Autonomous Cars 2015

“The Future of ADAS and Core Technologies for the Autonomous Car”

- From an ISO 26262 perspective

Dr. Hakan Sivencrona

ISO 26262 International expert 2007-2015

DELPHI

Innovation for the Real World

Disclaimer: The content does not represent Delphi opinion or Delphi technology approach

Overview of Content

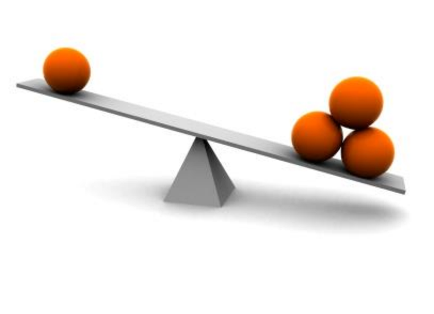
- Part 1 – Short Overview of Safety and Safe Autonomous Cars
  - Safety – Paradigm shift and Autonomous “level”
  - What the public sees and hears
  - Autonomous research, challenges
- Part 2 – Challenges with applying ISO 26262
- Part 3 – Verifying and validating - Autonomous system
- Part 4 – Core architectures
- Part 5 – Q&A

What is Safety in a Car context?

today and tomorrow

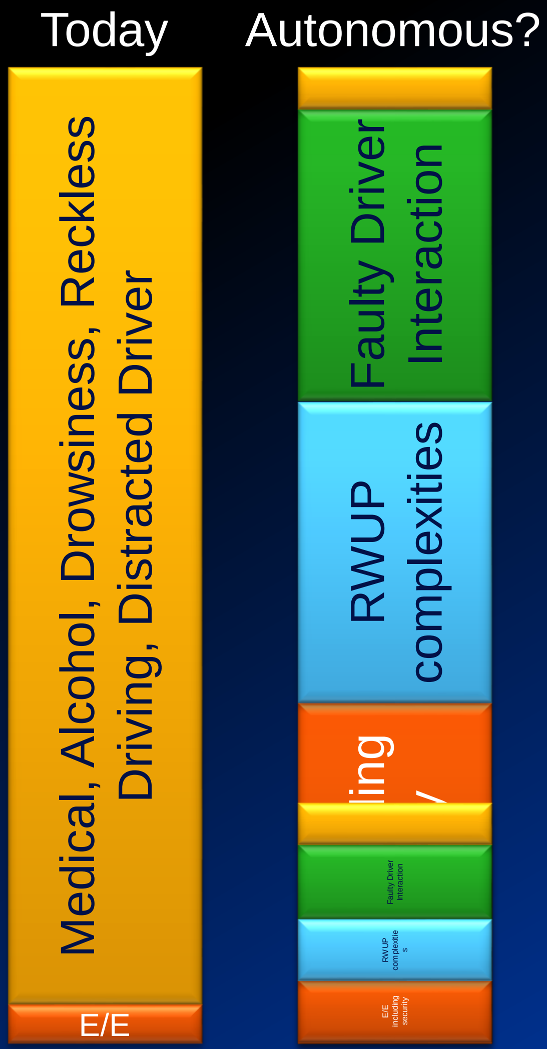
Distribution of Root Causes

- Safety is: Ability to protect the driver, passengers and environment from unreasonable damage, no matter origin
- ADAS addresses every day situations to decrease accidents and increase safety
- AD further contributes by deve
  that will increase safety even m
  - By avoiding hazardous situations



- **Note:** ADAS and AD products i
  hazardous events themselves
  - **PROS** and **CONS** would still deem this probability as acceptable, or?
  - **Hard to claim that it is acceptable that products malfunction due to bad designs and insufficient testing or analysis**
- How to tackle this within AD?

E/E faults



Delphi / Public

Roadrunner Makes History…

First-ever coast-to-coast **automated drive**

San Francisco to New York
March 22 - 30, 2015

- 3,400 miles
- 99% automated
- 3TB of data
- 15 states + D.C.
- 9 days

Not fully autonomous!

- navigated multiple lanes of traffic - up to 6 in high congestion areas as Los Angeles and Atlanta
- Radars performed well in all conditions
- Vision systems performed well in most lighting conditions
- Possible to operate in automated mode nearly 100% of the time on US Highways
- Successfully handled complex bridges with challenging steel girders and structures, as well as traffic congestion, heavy construction areas and large haulers with oversized loads

Delphi - In the News…

Delphi's demonstration underscores growing interest… could be production-ready by 2020. REUTERS

Race around the road of future Handelsblatt

Delphi's autonomous test car,… is far more advanced than anything on the road today… Detroit Free Press

Delphi Accomplishes Historic Milestone 咕噜网 cngulu.com 全原创科技媒体 关注互联网

…attered bugs across grille.

As competitors from Google to Mercedes-Benz all develop autonomous cars, this was a crowning moment for Delphi. In reality, the trip is in the rear-view mirror and the hard work now begins. autoblog

In reality, the trip

"The car never gets distracted even when the driver is." CNBC

Delphi will use the lessons it learned to improve the systems that will eventually be used in autonomous and even driverless cars. AP

"This amazing feat…underscores the great leaps this technology has taken in recent years." WIRED

…this achievement should accelerate their future product development…the global game for the autonomous crown is very much on. JALOPNIK

NHTSA definition of ADAS/AD (BAST similar)

No-Automation (Level 0): The driver is in complete and sole control of the primary vehicle controls

Function-specific Automation (Level 1): Automation at this level involves one or more specific control functions., e.g. vehicle automatically assists with braking to enable the driver to regain control of the vehicle or stop faster than possible by acting alone.

Combined Function Automation (Level 2): This level involves automation of at least two primary control functions designed to work in unison, e.g. adaptive cruise control in combination with lane centering.

Limited Self-Driving Automation (Level 3): Vehicles at this level of automation enable the driver to cede full control of all safety-critical functions under certain traffic or environmental conditions and in those conditions to rely heavily on the vehicle to monitor for changes in those conditions requiring transition back to driver control.

Full Self-Driving Automation (Level 4): The vehicle is designed to perform all safety-critical driving functions and monitor roadway conditions for an entire trip.

Source: © NHTSA

Some Interesting Autonomous Research

- How desirable are autonomous functions?
  - And how to find out?
- Swedish Research Project
  - Looking at the ordinary driver and how this driver reacts and what the driver think
- Investigating Methods for Designing Autonomous Systems
  - How to design systems that really support the driver?
  - How to measure that they really support the driver?
  - How to make sure the system is safe?

