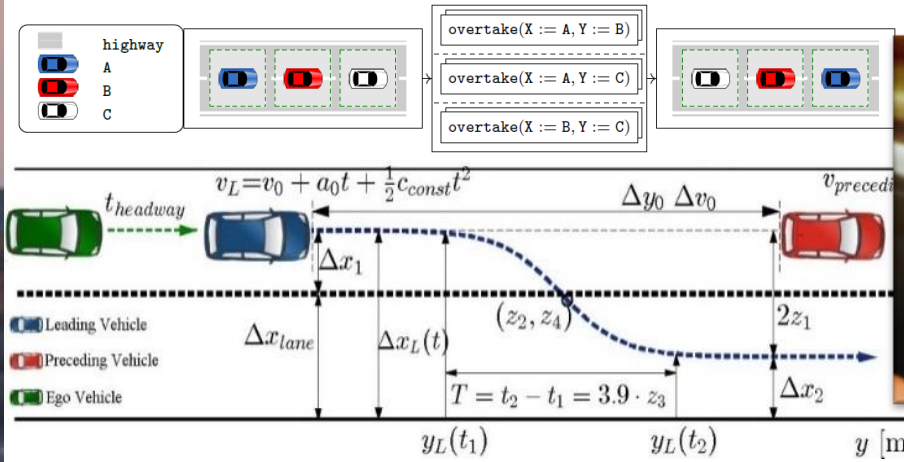
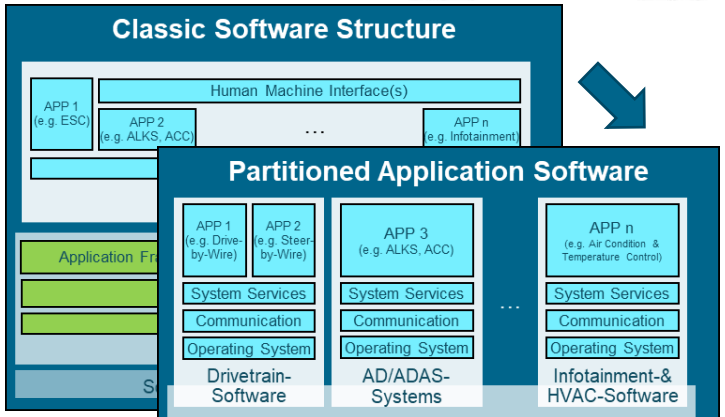
CBD4SDV

Contract Based Design for Software Defined Vehicles

Gregor Nitsche, Astrid Rakow, Ingo Stierand (DLR) & Janis Kröger (UOL)

DLR is (more than) Avionics and Rocket Science:
… we are also doing AD & SDV





Lessons Learned

background from & coop. w.

OFFIS

Carl von Ossietzky Universität Oldenburg

MULTIC Tooling

UP2DATE

TRANSACT

GAIA-X 4 AGEDA

Step-UP!CPS

V&V4NGC
Methoden, Prozesse und Werkzeugketten für die Validierung & Verifikation von Next Generation Car

DLR

Safety cannot be guaranteed by functional correctness alone.

