



**CISTER** - Research Center in  
Real-Time & Embedded Computing Systems

# How Realistic is the Mixed-Criticality Real-Time System Model?

Alexandre Esper, Geoffrey Nelissen, Vincent Nélis, Eduardo Tovar

# Introduction



Current status

MC model gradually gaining in sophistication



# Introduction



## Current status

MC model gradually gaining in sophistication



## Issue

Safety-related standards not freely accessible  
→ many academic works are building on top of previous models and claims

# Introduction



## Current status

MC model gradually gaining in sophistication



## Issue

Safety-related standards not freely accessible  
→ many academic works are building on top of previous models and claims



## Risk

Facilitates the propagation of misconceptions and drift from actual standards requirements



# Introduction



## Current status

MC model gradually gaining in sophistication



## Issue

Safety-related standards not freely accessible  
→ many academic works are building on top of previous models and claims



## Risk

Facilitates the propagation of misconceptions and drift from actual standards requirements



## Contribution

Elaborate on misinterpretations and discuss motivating arguments for future work



# System Design and Development Assurance Process

## Safety-Critical Systems

Aeronautics



Railway



Automotive



Space

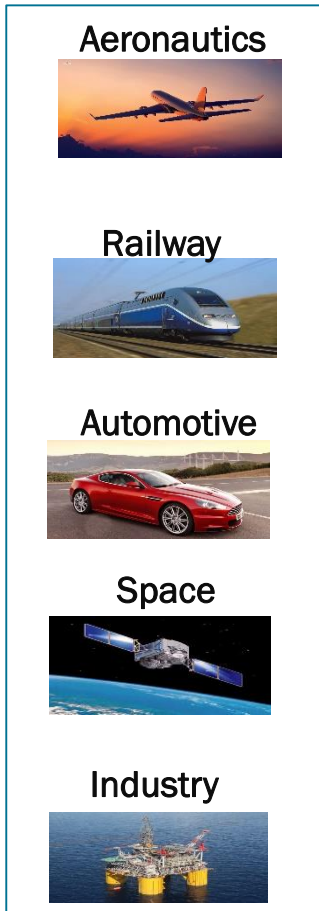


Industry

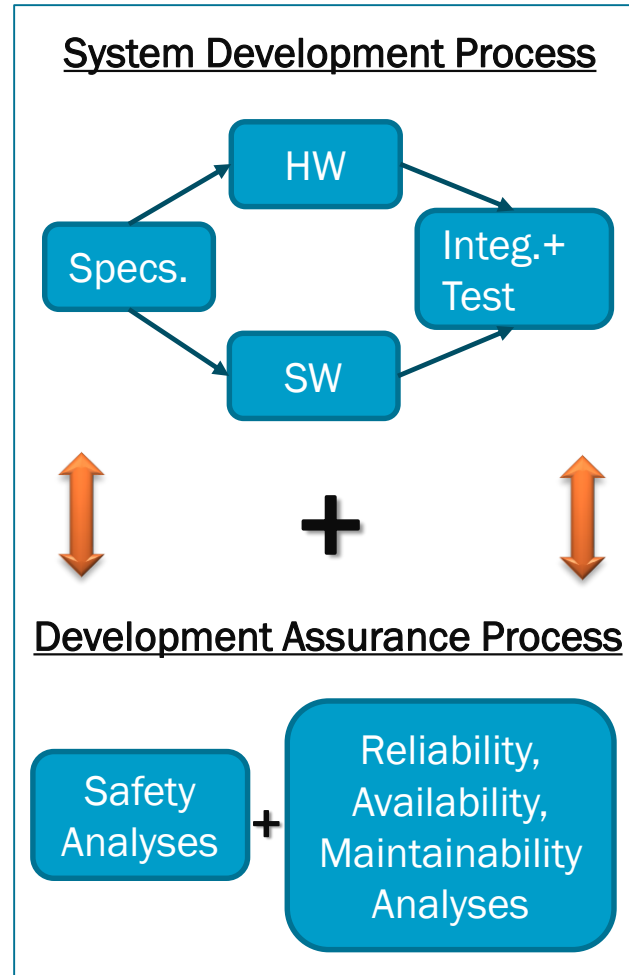


# System Design and Development Assurance Process

## Safety-Critical Systems

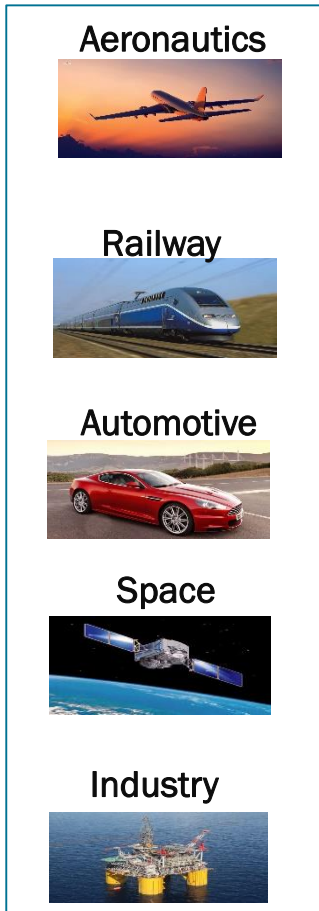


## Safety-Critical Systems Development Process



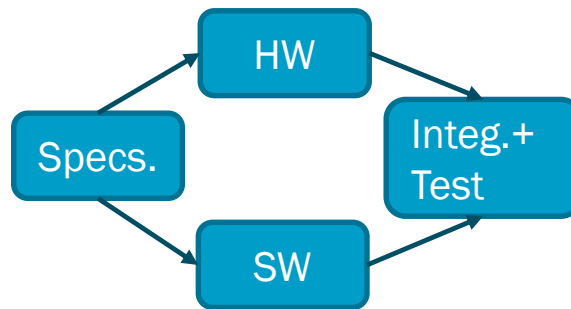
# System Design and Development Assurance Process

