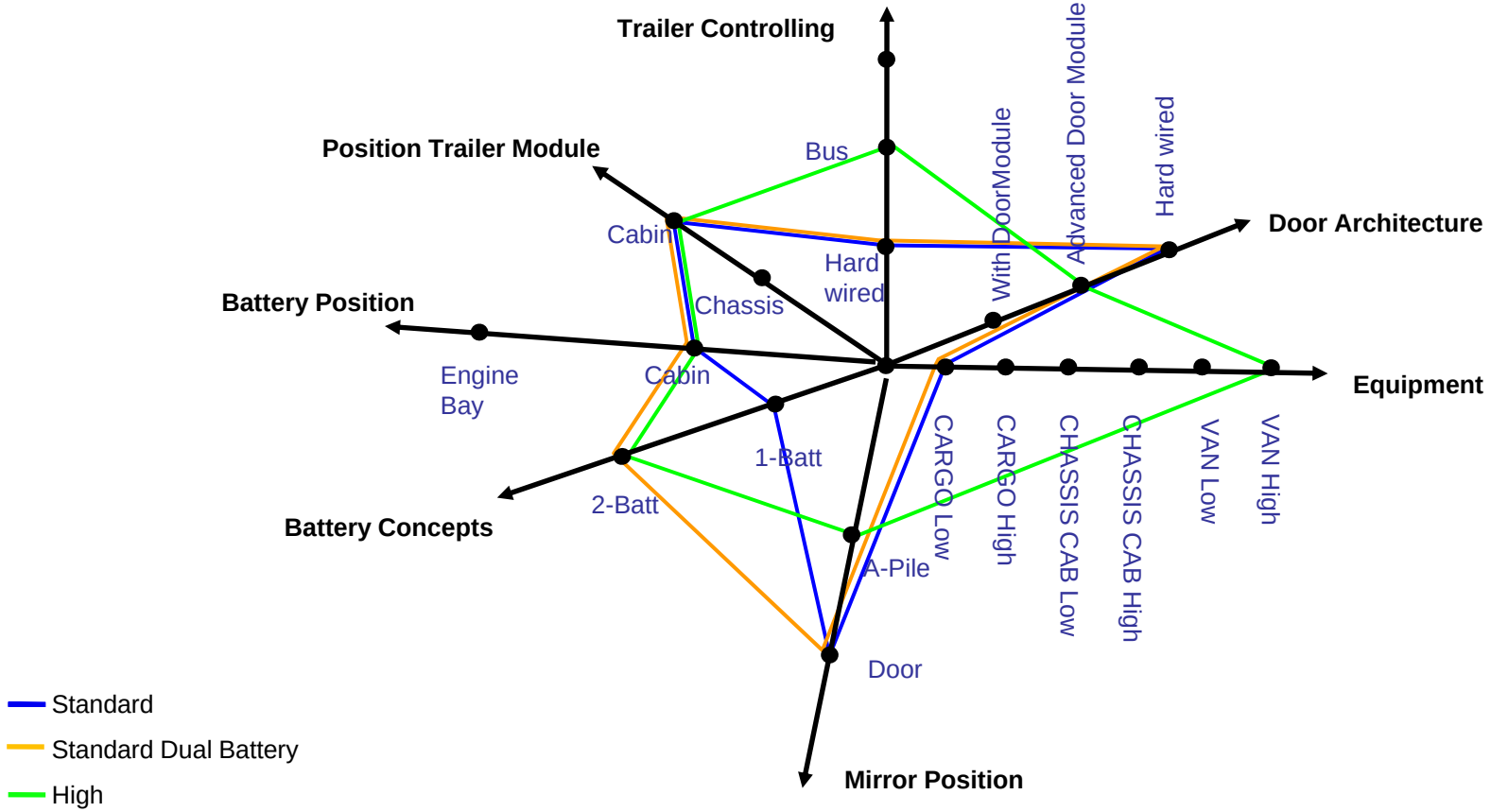
- Variant Structure
  (Hierarchy)
- Propagation Rules
- Variant Conditions





Beispiel

Architekturvarianten



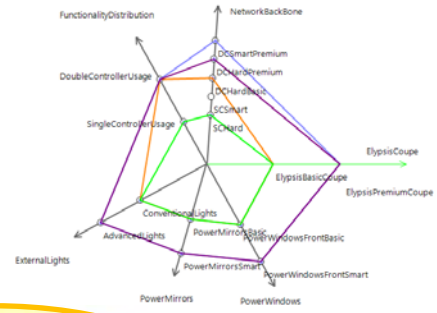
Use Case: Variant Management

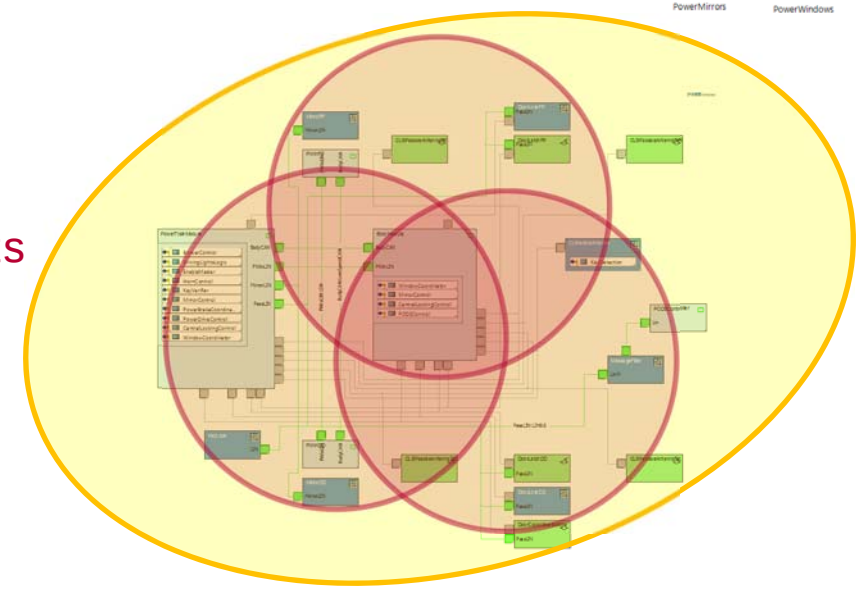
Super-Set Modeling

‘150% Modeling’ of Product Lines and Libraries

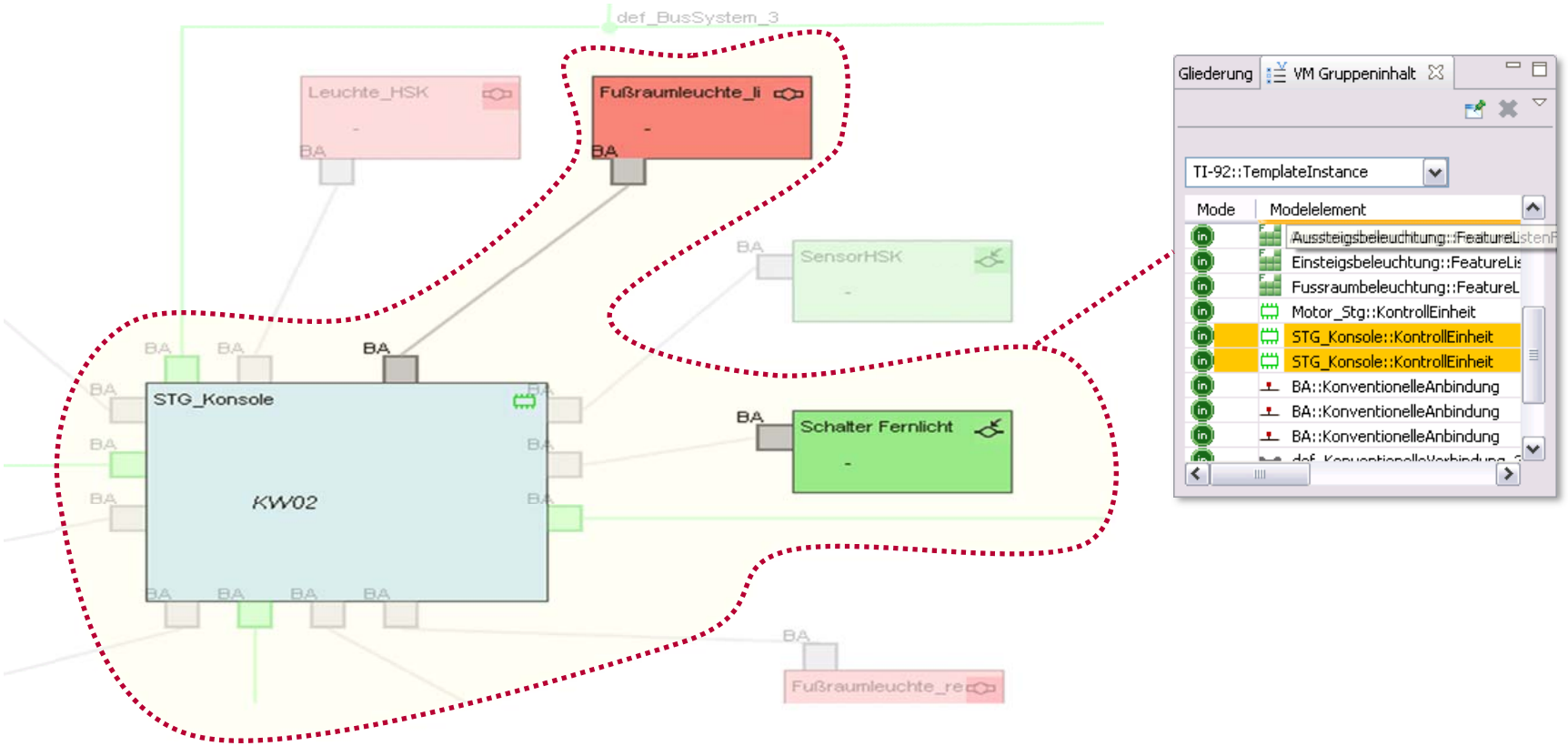
- **Technical Concepts**
  Definition and Selection
- **Equipment**
  Defining Configurations of Features
- **Building**
  Representative Architecture Variants