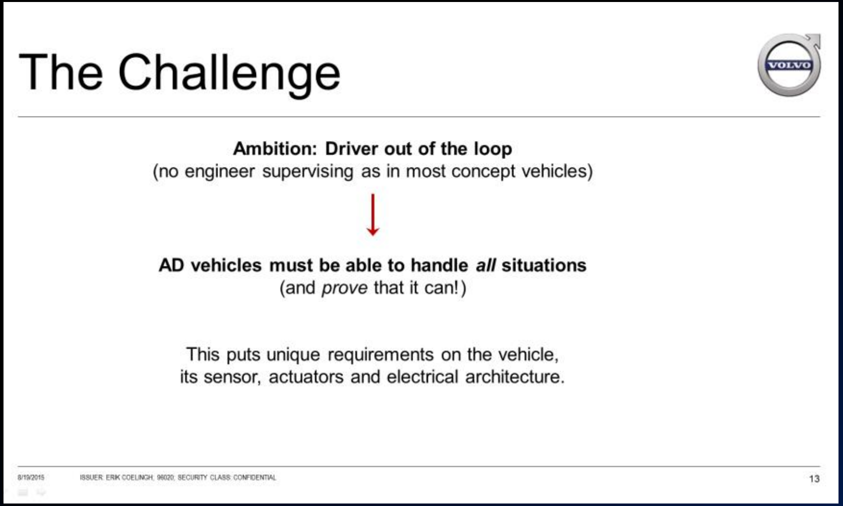
MODAS

Asiana Airlines

"Don't rely too much on cockpit automation!!!"

Volvo - DRIVE-ME - Research Questions

- Driver interaction (mode confusion, override *etc*.)
- Fail operational (controllability, safe maneuver, safe state)
- Relation to other items (certified road, other vehicle systems)
- Environment sensing (situation dependent, missed and false)
- Reasonable foreseeable misuse, product compliance (*e.g.* other traffic participants testing the autonomous vehicle)



The Challenge

Ambition: Driver out of the loop
(no engineer supervising as in most concept vehicles)

↓

AD vehicles must be able to handle *all* situations
(and *prove* that it can!)

This puts unique requirements on the vehicle,
its sensor, actuators and electrical architecture.

8/19/2015 ISSUER: ERIK COELINGH; 96020; SECURITY CLASS: CONFIDENTIAL 13

© Courtesy from DriveMe

- Scope
  - 360 degree fusion
  - Own position
  - Obstacles

From Research, over Advanced to Production

- Current “Full” Autonomous Concept Cars are not yet in production status.
    - NOTE According to Wikipedia: Google's robotic cars have about $150,000 in equipment including a $70,000 Lidar system, to allow a safe driving)
    - Useful for data collection and concept evaluation to develop features
    - Google had driven more than 270 000 km/ with their 23 special Lexus-SUVs, of these 160 000 km in autonomous mode. (According to information early summer 2015) – Other say million miles
- AP reveals that autonomous cars have been involved in 11 incidents
    - "light damage, no injuries" — and happened over 1.7 million miles of testing, including nearly 1 million miles in self-driving mode.
- Future autonomous cars will inevitable have to suffice with sensor systems having a BOM cost between 100-1000 dollars a car shared by many supplier?
    - And share the most immediate road with multiple other AD

ADAS logging - 6 months – example

- Driving the cars generate 3,1GB of data per 70 sec (43,43MB/s)
- About 5,5h of driving in one 8 hour driving shift. (19.800s)
- 3 shifts per day per car
- 7 cars used over 6 months (182 days)
- 43 MB/s * 18000 = 859.91GB per shift * 3 shifts per day = 2.5797TB per day per car
- 2.5797TB * 182 days = 496.5TB per car
- 7 * 496.5TB = 3.2865 petabytes data collected for a project over 6 months.
- That's 13146 laptop’s (250GB drives) that makes one 420,7m high tower if placed on each other **(or 1.4 Eiffel Towers)**
- Add to this metadata extract, resims and storage overhead. (~25% per resim)
- **And this only validates the nominal function and not at all possible systematic faults in SW and random faults in HW**

Movement of data.

- The data needs to be moved to some were where it can be processed .
- Using 50Mbit/s connections the move of just one day worth of raw logs will take 4days and 18h
- Using DHL to move one day of logs will take 1 day from Gothenburg to any European city. Giving DHL a speed of 238Mbits/s.
- “Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.”

—*Tanenbaum, Andrew S. (1989). Computer Networks. New Jersey: Prentice-Hall. p. 57. ISBN 0-13-166836-6.*

- Adjusted to today's technology, Tanenbaum's quote would be thus: **"Never underestimate the bandwidth of an Boeing 777 filled with 64 GB MicroSD cards"**

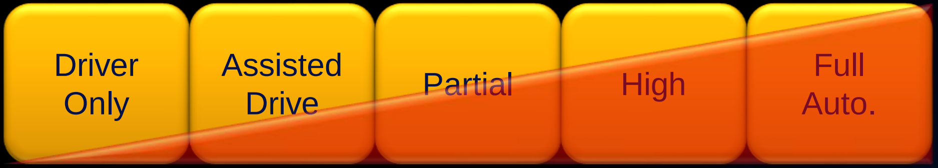
Challenges with building AD cars under ISO 26262 – Functional Safety Standard for road vehicles

DELPHI

Innovation for the Real World

Moving from just "Safety Related" to Autonomous

- ISO primary scope, faults in E/E system
- What happens to Functional Safety, when moving to more autonomy?
  - How to define it? (Looking definiti
    - sc
  - How to for arch divisio
    - Many domains
  - How to prove it? (Demand for new compositional safety arguing)
    - Not only testing


Driver Only

Assisted Drive

Partial

High

Full Auto.

- Driver only – ISO 26262 fully applicable

are

is

or

- Full autonomy, Cloud, V2X and GPS (maps) are in the loop
  - introducing security where faults are hard to quantify and evaluate

Higher levels of autonomy for automated driving will decrease the driver's ability to control the consequences of a hazardous event (=> controllability) (see SAE J3016

How is ISO 26262 standing?

- Two types of reasons why ISO26262 becomes problematic
  - Things are (much) more complicated
    - Item definition for extremely complex functionalities
    - FSC and TSC much more complex to make formal
- Things are fundamentally different
  - Manual driver not directly in the loop (controllability decreasing)
  - Infrastructure is not fully prepared
  - Many more RWUPs are introduced (exposure)
- -> Addressed by the SOTIF group in ISO 26262
  - One part or integrated with the current?

Challenge - Hazard Analysis and Risk Assessment - HARA

- While hazardous events for items in current ISO 26262 scope are fairly well known
- ADAS and AD are quite different
  - Many more situations that need to be taken into account
  - Many more error sources to analyze
  - From internal E/E, Function, and external info
- Safety Goals with ASIL D
  - Leaving lane, (road)
  - How to consolidate?