## Safety-Critical Systems

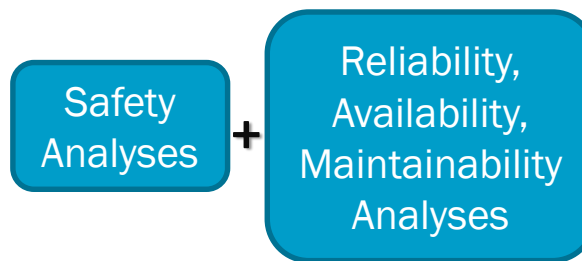


## Safety-Critical Systems Development Process

### System Development Process



### Development Assurance Process



## System Operation



## Certification Process

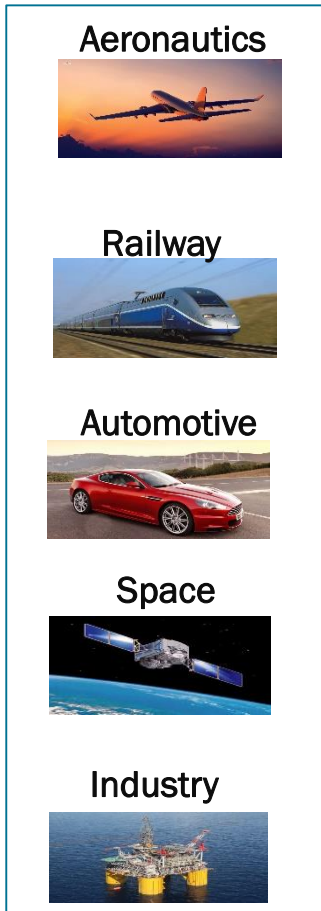
Certification Authority





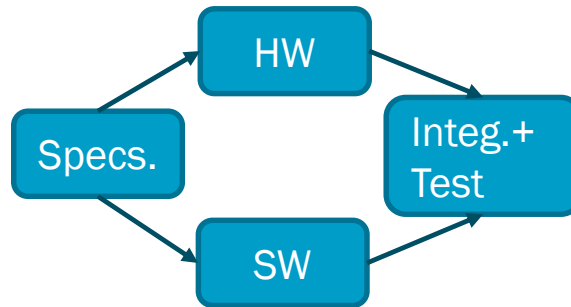
# System Design and Development Assurance Process