- Set-based Structuring
- System Integration

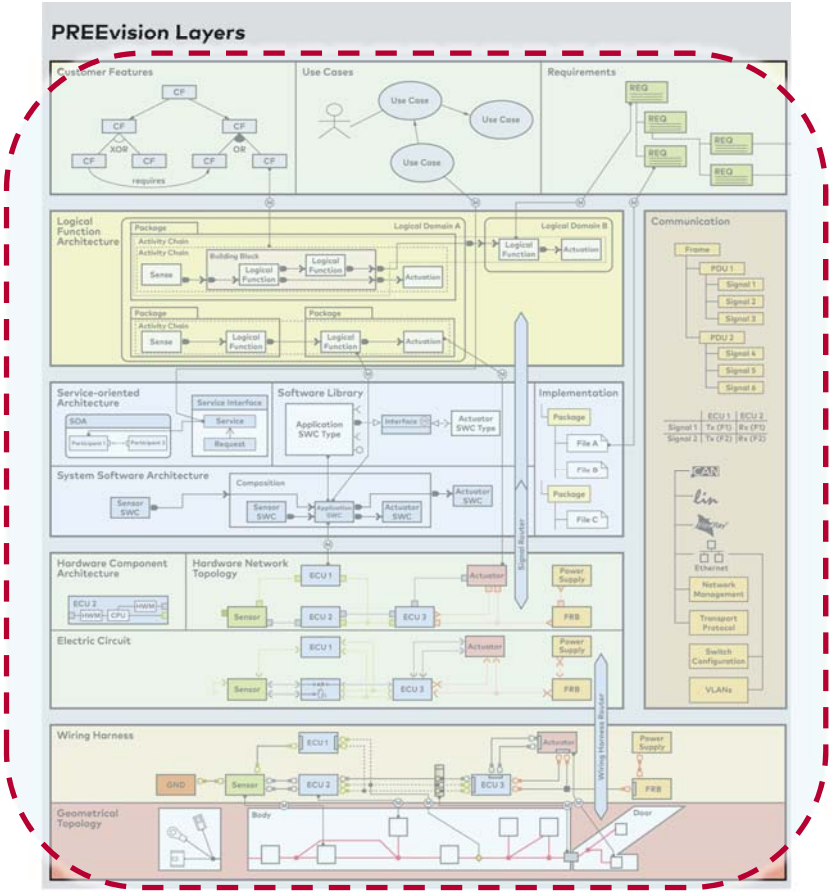




Bilden von Varianten



Metriken


PREEvision Layers
Customer Features
Use Cases
Requirements
Logical Function Architecture
Communication
Service-oriented Architecture
Software Library
Implementation
System Software Architecture
Hardware Component Architecture
Hardware Network Topology
Electric Circuit
Wiring Harness
Geometrical Topology

Bewertungskriterien einer E/E-Architektur

Qualität

- Wiederverwendbarkeit
- Robustheit
- EMV-Klassifikation
- Erweiterbarkeit
- Testbarkeit von Feature / Funktionen
- Unabhängigkeit der Teilfunktionen
- Kompatibilität
- Abhängigkeit zu einzelnen Zulieferern
- Einsatz bewährter Technologie (Anteil)
- Wiederverw. bestehender Komponenten
- Redundanz (SIL)

Kosten

- Gesamtsystem
  - Entwicklungskosten
    - Steuergeräte
    - Leitungssatz
    - Einsparungen
    - Wiederverwendung
  - Materialkosten
    - Leitungssatz
    - Komponenten
  - Zeitaufwand Montage
    - Teile, Baugruppen
- Min-, Max-, Eck-Typen
- Feature
  - Systemkosten
  - Garantie / Wartung
- Life Cycle

Mech. Eigenschaften

- Gewichte
  - Steuergeräte
  - Aktoren
  - Sensoren
  - Leitungssatz
  - Gewichtsverteilung
- Dimensionen
  - Leitungslängen
  - Komponentengrößen
  - Befüllung Segmente

E/E

- Bordnetz
  - Netzmanagement-Ereignisse
  - Nachlauf
  - Ruhestrom
  - Versorgung Komponenten
  - Sicherung Komponenten
- Informationsaustausch
  - Anzahl Bussysteme
  - Buslasten
  - Anzahl konv. Verb.
- Diagnoseeigenschaften
- Beiträge zu Fehlerspeicher-Einträgen
- Gateways

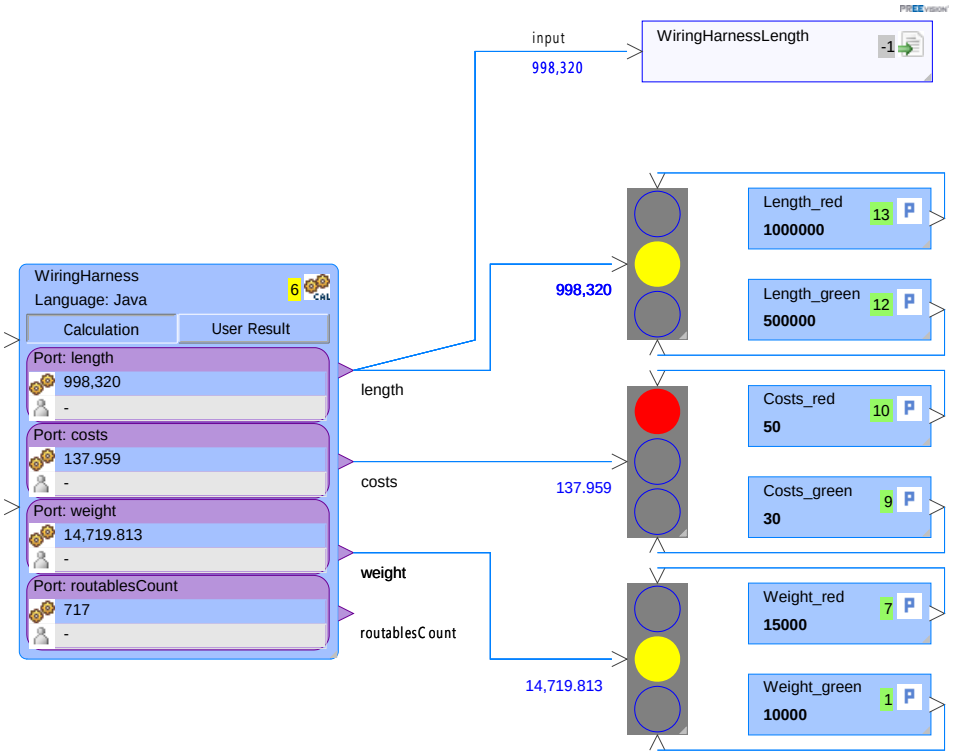
Dokumentation

- Entwicklungsstand
  - Funktionsbeschreibung
  - SG- / Systemsteckbriefe
- Versionsverwaltung
- Dokumentationsunterlagen
  - Präsentationen
  - Topologie-Schaltbild
  - Architekturbewertungen
  - Architekturvergleiche
- Lastenheft
- Unterlagen für Audits

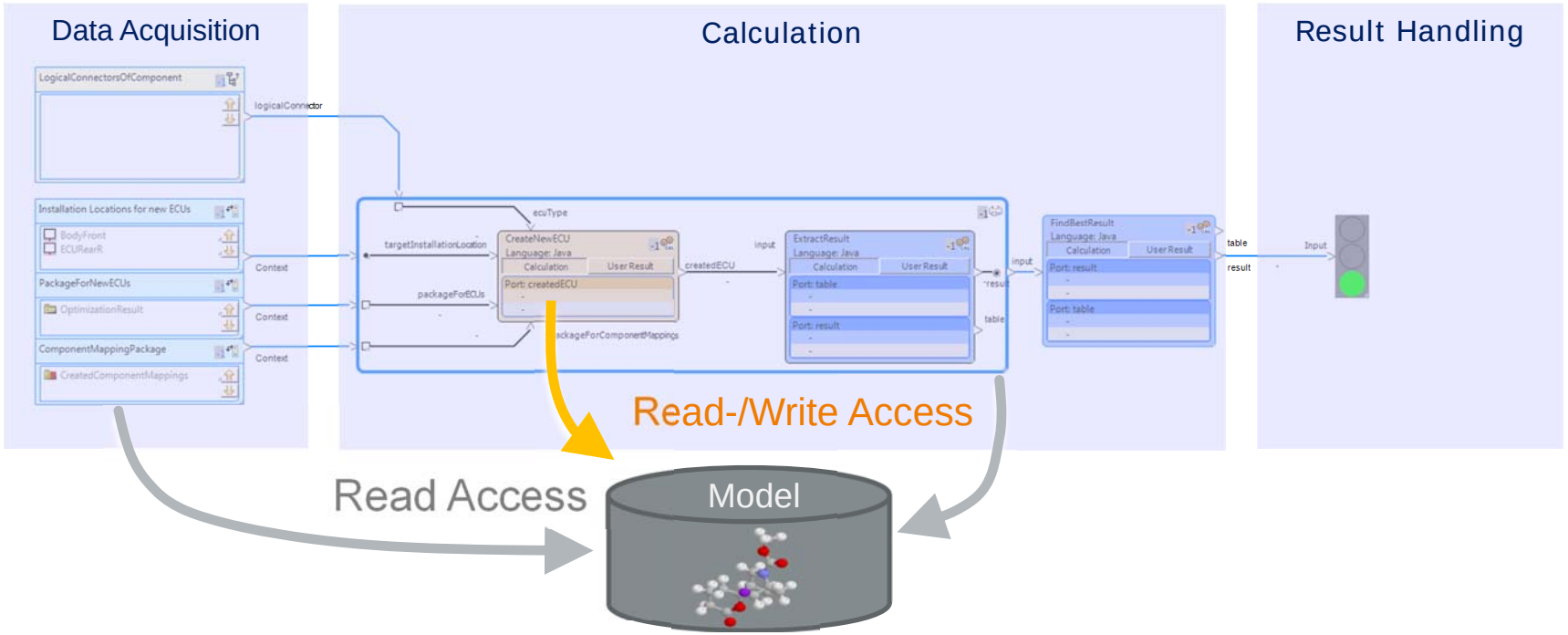
Machbarkeit

- Physikalische Baubarkeit, logisches Vernetzungskonzept
- Steuergeräte
- Bauräume, Verkabelung, Komponenten
- Gesetzliche Vorschriften
- Stückliste
- Reifegrad der Funktionen