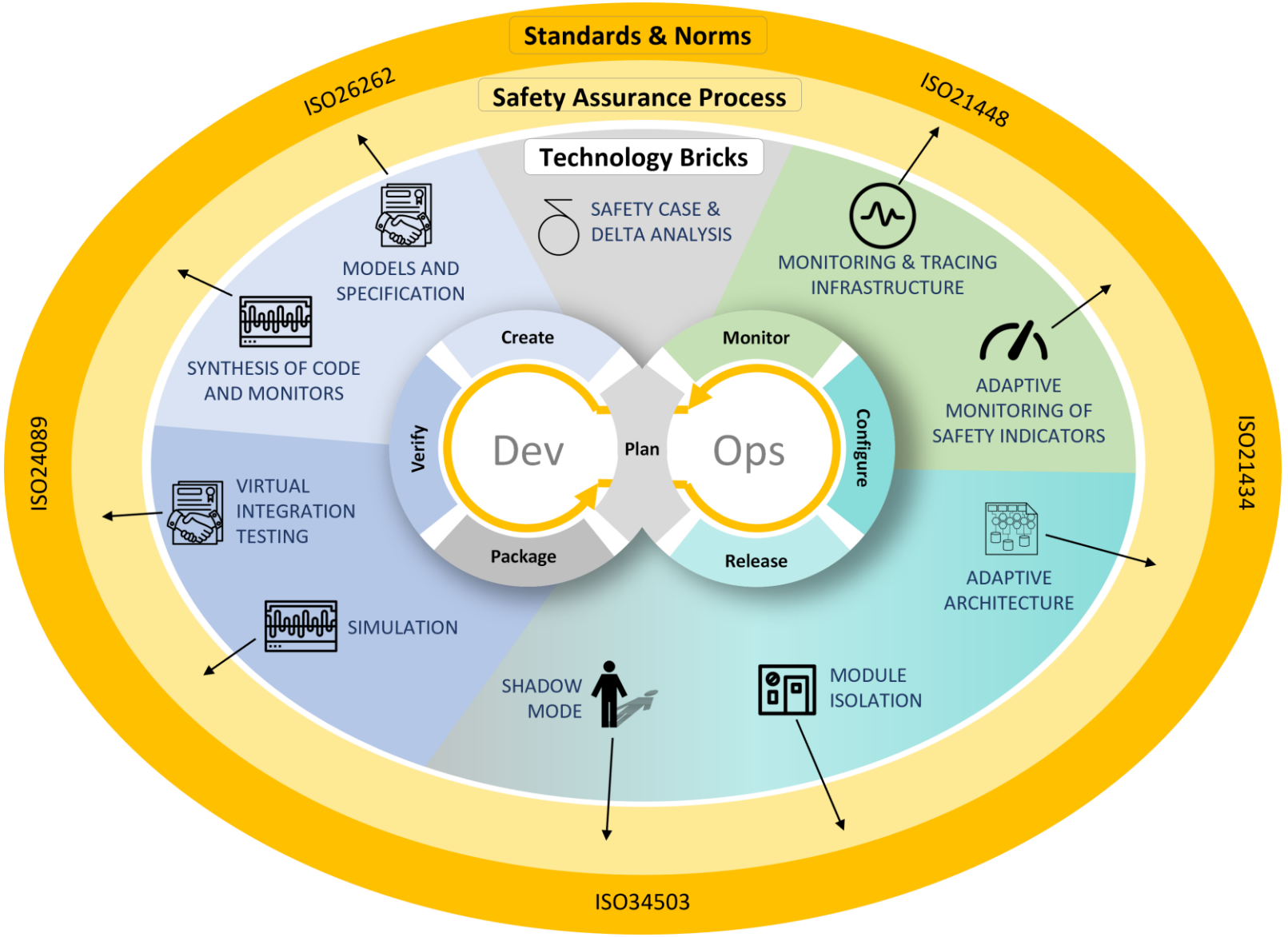
Contracts have successfully been used to specify various system properties:

- Fail-safe operation modes
- Updateability, Phases of update process
- Services of legacy system and cloud-edge extension
- Functional, quantitative and contextual requirements

→ Contracts allow **to decouple** the design and structure the design process
  → decomposition of **components, context and safety arguments**

→ (Semi-formal) contracts can **capture & relate multiple dimensions** of requirements
  → better **traceable, processable & analyzable**