## Safety-Critical Systems

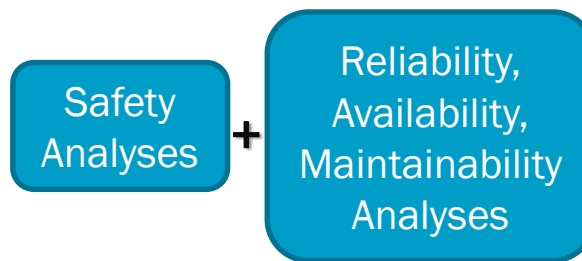


## Safety-Critical Systems Development Process

### System Development Process



### Development Assurance Process

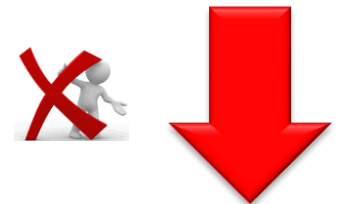


## System Operation



## Certification Process

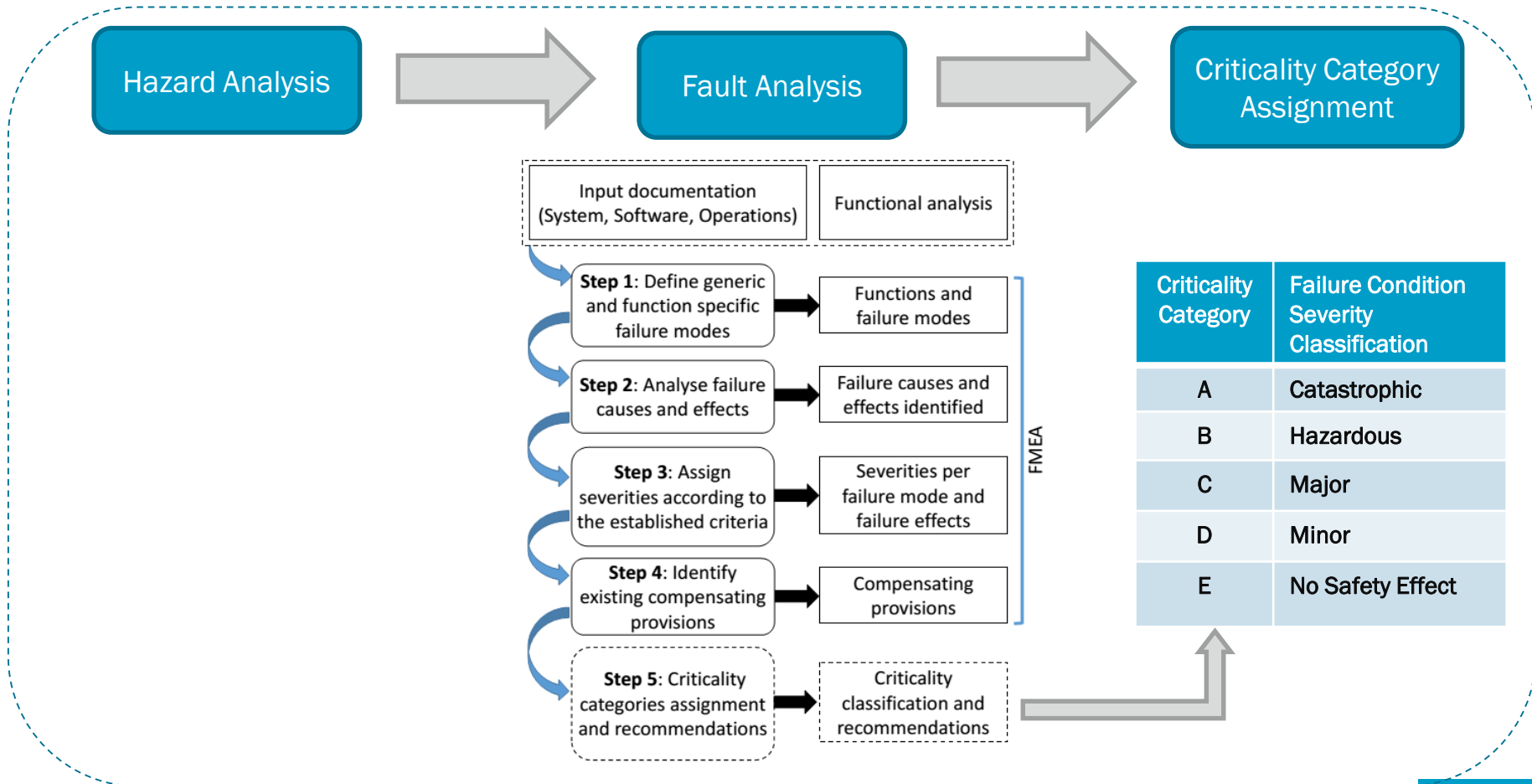
Certification Authority



Redesign

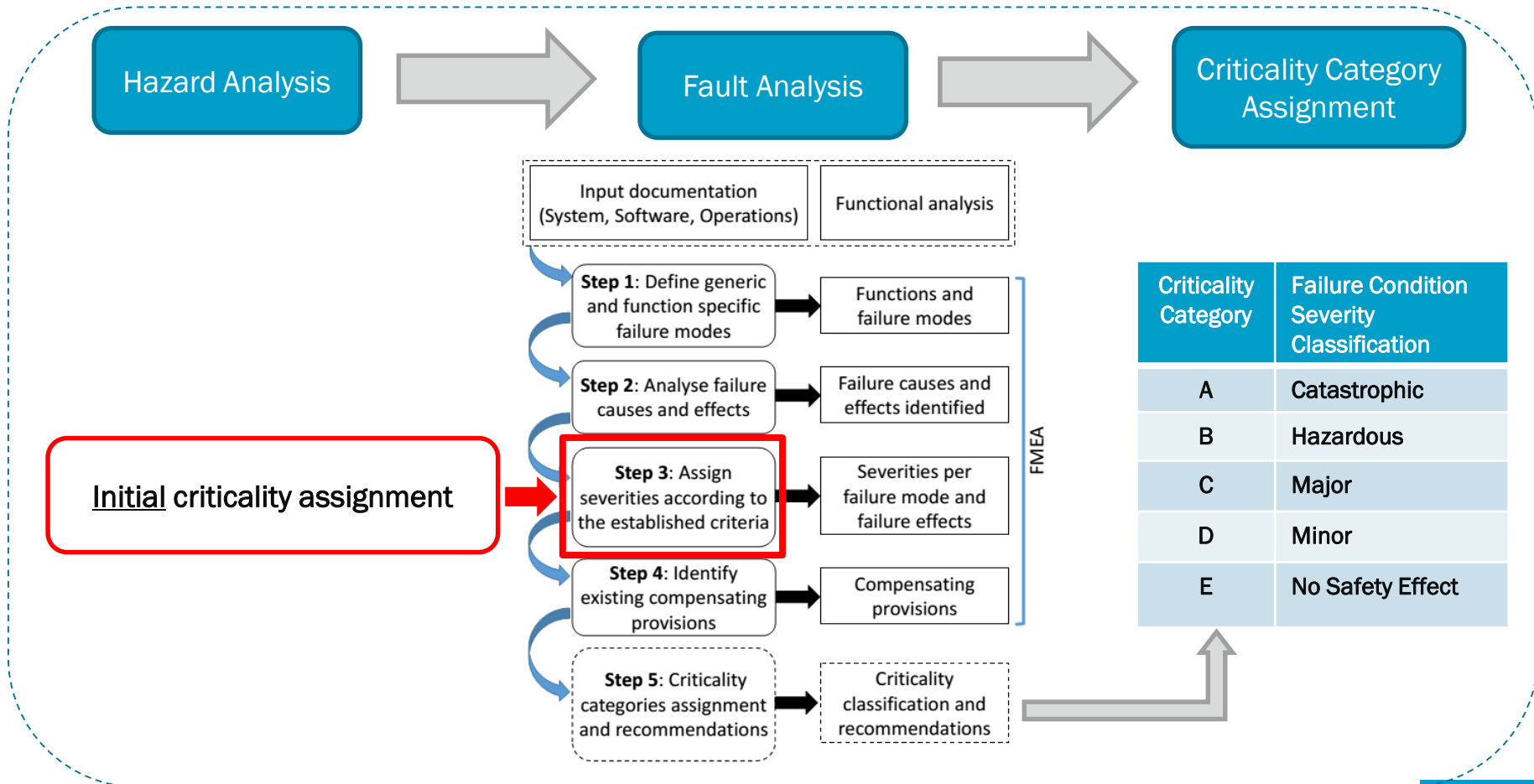
# The Notion of Mixed-Criticality Systems