Driver:

Fail to:

Detect information

Attend to information

Comprehend information

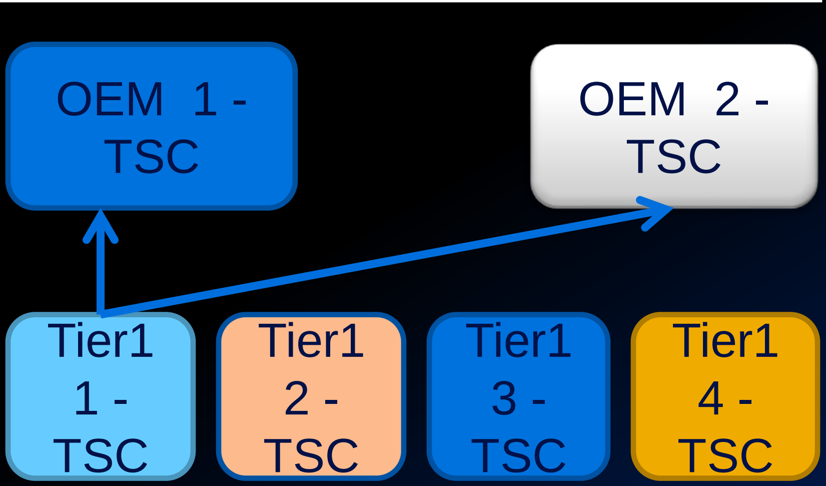
Take right decision and actions

Interpret feedback

To trust the system

Challenge OEMs vs Tier 1/2

- OEMs wish to have their view and functions supported, their HARA and TSC
  - Using existing platforms and EUCs
- Tier 1s think up and down and must also support several OEMs, based on own HARA and SEooC TSC
  - Wish to reuse their IP
- Tier 2/3s think bottom up but must provide good “TSC” and testing
  - Focus on “safe” components
  - Having an assumed use case, SEooC



- Conclusion: Formal Architectures need to be even more precise and used between the stake holders, which are “expensive” to maintain
  - CORE ARCHITECTURES

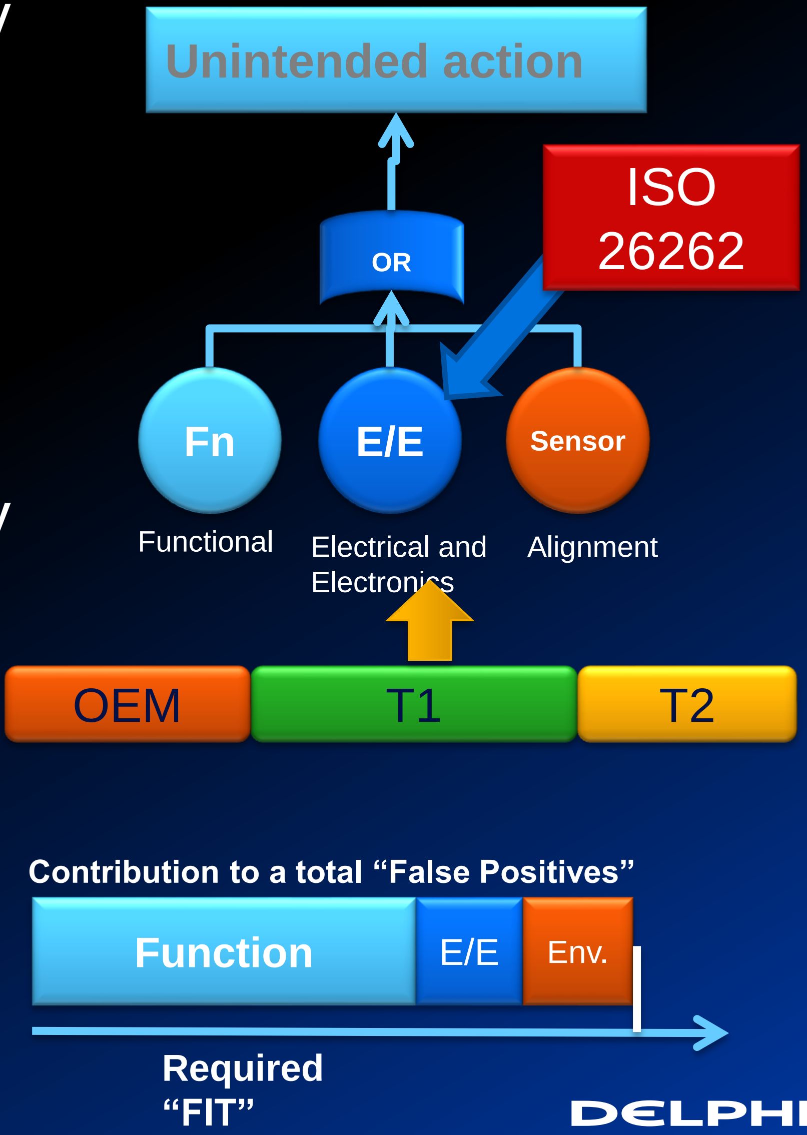
Intended, Unfortunate, Unintended and Unknown Function

- The intended function is to travel from A to B tak situation into account that is plausible
  - **Intended function** will resolve the s der the valid fault assumption or it
- With a sensor syst or the vehicle and functionally situations that will cause false
  - on means that the situation was correctly was resolved externally to ego vehicle but the actual g will still manifest as a false activation
  - **Unintended function**, means that a misconception, imperfection in describing the reality lead to a design that can cause a false activation or no activation
  - **Unknown function**, meaning that there will be situations impossible to forecast the behavior because we do not know them.