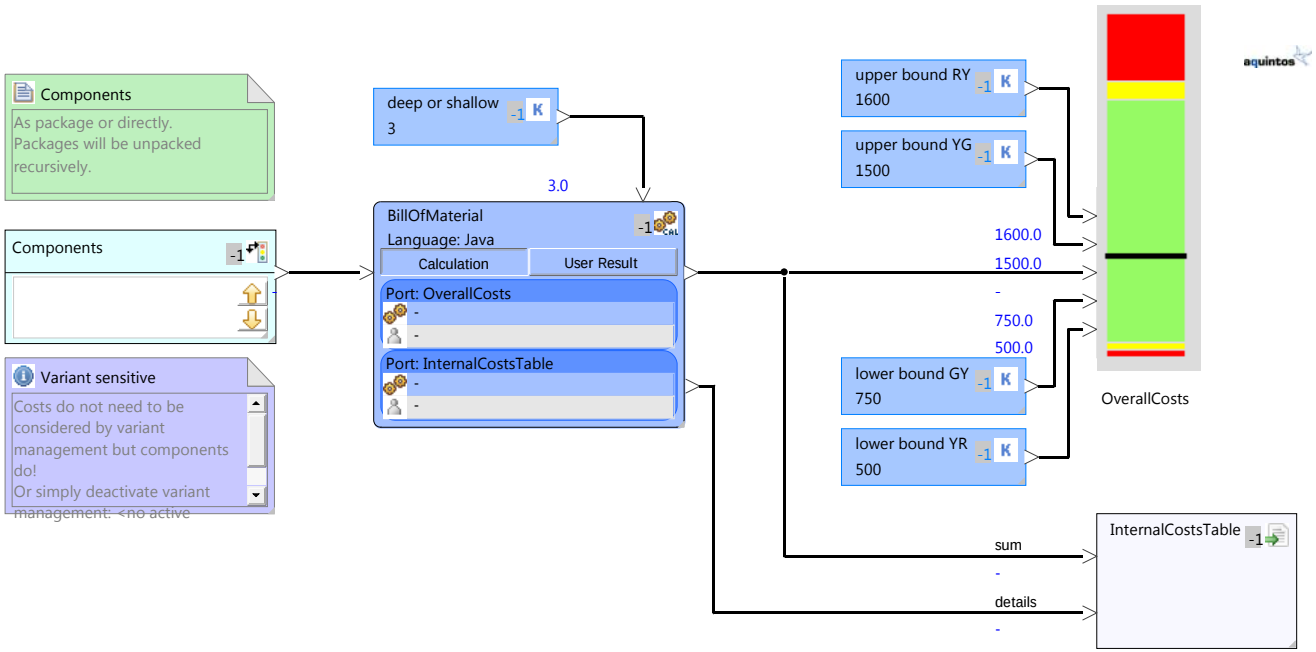
- Are used to perform **calculations** on the data model for analysis and optimization.
- Can be widely used to enrich tool with customer **individual** IP
- Always run on the full model – the result is up to date.
- Output of metrics is streamed into **Reports** or into **GUI** as lights, scales or values.



- Are based on a graphical notation and can be expanded by Java-code

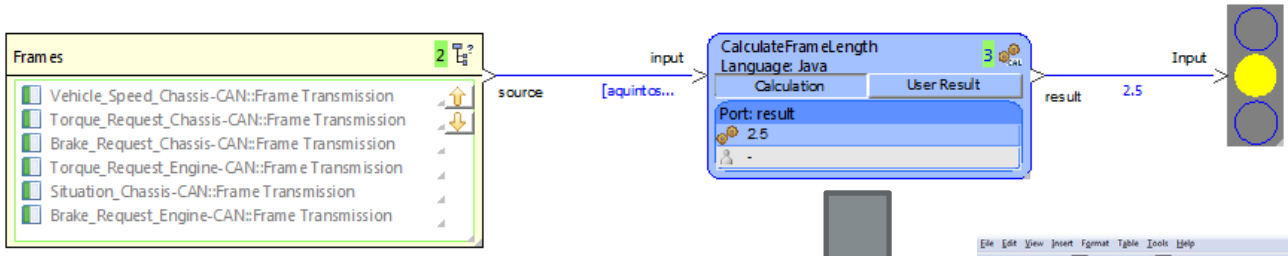


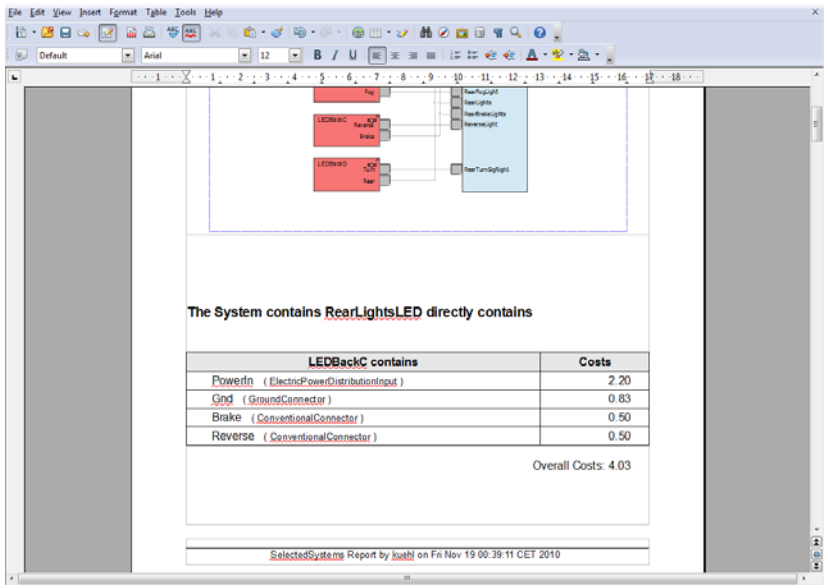
Graphical Data-Flow Oriented Language



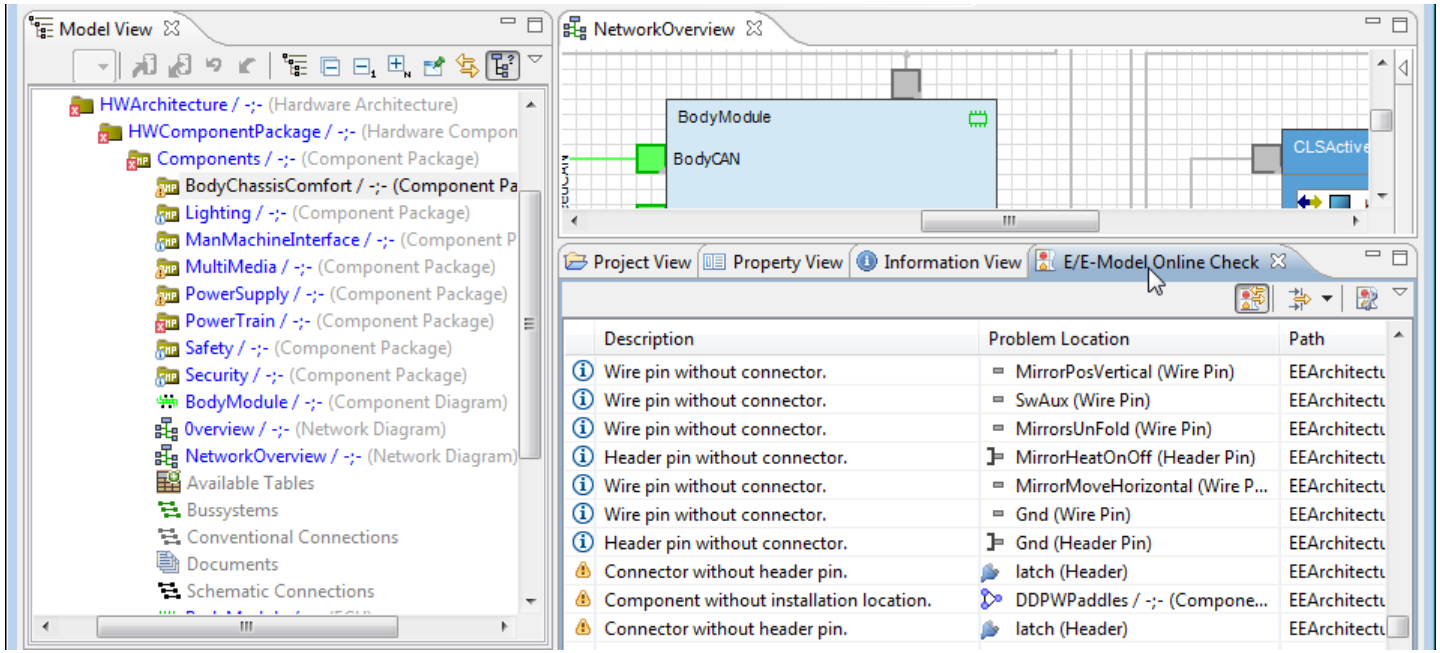
- Metrics are used to compute different evaluation criteria
- Output of metrics is streamed into Reports or into GUI