## System Safety Assessment Process



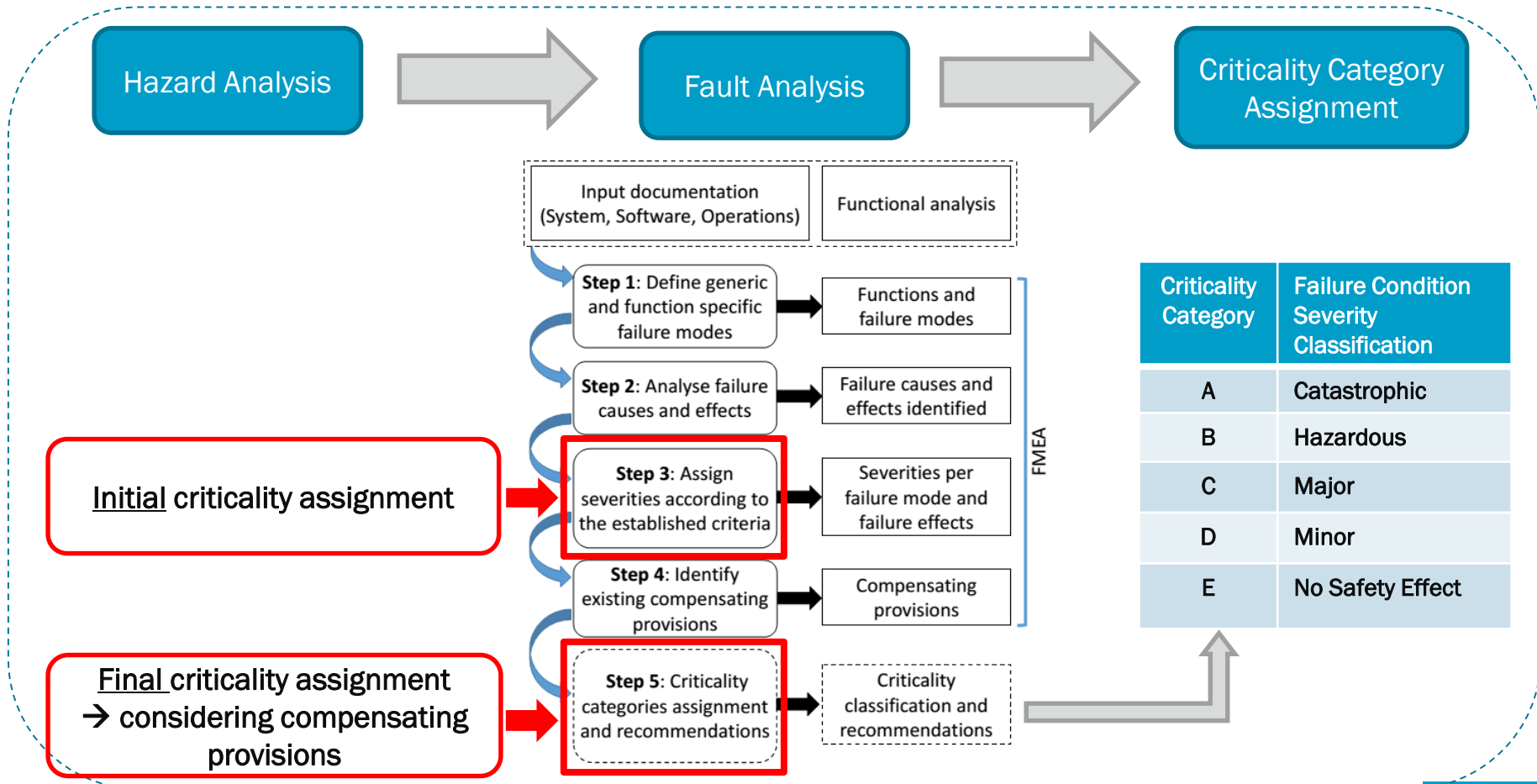
# The Notion of Mixed-Criticality Systems

## System Safety Assessment Process



# The Notion of Mixed-Criticality Systems

## System Safety Assessment Process



# Safety-Related Industrial Standards

## Industry

### IEC 61508

Functional safety of E/E/PE safety-related systems

### IEC 61511

Functional safety – Safety instrumented systems for the process industry sector

### IEC 62061

Safety of machinery – Functional safety of electrical, electronic and programmable electronic control systems

## Aeronautics

### ARP 4761

Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment

### DO-178B/C

Software Considerations in Airborne Systems and Equipment Certification

### ARP 4754

Certification Considerations for Highly-Integrated or Complex Aircraft Systems

### DO-254

Design Assurance Guidance for Airborne Electronic Hardware

## Automotive

### ISO 26262

Road vehicles – Functional safety

Development Assurance – Safety Standards

## Railway

### EN 50126

Railway applications – Specification and demonstration of reliability, availability, maintainability and safety

