# Challenges to Improve the Confidence in Cyber-Physical Systems

**Tetsuya TOHDO**

**ePF Advanced R&D Dept.**
**DENSO CORPORATION**

DENSO

# DENSO CORPORATION



| | |
|---|---|
| **Established** | December 16, 1949 |
| **Capital** | 187.4 billion yen (US$1.6 billion) |
| **Revenue** | |
| Consolidated basis | 4,524.5 billion yen (US$40.2 billion) |
| **Operating Profit** | |
| Consolidated basis | 315.7 billion yen (US$ 2.8 billion) |
| **Employees** | |
| Consolidated basis | 151,775 |
| Non-consolidated basis | 38,490 |
| **Consolidated Subsidiaries** | 188 |
| (Japan 62, North America 28, Europe 34, Asia 58, South America/Others 6) | |
| **Affiliates under the Equity Method** | 36 |
| (Japan 13, North America 4, Europe 4, Asia 13, South America/Others 2) | |

Notes:
U.S.dollar amounts have been translated, for convenience only, at the rate of 112.68 yen = US$1,
the approximate exchange rate prevailing on March 31, 2016. Billion is used in the American sense of one thousand million.

/ as of March 31, 2016

# Products



## Environment

Hybrid and electric vehicle components,
Products for fuel cell vehicles,
gasoline engine management system,
diesel engine management system,
starter, alternator, radiator, etc.

## Safety

Sensing technologies for driving assist systems,
actuator & computer for antilock brake system (ABS) / electronic stability control (ESC),
adaptive front-lighting system (AFS),
airbag sensors & electronic control units,
periphery monitoring system, instrument cluster,
rain sensor for automatic windshield wiper, etc.

## Comfort & Convenience

Car air-conditioning system,
air conditioner for buses, air purifier,

Car navigation system,
electronic toll collection system (ETC),
remote security system,
remote touch controller, smart key,
advanced vehicle operation system (AVOS), etc.

## Non-Automotive Fields

Home Appliances, Heating and Cooling Equipment,
Auto ID Data Capture Devices,
Factory Automation Products

DENSO Global Site

# Outline



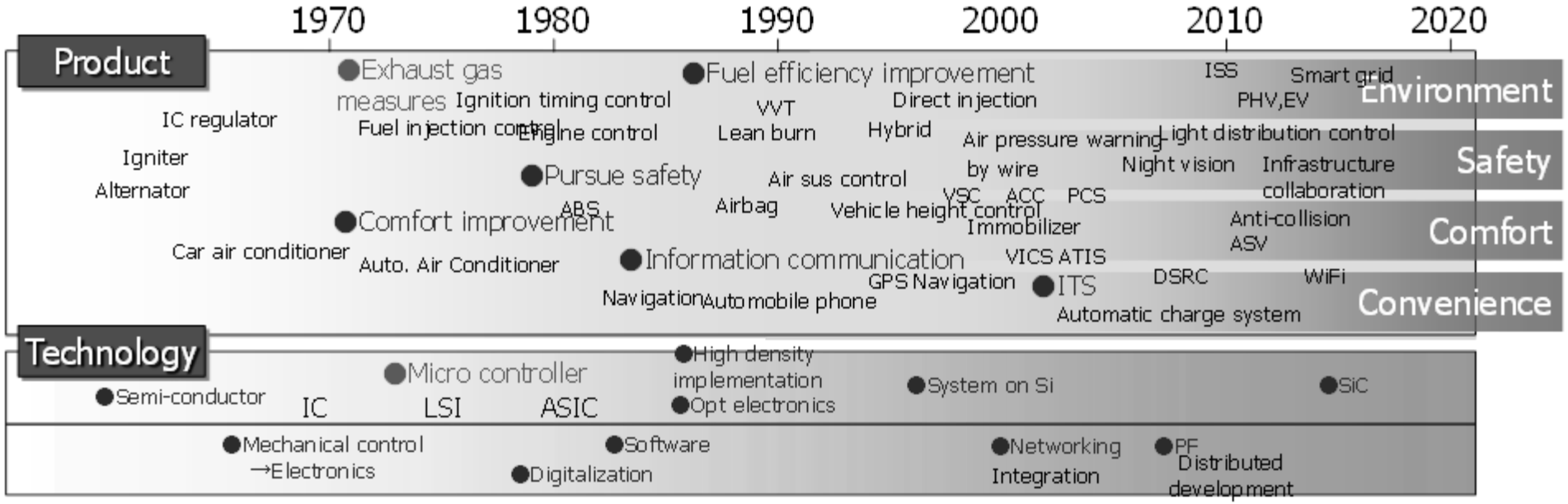
- Introduction
- MBD (Model-Based Development)
- Functional Safety
- Advanced Topics
- Testing
- Summary / Conclusion

# Introduction

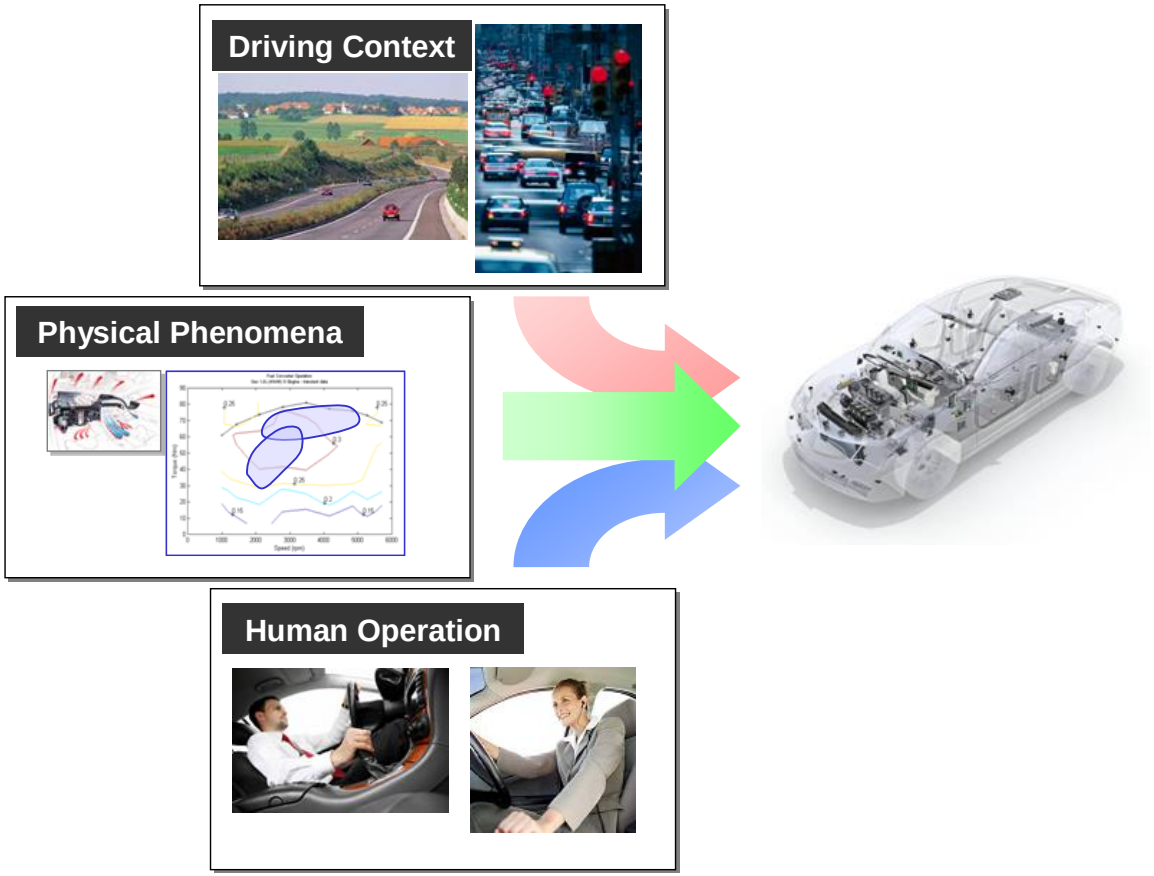
# Electronics

add new features

improve control performance (environment, safety)



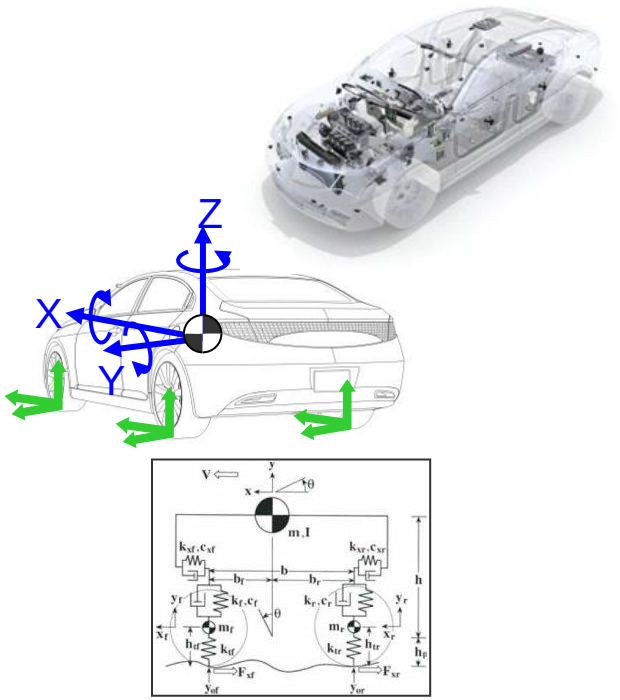
## Automotive E/E Systems

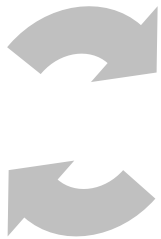


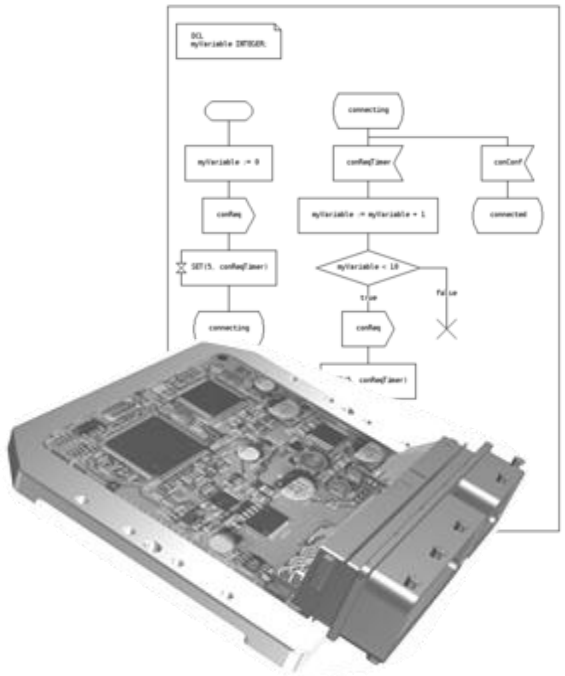
# As Cyber-Physical Systems

- **Physical**

  physics (chemical, …)