Use Case: Report Generator



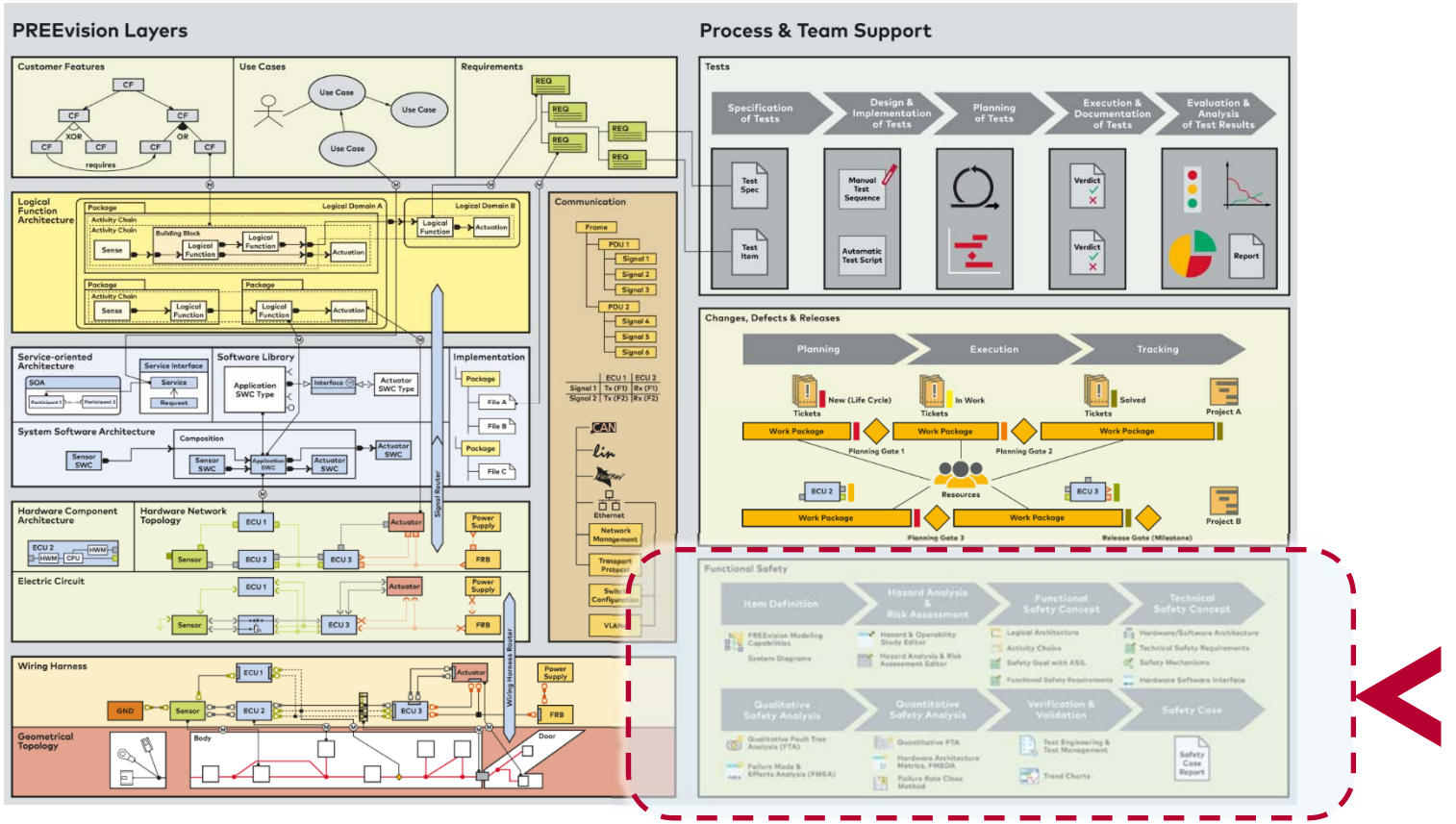


Consistency Checker

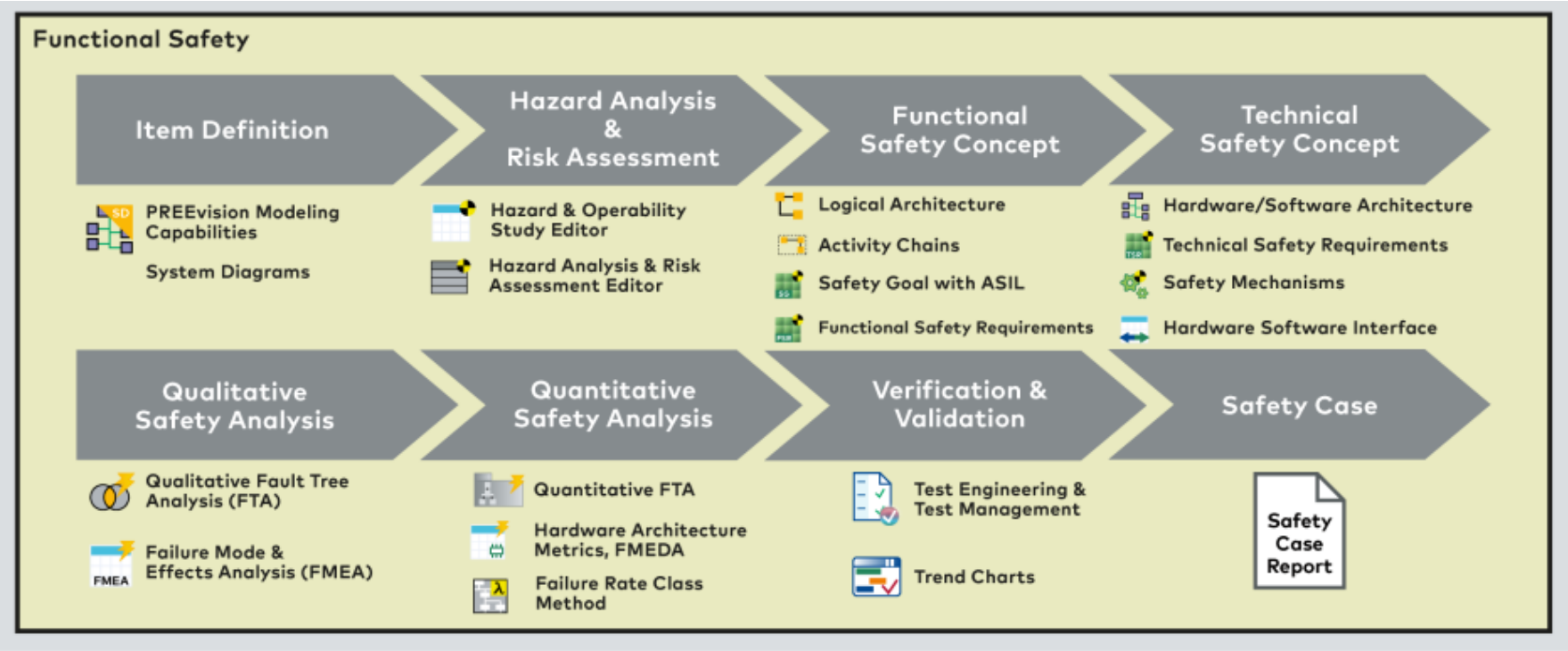


- Errors, Warnings, Information on Architecture or selected Parts
- Relevant Consistency Checks to be selected/extended
- Interactive ToDo List

Safety


PREEvision Layers
Process & Team Support
Customer Features
Use Cases
Requirements
Logical Function Architecture
Communication
Service-oriented Architecture
Software Library
Implementation
System Software Architecture
Hardware Component Architecture
Hardware Network Topology
Electric Circuit
Wiring Harness
Geometrical Topology
Tests
Specification of Tests
Design & Implementation of Tests
Planning of Tests
Execution & Documentation of Tests
Evaluation & Analysis of Test Results
Changes, Defects & Releases
Planning
Execution
Tracking
Functional Safety
Item Definition
Hazard Analysis & Risk Assessment
Functional Safety Concept
Technical Safety Concept
Qualitative Safety Analysis
Quantitative Safety Analysis
Verification & Validation
Safety Case

Safety


Functional Safety
Item Definition
Hazard Analysis & Risk Assessment
Functional Safety Concept
Technical Safety Concept
PREEvision Modeling Capabilities
System Diagrams
Hazard & Operability Study Editor
Hazard Analysis & Risk Assessment Editor
Logical Architecture
Activity Chains
Safety Goal with ASIL
Functional Safety Requirements
Hardware/Software Architecture
Technical Safety Requirements
Safety Mechanisms
Hardware Software Interface
Qualitative Safety Analysis
Quantitative Safety Analysis
Verification & Validation
Safety Case
Qualitative Fault Tree Analysis (FTA)
Failure Mode & Effects Analysis (FMEA)
FMEA
Quantitative FTA
Hardware Architecture Metrics, FMEDA
Failure Rate Class Method
Test Engineering & Test Management
Trend Charts
Safety Case Report

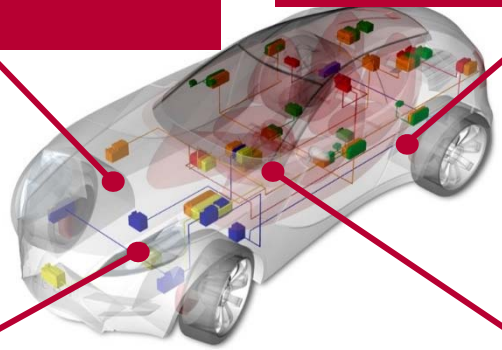
Motivation: Many Functions are Safety-Critical