### EN 50128

Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems

### EN 50129

Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling

## Space

### ECSS series

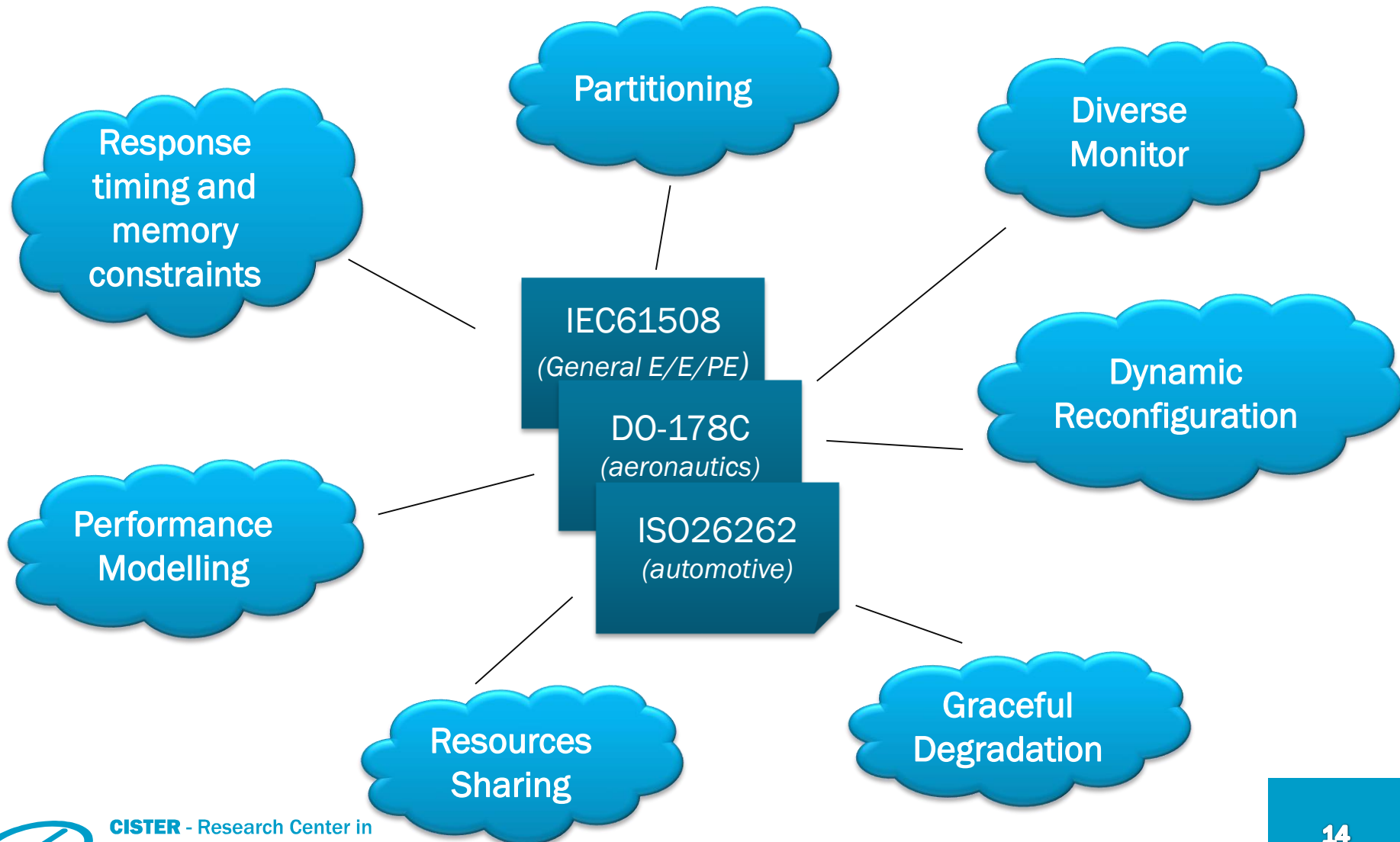
Processes for project management, engineering and product assurance in space projects and applications

### NASA-STD-8719.13B

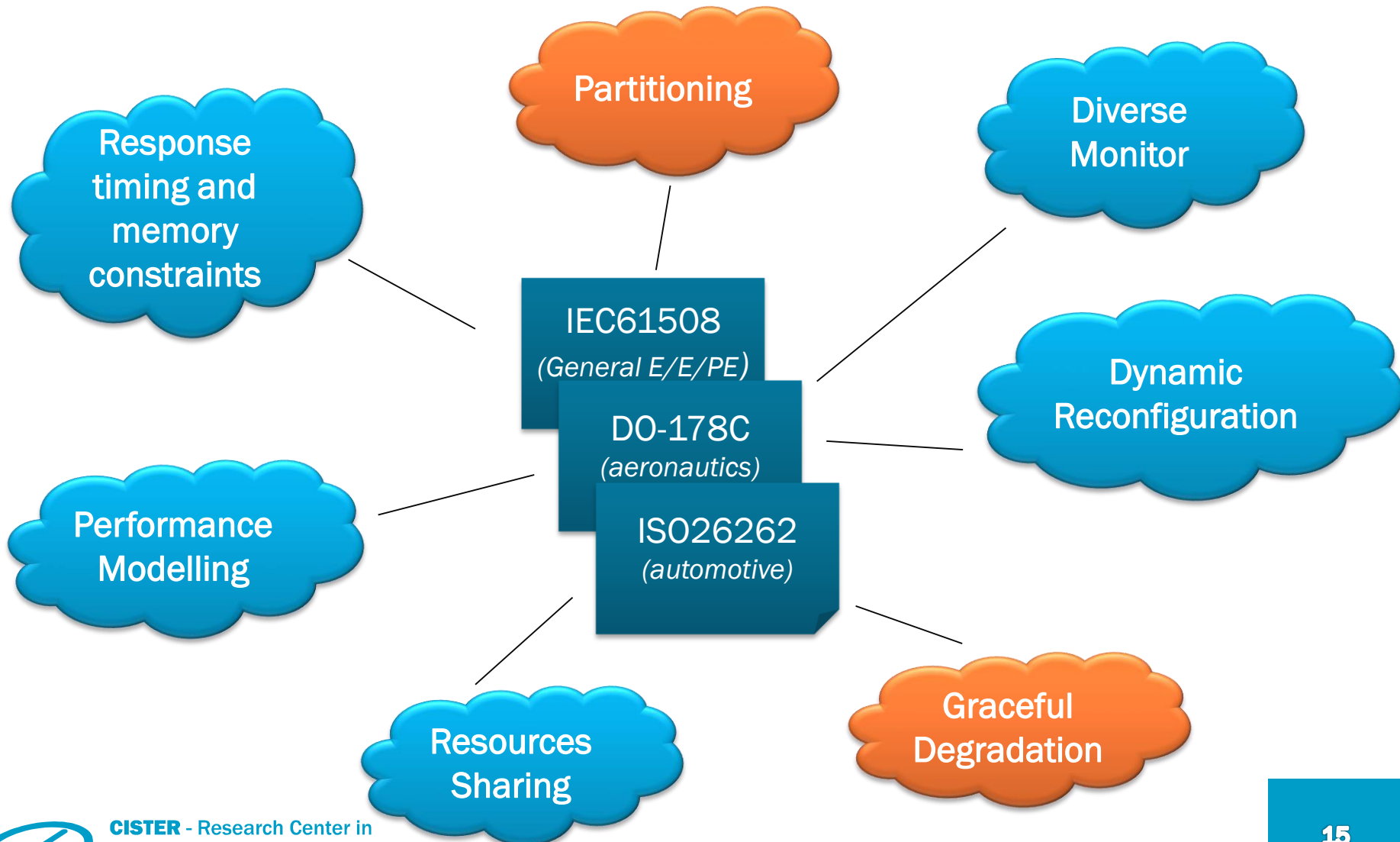
Software Safety Standard – NASA Technical Standard