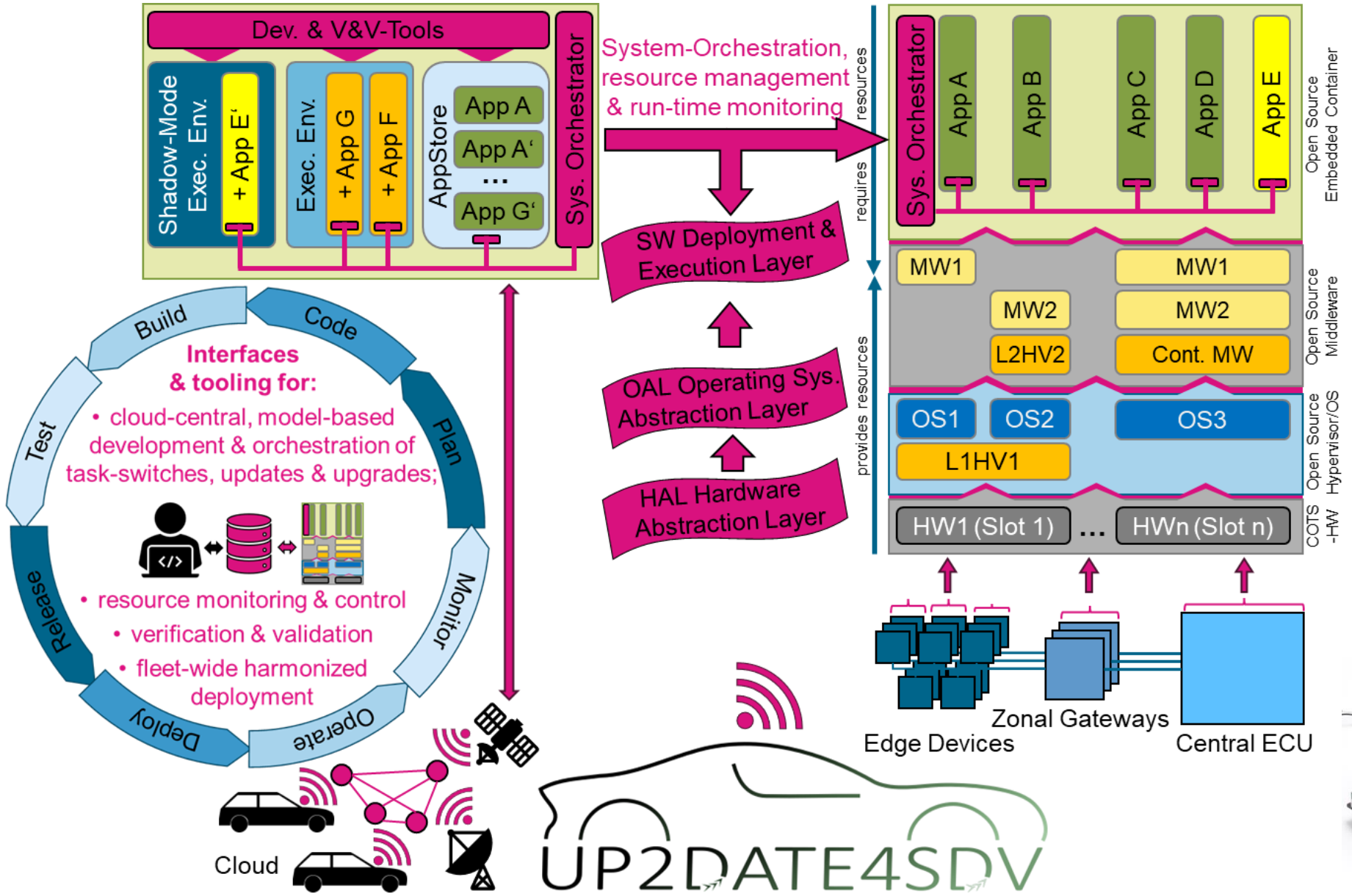
Contracts Linking System Design and Safety Argumentation



Methods and tools for DevOps:

- → specifying multi-dimensional (semi-formal) contracts
- → managing their evolution
- → linking contracts to assurance evidence

Current Work



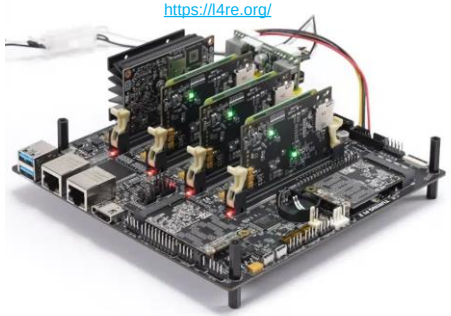
https://carla.org/img/carla.jpg

https://picx.zhimg.com/v2-65dc3d9fef7f05f5c4cec2bb1e4b52ea_720w.jpg?source=172ae18b

https://docs.ros.org/en/foxy/_static/favicon.ico

L4Re

https://l4re.org/



https://www.notebookcheck.net/fileadmin/Notebooks/News/_nc4/turingpi25.jpg

CBD4SDV - Conclusion & Vision

CBD

- supports horizontal and vertical compositional reasoning
  → key enabler for hierarchical designs and design-processes

4SDV we want to further develop

- developer-friendly domain-specific syntax & semantics
- domain specific tayloring of theories & tools for industrial applications
- multidimensional links between all (evolutionary) phases of the DevOps-cycle
- tools for the specification, tracing, management, and monitoring of contracts
- blueprints for the application in modular safe & secure E/E-architectures

