Safety in Cooperative Autonomous Systems

Vision of Dynamic Safety Management

Dr. Daniel Schneider

Ms. Sc. Emilia Cioroaica

Fraunhofer IESE

Fraunhofer IESE

The institute for software and systems engineering methods

- Founded in 1996, headquartered in Kaiserslautern
- Over 155 full-time equivalents (FTEs)
- Our most important business areas:
  - Automotive and Transportation Systems
  - Automation and Plant Engineering
  - Health Care
  - Information Systems
  - Energy Management
  - E-Government

ESQ – Focus on Safety

Model-based Safety Engineering

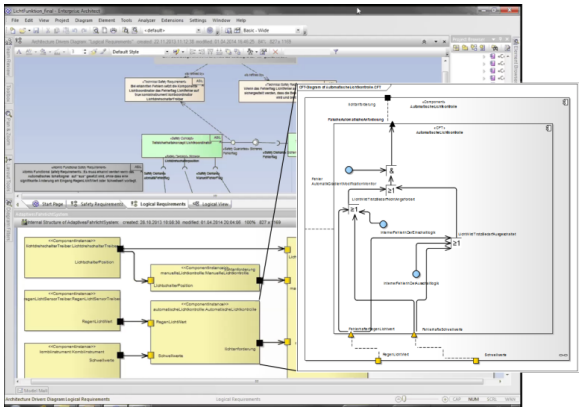
- Hazard- and Riskanalyses
- Safetyanalyses (FMEA, FTA, CFT etc.)
- Safety Concepts
- Tools and methods (in particular **safeTbox**)

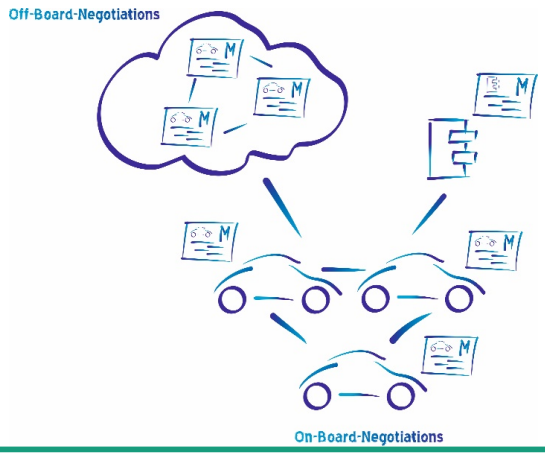
Engineering of Safety-related Solutions; e.g.:

- Controlling critical functions with mobile devices
- Safe CE(/NAC …) Hardware

Research Topics

- Modular Certification
- Runtime Certification in open adaptive systems
- Security for Safety
- Safety of autonomous systems
- Automated interference analyses and Embedded apps



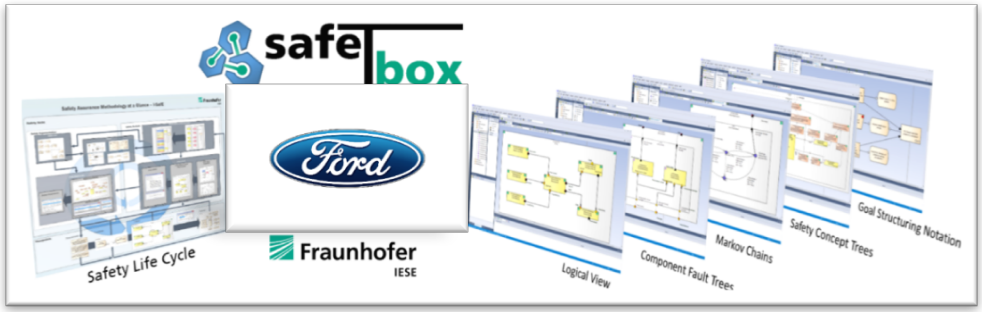


Fraunhofer

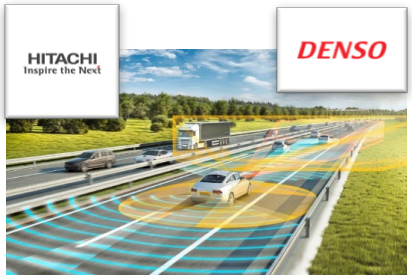
Examples of recent industry projects (since 2016)

Complete safety engineering for e-vehicle

e.GO



Model-based safety engineering with safeTbox



Markus Schweizer, Projektleiter der zentralen Forschung der Robert Bosch GmbH: *„Durch die mit dem Fraunhofer IESE realisierte Lösung zur sicheren Erkennung sporadischer Fehler können wir auf teure Spezialhardware verzichten und erreichen dennoch Sicherheit. Wir sind von den Ergebnissen der Kooperation mit dem Fraunhofer IESE so überzeugt, dass wir sie in die Serienproduktion übernehmen werden.“*

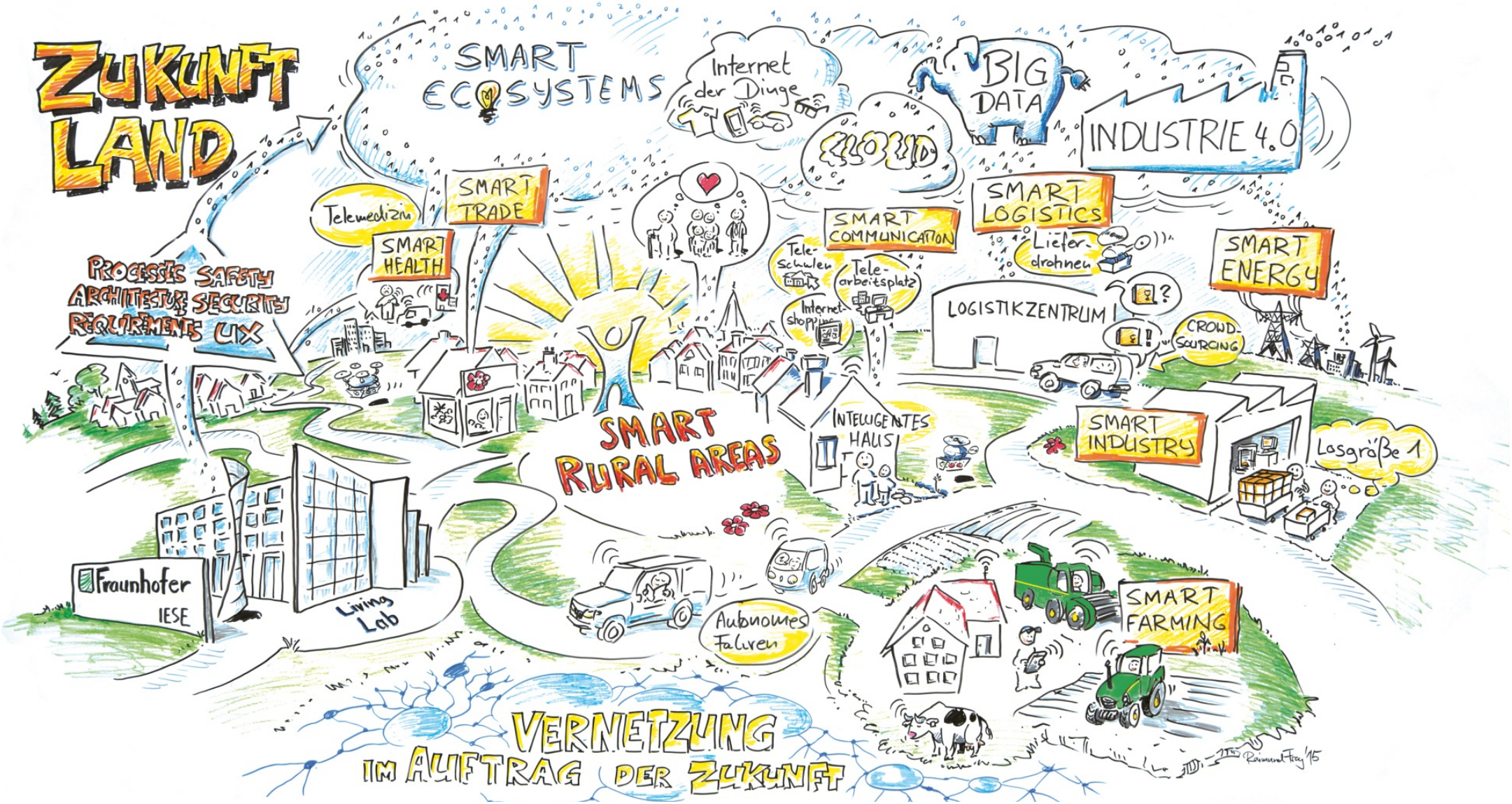
BOSCH

Innovative safety Architectures

--- Our Vision of Dynamic Safety Management ---

Fraunhofer

Smart everything – Diverse (yet interlinked) domains, similar challenges



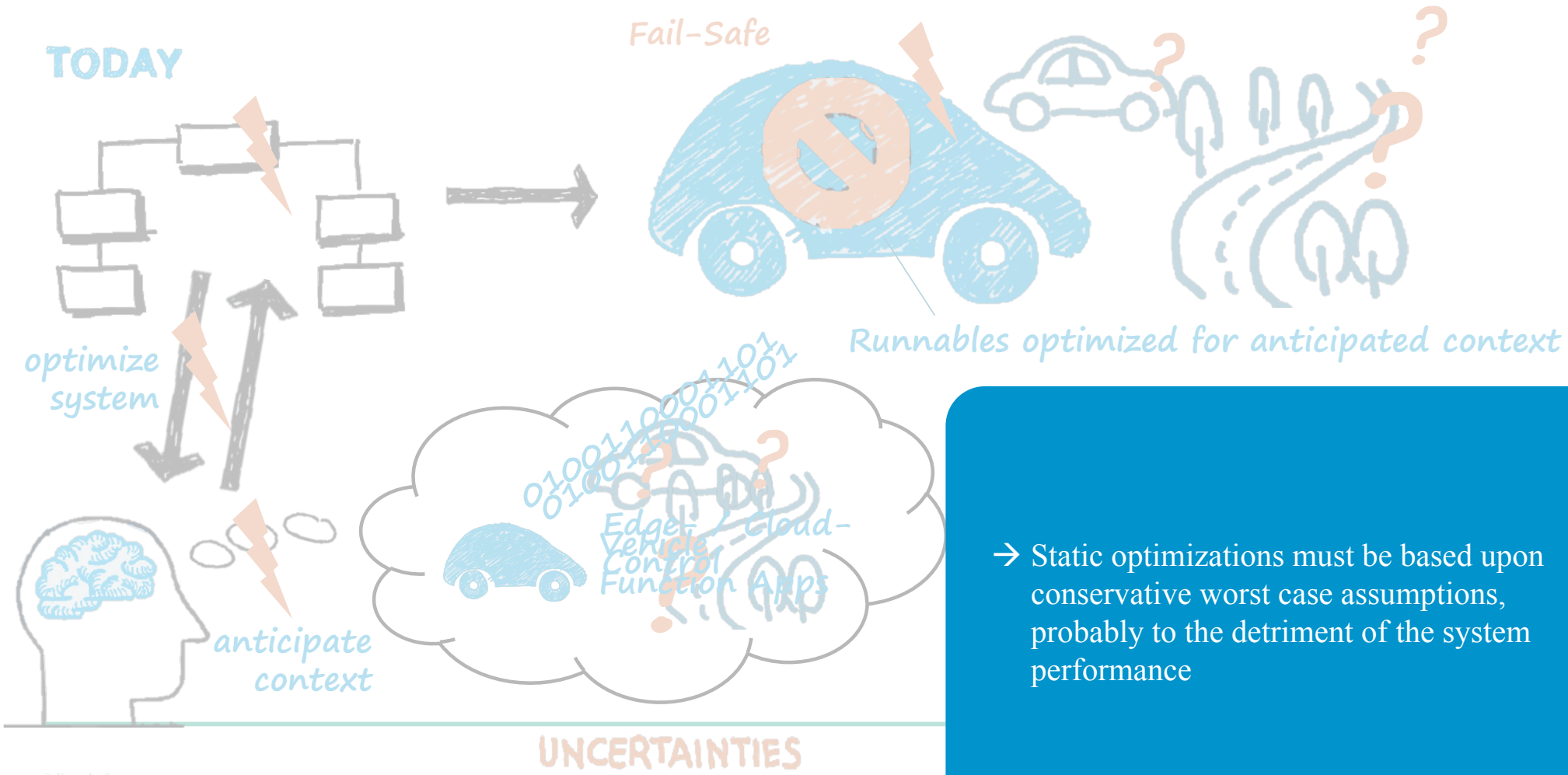
A networked and automated world

Making this vision reality requires:

Autonomy – humans incapable to manually control this

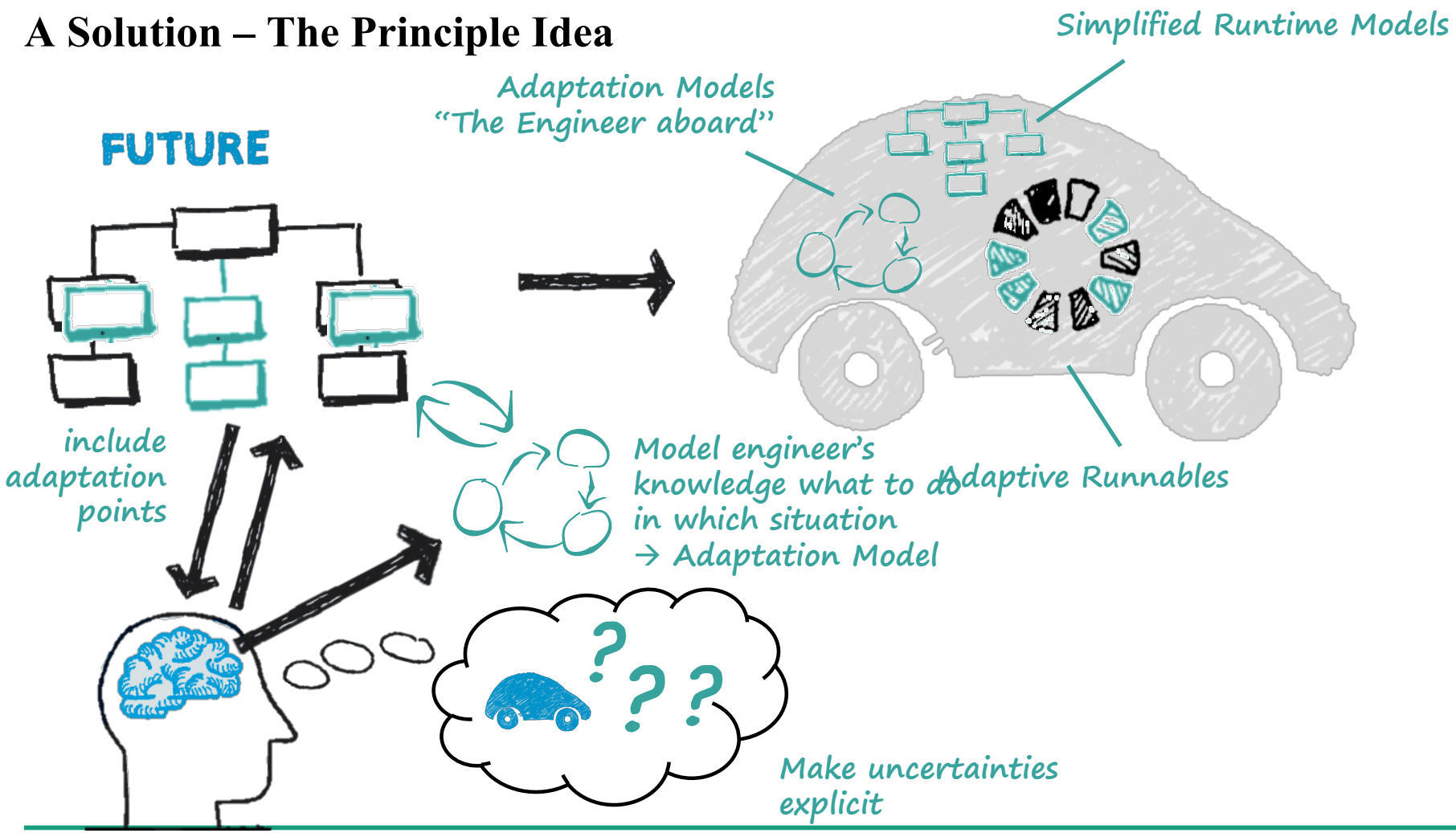
Openness – systems need to be open to exchange information and cooperate

The Challenge – From an Engineering Point of View

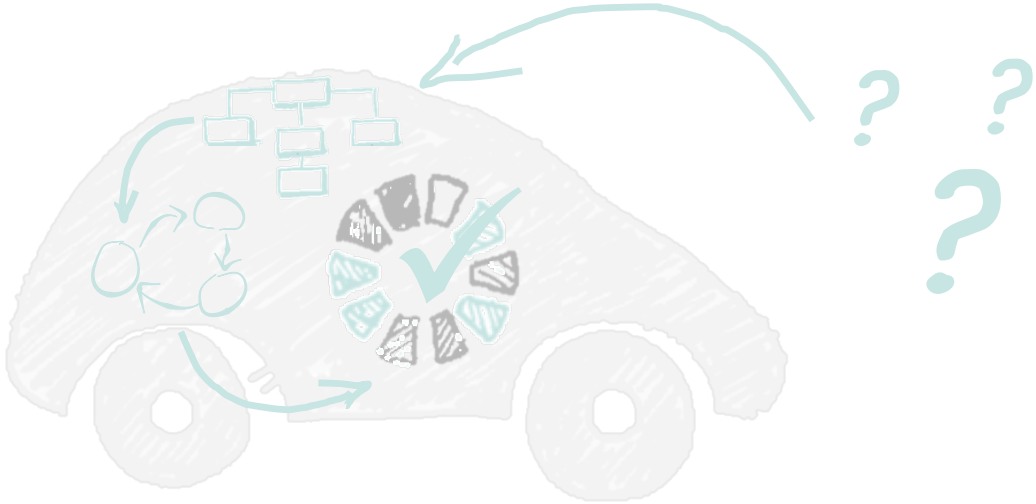


→ Static optimizations must be based upon conservative worst case assumptions, probably to the detriment of the system performance

A Solution – The Principle Idea



A Solution – The Principle Idea



1. Monitor context
2. Reflect context and state in runtime models
3. "Artifical Onboard Dependability Engineer" decides what needs to be done in a concrete situation based on runtime models and adaptation models (assurability)
4. Adapt System (fault-tolerance, graceful degradation, fail-operational)