# Requirements of Safety-Related Industrial Standards



# Requirements of Safety-Related Industrial Standards





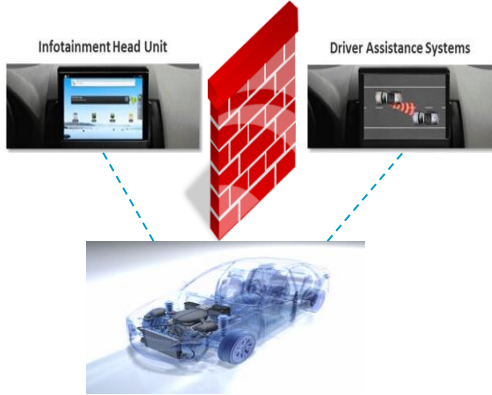
# MCS and the Challenge of Compliance to Safety-related Standards

Safety-related Industrial Standards



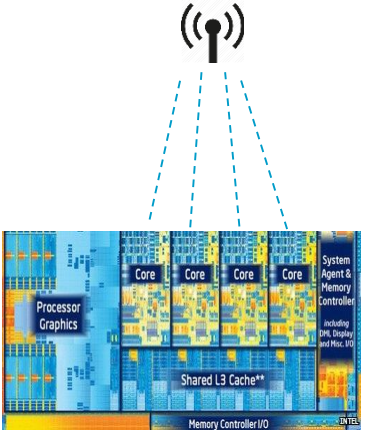
No explicit MCS requirements

...but specify stringent safety requirements



→ *isolation* and *independence* between applications.

Additional challenges



Multicore + Shared Resources



# MCS and the Challenge of Compliance to Safety-related Standards

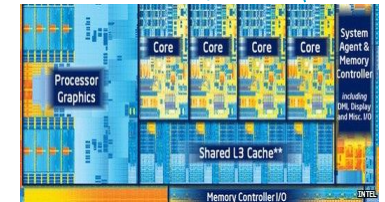
Safety-related Industrial Standards



...but specify stringent **safety requirements**



Additional challenges



Multicore + Shared Resources

Industrial Solutions  
**ARINC-653 & AUTOSAR**

→ *isolation and independence* between applications.

# The Theoretical MC Model and its Common Misconceptions

Most MCS works are Based on the Vestal Model:



Several modes of execution ( $1, 2, \dots, L$ )

tasks  $\rightarrow$  period, deadline, WCET and an assurance level

System running in mode  $k$

Budget of a task is overshot

System switches to mode  $k + 1$

All the tasks of criticality not greater than  $k$  are suspended (potentially reactivated)

# The Use of the Word “Function”

## Safety-related Industrial Standards



- Used at system level
- System functionality (HW + SW)

“Function”



## Academic Publications



- Used like a pure SW function
- E.g.: C function or real-time task

# Mismatch of Interpretation of the Concept of “System Criticality”

## Safety-related Industrial Standards



- Level of assurance (e.g. DAL, SIL,...)
- Safety functions

“System Criticality”



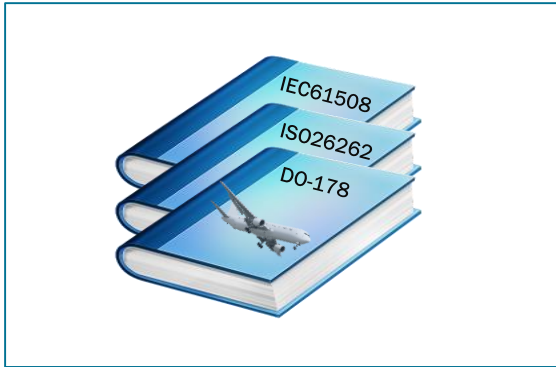
## Academic Publications



- Based on Vestal
- Modes of execution
- E.g. high and low criticality

# The Misalignment of Terminology

Safety-related  
Industrial Standards



≠

Academic  
Publications



Although **not fundamentally wrong**, it creates confusion in the context of industrial MCS

→ leads the two communities to **misunderstand** each others' work

# Confusion Between the Notions of Criticality and Importance

## Function 1

**Severity:**

Car unusable

**Probability:**

Probable

**Controllability:**

Driver can keep the car on the road



## Function 2

**Severity:**

Car unusable

**Probability:**

Probable

**Controllability:**

Uncontrollable





# Confusion Between the Notions of Criticality and Importance

## Function 1

### Severity:

Car unusable

### Probability:

Probable

### Controllability:

Driver can keep  
car on the road

## Function 2

### Severity:

Car unusable

### Probability:

Probable

### Controllability:

Uncontrollable

Both are important!