ESP

Unintended, single-sided brake effect on straight lane

Electronic Park Brake

Unintended activation in motion



Collision Avoidance

Acceleration instead of deceleration in traffic

Airbag

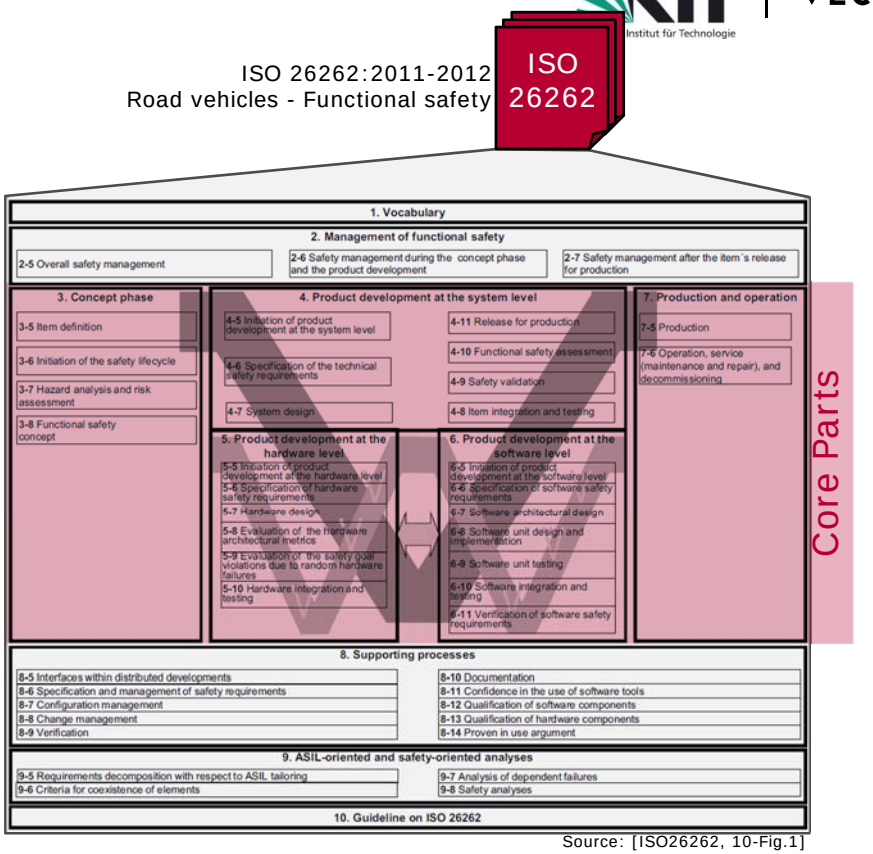
Delayed deployment after crash detection

Reality or science fiction – Is such a scenario realistic?

Challenges

- 10 Parts
- 43 Chapters
- 100 Work products
- 180 Engineering methods
- 500 Pages
- 600 Requirements

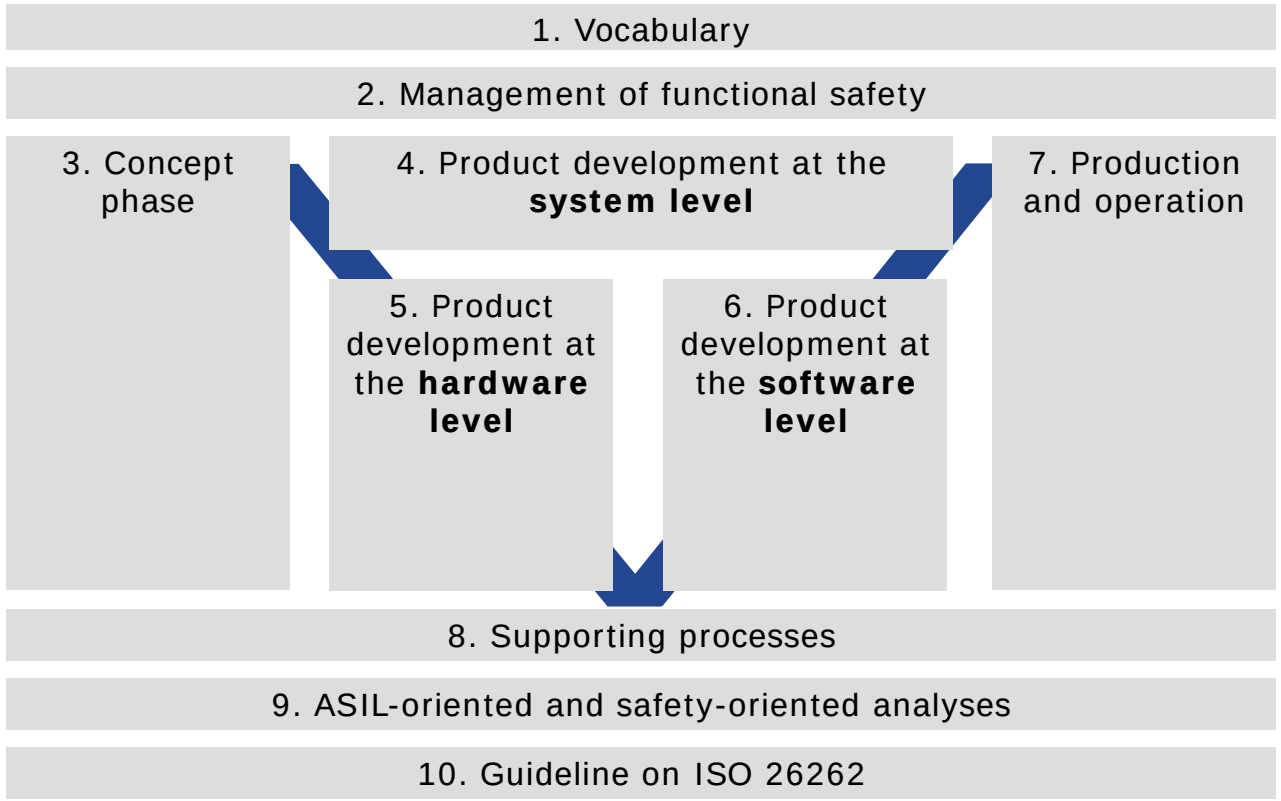
ISO 26262:2011-2012
Road vehicles - Functional safety



Source: [ISO26262, 10-Fig.1]

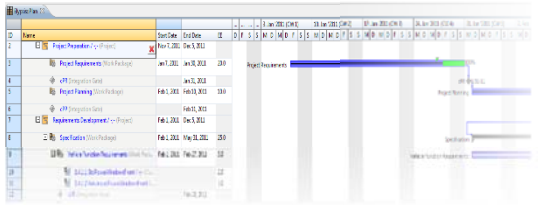
▶▶ Complex standard → Risk of overheads and costs if applied ad hoc

Parts of ISO 26262



Integrated Model Based System Engineering Platform

Safety Plan



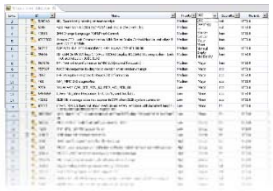
Safety Analysis Methods

Cost efficient consistency and traceability

Requirements Management

System / Function / HW / SW Design

Test Management



Change Management

The safety lifecycle according to ISO 26262

1

Item Definition

Definition of features and their interactions, operating modes, vehicle states, etc.

2

Hazard and Risk Analysis

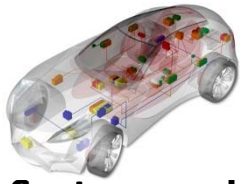
Identification and classification of hazardous events and derivation of appropriate system safety goals.

3

System safety concept

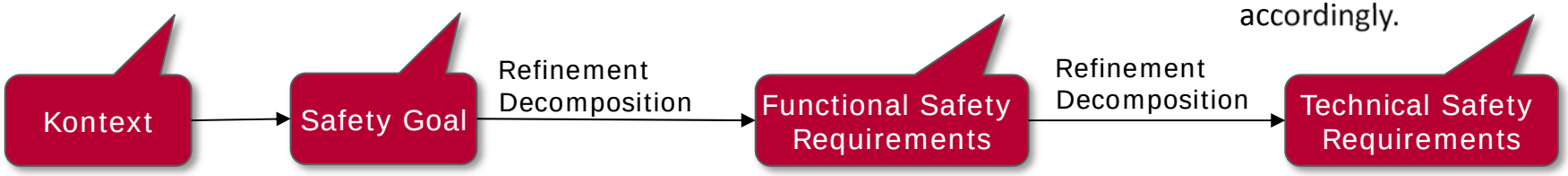
Design of a system concept for implementing (in LA) the safety goals (Req), for example on the basis of diagnostic or redundancy measures.

4



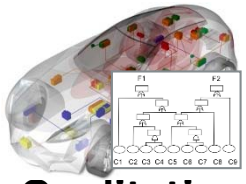
System and component design

Design of technical system and component concepts including the derivation and implementation of technical safety requirements accordingly.



The safety lifecycle according to ISO 26262

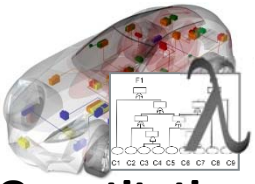
5



Qualitative Safety Analyses

Application of deductive and inductive safety analysis techniques (e.g. FTA, FMEA) to validate the ability of the design to meet the system safety goals.

6



Quantitative Safety Analyses

Calculation of the probability of the system failing to meet the safety goals and confirmation that the failure rate and diagnostic coverage targets are met.