➔ Dynamic Dependability Management

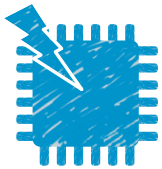
➔ Resilient Architecture

Dependability Awareness

Information

External Context

Check Results

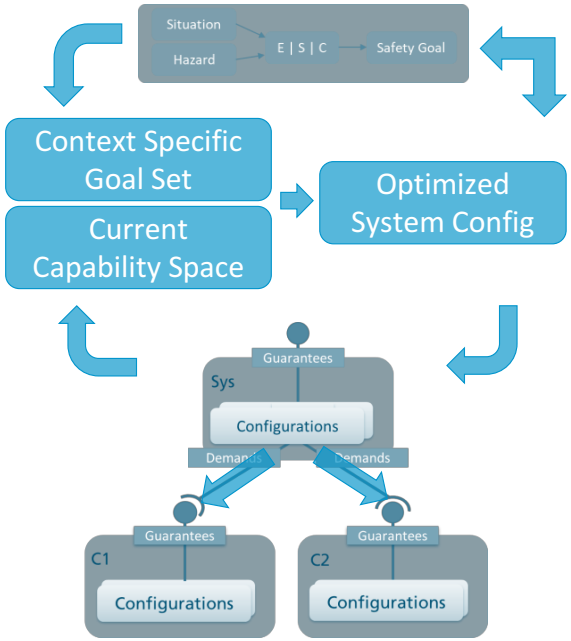


Internal Errors

Knowledge

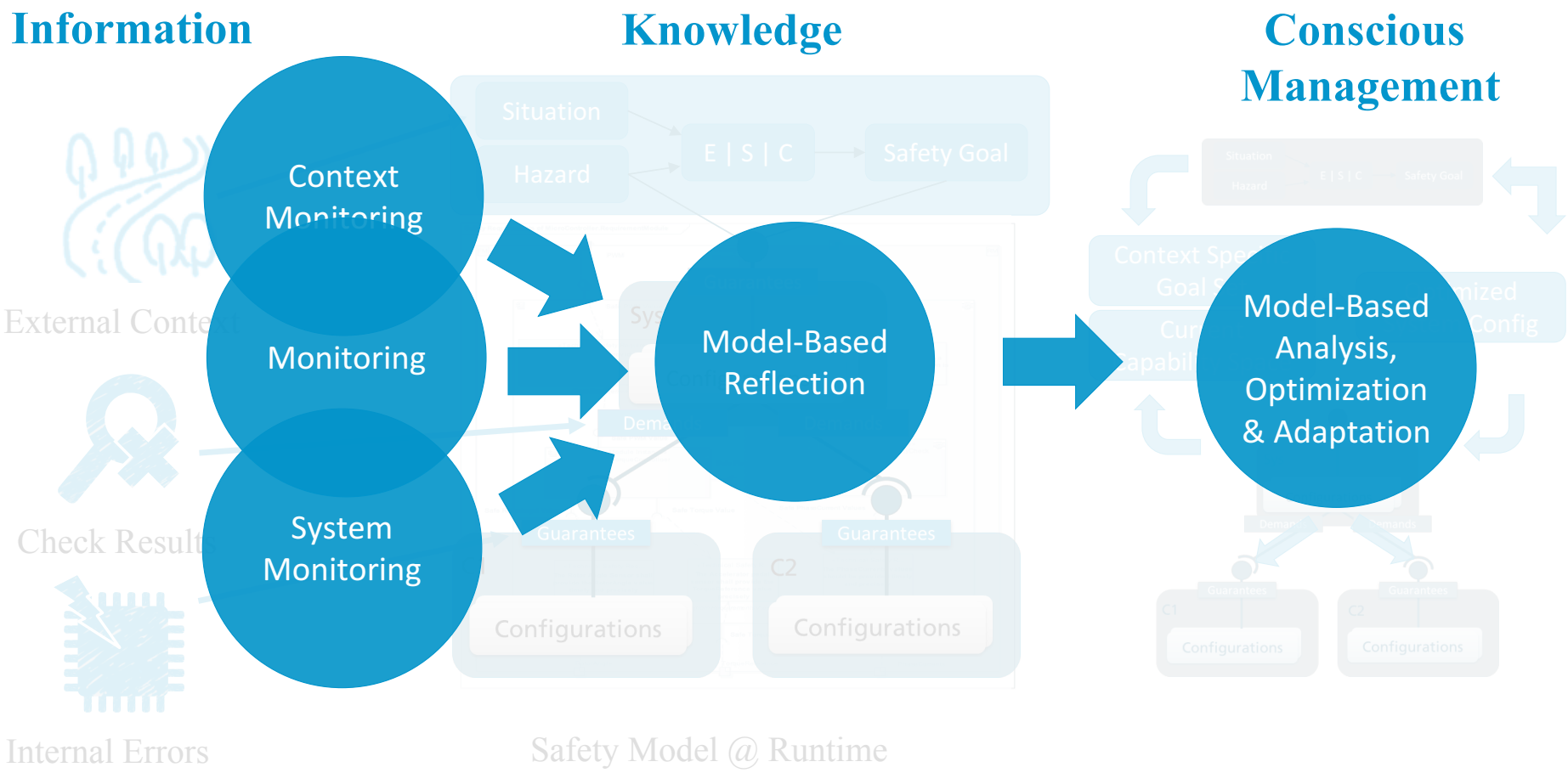
Safety Model @ Runtime

Conscious Management



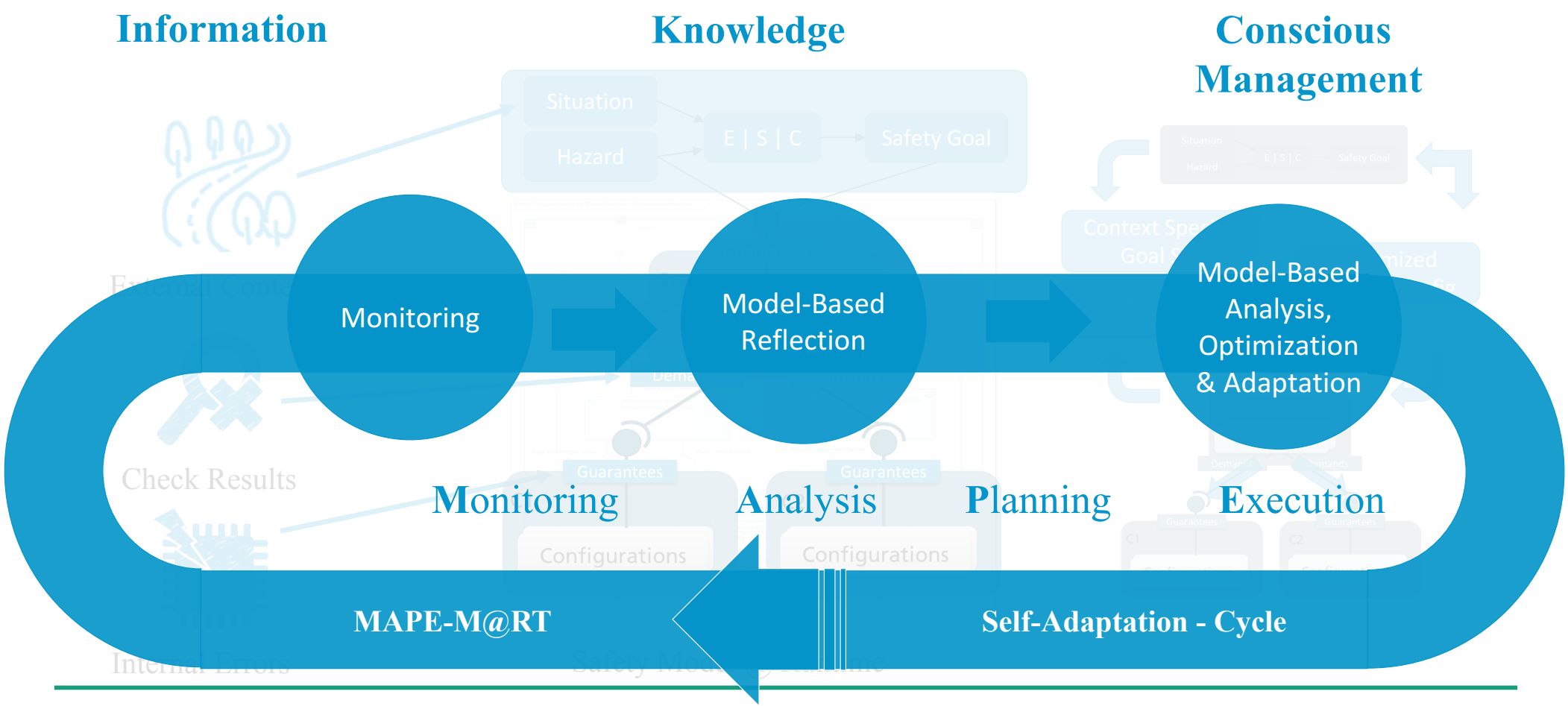
Fraunhofer

Dependability Awareness



Fraunhofer

Dependability Awareness



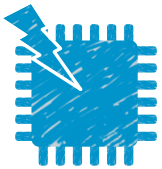
Fraunhofer

How to: (runtime) models for dependability management

Information

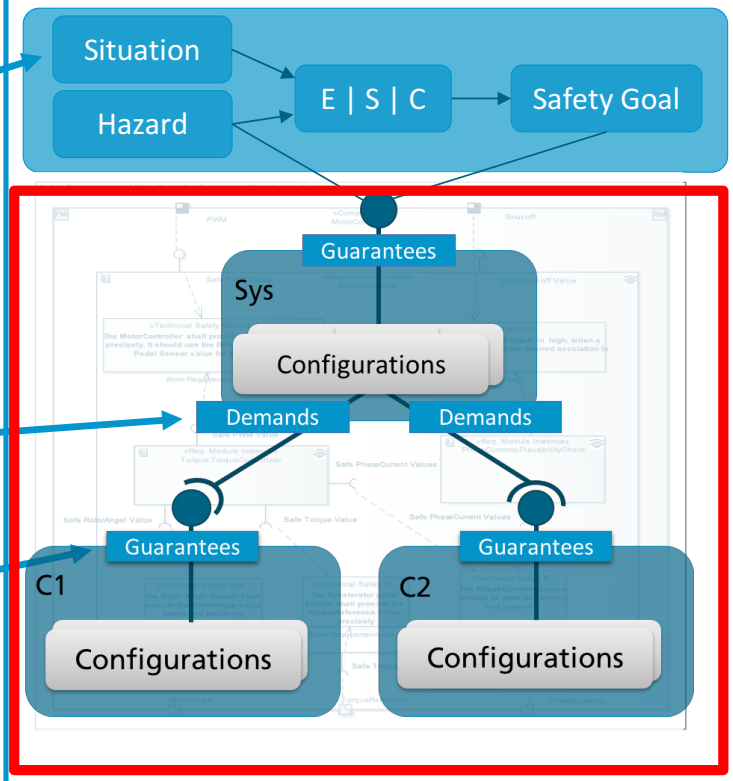
External Context

Check Results



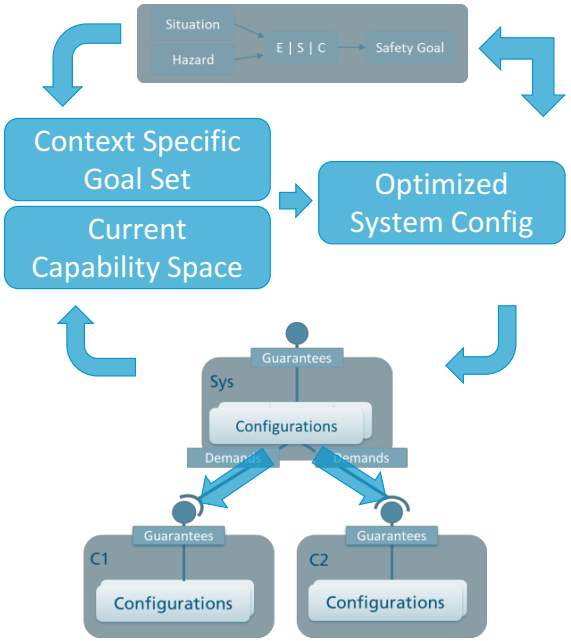
Internal Errors

Knowledge



Safety Model @ Runtime

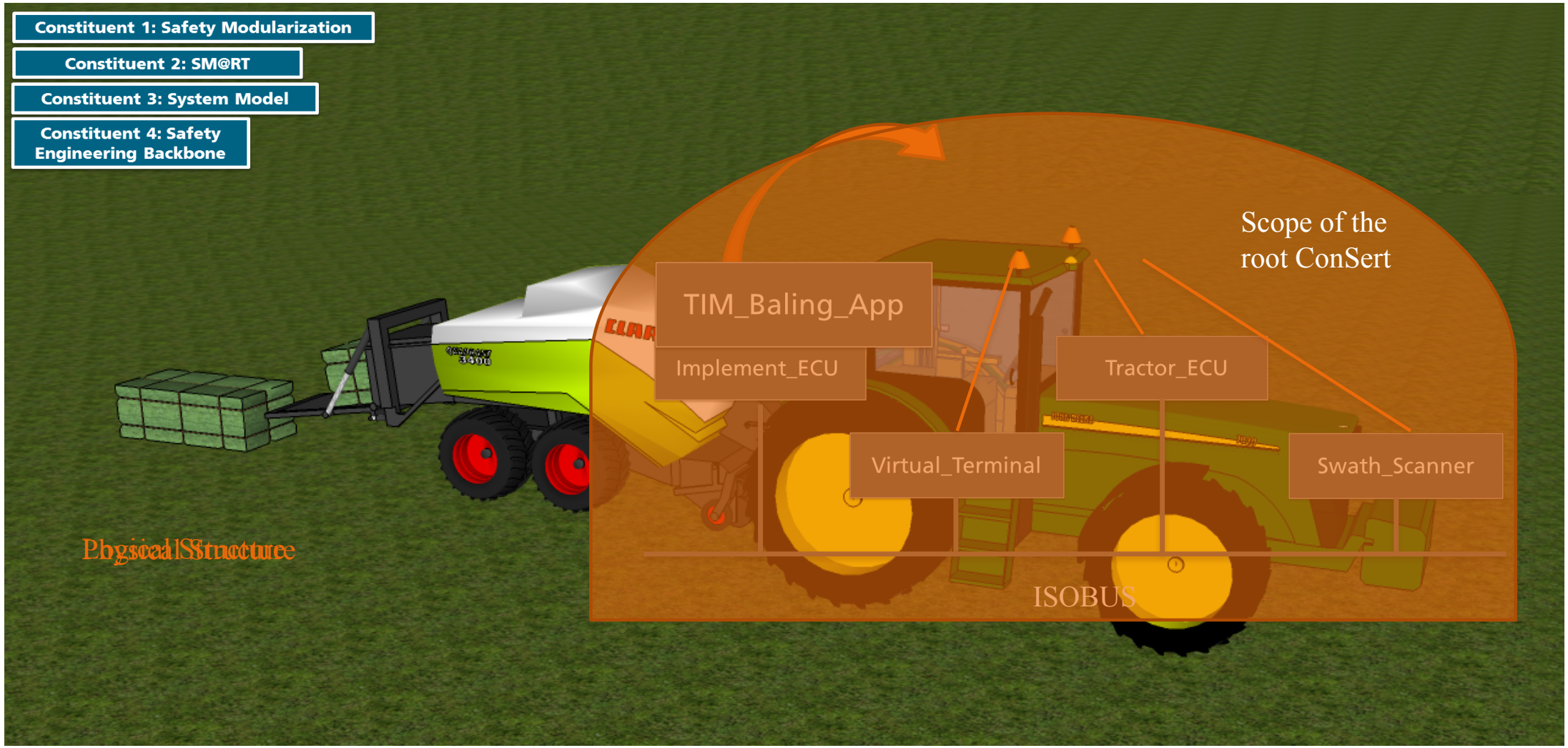
Conscious Management



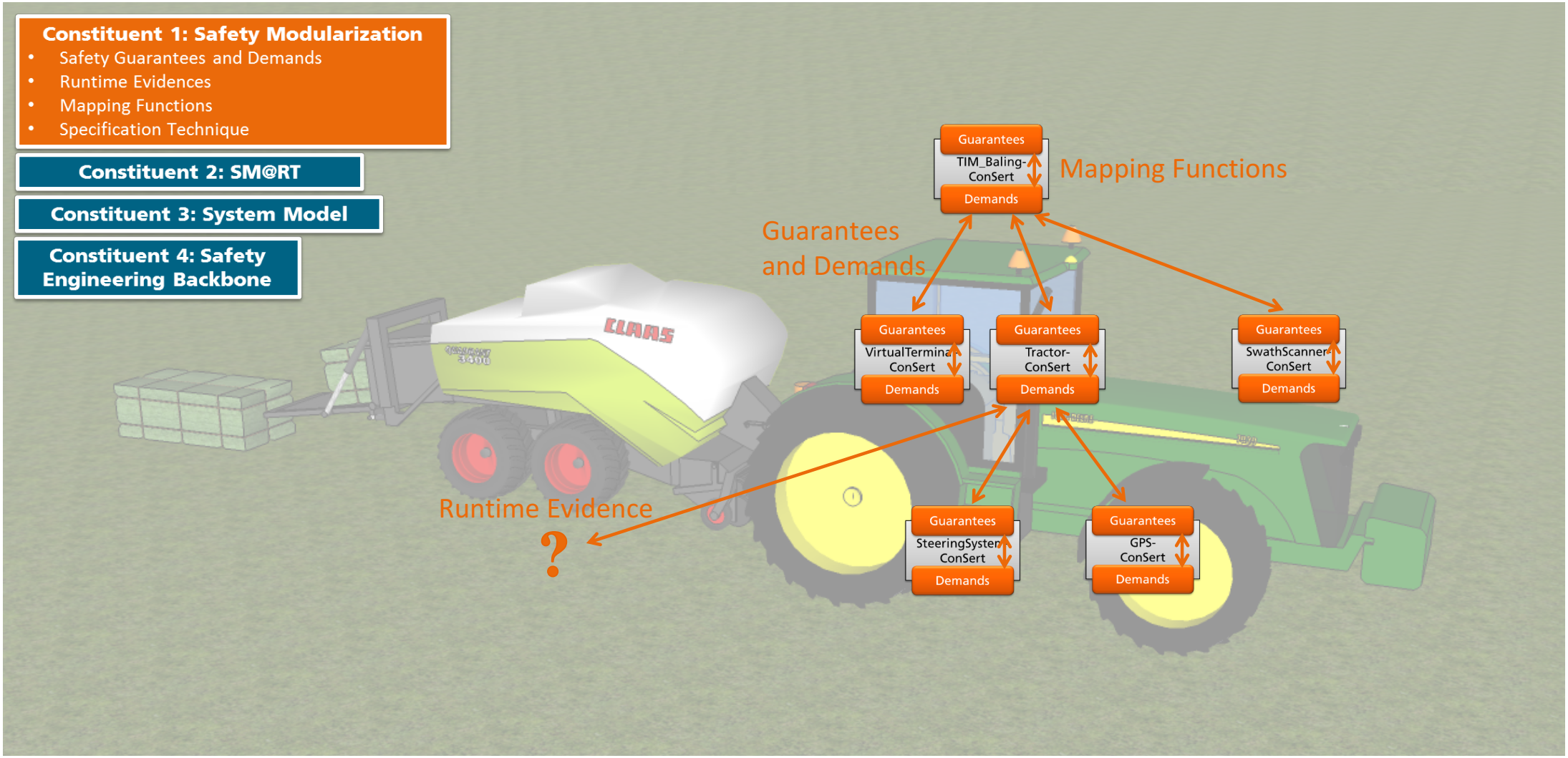
Fraunhofer

ConSerts Overview

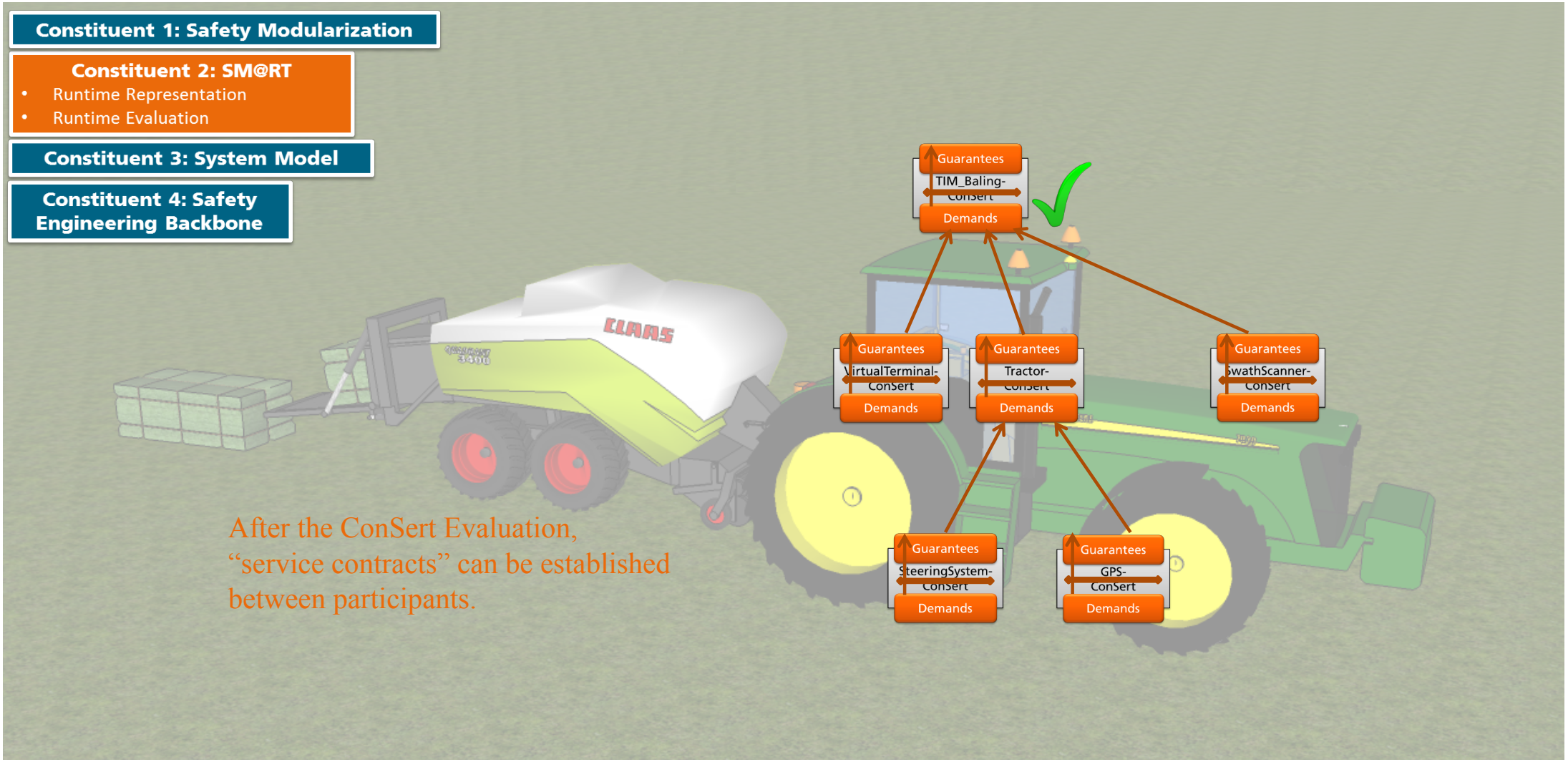
ConSerts Overview – Dynamic Hierarchies



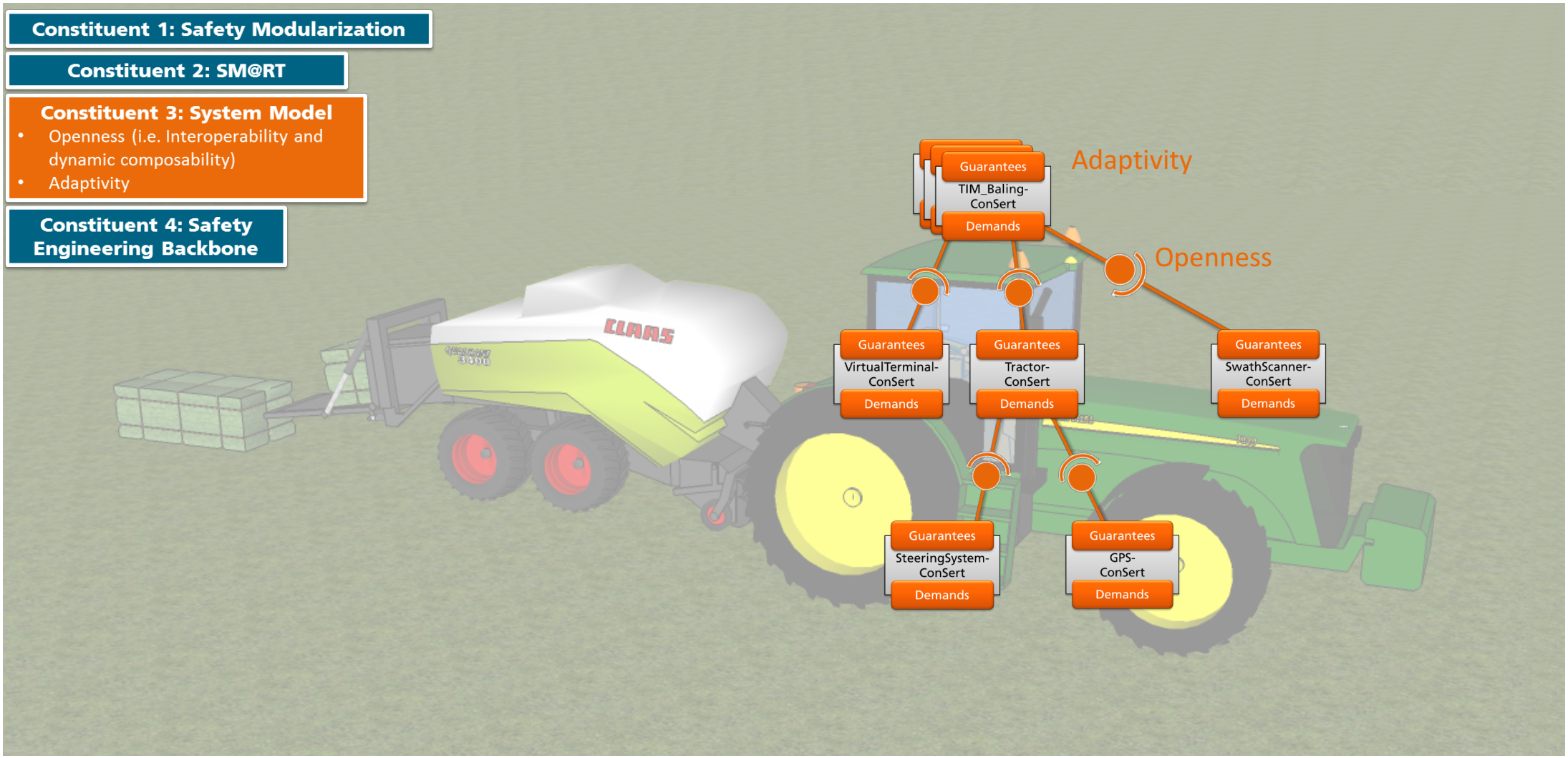
ConSerts Overview – Safety Modularization