computation

- **Cyber**

# Embedded Software

- **Complex system behavior**

  Need confidence in design

- **Technologies**

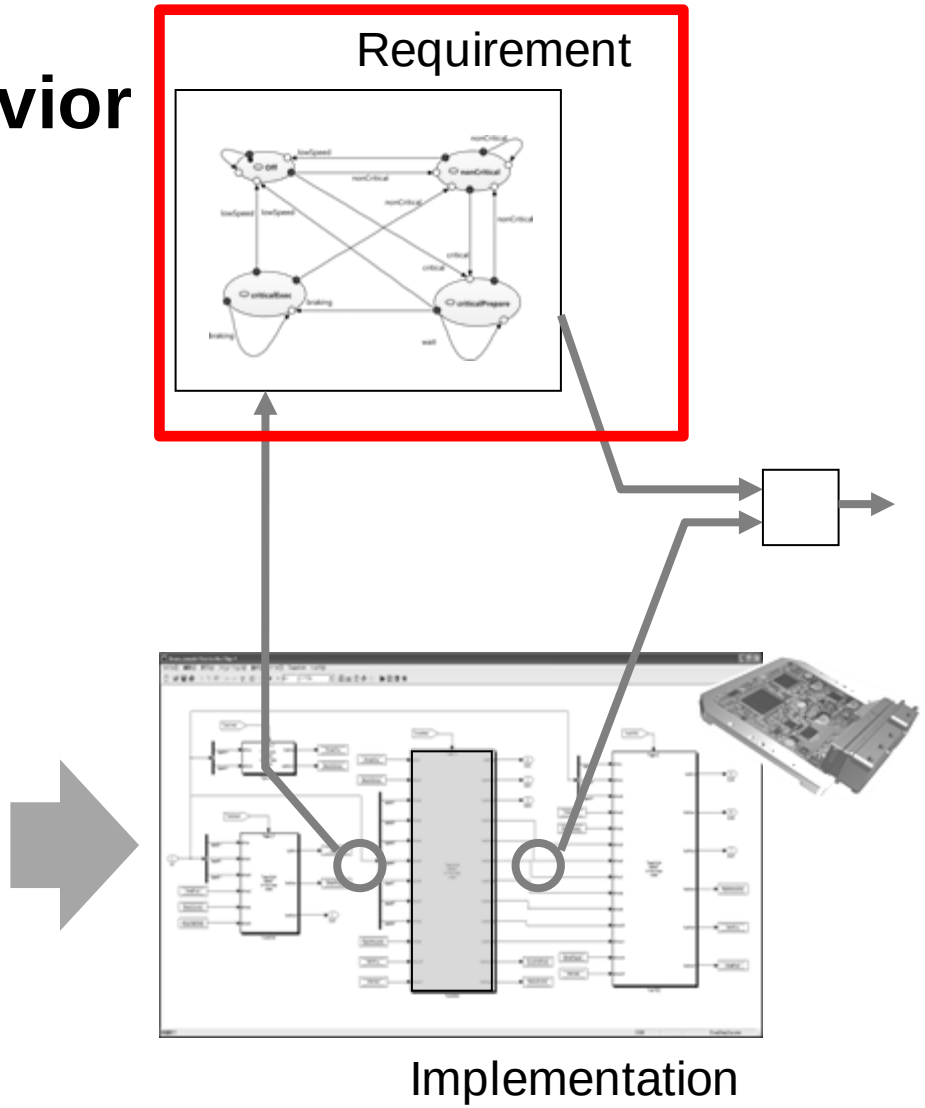
  Simulation

  Formal Verification

  Test Vector Generation

- **Experience**

  Test oracle is hard



- **Cyber-Physical**

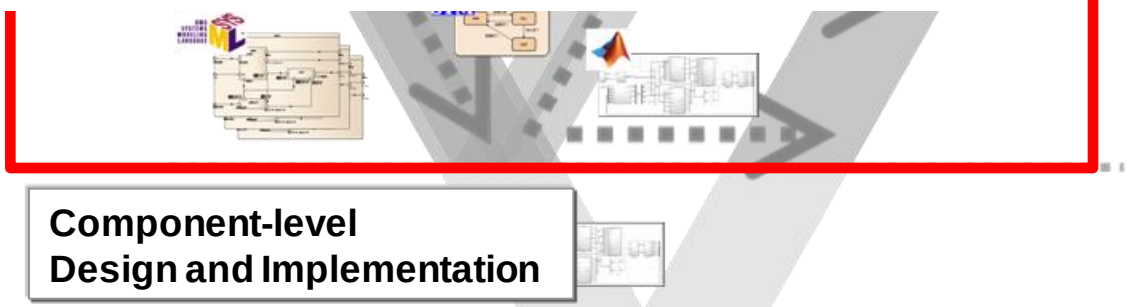
  Functionality of SW designed at system-level

  Various disciplines relate to each other

- **This talk**

  Design, analysis and testing

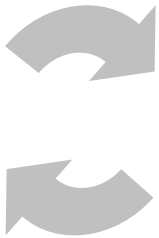
  System and software level



# MBD

Model-based Development

- **Originally, model-based (control) design**
  - Plant model
  - Controller model