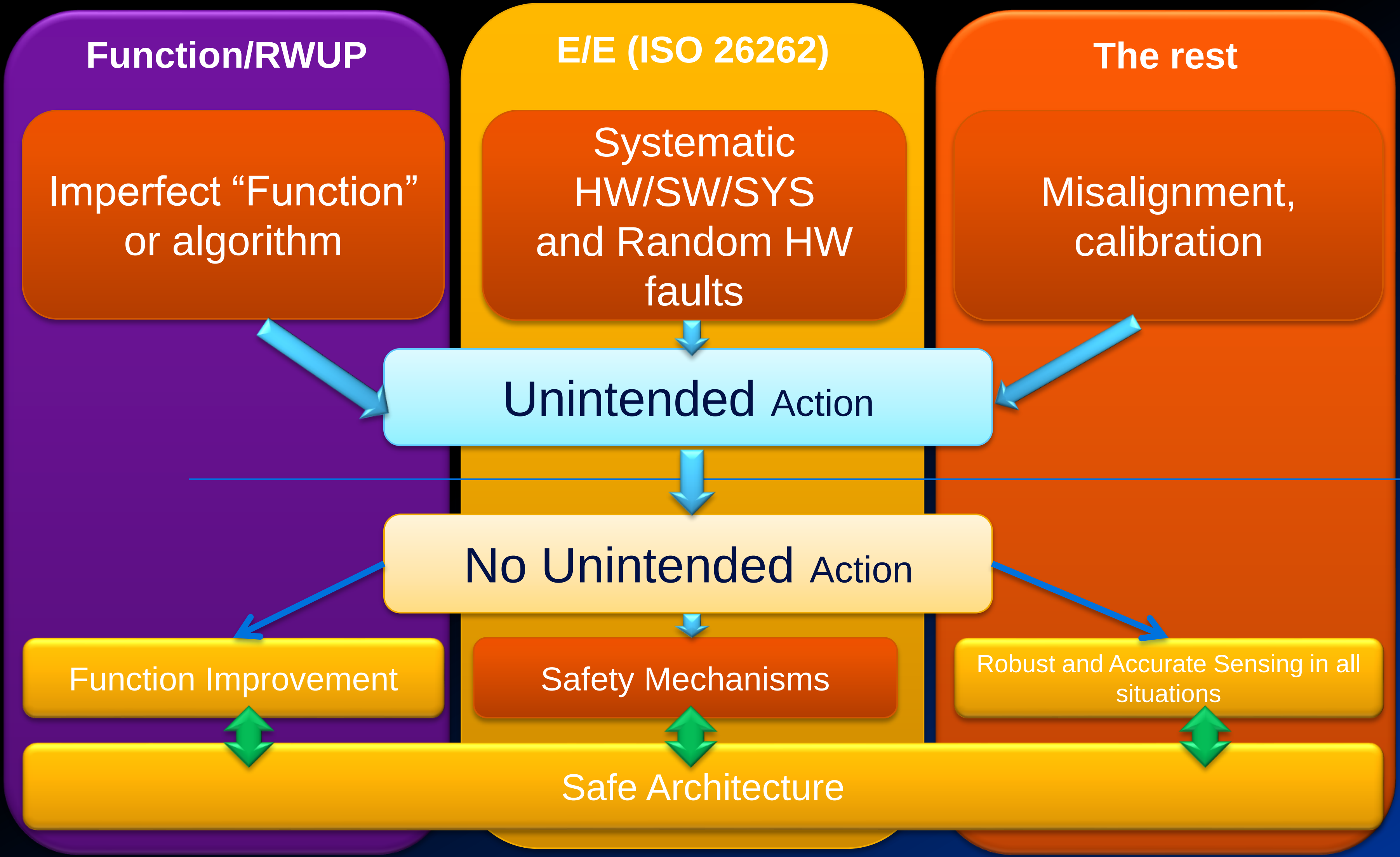
And this is in a ”fault” free environment – SW and HW faults not taken inte account

Challenge – Errors causing unintended actions

- Not only E/E faults cause “Safety Goal Violations”
  - ”Surprise”
- Still - System must be safe
  - Considering realistic RWUPs, e.g. during lane changes
- Safety is a Shared Responsibility
  - Function Owners
  - Functional Safety
  - System Functions
- How to make one ”Technical Concept” addressing safety?



Overview of the domains that must be addressed!



Function/RWUP

E/E (ISO 26262)

The rest

Imperfect “Function” or algorithm

Systematic HW/SW/SYS and Random HW faults

Misalignment, calibration

Unintended Action

No Unintended Action

Function Improvement

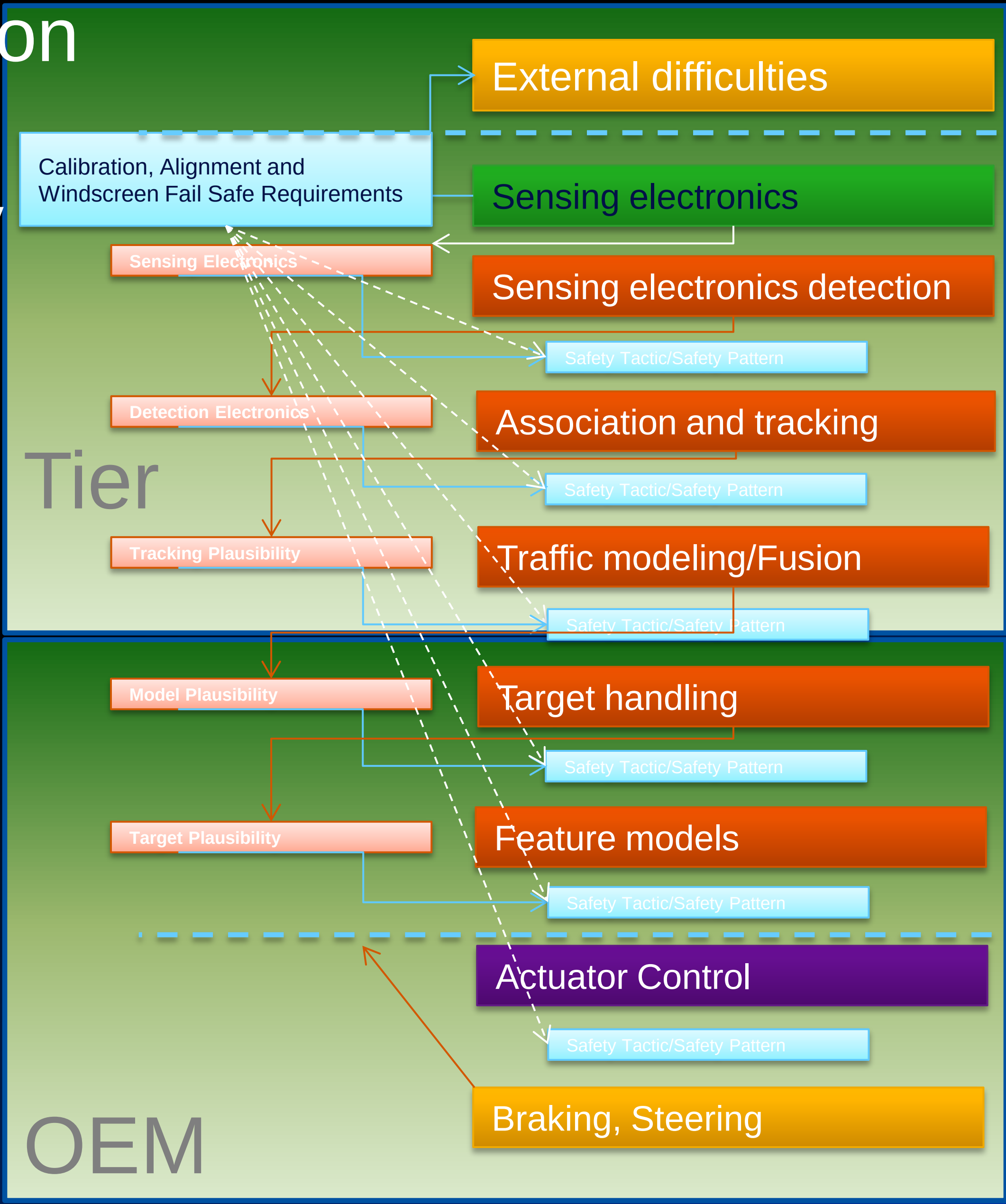
Safety Mechanisms

Robust and Accurate Sensing in all situations

Safe Architecture

Levels for Implementation of Safety Patterns

- The approach to comply to safety goals/safety requirements, no matter origin, needs to be consolidated and implemented at the “right level” with the right:
  - Safety Tactics
  - Safety Patterns
- Support for the system engineering to avoid human errors impact