ConSerts Overview – SM@RT



ConSerts Overview – System Model



ConSerts Overview – Engineering Backbone

Constituent 1: Safety Modularization

Constituent 2: SM@RT

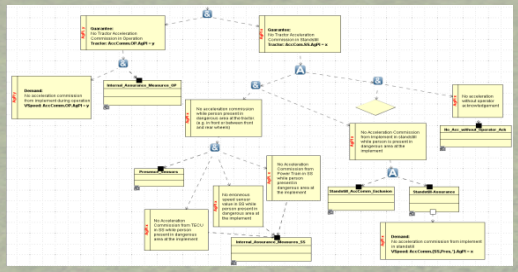
Constituent 3: System Model

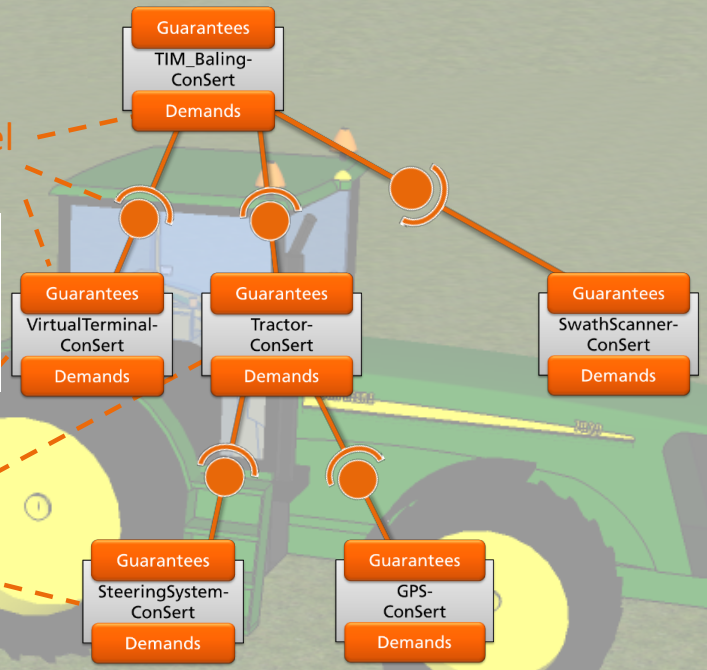
Constituent 4: Safety Engineering Backbone
- Domain-Level Engineering
- System-Level Engineering

Domain-Level Engineering

No.	Interface Function	Guideword	Deviation from correct function (effect that is associated with guideword)	Possible Causes	Possible consequences in the TIA baling scenario
1	**Auxiliary Valves – Valve state**	Omission	No hydraulic flow even though it would be required	Communication failure. Sensor failure. ECU (/SW) failure (tractor or implement).	Rear gate (of the round baler) does not open (or close) even though it must.
2		Commission	Hydraulic flow commissioned even though it is not wanted	Sensor failure. ECU (/SW) failure (tractor or implement). VT failure. "Third party" device issuing unwarranted request.	Rear gate does open (or close) even though it must not.