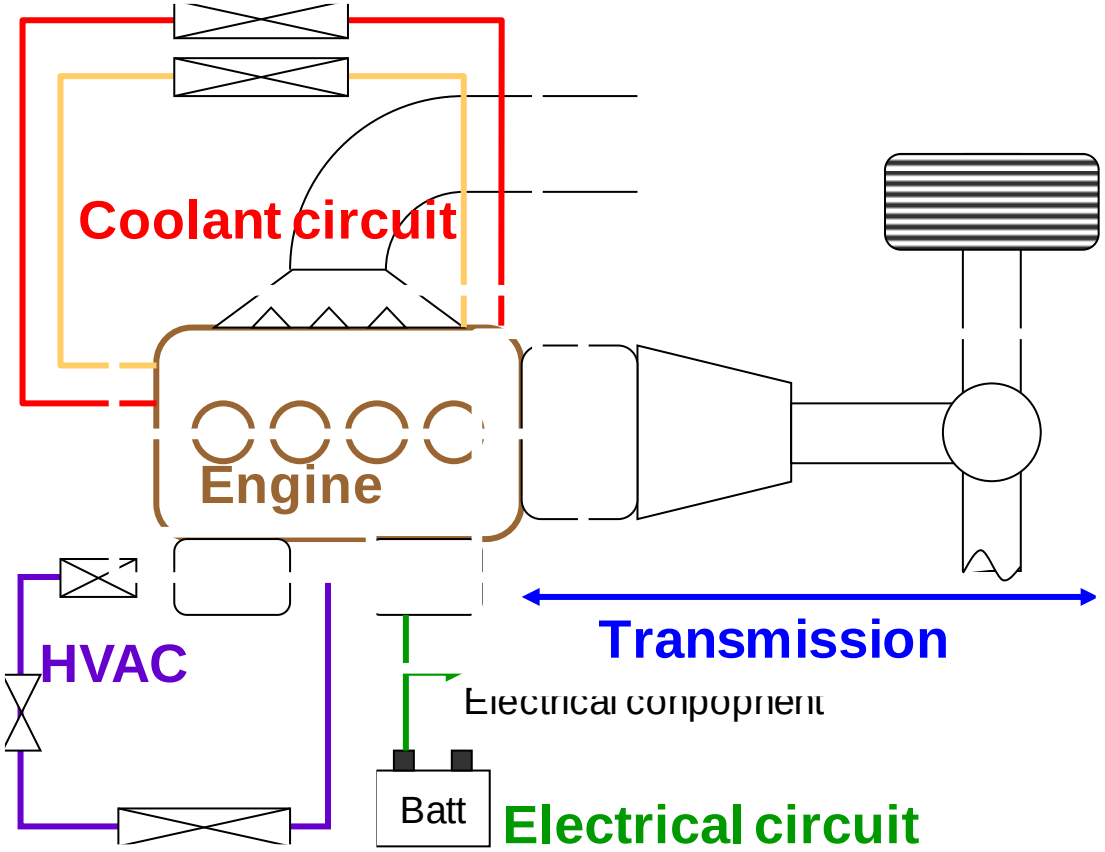


Physical constraints

- **Now, use of simulation technologies**

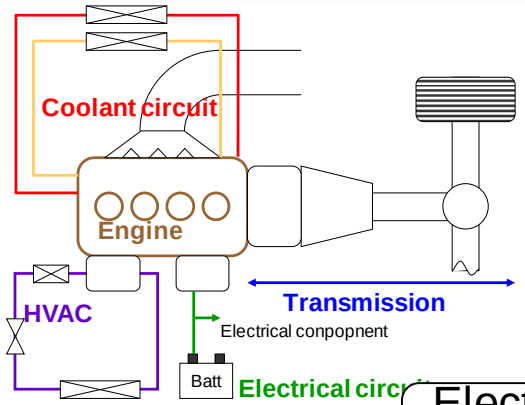
→ Model-based development

## Vehicle Energy Flow

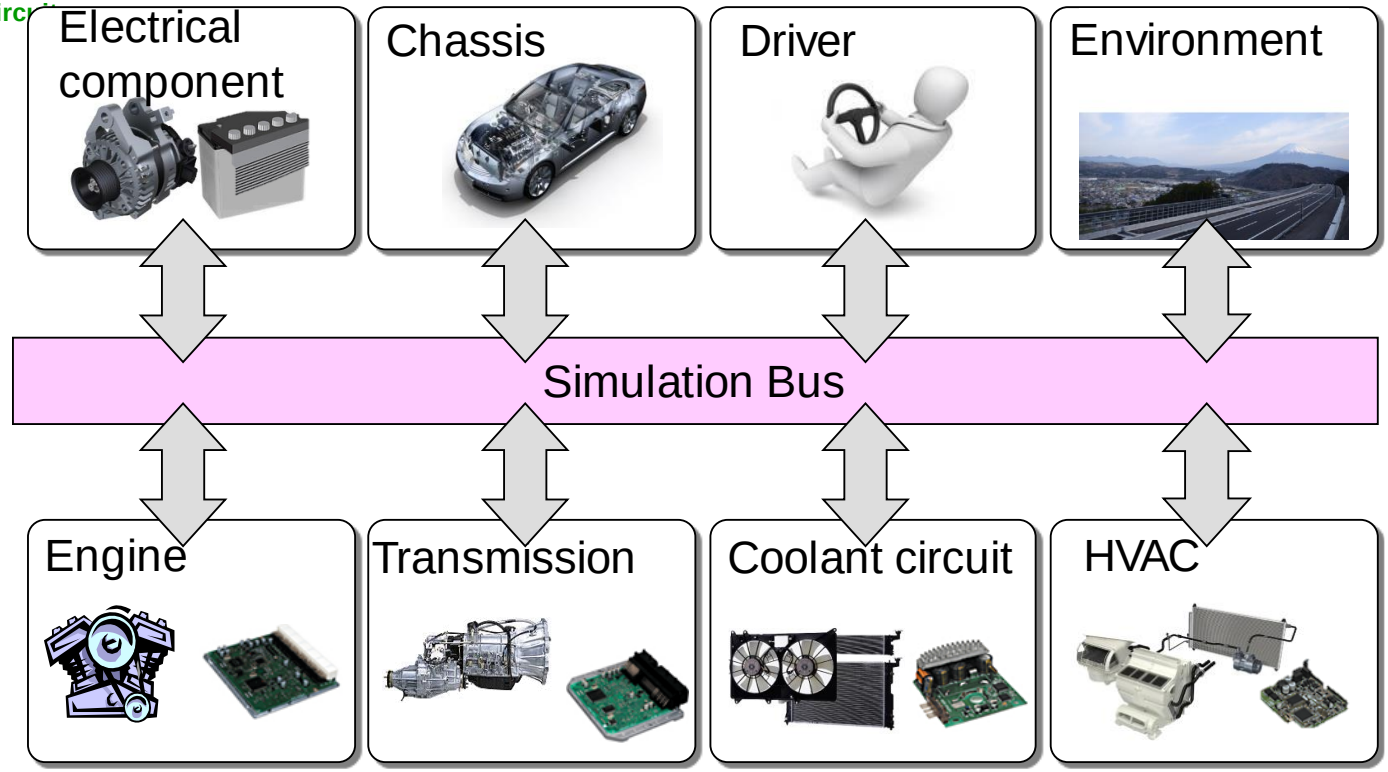


# Simulation : Example





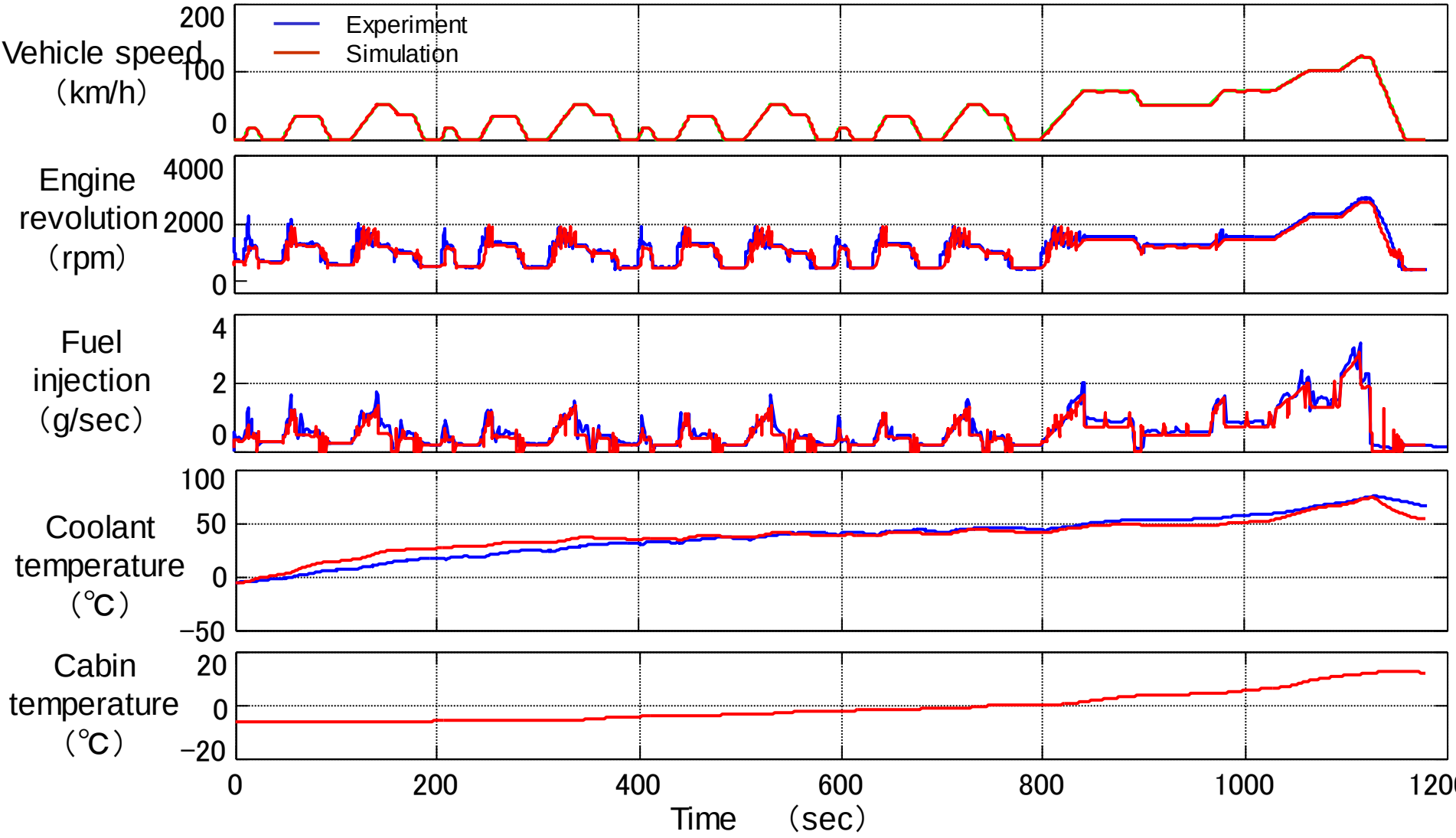

- **Domain → specific simulation**
- **Tool Integration**



# Simulation : Example



## Result



# MBD Technologies

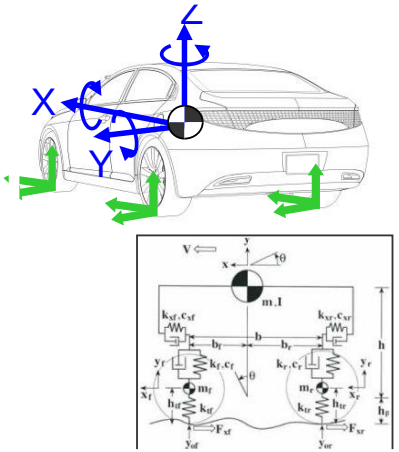


**Plant**

**Controller**

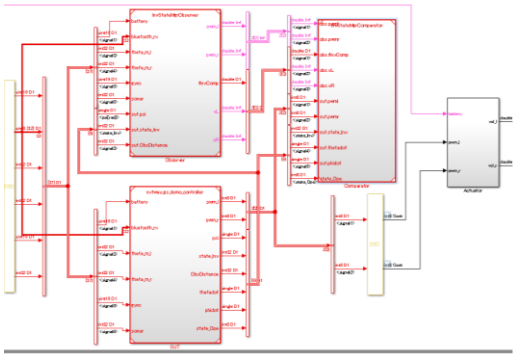
# MBD Technologies

## Plant



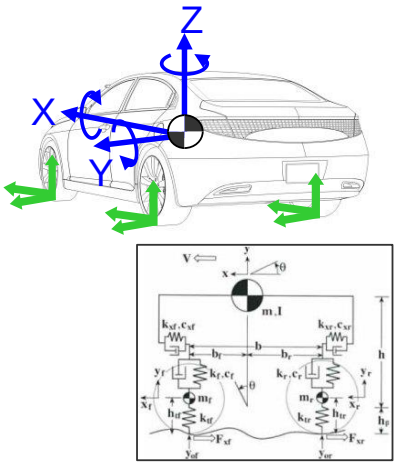



## Controller

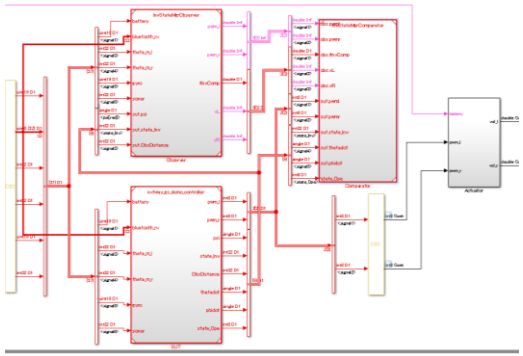


## Simulation

### Plant



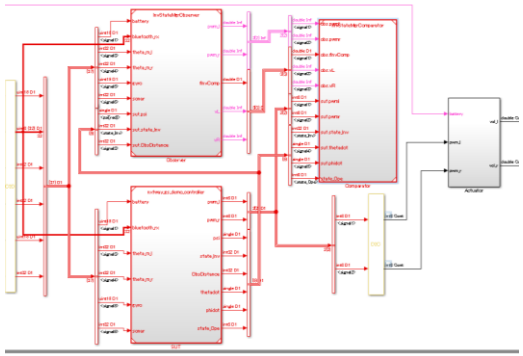

### Controller



- Flexible

## Rapid Prototyping

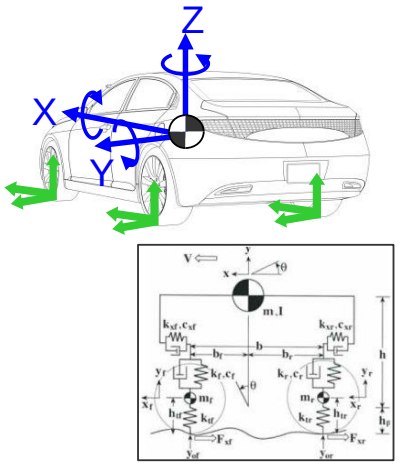
**Plant** **Controller**



- Precise dynamics

## HIL (Hardware in the Loop) simulation

**Plant**




**Controller**

- Reduce space and time
- Extreme condition

# MBD Contribution



- **Efficient development**

  Replace prototype (vehicle …) with simulation

- **Improve performance (quality of control)**

  Flexible setting including extreme conditions

# MBD Side Effect



- **Demand**

  Precision　　plant simulation

  Performance　　designed control

- **Iterative improvement**

  Models → complex and detailed

- **Confusing role**

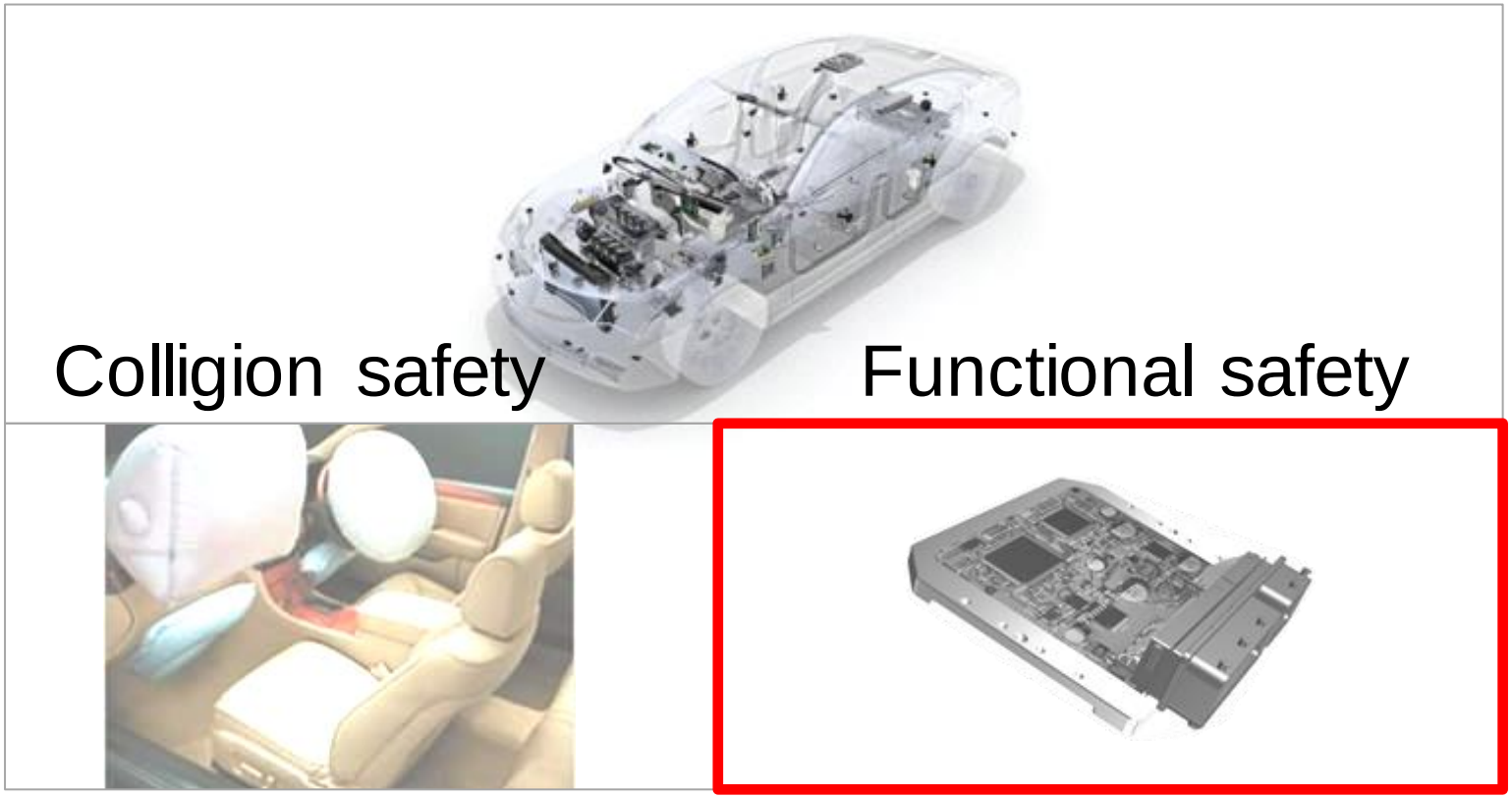