External difficulties
Calibration, Alignment and Windscreen Fail Safe Requirements
Sensing electronics
Sensing Electronics
Sensing electronics detection
Safety Tactic/Safety Pattern
Detection Electronics
Association and tracking
Safety Tactic/Safety Pattern
Tier
Tracking Plausibility
Traffic modeling/Fusion
Safety Tactic/Safety Pattern
Model Plausibility
Target handling
Safety Tactic/Safety Pattern
Target Plausibility
Feature models
Safety Tactic/Safety Pattern
Actuator Control
Safety Tactic/Safety Pattern
Braking, Steering
OEM

Examples of different levels of safety mechanisms

- Confidence of a road – e.g. two lane markers vs one lane marker and the distance between them
  - Adding ”robustness but maybe lowering availability”
- Radar
  - Using two (redundant sweeps), adding possibility for higher plausibility
- Camera
  - Different fail safes, too dark…
- Fusion
  - Establishing confidence through comparing objects
- Communication - Protecting produced Data
  - Corrupted messages with data, safety concepts such as E2E
- Platform, ensuring freedom from interference
- Vehicles Models
- The entire vehicle architecture

How then to Verify and Validate ADAS?

Verification and Validation

- As discussed, there are millions of miles collected from ADAS & AD application validation using powerful and intelligent tools.
  - Mainly to understand why false negatives occur and understand false positives wrt RWUP -> deedback to FO
  - This data is used to “validate” todays RWUPs to meet NHTSA and Euro NCAP assessment requirements
  - **Still far from validating safety mechanisms**

Challenge - Verification cont

- Faults caused by systematic software faults and random hardware faults causing **false negatives/false positives** should typically be in the range of 10-8(9) faults per hour for an autonomous car.

- May be possible to show argumentation for suc … o development process, testing and hard …

- Verification of blue prints an … gh to prove the building is representi … ers and OEMs expectations
  - Also, th … simply too complex to allow a straight f … the system components.

- H … that behavior on the top level, i.e. that the safety goals ar … ved
  - Techniques for validation are there but need to mature – What do we validate?
  - Negative testing and Fault injection is a must

More in the Panel: Defining Safety Priorities with Functional Safety and ”Safety Critical Systems”

One solution – Establish Formal Architectures/Cores

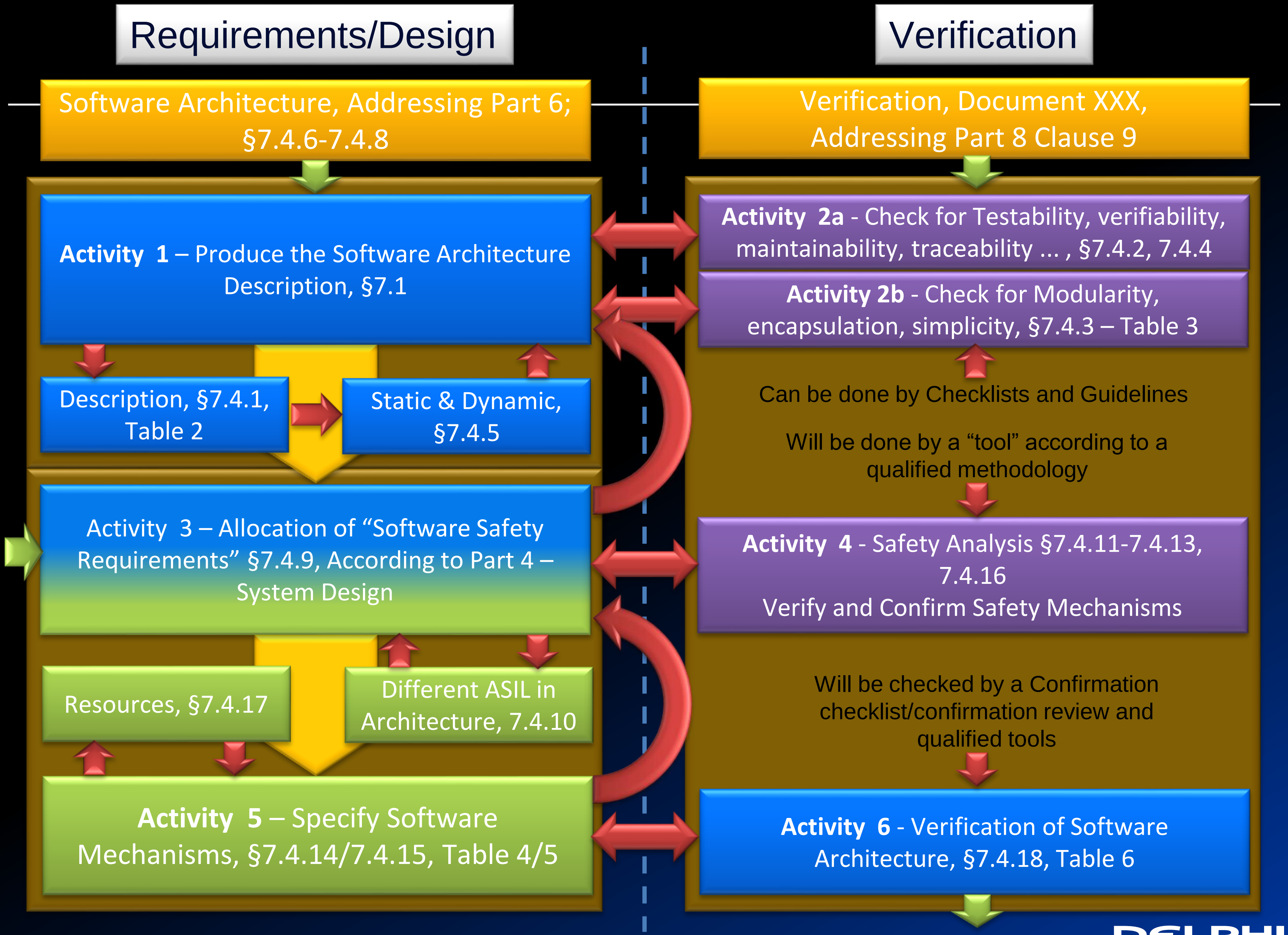
Challenges with todays ADAS->AD

- Why ISO 26262 focuses on formalism and architectures
- “It is not easy to get time to build maintainable and formal Function- System- and SW architectures”
- “I wish there was an stringent way to reuse requirements and software between projects”
- “It is not easy to add new functionality when you do not know what you already have”
- “I cannot analyze the design to a sufficient degree”
- “Impossible to push back customers “similar” requirements without good design”
- “I wish that the customer had more realistic expectations on our sensor system”