T6.2 System-Level Safety&Security Co-Engineering



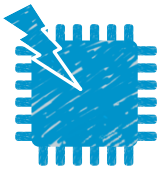


How to: (runtime) models for dependability management

Information

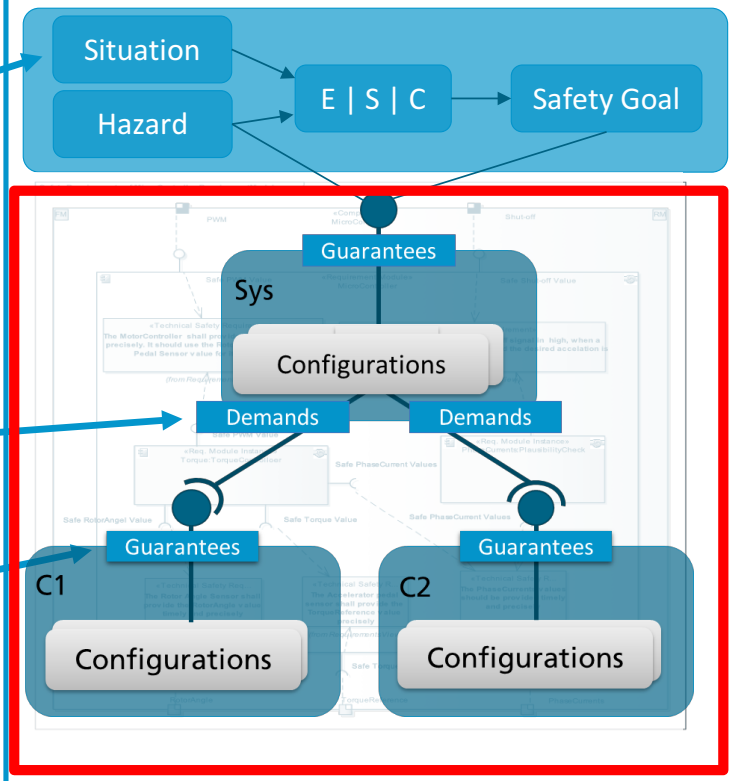
External Context

Check Results



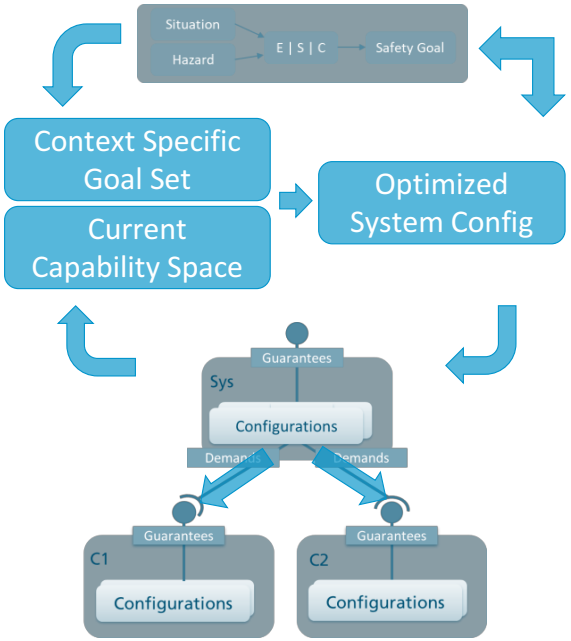
Internal Errors

Knowledge



Safety Model @ Runtime

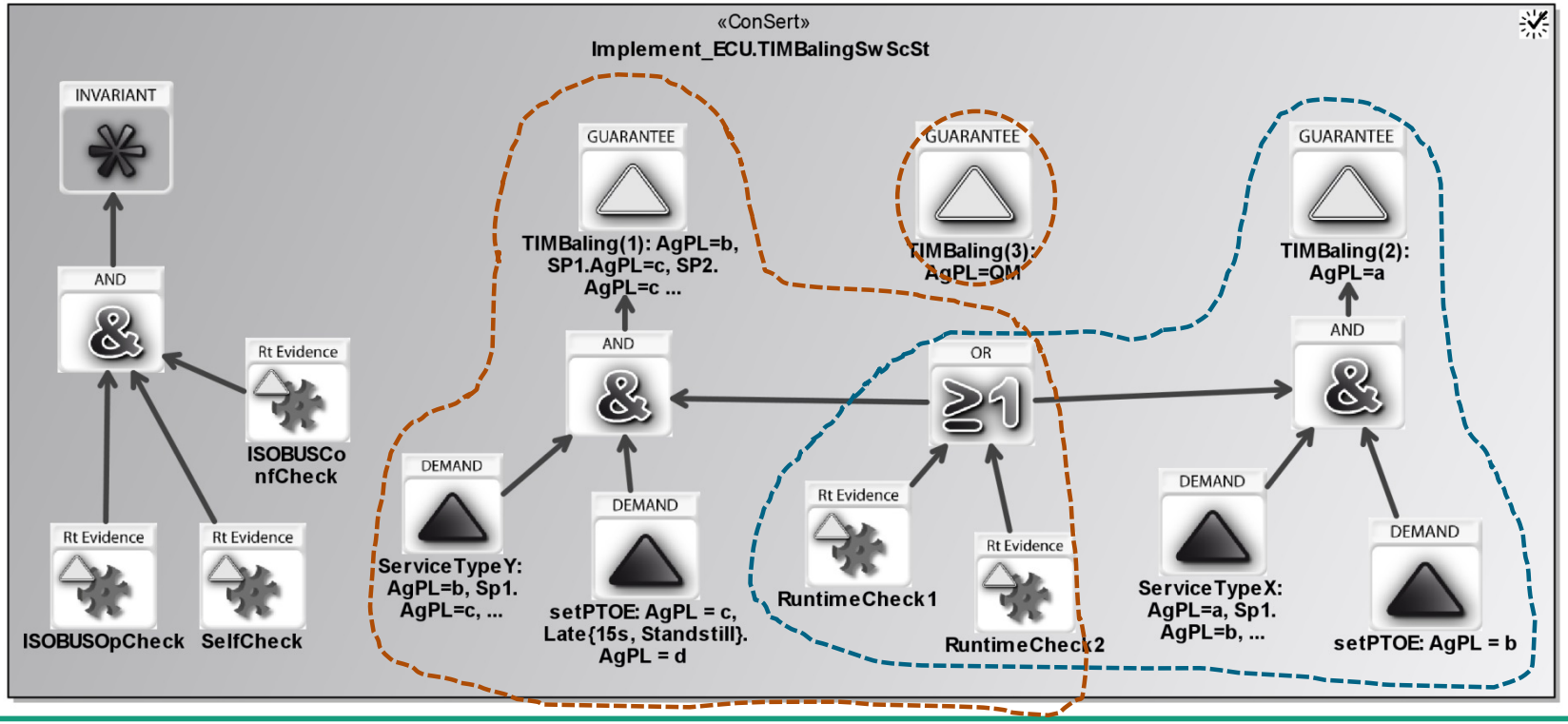
Conscious Management



Fraunhofer

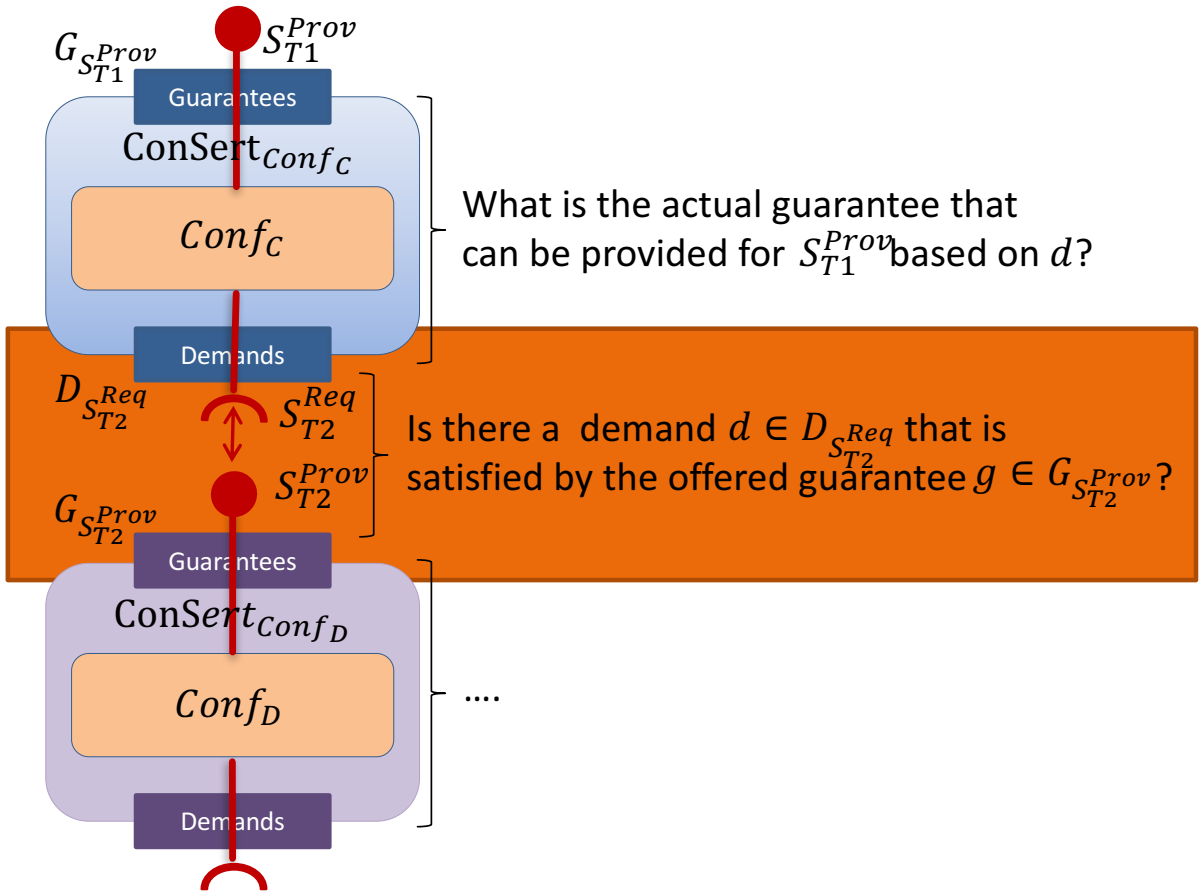
Specification of ConSerts

- For each potential safety guarantee of each provided service, there is a separate ConSert Tree (CST)



Fraunhofer

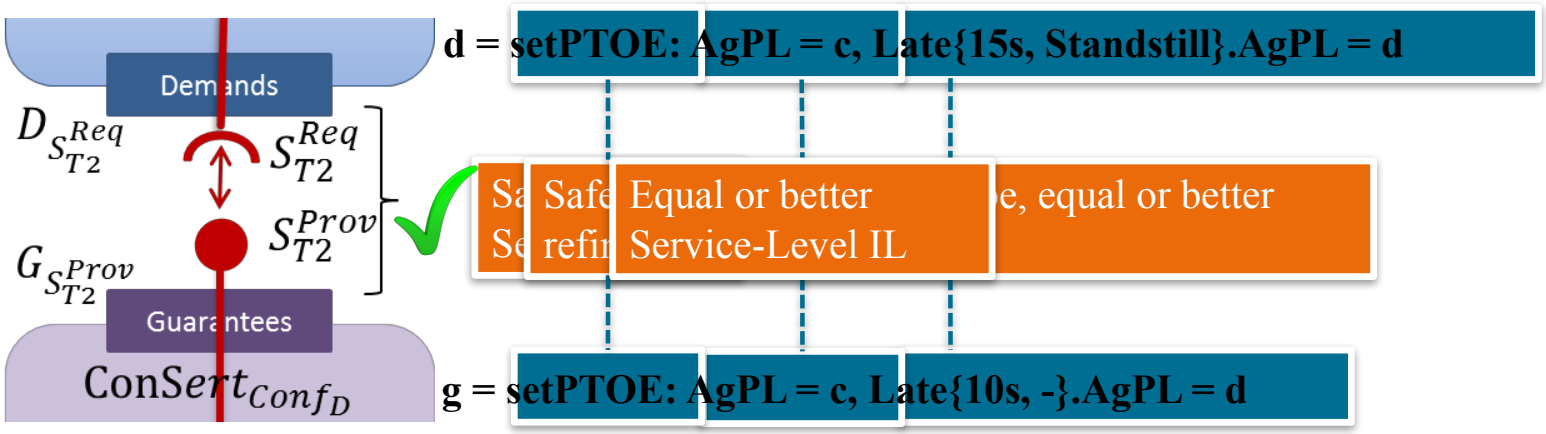
Matching and Mapping of Guarantees and Demands



Fraunhofer

Matching of Guarantees and Demands

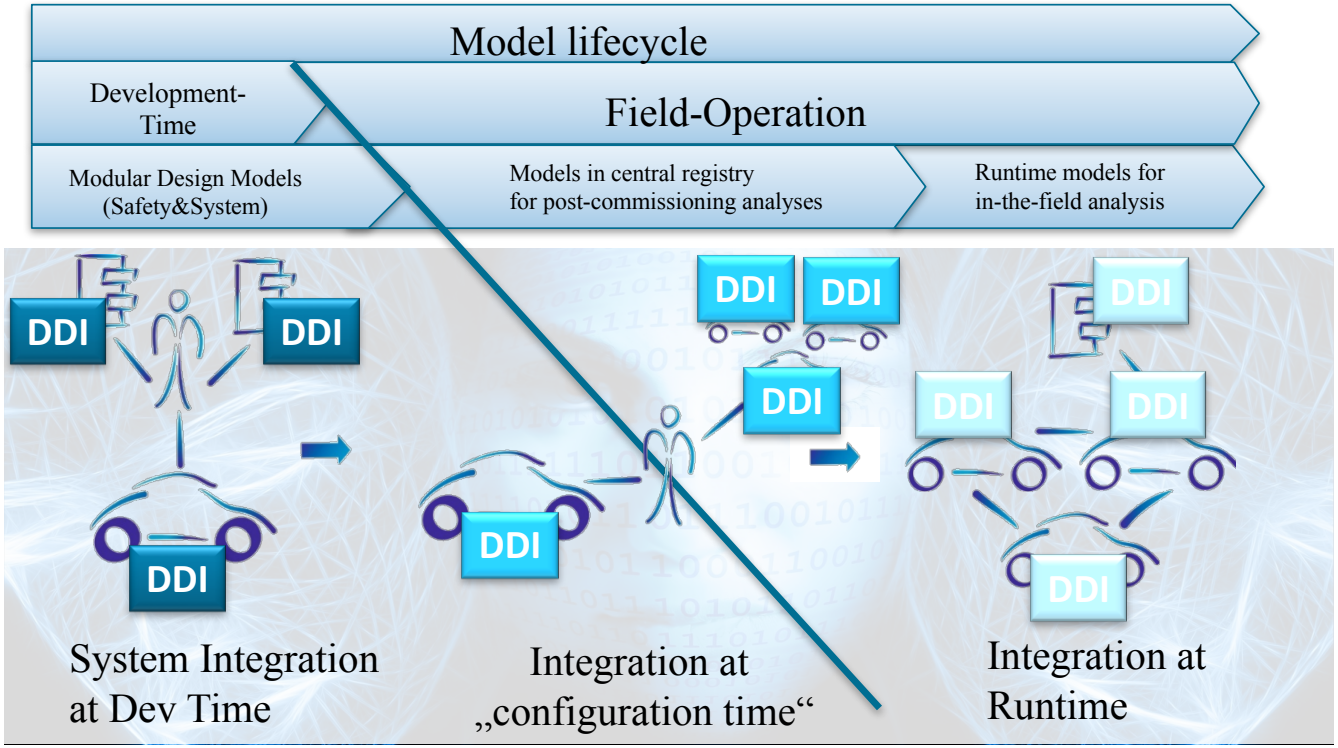
- When is a demand satisfied by a guarantee?



--- The DEIS Project ---

Fraunhofer

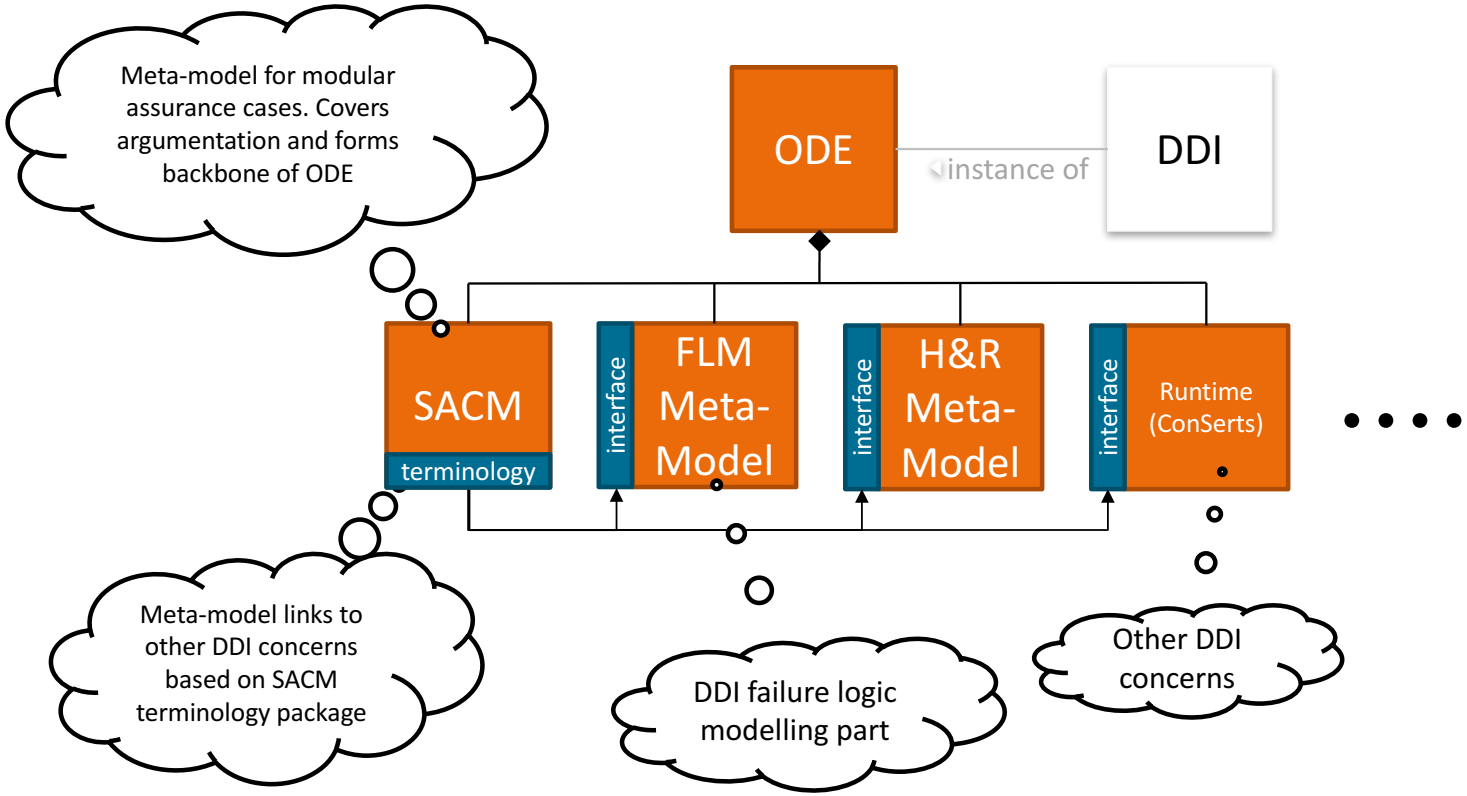
DEIS Project: Digital Dependability Identity