  requirement?　implementation?

  → Need to keep concepts (requirements)

# Functional Safety

## Safety of E/E systems

Vehicle : safety critical various safety (by use case)



# Functional Safety: Background



- **In-vehicle network**

  E/E subsystems connected (2000~)

- **Functional safety for automotive (research)**

  Apply IEC61508 (2000~)

  Advanced development methods

- **ISO26262**

  WG (2005~) → published 2011

- **State**

  without cause of hazard, ideally

  Kill or injury person

- **Social acceptance** ISO/IEC Guide 51

  tolerable (acceptable) risk

$$\text{risk} \triangleq \sum \text{damage} \times \text{probability}$$

- **What is the safest car?**

  ➔ **Safety is prior to functionalities**

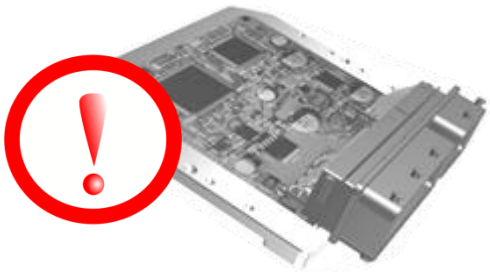
## Safety of E/E systems (ISO26262-1:2011)

absence of **unreasonable risk** due to **hazards** caused by **malfunctioning behaviour** of **E/E systems**

Example

Context: Driving (highway)

Hazard: Air bag(inflating by **failure**)



# Failures (Faults)



- **Random hardware faults**
- **Systematic failures**

  Today's topic → Software

  Focus on "systematic failures" only

## Concept Phase

- Vehicle level use case
- Hazard
- Functional Safety Concept

- Safety requirements
- Architectural Assumption

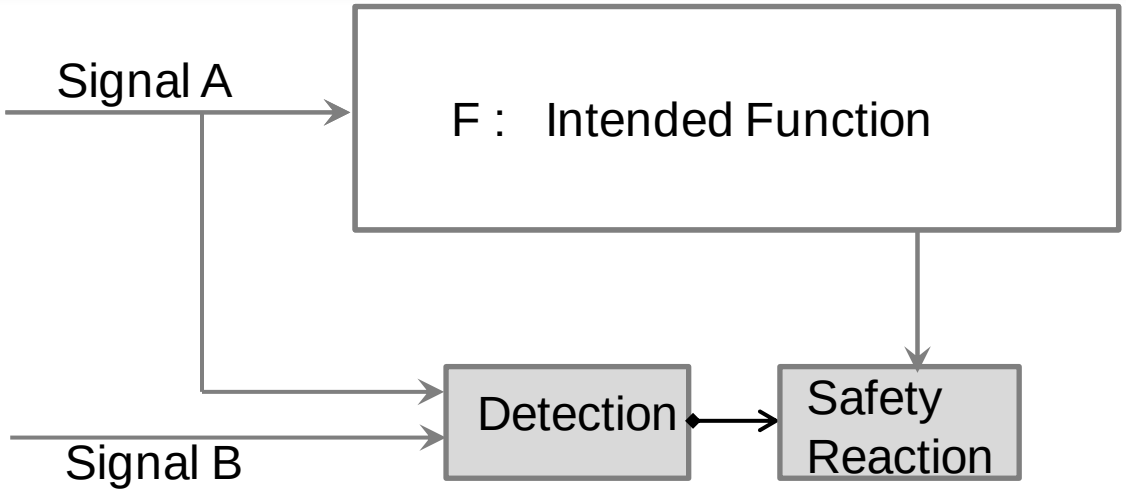
## Design Phase

**Hierarchical Activities**

- Decompose requirements
- Allocate requirements to sub components
- Safety Analysis