Core Architectures

- Importance to have your product defined and described, done through formal interfaces/APIs
  - Easier to connect to SW arch
  - ”ISO” precicely described interfaces
- Entreprise Architect Models (UML) or in Medini analyze
- Ensuring an understanding of the product
- Analysis results can be reused

> NOTE Architectural constraints described in ISO 26262-5:2011, Clauses 8 and 9, are not directly applicable to COTS parts and components. This is because suppliers usually cannot foresee the usage of their products in the end-item and the potential safety implications. In such a case, basic data such as failure rate, failure modes, failure rate distribution per failure modes, built-in diagnosis, etc. are made available by the part supplier in order to allow the estimation of architectural constraints at overall hardware architecture level.



Requirements/Design

Verification

Software Architecture, Addressing Part 6; §7.4.6-7.4.8

Verification, Document XXX, Addressing Part 8 Clause 9

Activity 1 – Produce the Software Architecture Description, §7.1

Activity 2a - Check for Testability, verifiability, maintainability, traceability … , §7.4.2, 7.4.4

Activity 2b - Check for Modularity, encapsulation, simplicity, §7.4.3 – Table 3

Description, §7.4.1, Table 2

Static & Dynamic, §7.4.5

Can be done by Checklists and Guidelines

Will be done by a “tool” according to a qualified methodology

Activity 3 – Allocation of “Software Safety Requirements” §7.4.9, According to Part 4 – System Design

Activity 4 - Safety Analysis §7.4.11-7.4.13, 7.4.16
Verify and Confirm Safety Mechanisms

Resources, §7.4.17

Different ASIL in Architecture, 7.4.10

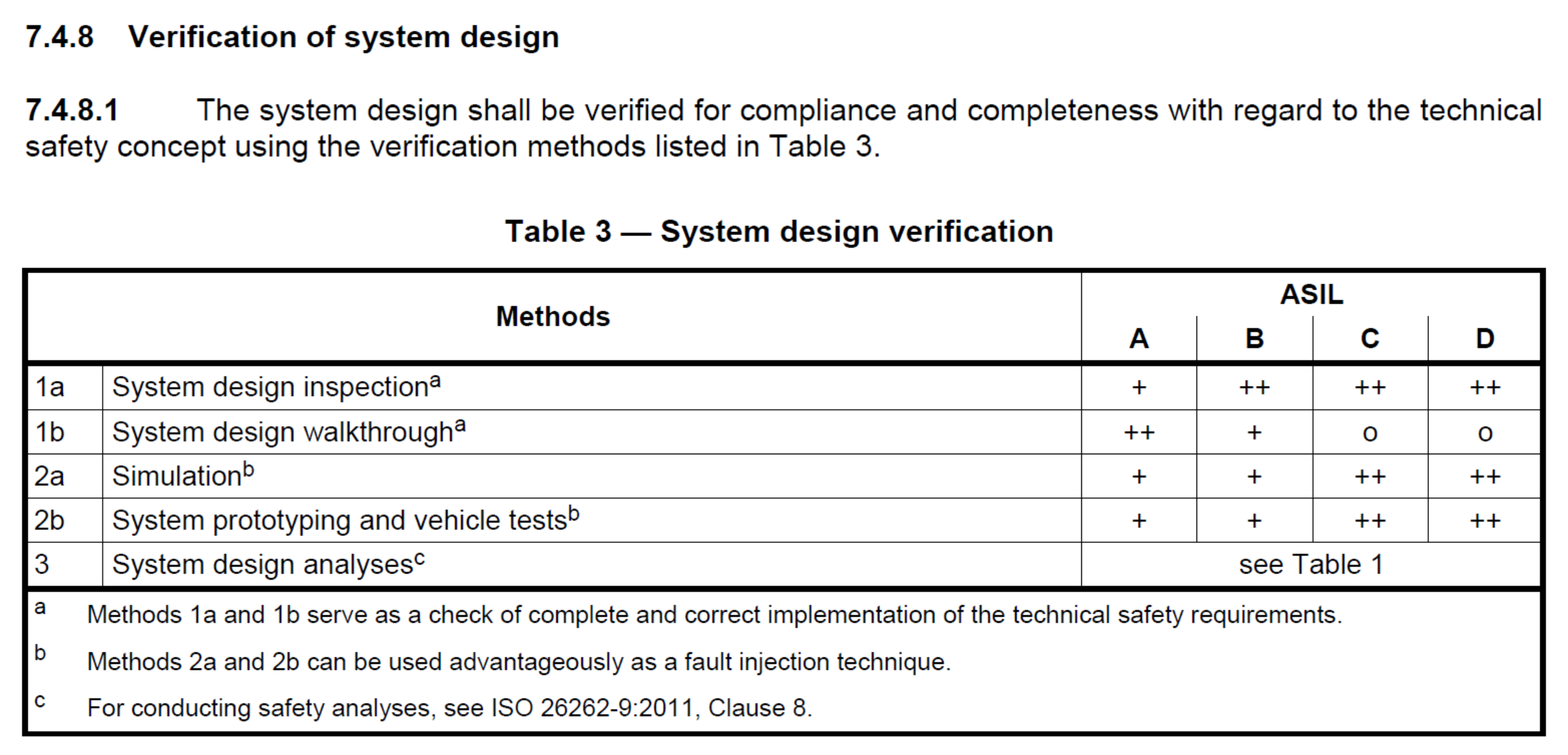
Will be checked by a Confirmation checklist/confirmation review and qualified tools

Activity 5 – Specify Software Mechanisms, §7.4.14/7.4.15, Table 4/5

Activity 6 - Verification of Software Architecture, §7.4.18, Table 6

ISO on System Verification

- “Safety analyses on the system design to identify the causes of systematic failures and the effects of systematic faults shall be applied “



7.4.8 Verification of system design

7.4.8.1 The system design shall be verified for compliance and completeness with regard to the technical safety concept using the verification methods listed in Table 3.