# Confusion Between the Notions of Criticality and Importance

## Function 1

**Severity:**

Car unusable

**Probability:**

Probable

**Controllability:**

Driver can keep the car on the road

## Function 2

**Severity:**

Car unusable

**Probability:**

Probable

**Controllability:**

Uncontrollable

**But ...**

**ASIL =  
Severity +  
Probability +  
Controllability**

Severity class	Probability class	Controllability class		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D



# Confusion Between the Notions of Criticality and Importance

## Function 1

ASIL C

### Severity:

Car unusable

### Probability:

Probable

### Controllability:

Driver can keep the car on the road

## Function 2

### Severity:

Car unusable

### Probability:

Probable

### Controllability:

Uncontrollable

But ...

ASIL =  
Severity +  
Probability +  
Controllability

Severity class	Probability class	Controllability class		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D



# Confusion Between the Notions of Criticality and Importance

Function 1

ASIL C

Severity:

Car unusable

Probability:

Probable

Controllability:

Driver can keep the car on the road

Function 2

ASIL D

Severity:

Car unusable

Probability:

Probable

Controllability:

Uncontrollable

But ...

ASIL =  
Severity +  
Probability +  
Controllability

Severity class	Probability class	Controllability class		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D



# Confusion Between the Notions of Criticality and Importance

## Function 1

ASIL C

### Severity:

Car unusable

### Probability:

Probable

### Controllability:

Driver can keep the car on the road

## Function 2

ASIL D

### Severity:

Car unusable

### Probability:

Probable

### Controllability:

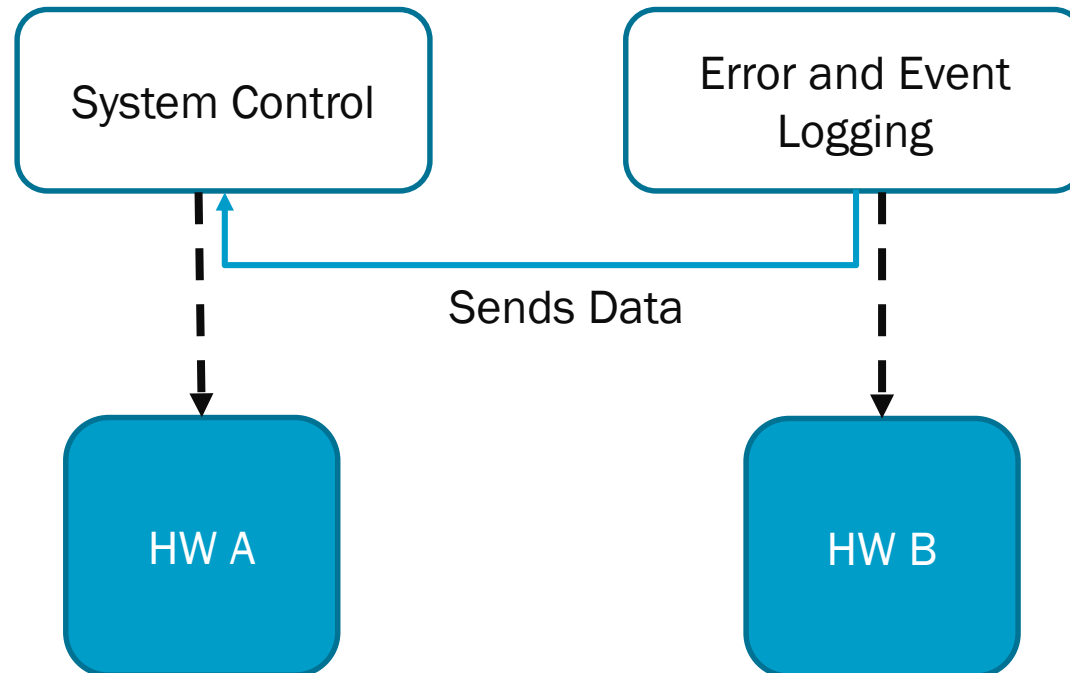
Driver can keep the car on the road

We cannot always stop lower criticality tasks in favour of higher criticality ones