# Example



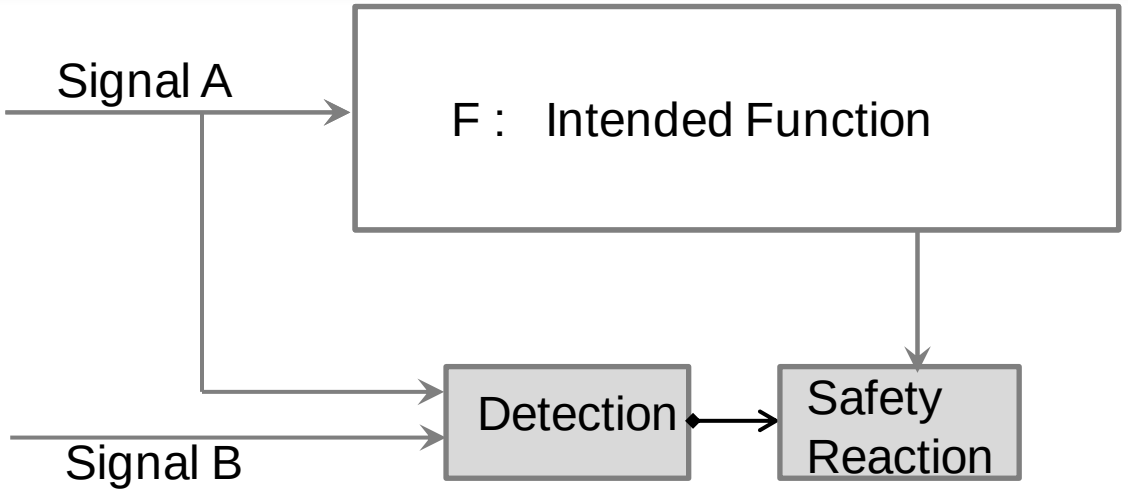



- **Safety Concept**

  use redundancy to detect failure

# Software Safety Requirement



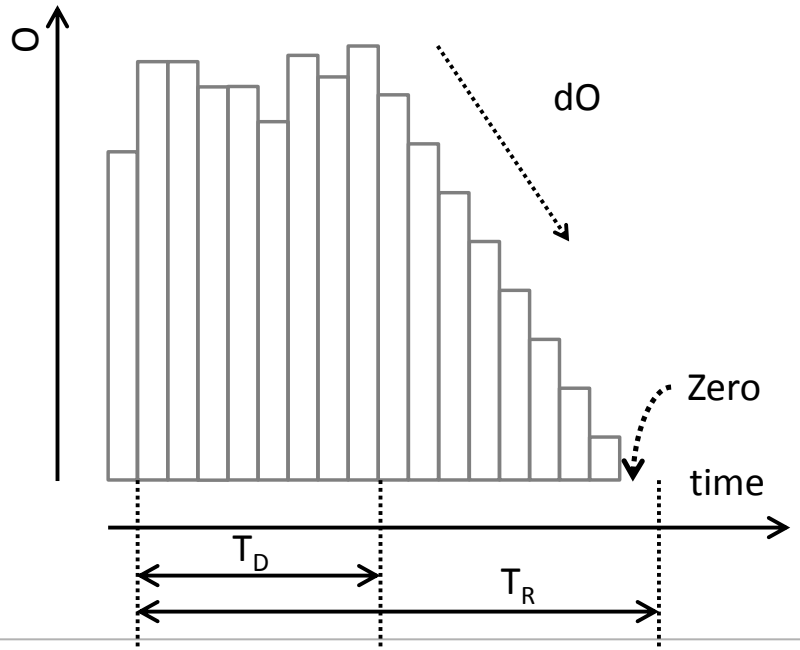



- **Functionalities**

  allocated to Software

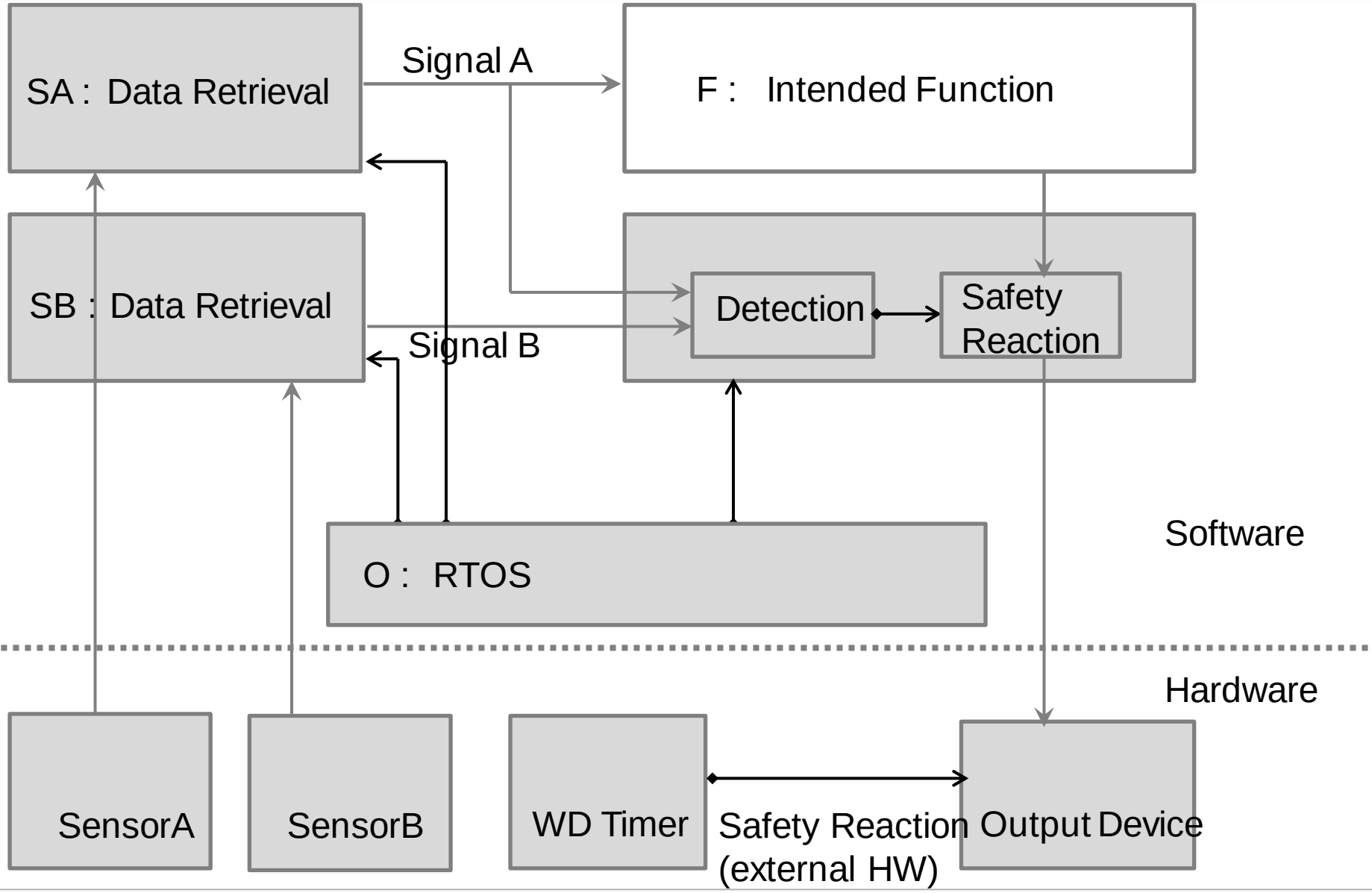
e.g.

“Complete safety reaction within specified time”



# Software Architecture (w/ related HW)





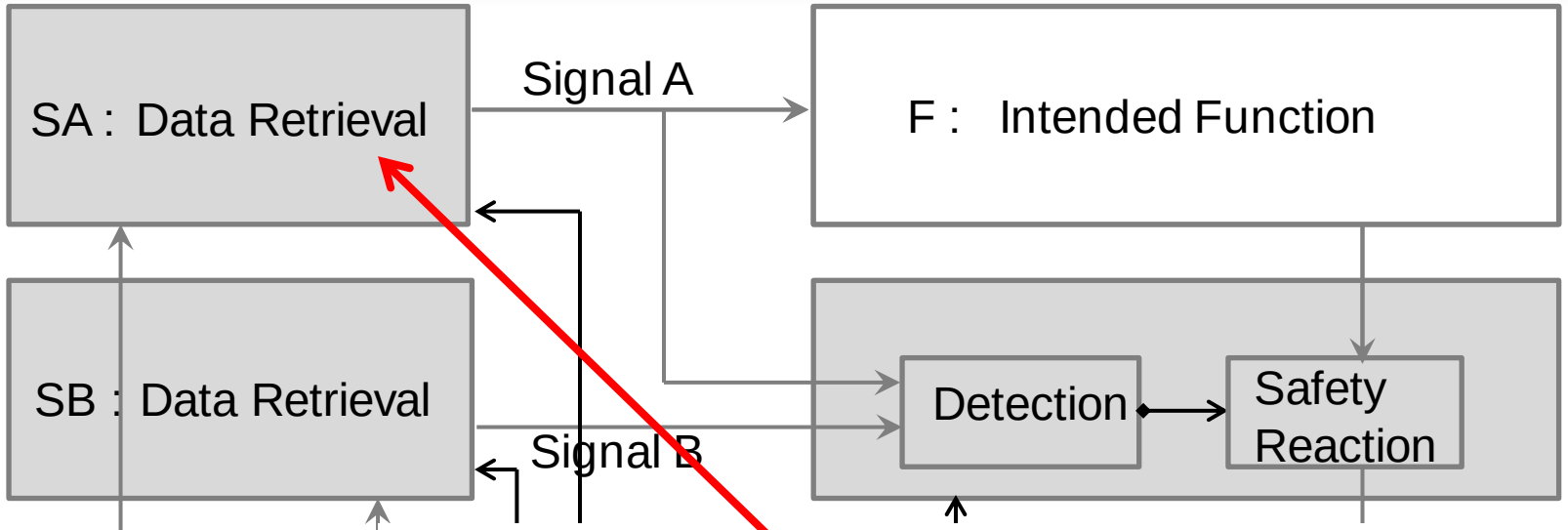
# Software Safety Analysis



- **Consider Failure**
- **Analyze Influence**
  - confirm safety
  - add safety mechanism, if necessary

# Software Safety Analysis






- Div by Zero

→ **add guard**

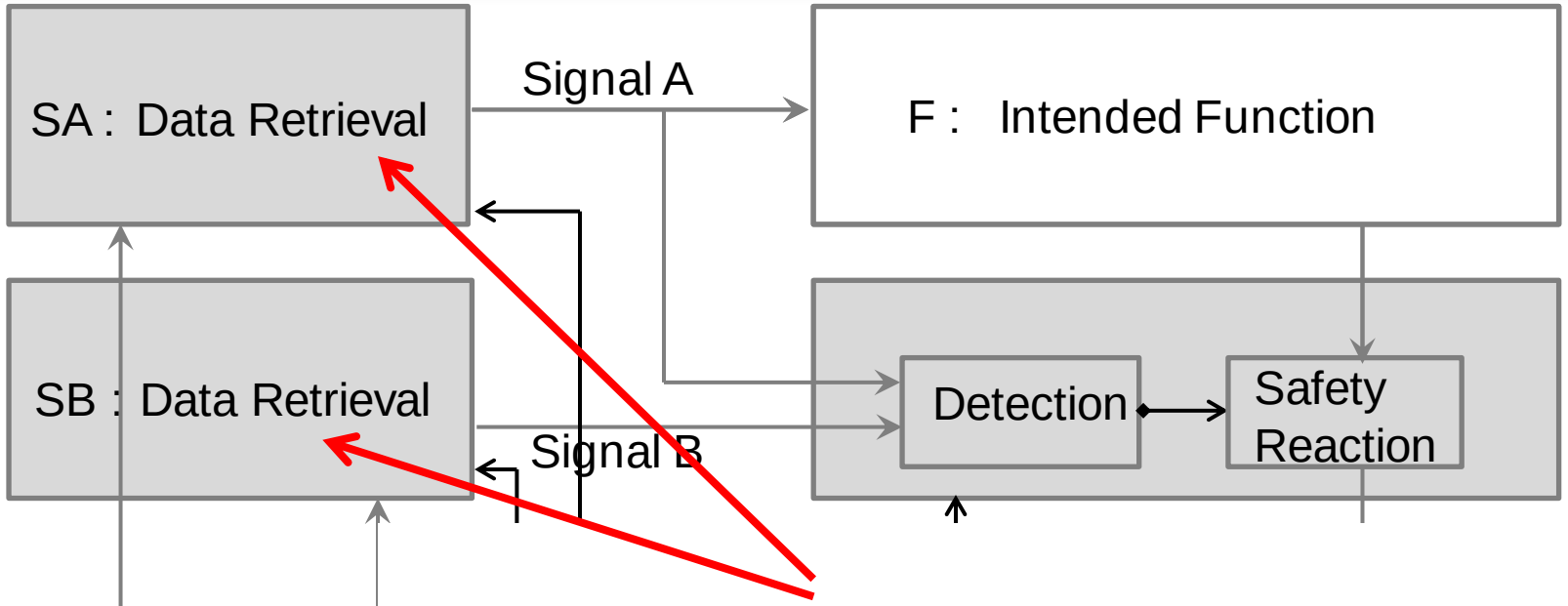
  - covered by design rule
  - static verification

SensorA

(external HW)