7

Verification and Validation

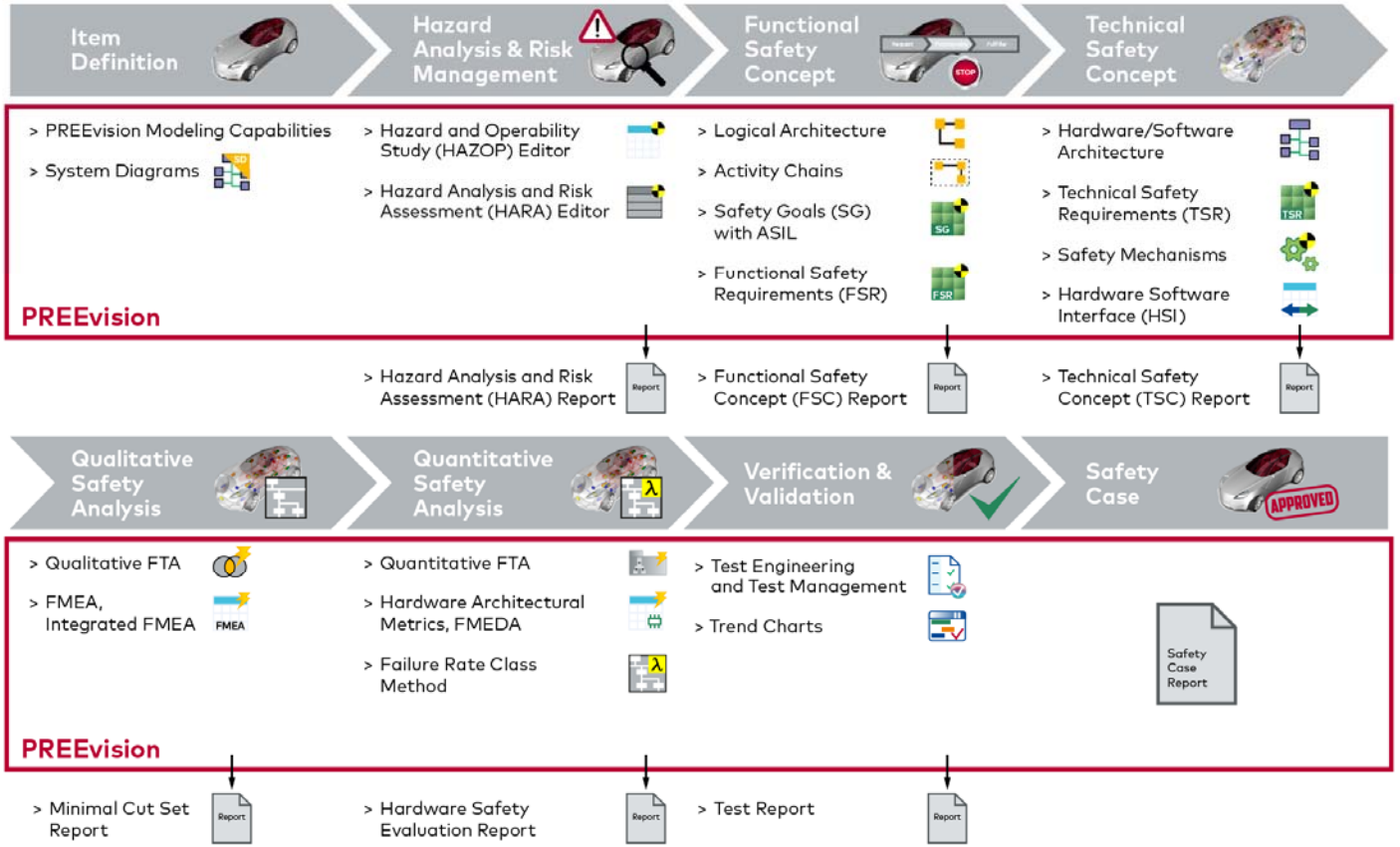
Confirmation through review, analysis and test that all safety requirements are correctly implemented in the delivered system and that all assumptions made in the safety concept are valid.

8

Safety Case

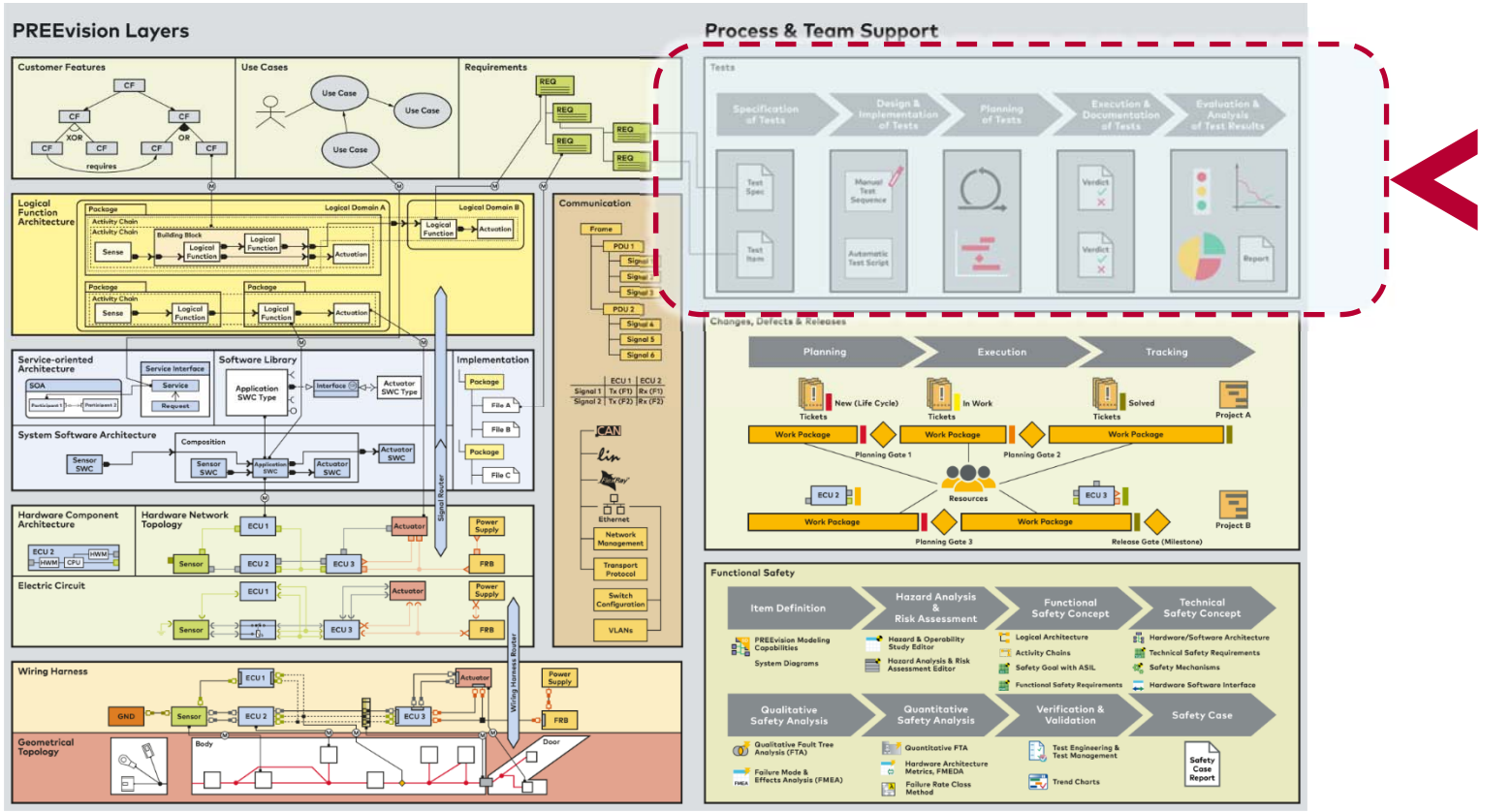
Construction of a structured, coherent, complete and convincing argument that the system meets all its safety goals and appropriate regulations.

Supported steps of the safety lifecycle



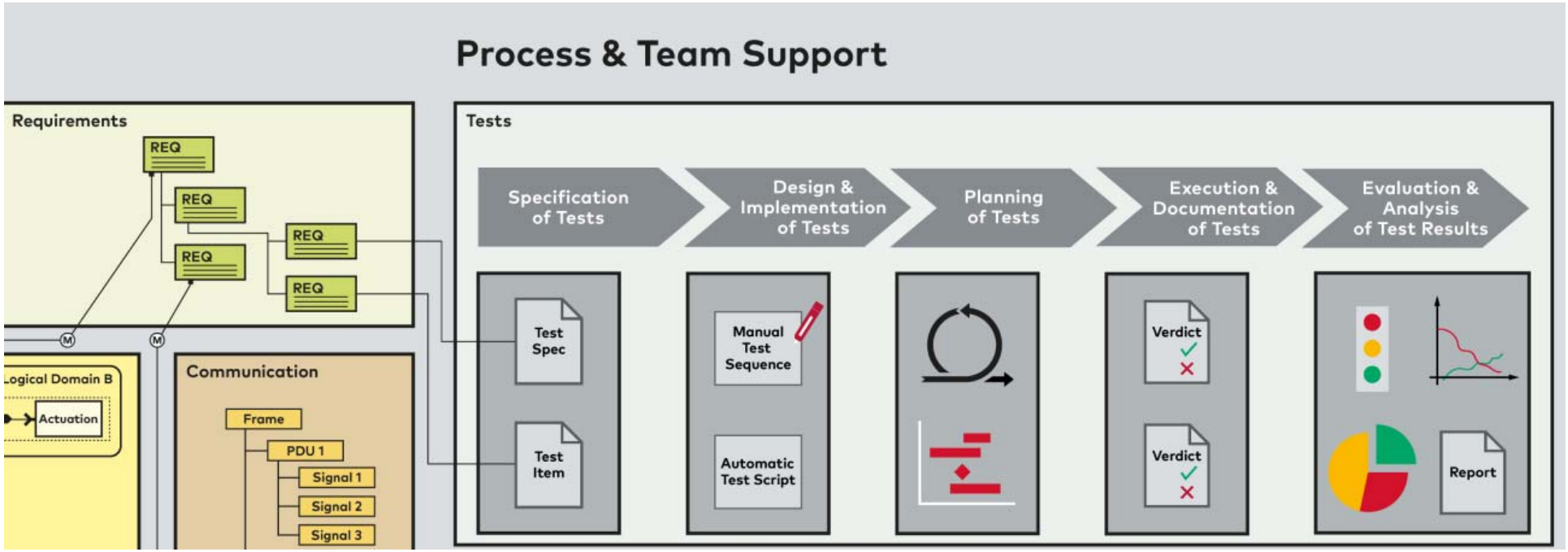
PREEvision Collaboration Platform (IX)