QUESTIONS AND DISCUSSION

Quelle: https://pixabay.com/de/photos/frage-fragen-fragezeichen-einzel-6387294/

Imprint

Topic:	CBD4SDV – Contract Based Design for Software-Defined Vehicles
Date:	2025-11-18
Author:	Gregor Nitsche Astrid Rakow, Ingo Stierand, Janis Kröger (UOL)
Institute:	Systems Engineering for Future Mobility (SE)
Image sources:	All images “DLR (CC BY-NC-ND 3.0)” unless otherwise stated

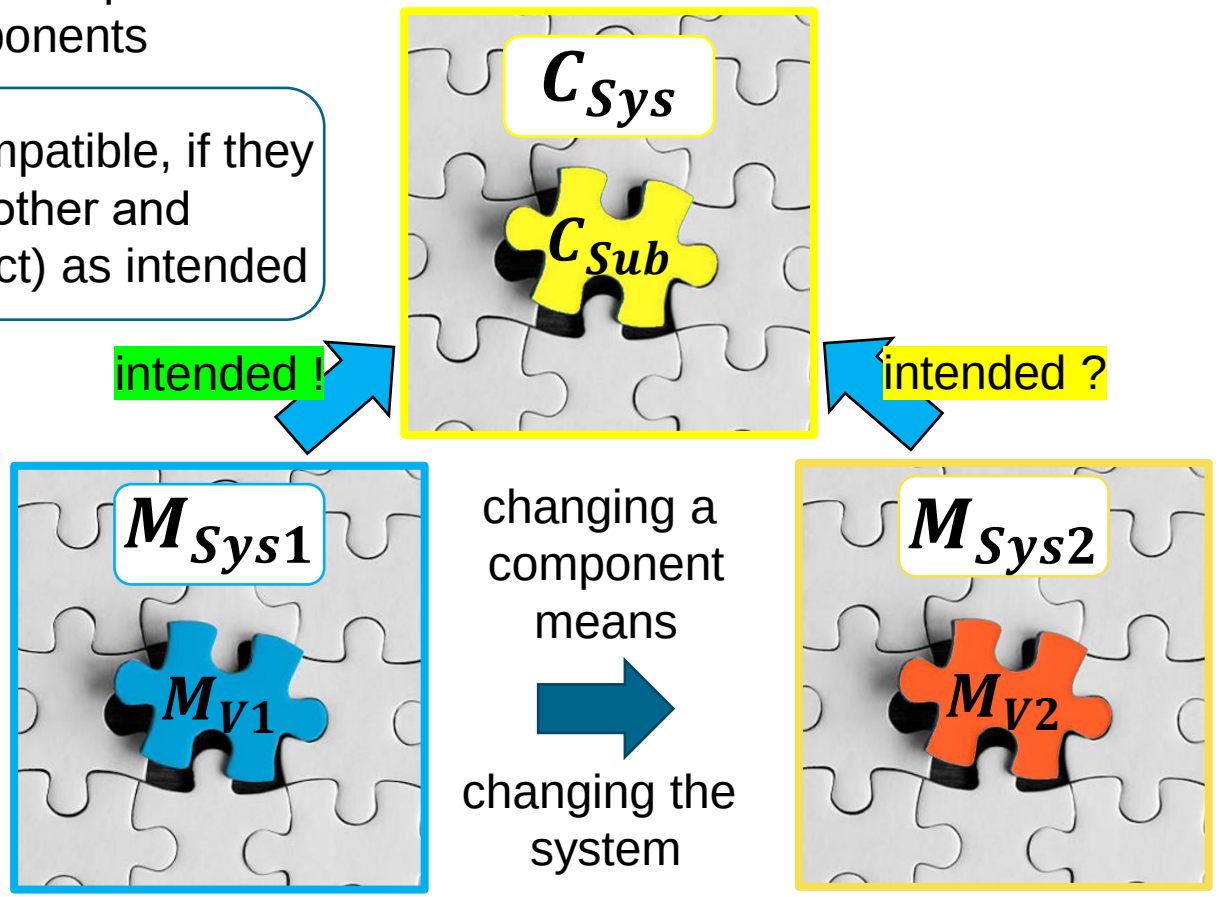
Contract-Based Compatibility (Modularity)

Compatibility is:
- based on components & composition
- a relation between components

Def: Components are compatible, if they
1. don‘t harm each other and
2. cooperate (interact) as intended

Components don‘t harm each other iff:
- a component‘s assumptions are not violated by its environment
- the component‘s guarantees don‘t lead to violated assumptions within its environment

→ Check satisfaction between assumptions and (composed) guarantees



→ Check refinement between a system-specification and different versions of system-compositions