# Software Safety Analysis






- Round Error Accumulation

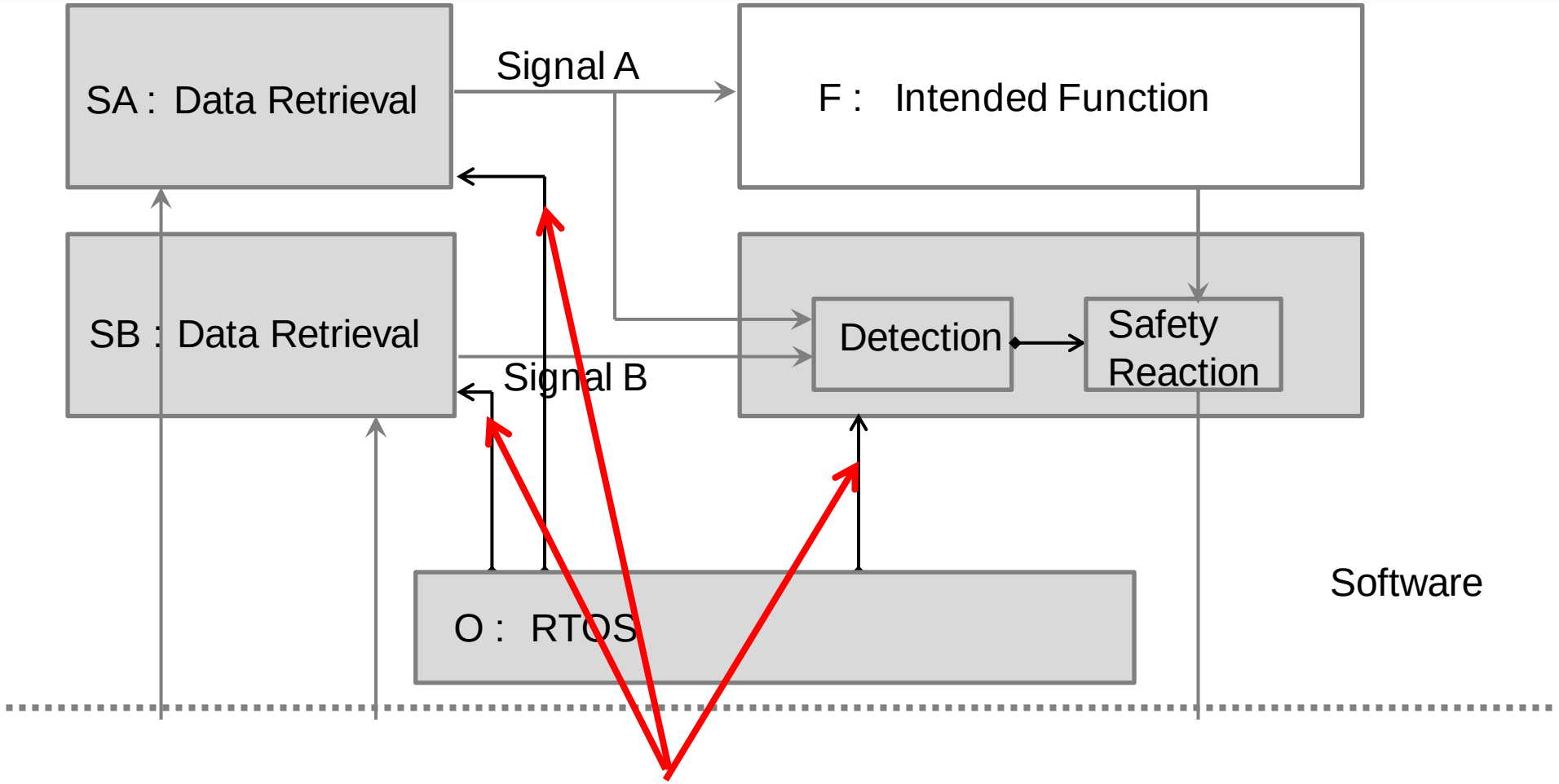
→ **already covered**

  - redundancy with diverse signals

(external HW)

# Software Safety Analysis



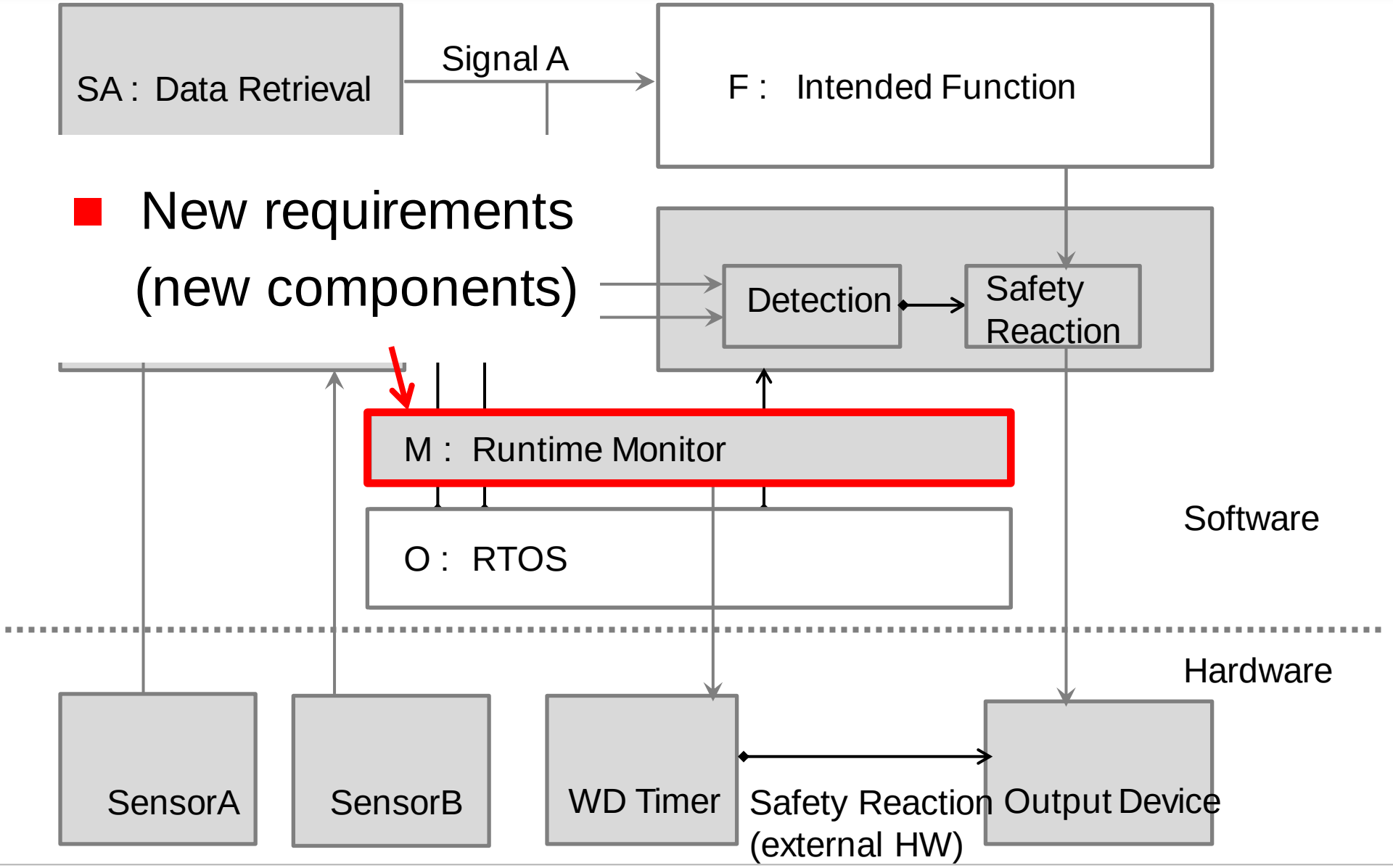



- Slipped Triggers (deadline miss)

(external HW)

# Software Safety Analysis





# Lessons learned from Functional Safety



- **Design → Hierarchical Activities**
  - Decomposition
  - Allocate requirements
- **Component may fail**
  - Not trust too much
- **Analysis at architectural abstraction**
  **→ systematic mitigation using both**
  - Runtime mechanisms
  - Design time measures

# Advanced Topics

# Advanced Technology



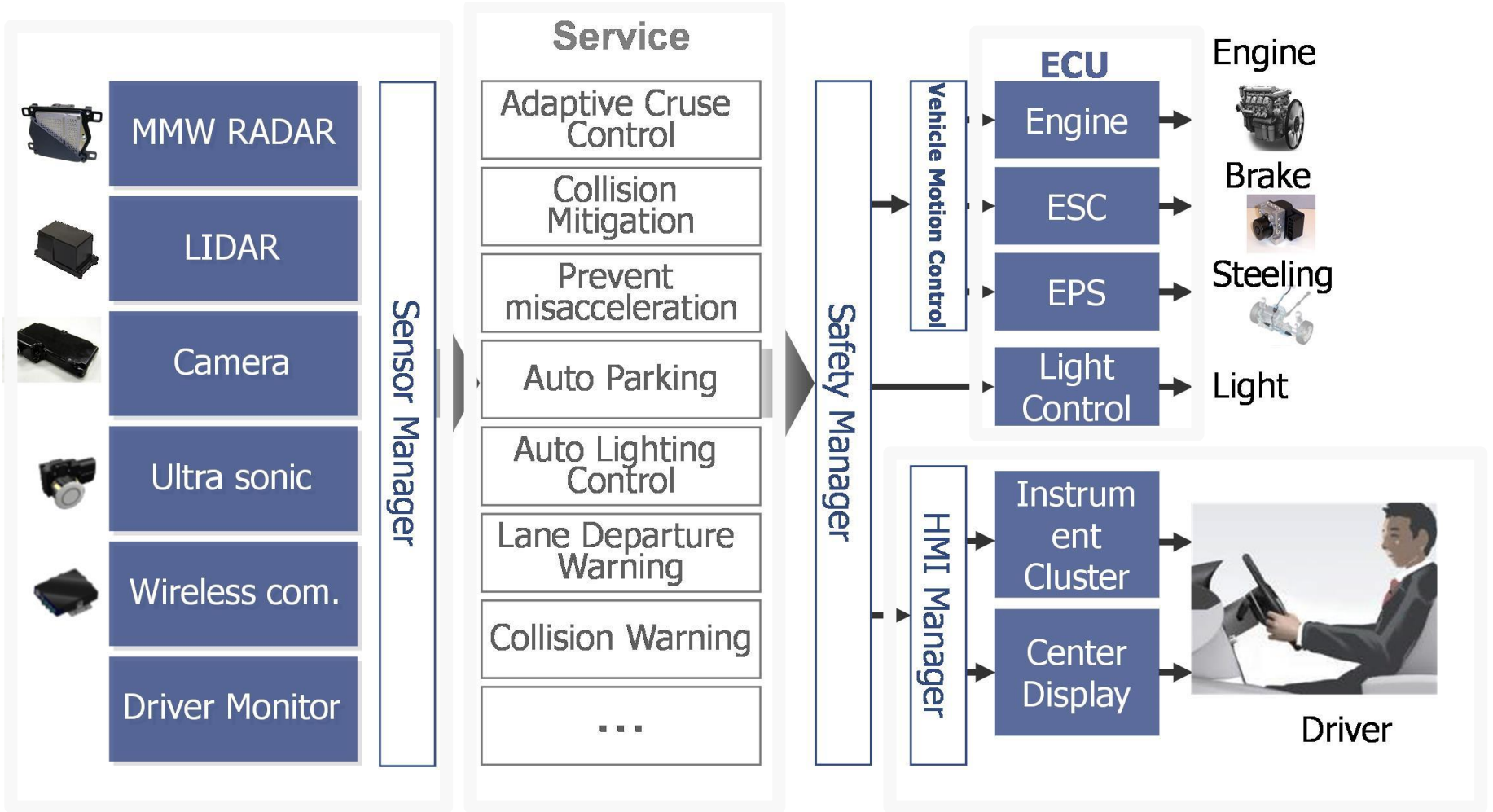
- **ADAS** Advanced Driving Assistance System
- **AD** Automated Driving