Test Daten Mangement



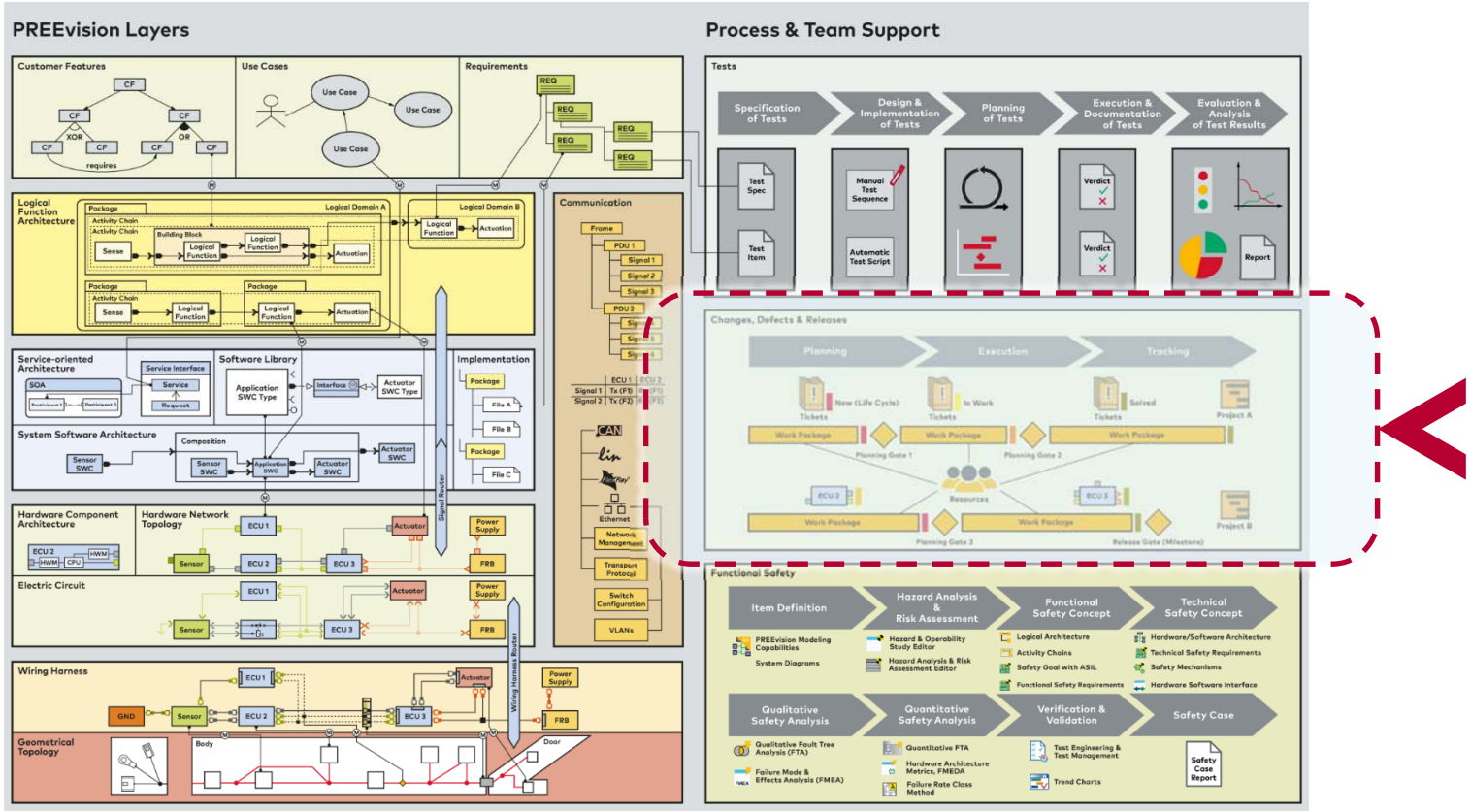
PREEvision Collaboration Platform (IX)

Test Daten Management



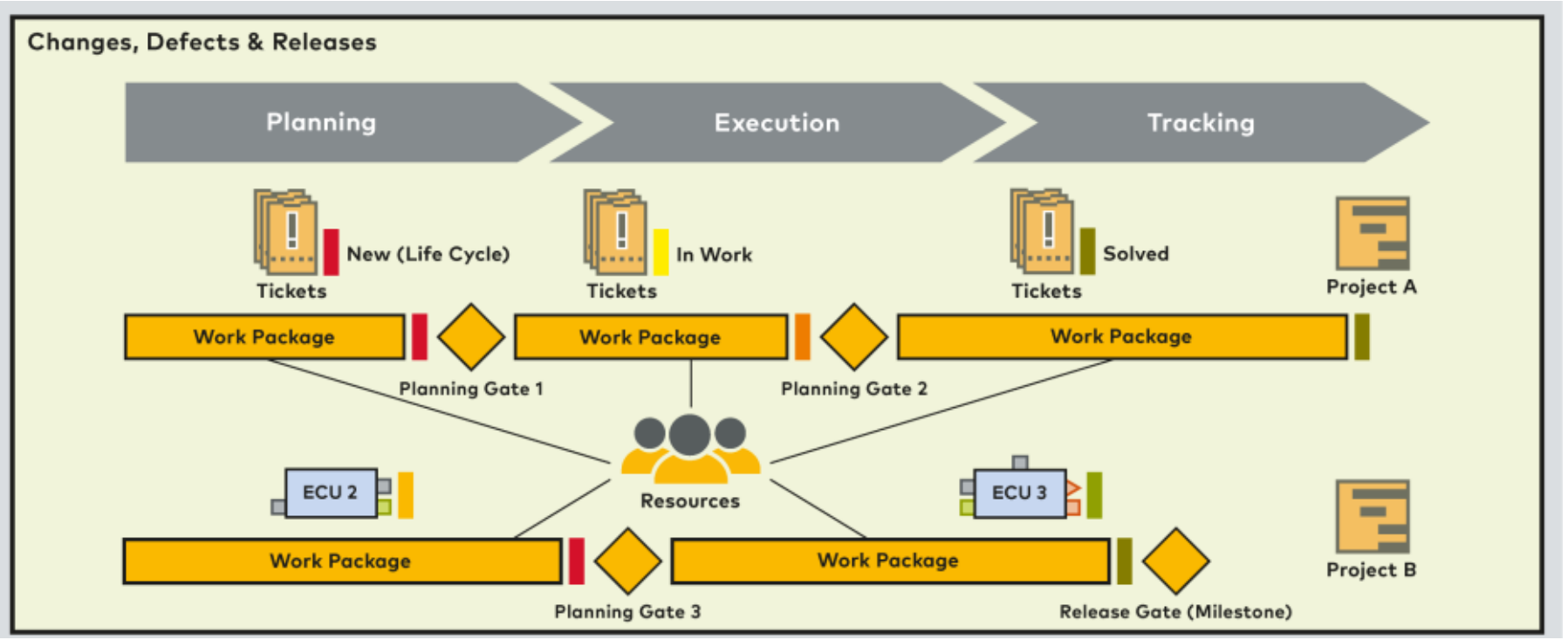
PREEvision Collaboration Platform (IX)

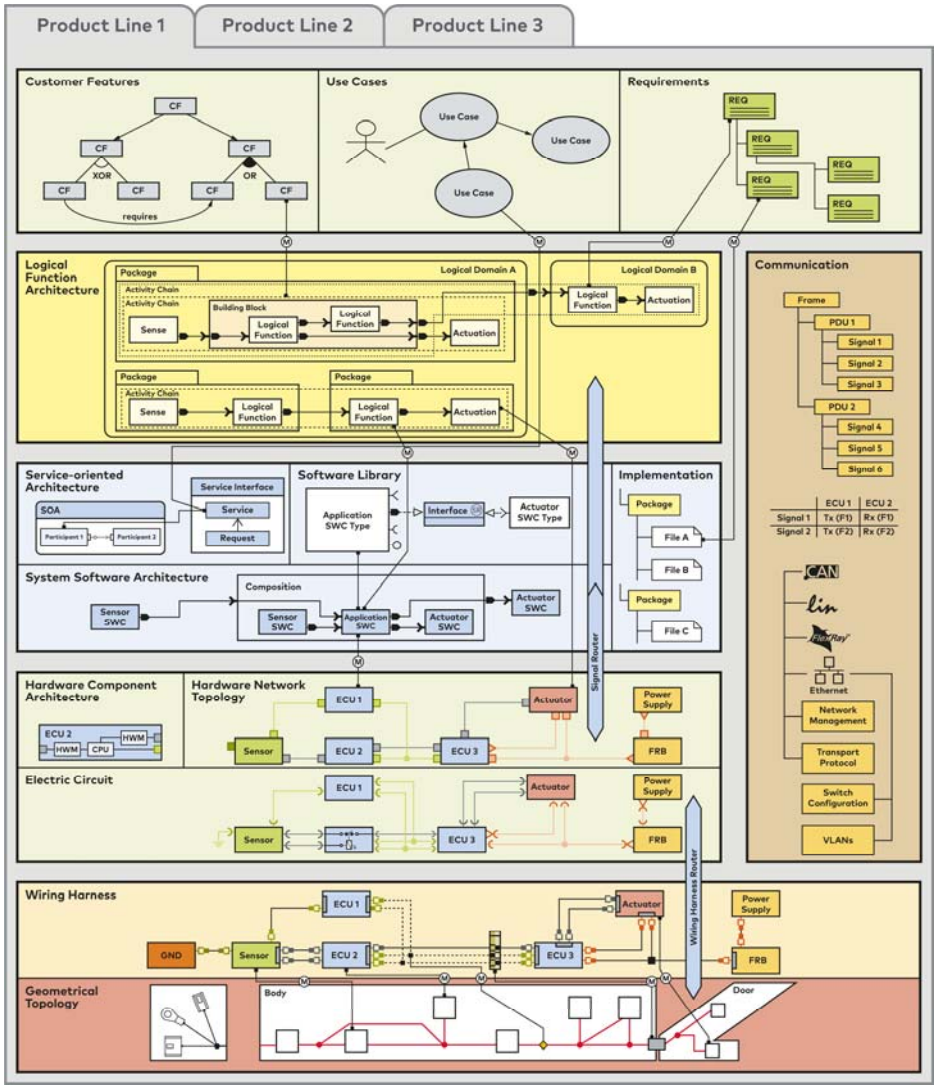
Change Management



PREEvision Collaboration Platform (IX)