Table 3 — System design verification

Methods		ASIL			
		A	B	C	D
1a	System design inspection[a]	+	++	++	++
1b	System design walkthrough[a]	++	+	o	o
2a	Simulation[b]	+	+	++	++
2b	System prototyping and vehicle tests[b]	+	+	++	++
3	System design analyses[c]	see Table 1			

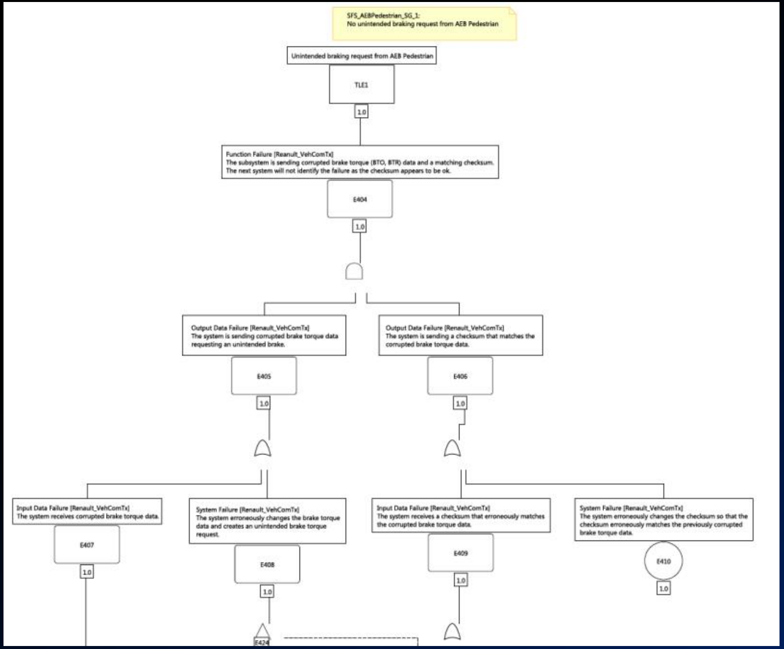
[a] Methods 1a and 1b serve as a check of complete and correct implementation of the technical safety requirements.

[b] Methods 2a and 2b can be used advantageously as a fault injection technique.

[c] For conducting safety analyses, see ISO 26262-9:2011, Clause 8.

Challenge - Safety Analysis Reuse

- Safety Analyses over multiple customer and ”sub” system interfaces – complicated and expensive
  - Sensor FTAs with TLE reporting to FUSION as Basic events
  - Fusion reporting TLE to AD models as basic events
  - TLE from Models will be basic event etc
  - …
  - Finally, Basic events into Brake Node will be TLE to the brake actuator
- How to re-use safety analysis work, how to ”AND” hard to analyze elements of the system



SFS_AEBPedestrian_SG_1:
No unintended braking request from AEB Pedestrian

Unintended braking request from AEB Pedestrian

Function Failure [Reanult_VehComTx]
The subsystem is sending corrupted brake torque (BTO, BTR) data and a matching checksum. The next system will not identify the failure as the checksum appears to be ok.

Output Data Failure [Renault_VehComTx]
The system is sending corrupted brake torque data requesting an unintended brake.

Output Data Failure [Renault_VehComTx]
The system is sending a checksum that matches the corrupted brake torque data.

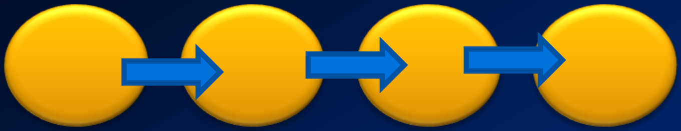
Input Data Failure [Renault_VehComTx]
The system receives corrupted brake torque data.

System Failure [Renault_VehComTx]
The system erroneously changes the brake torque data and creates an unintended brake torque request.

Input Data Failure [Renault_VehComTx]
The system receives a checksum that erroneously matches the corrupted brake torque data.

System Failure [Renault_VehComTx]
The system erroneously changes the checksum so that the checksum erroneously matches the previously corrupted brake torque data.

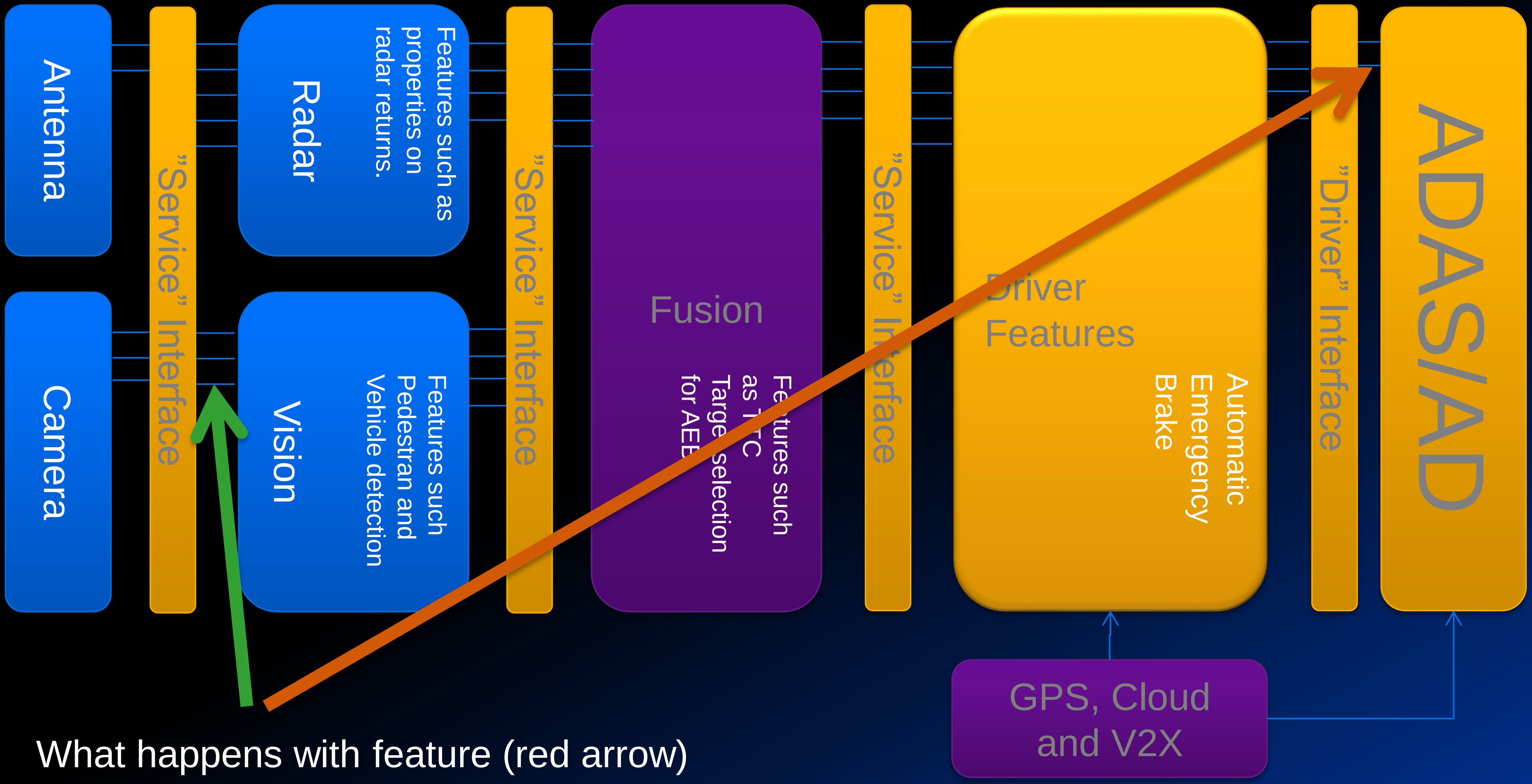
- Safety concepts need to be precise and simple to validate