**www.denso.com**

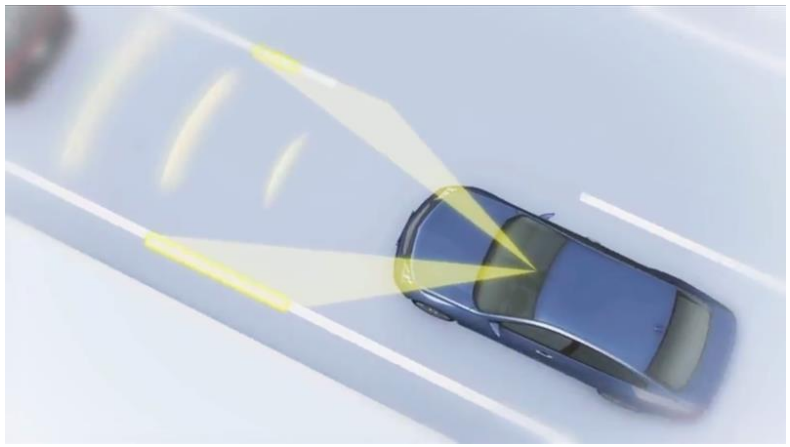
Test on public road & Demonstration driving (2014-)

# ADAS / AD Architecture





## Identifying Objects



Performance required depending on use case

# Level of Automation



Source : SAE J3016™ Sep2016 (simplified)

| SAE Level | | Control | Monitor | Recover | Usecase |
|---|---|---|---|---|---|
| 0 | No Driving Automation | Driver | Driver | Driver | n/a |
| 1 | Driver Assistance | Driver and System | Driver | Driver | Limited |
| 2 | Partial Driving Automation | **System** | Driver | Driver | Limited |
| 3 | Conditional Driving Automation | System | **System** | **Fallback-ready user** | Limited |
| 4 | High Driving Automation | System | System | **System** | Limited |
| 5 | Full Driving Automation | System | System | System | **Unlimited** |

# Various Service



- **Functionalities depend on contexts**

  **same sensors, same actuators,**

  **different use-cases**

  **e.g. highway driving / parking**

# Advanced Technologies: Challenge



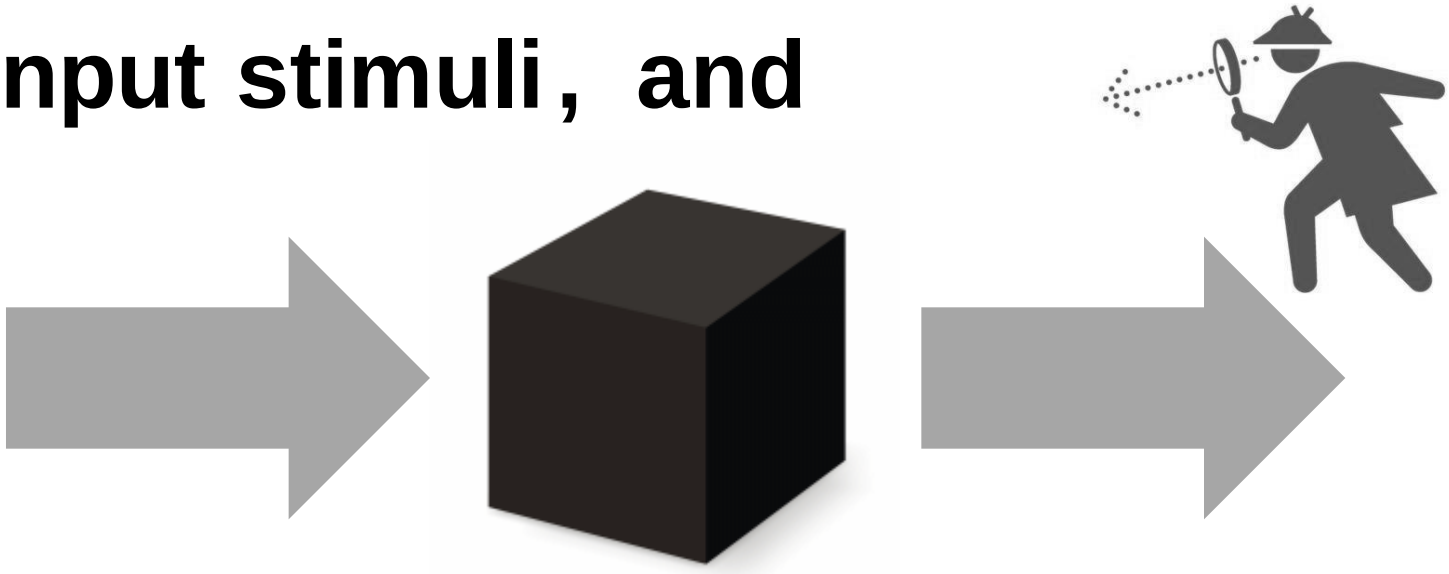
- **Complex Environment**
  - Various Contexts
  - Human Factors

# Testing

# What is testing?



- Input stimuli，and

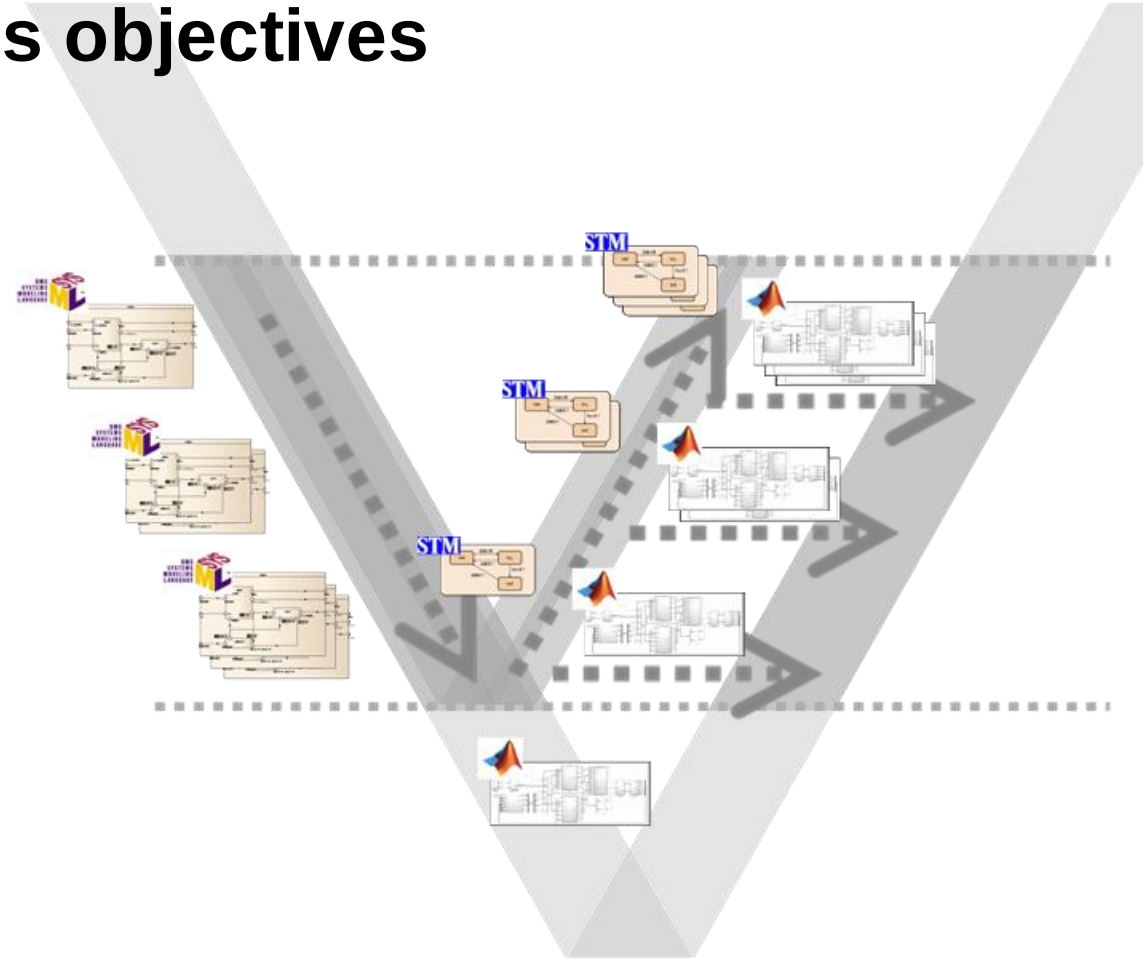


- Observe

# What is testing?



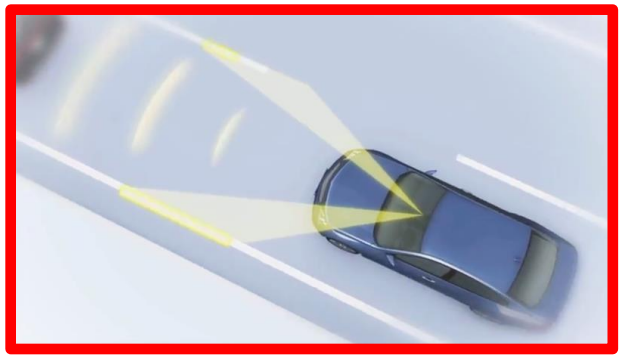
- Input stimuli and observe
- Used for various objectives