Change Management

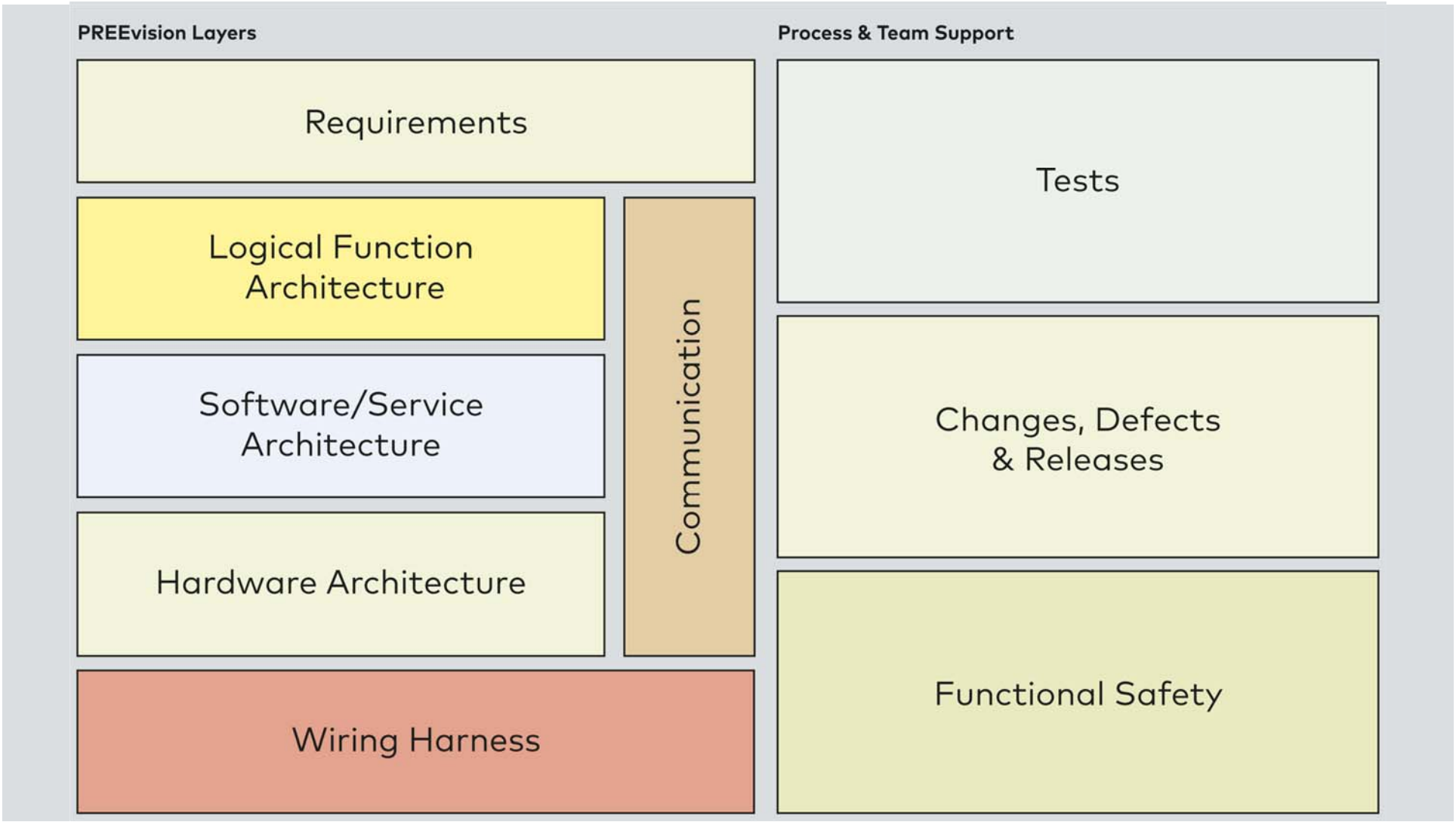


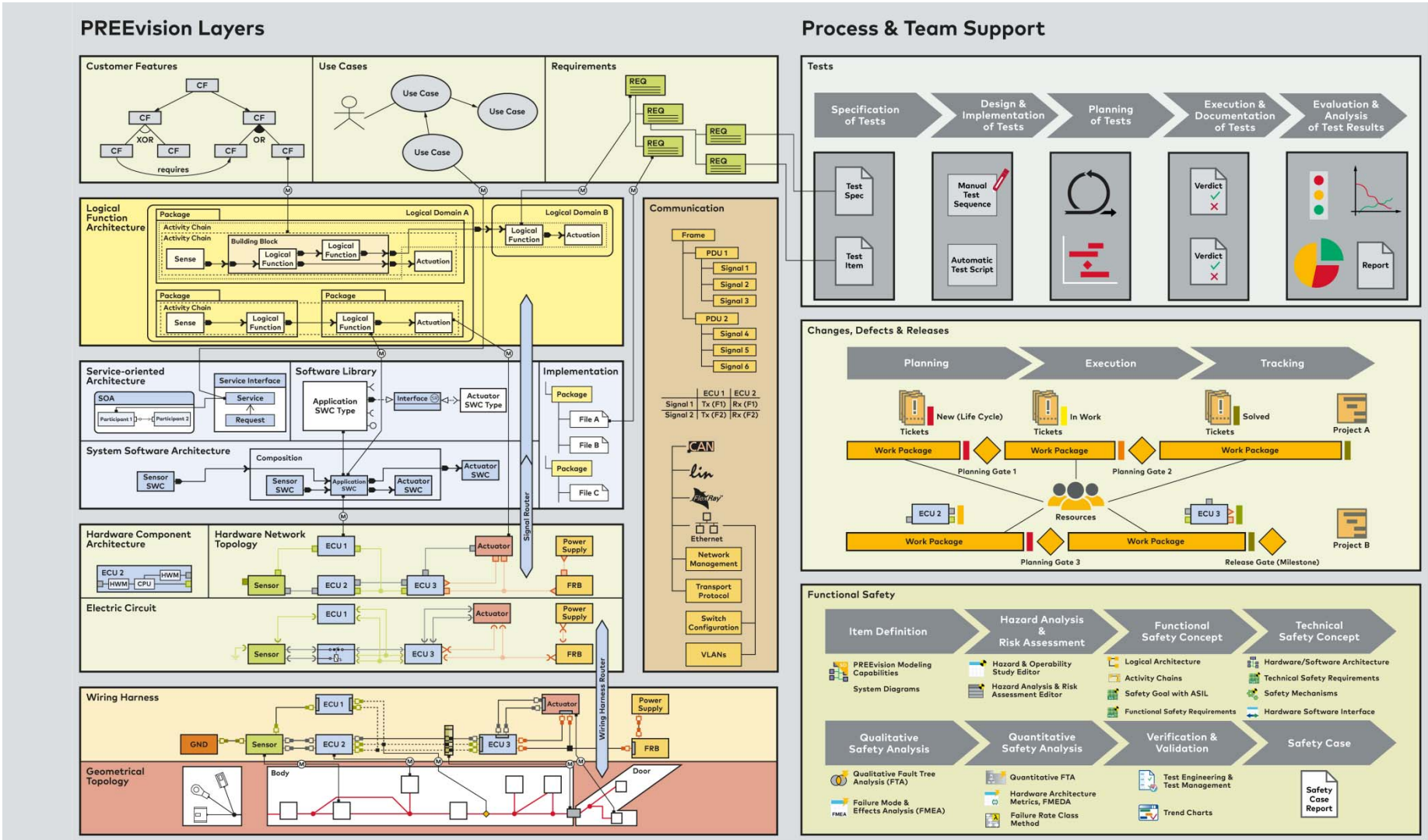


PREEvision Layers

- Requirements
- Logical Function Architecture
- Software/Service Architecture
- Hardware Architecture
- Wiring Harness

Communication


PREEvision Layers
Requirements
Logical Function Architecture
Software/Service Architecture
Hardware Architecture
Communication
Wiring Harness
Process & Team Support
Tests
Changes, Defects & Releases
Functional Safety


PREEvision Layers
Customer Features
CF
XOR
OR
requires
Use Cases
Use Case
Requirements
REQ
Logical Function Architecture
Package
Logical Domain A
Logical Domain B
Activity Chain
Building Block
Sense
Logical Function
Actuation
Communication
Frame
PDU 1
Signal 1
Signal 2
Signal 3
PDU 2
Signal 4
Signal 5
Signal 6
ECU 1
ECU 2
Tx (F1)
Rx (F1)
Tx (F2)
Rx (F2)
CAN
lin
FlexRay
Ethernet
Network Management
Transport Protocol
Switch Configuration
VLANs
Service-oriented Architecture
SOA
Service Interface
Service
Request
Software Library
Application SWC Type
Interface
Actuator SWC Type
Implementation
File A
File B
File C
System Software Architecture
Composition
Sensor SWC
Application SWC
Actuator SWC
Signal Router
Hardware Component Architecture
HWM
CPU
Hardware Network Topology
Electric Circuit
Sensor
Actuator
Power Supply
FRB
Wiring Harness
GND
Wiring Harness Router
Geometrical Topology
Body
Door
Process & Team Support
Tests
Specification of Tests
Design & Implementation of Tests
Planning of Tests
Execution & Documentation of Tests
Evaluation & Analysis of Test Results
Test Spec
Test Item
Manual Test Sequence
Automatic Test Script
Verdict
Report
Changes, Defects & Releases
Planning
Execution
Tracking
Tickets
New (Life Cycle)
In Work
Solved
Project A
Project B
Work Package
Planning Gate 1
Planning Gate 2
Planning Gate 3
Release Gate (Milestone)
Resources
Functional Safety
Item Definition
Hazard Analysis & Risk Assessment
Functional Safety Concept
Technical Safety Concept
PREEvision Modeling Capabilities
System Diagrams
Hazard & Operability Study Editor
Hazard Analysis & Risk Assessment Editor
Logical Architecture
Activity Chains
Safety Goal with ASIL
Functional Safety Requirements
Hardware/Software Architecture
Technical Safety Requirements
Safety Mechanisms
Hardware Software Interface
Qualitative Safety Analysis
Quantitative Safety Analysis
Verification & Validation
Safety Case
Qualitative Fault Tree Analysis (FTA)
Failure Mode & Effects Analysis (FMEA)
Quantitative FTA
Hardware Architecture Metrics, FMEDA
Failure Rate Class Method
Test Engineering & Test Management
Trend Charts
Safety Case Report

Thank you for your attention.

For detailed information about Vector
and our products please have a look at:
www.vector.com

Authors

Dr. Clemens Reichmann
Productline Process Tools
Vector Informatik GmbH, Stuttgart
aquintos GmbH, Karlsruhe