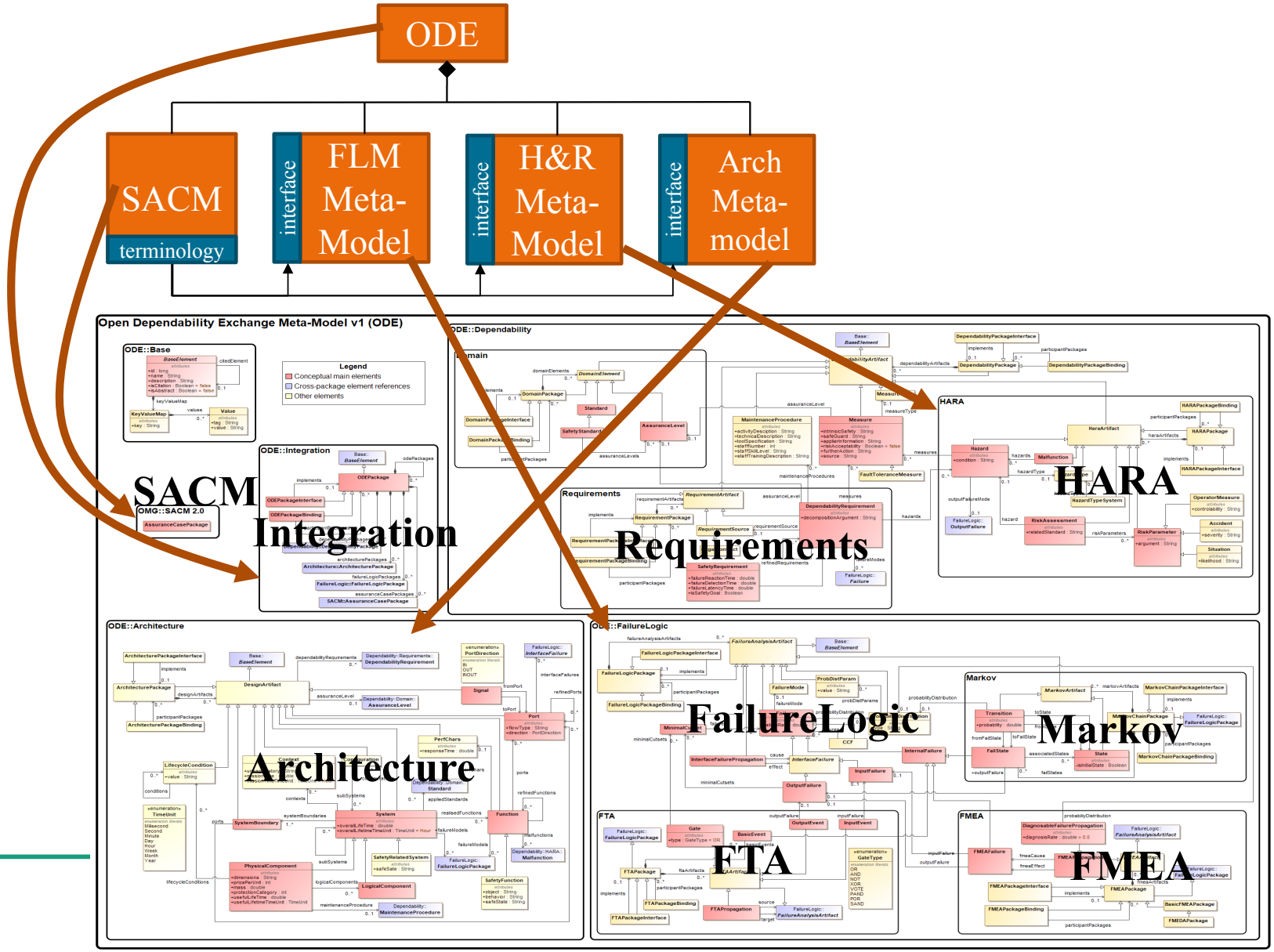
DDI DDI DDI Color encodes level of abstraction /amount of information

Fraunhofer

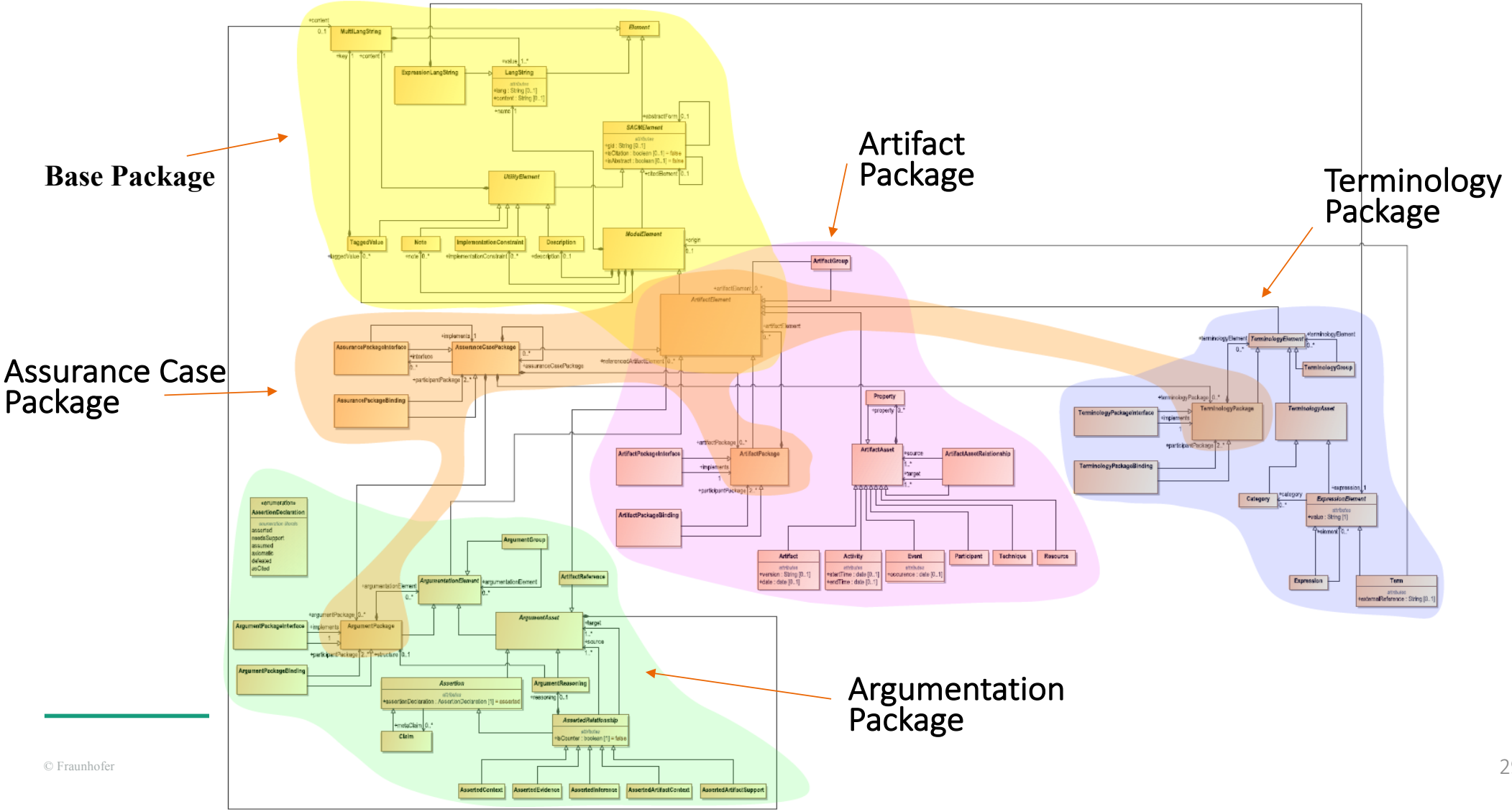
DDI Meta-Model (Open Dependability Exchange (ODE))



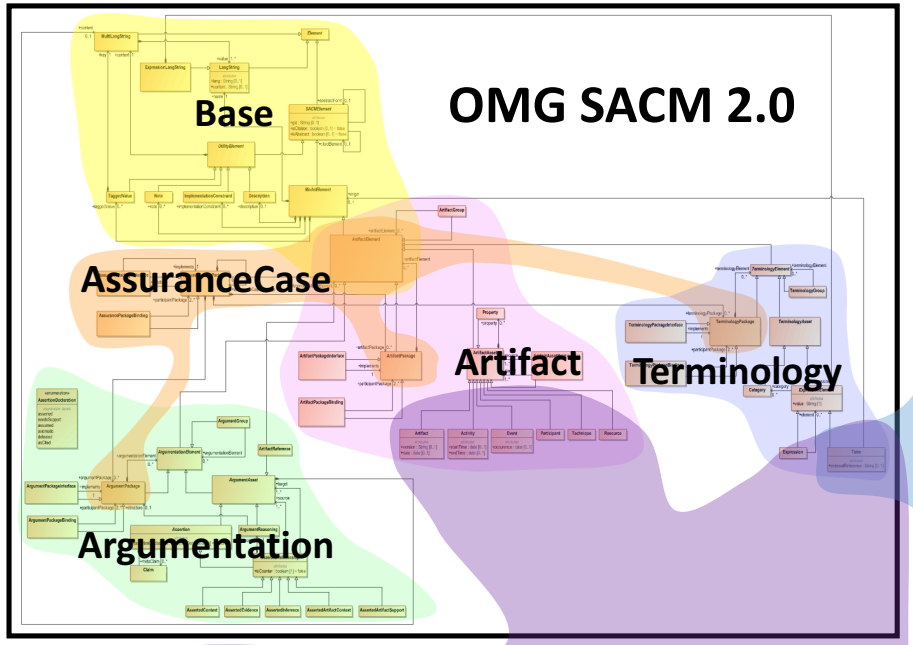
ODE v1



SACM 2.0 metamodel



ODE Meta-model v2



To which standards and processes is the argument refering to?

ODE::Certification

- Process
- Dependability Standard

OMG SPEM 2.0

Common Assurance and Certification Metamodel (CACM)
by OPENCOSS/ AMASS

ODE::Mapping

How can tailored certification activities and their inputs/output artifacts be mapped to tailored dependability models?