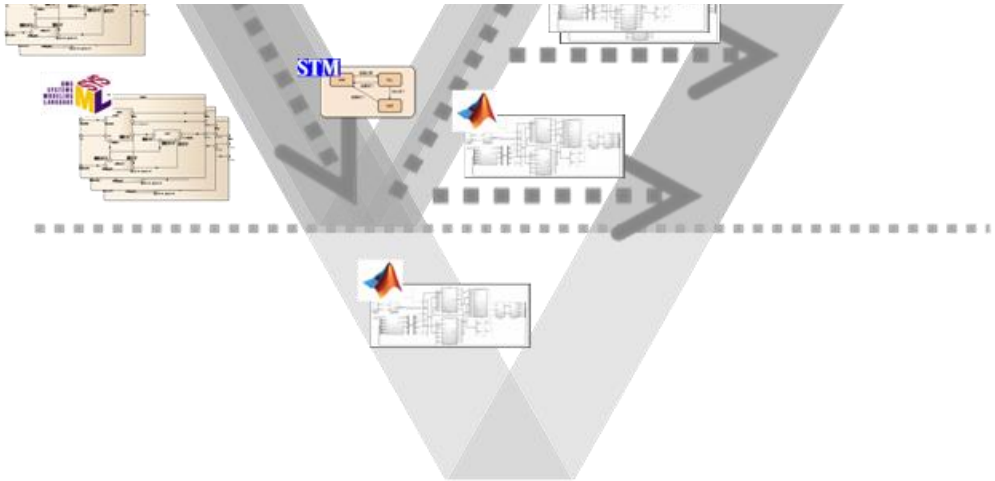
# Testing : Usage



- Input stimuli and observe
- Used for

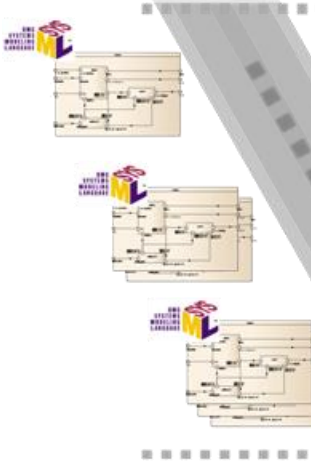


**Prototyping Advanced Dev.**

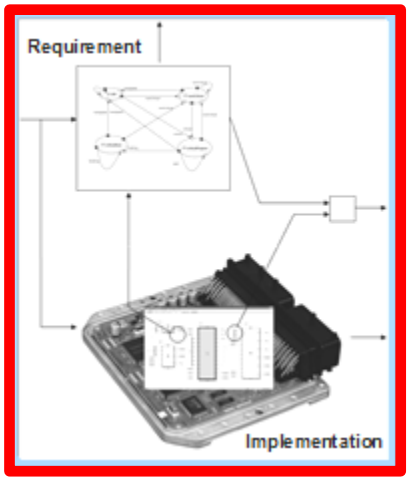


# Testing : Usage

- Input stimuli and observe
- Used for



## Verifying design

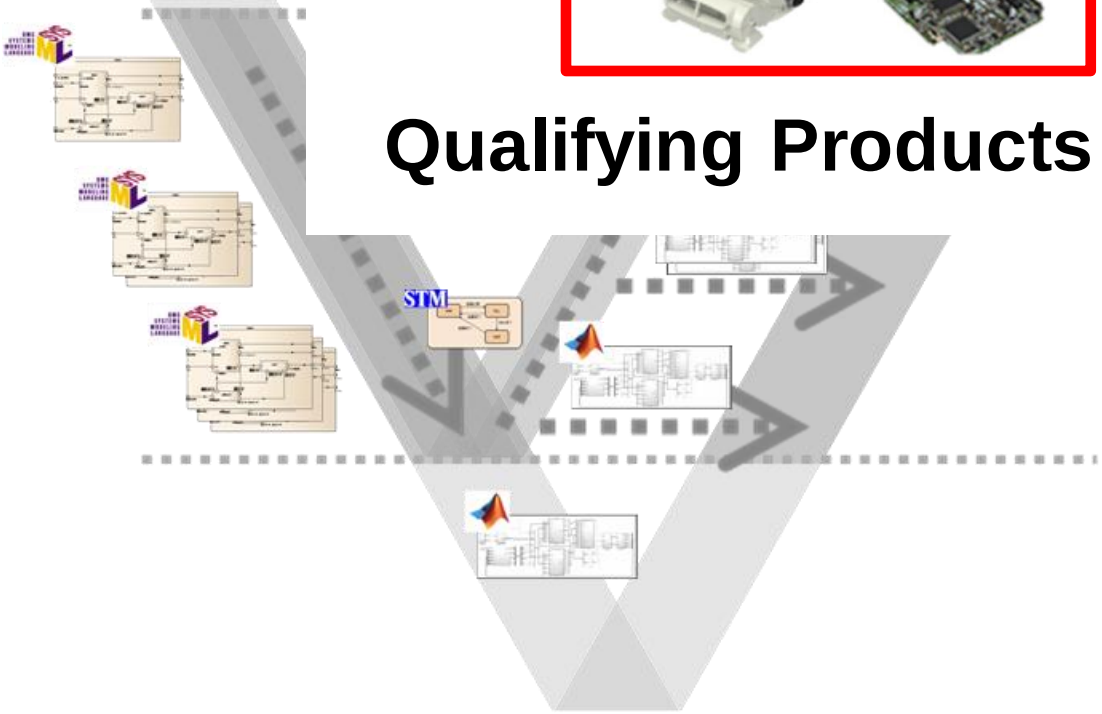


# Testing : Usage



- Input stimuli and observe
- Used for

Qualifying Products



# Testing : Usage



- **Validation**

  Correct Requirements?

**Requirements**

- **Verification**

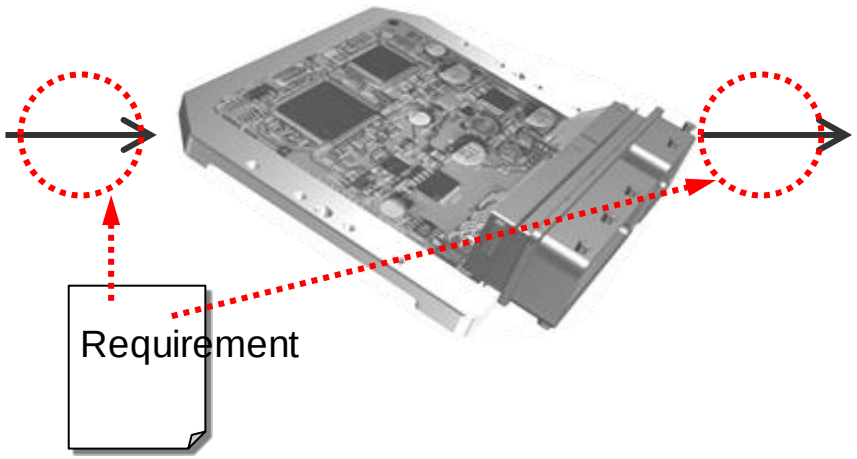
  Correct Product?

# Quality of Requirements



- **Be verifiable**
- **Should specify**

  “What to achieve” for the component




- **In reality,** often “how to calculate”

# Test Oracle



Gaudel, M.C.: Testing can be formal, too.
In: TAPSOFT. LNCS 915 (1995)

- **Selection**

  **Design space of test case**

  with hypothesis

- **Observation**

  **Distinguish correct and bad**

# My View



- **Selection**

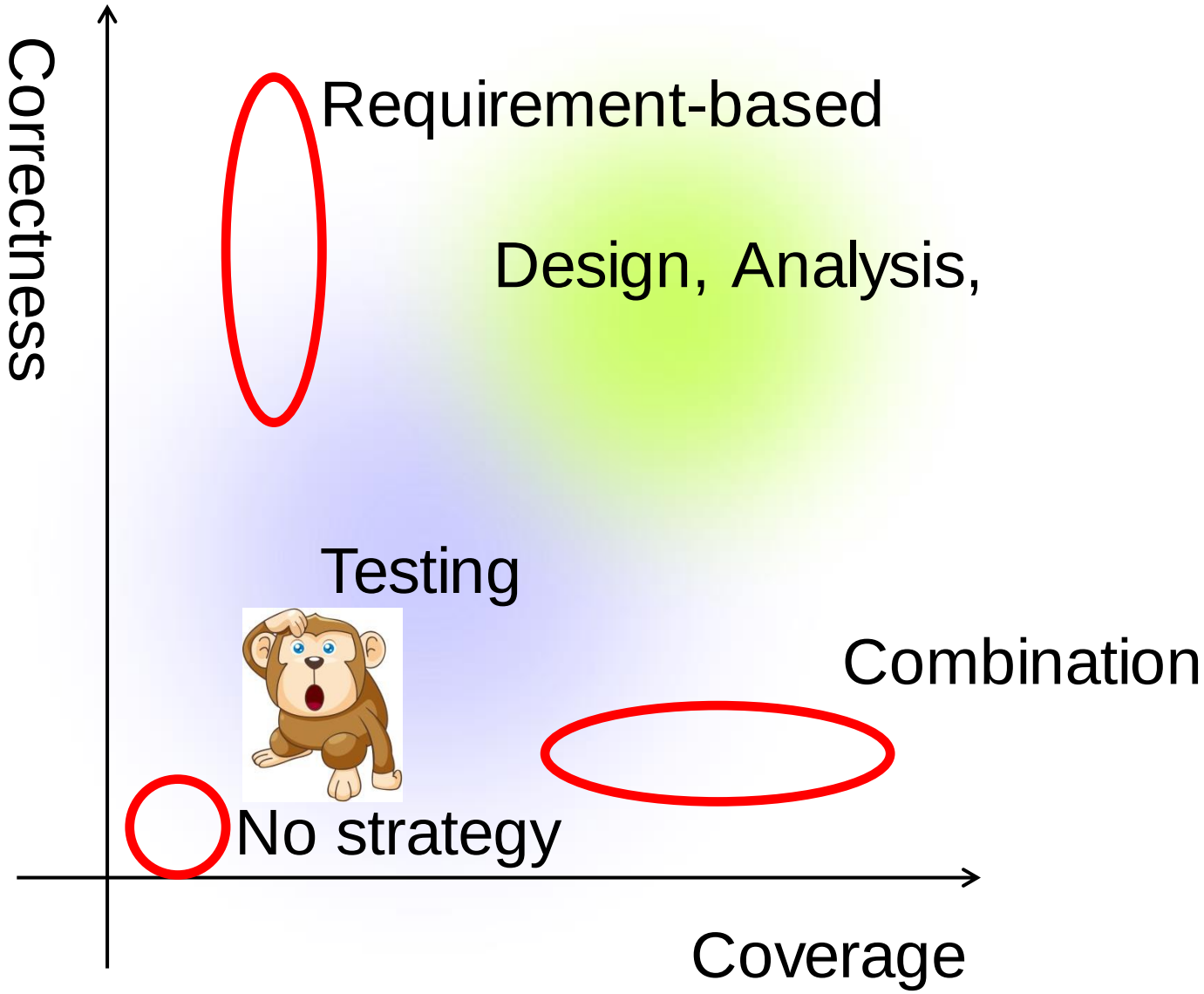
  **Coverage** **(incl. Assumptions)**

- **Observation**

  **Requirements** **(Correctness)**

# Vision





# Summary



- Automotive E/E systems
  **Cyber-Physical Systems**
- Evolution
  **Enabled by Simulation technologies (MBD)**
- Safety
  **Managing complexity by architectural design**
- Advanced challenge
  **Need to overcome complex environment**
- Testing → provide confidence
  **by integration with design and analysis**

## Cyber-Physical Systems

- Embedded software needs safety and reliability
- Software brings new values and services

  → **Various disciplines, related**

- Expertise will contribute in the industrial context, combined with other knowledge

DENSO
Crafting the Core