Need for OEM support for SEooC

- OEMs support
  - A high degree of Formalism in architectures and system designs are needed to comply with ISO 26262
  - Core building blocks (SEooC) with its own verification/validation strategy/evidence and Technical Safety Concepts – will increase the insight of sub systems
  - “Standard Interfaces” and formal properties of “objects” will support composable systems
  - Need for common methodologies for how safety analysis results can be coupled between stake holders

Challenge - Moving from ADAS architectures to AD



Antenna

Camera

"Service" Interface

Radar

Features such as properties on radar returns.

Vision

Features such Pedestrian and Vehicle detection

"Service" Interface

Fusion

Features such as TTC Target selection for AEB

"Service" Interface

Driver Features

Automatic Emergency Brake

"Driver" Interface

ADAS/AD

GPS, Cloud and V2X

What happens with feature (red arrow) if sensor property (green) is changed?

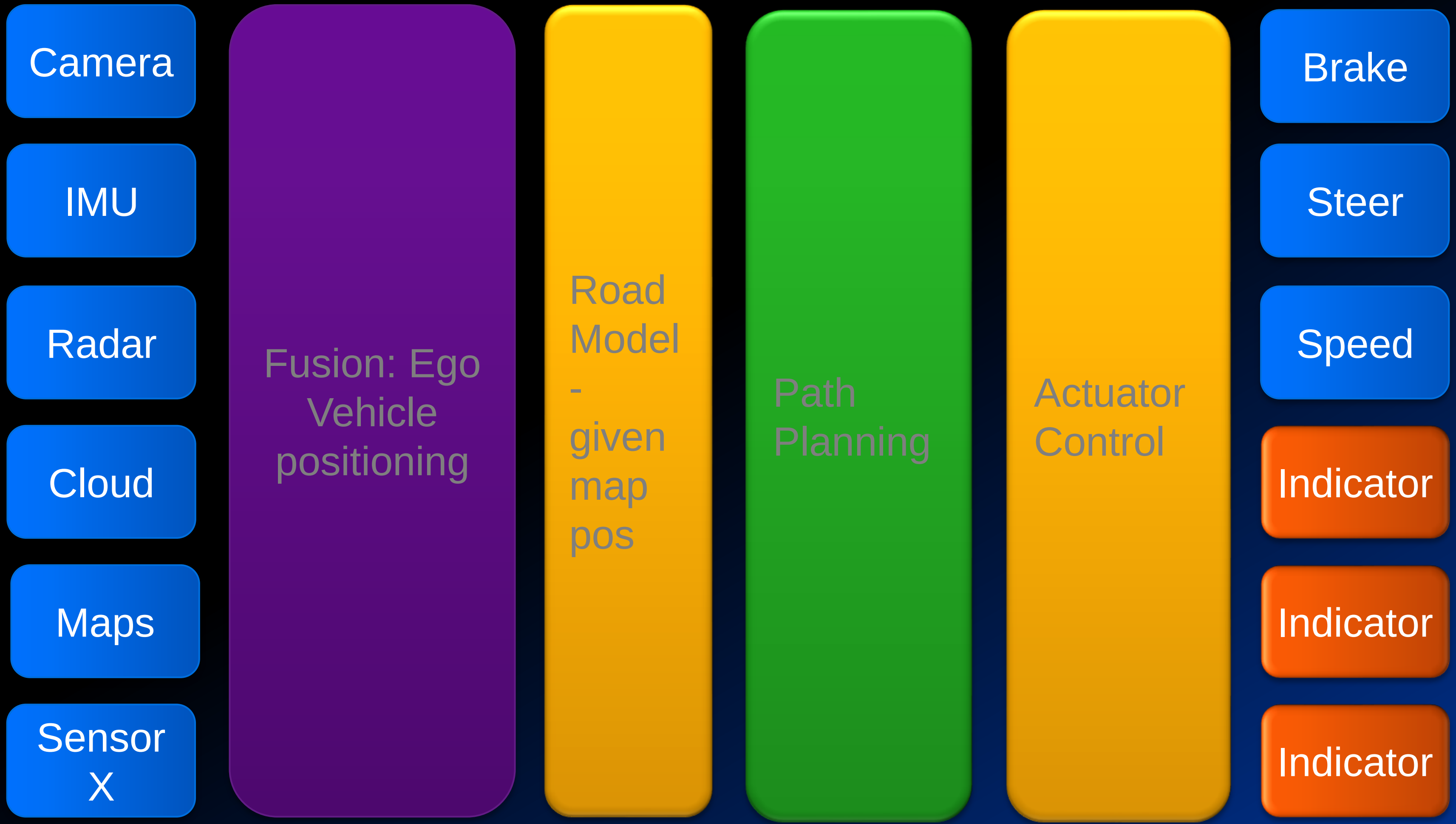
A challenge to design architectural elements reusable

Challenge - Moving existing products from ADAS to AD – Paradigm Shift

- High rate of True positives are key (ADAS best effort)
- Low False Positives rate, as today but for ons
- False Negatives, you avoid or enter safe state, not f
- True negativ
- Result:
  - Sensing systems must develop further
  - Verification and Validation will need to be sharpened up, significantly
  - Fault Injection on various levels will be needed to validate safety

Means that reuse of ADAS ”components” need to be ”handle with care”

AD ”System” Overview even more complicated, ADAS++



DELPHI

Some more Drivers for Core SW/System/Function ADAS

1. Need to feel confident about a liability investigation
   a. The principal safety “function” implementation needs to reflect state of the art
2. Need to create a SW/System/Function structure adapted for efficient and accurate testing
   a. If no efficient testing capability the product quality will be low
3. Need to meet the engineering cost level defined by the business
   a. Without high technology reuse between projects it will be difficult
4. Need to have advance projects applying state of the art development procedures and tooling
   a) “Hard to make a swan out of a goose”

Thanks for listening!

Questions?

DELPHI
Innovation for the Real World