ODE::Product

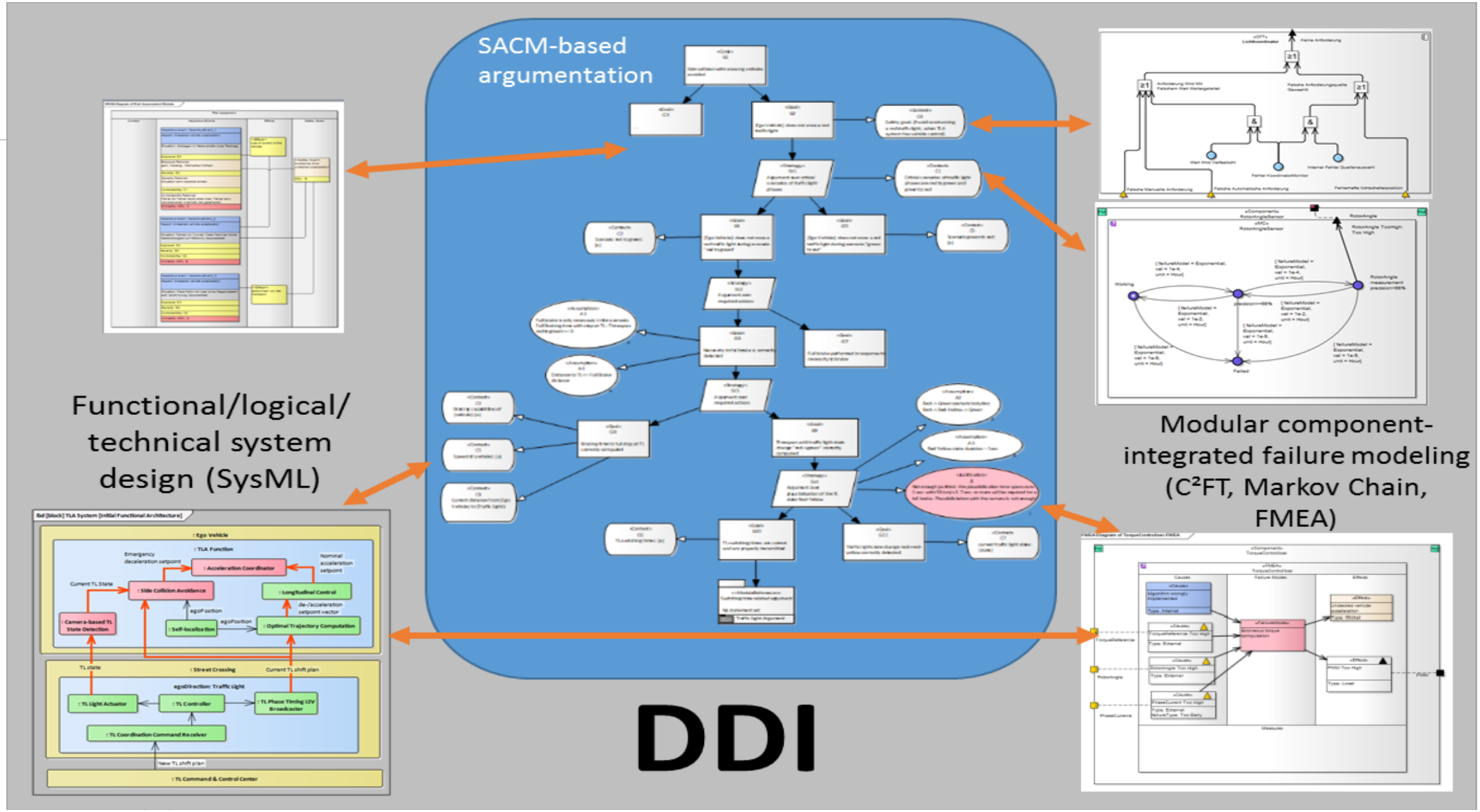
- Design
- FailureLogic
- HARA
- TARA
- AttackTree
- Measure

EAST-ADL

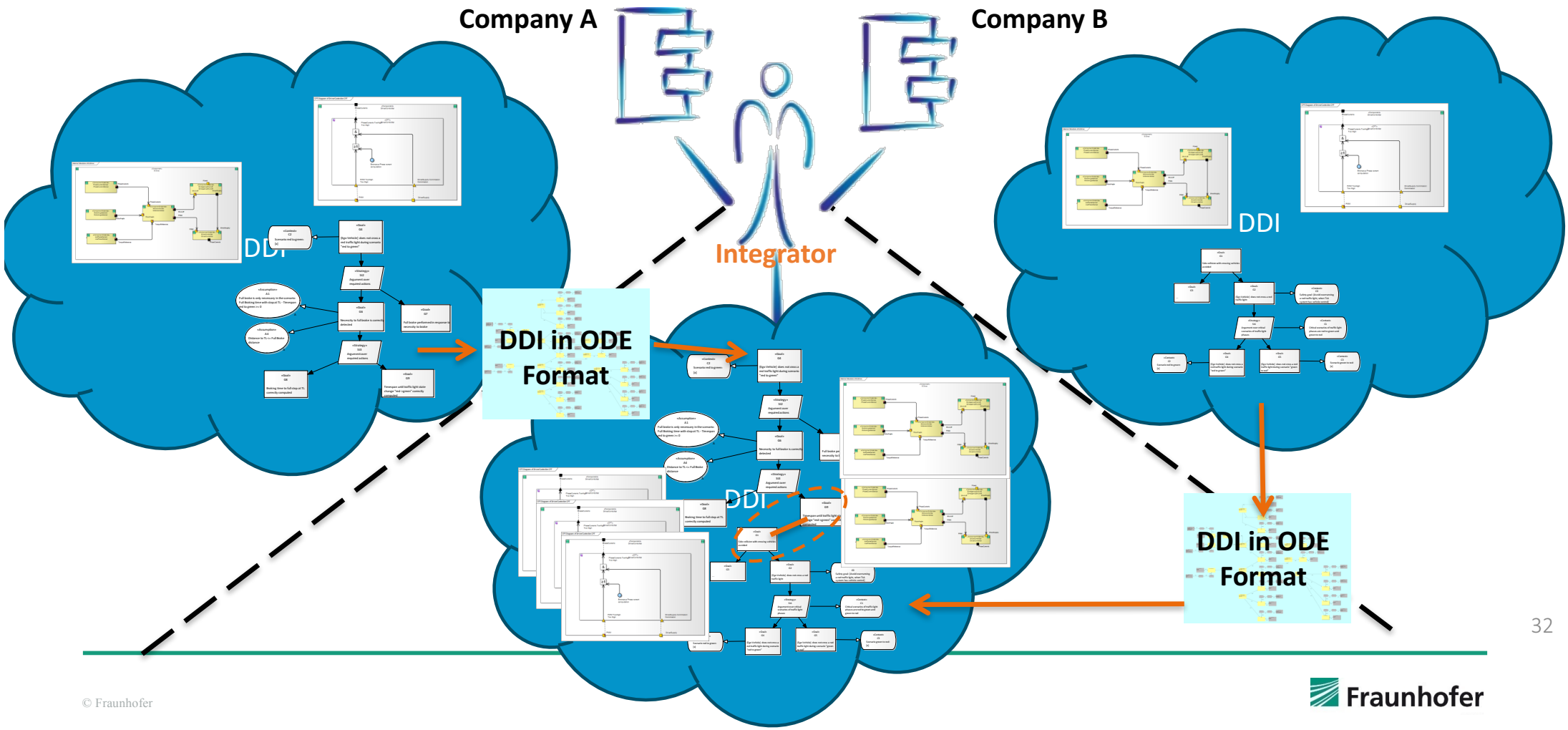
SafeML 1.1

Open Safety Meta-Model

How are dependability models that are being used as evidence or referenced in the argument, interrelated?


SACM-based argumentation
Functional/logical/ technical system design (SysML)
Modular component-integrated failure modeling (C²FT, Markov Chain, FMEA)
DDI

Development Time DDI Use Case



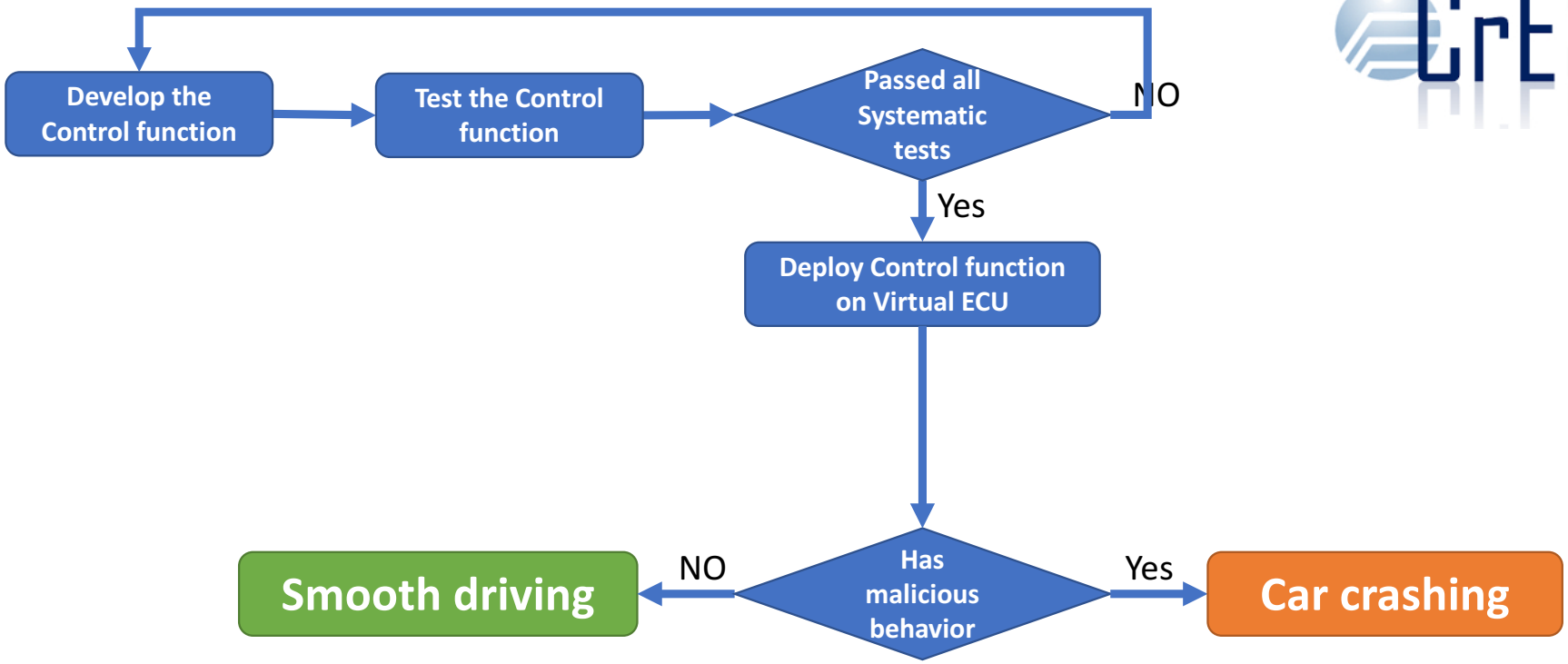
Fraunhofer

--- (Do Not) Trust in Ecosystems---

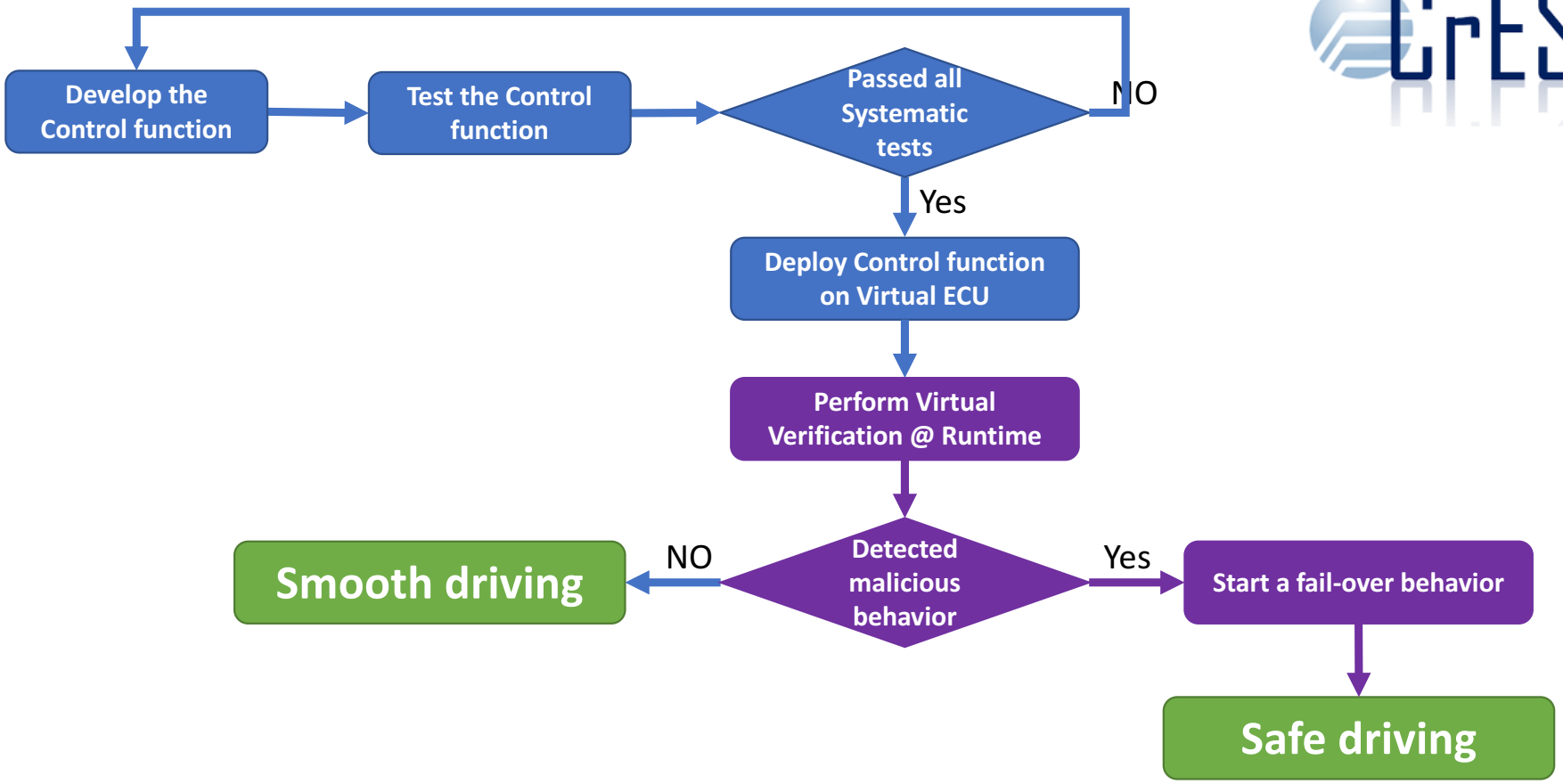
Fraunhofer

TOWARDS BUILDING TRUST

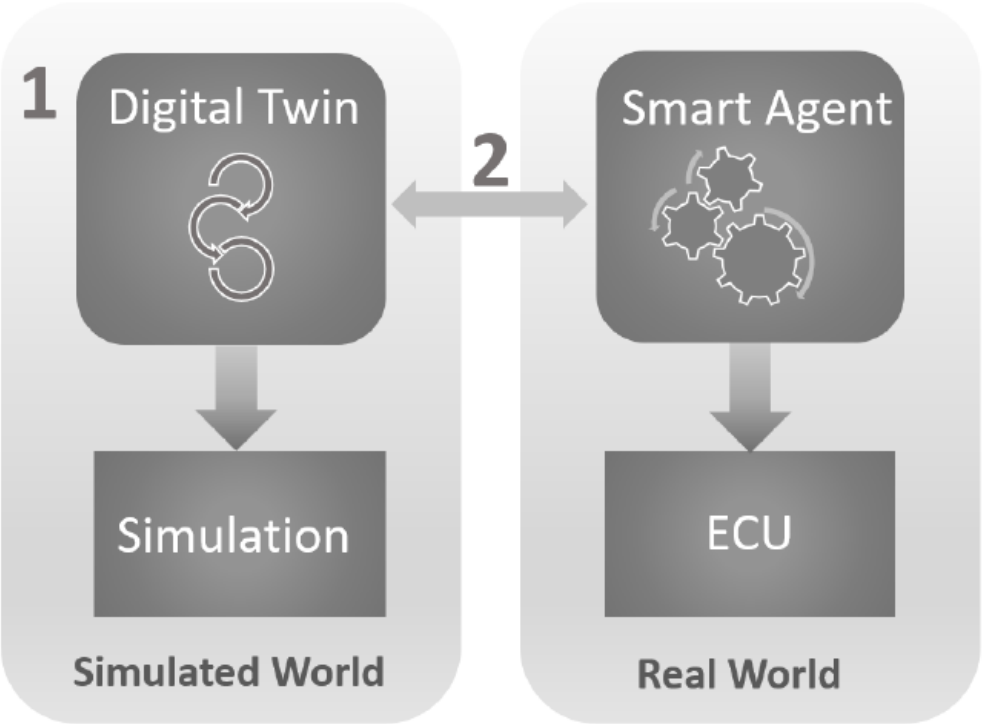
- Ecosystems are not formed from scratch,
  - Download of Smart Agents
  - Verification of Smart Agents
    - Requires code execution on the ECU
- Evaluation of DT of the Control Algorithm
  - Build Reputation and Trust based on DT evaluation



- Agent contains malicious code that causes car to run into another car with low probability.
- Systematic testing is not enough, verification through multiple execution of behavior is needed.

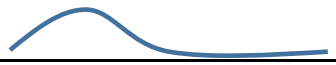
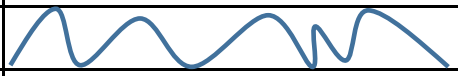
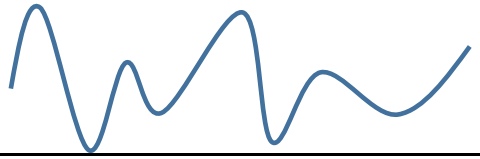
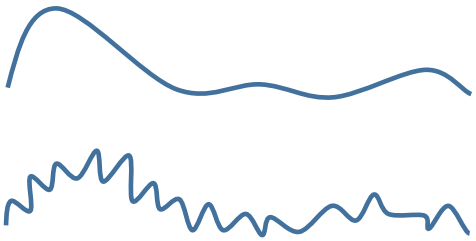

Develop the Control function
Test the Control function
Passed all Systematic tests
NO
Yes
Deploy Control function on Virtual ECU
Perform Virtual Verification @ Runtime
Detected malicious behavior
NO
Smooth driving
Yes
Start a fail-over behavior
Safe driving

METHOD FOR BUILDING TRUST



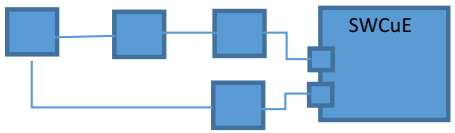
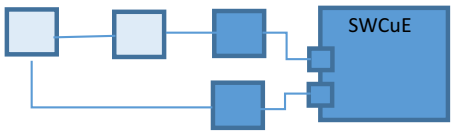
Fraunhofer

Technical Challanges

Input values linked to	Source of problem	Real World	Virtual World
Sensor Data	1 .Frequency		
	2. Value Range		
Can Bus Signals	3. Noise		

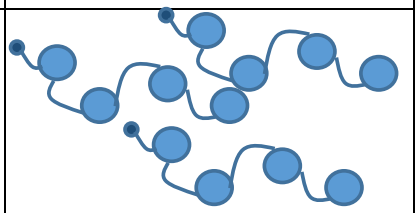
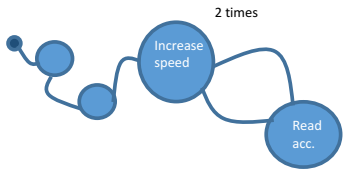
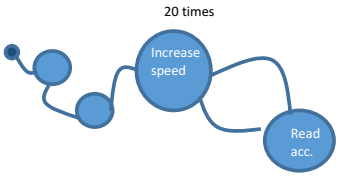
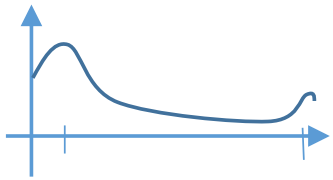
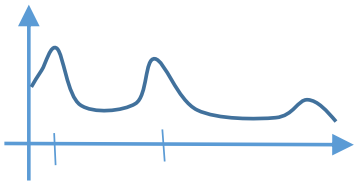
Fraunhofer

Technical Challanges

Deployment of SWCuE **linked to**	**Source of problem**	**Real World**	**Virtual World**
System Architecture	4. Abstraction Problem		

Fraunhofer

Technical Challanges

Behavior of SWCuE **linked to**	**Source of problem**	**Real World**	**Virtual World**
Software Behavior	5. Retain state		
			
	6. Monitor the passing of time		
	7. Observe the passing of time	time is passing linearly 	time is not passing linearly 

Fraunhofer

BUILD TRUST IN ECOSYSTEMS AN COMPONENTS

- Concept for Platform that enables building of trust :
  - Trust the Virtual Evaluation.
    - **Output Proccessor:**

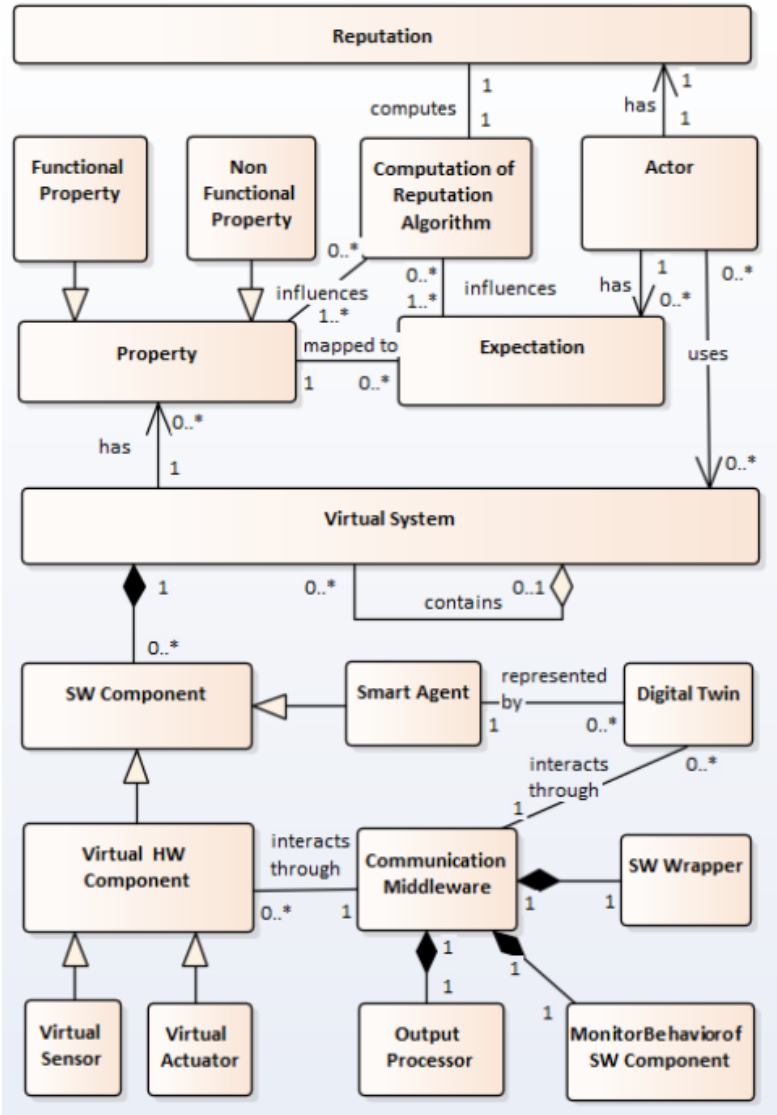
    outputs {frequency, value range of data, noise signals}
    - **Monitor** :

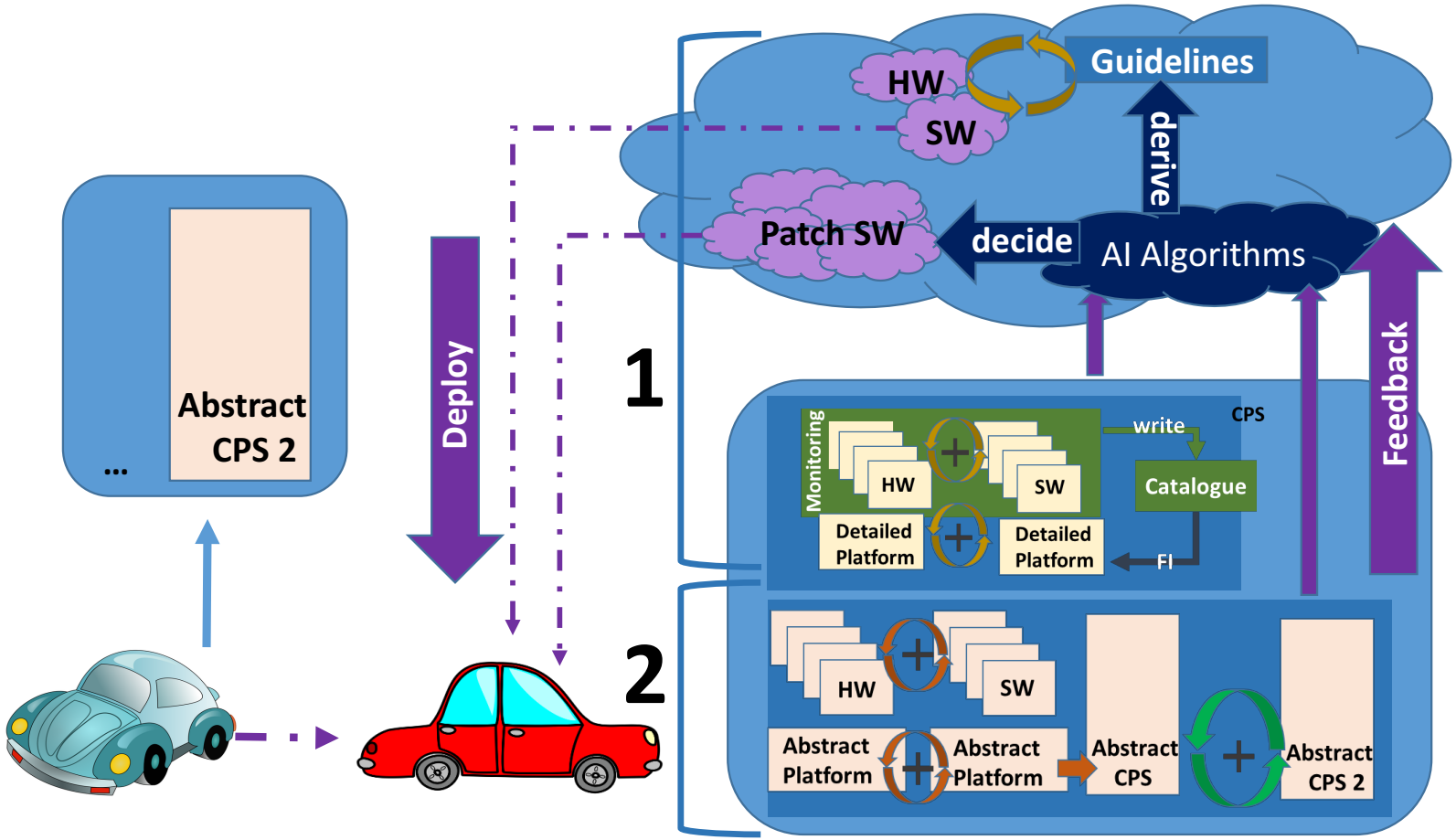
    detect suspicious interaction patterns.
    - **The Software Wrapper**

    assures that the behavior of software component

    is not able to retain the state,

    neither to monitor or observe the passing of time.
  - Components that enables computation of reputation.




Guidelines
HW
SW
derive
Patch SW
decide
AI Algorithms
Feedback
Deploy
Abstract
CPS 2
...
1
2
Monitoring
HW
SW
write
CPS
Catalogue
Detailed
Platform
Detailed
Platform
FI
HW
SW
Abstract
Platform
Abstract
Platform
Abstract
CPS
Abstract
CPS 2

Thank you for your interest

Contact:

Dr. Daniel Schneider
daniel.schneider@iese.fraunhofer.de
Tel.: +49 (0) 631 / 6800-2187
Fax.: +49 (0) 631 / 6800-9-2187
Mobile: +49 (0) 151 / 649 530 70