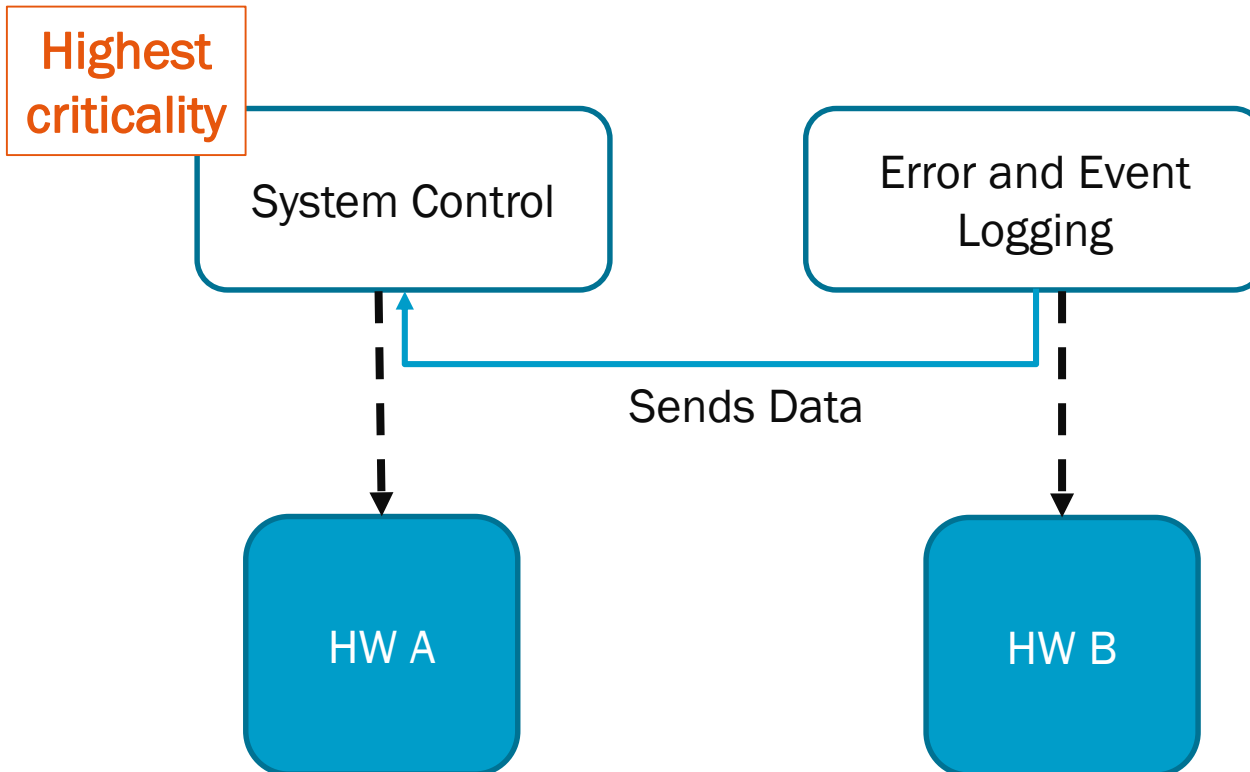
# Confusion Between the Notions of Criticality and Importance

Example from IEC61508



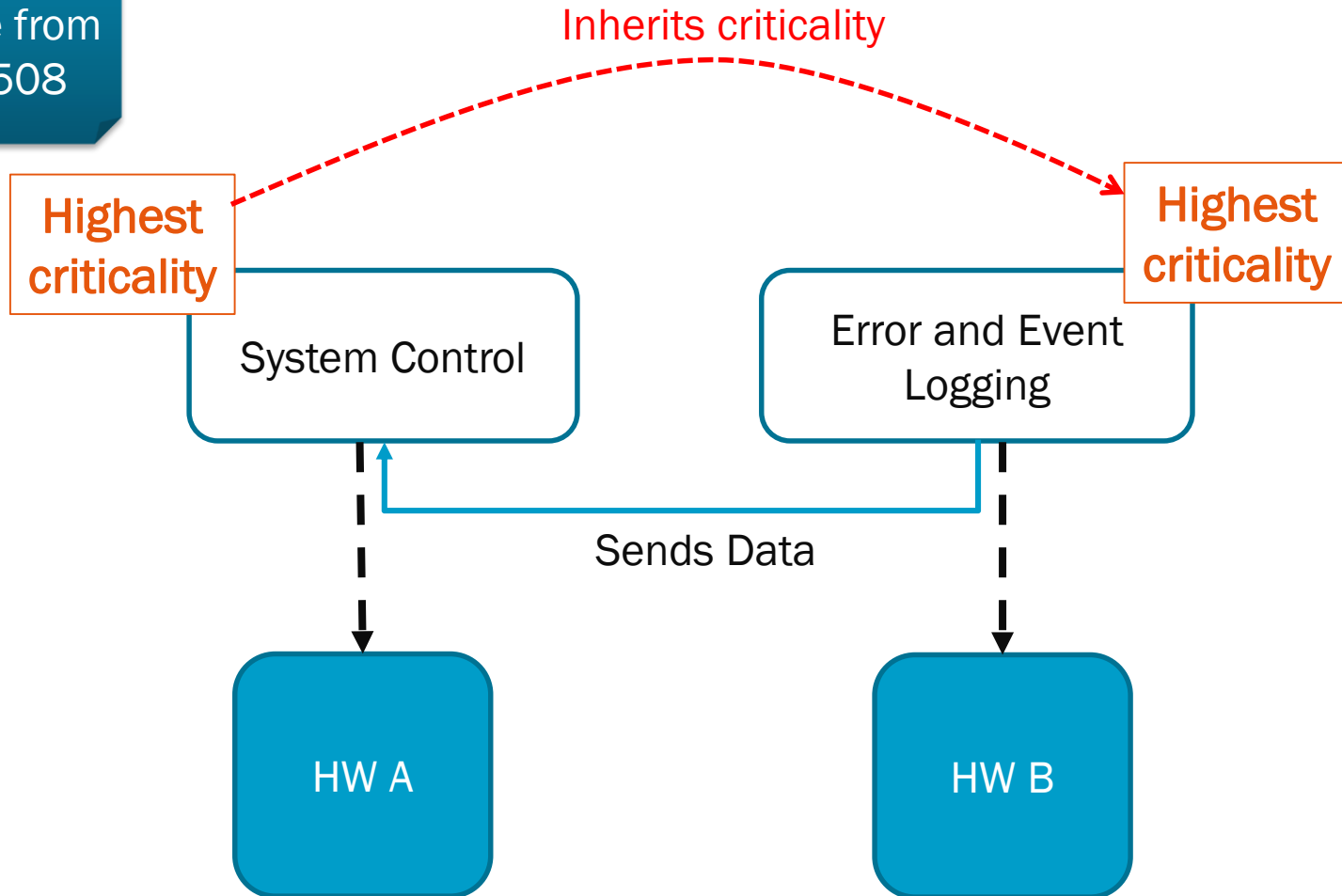
# Confusion Between the Notions of Criticality and Importance

Example from IEC61508



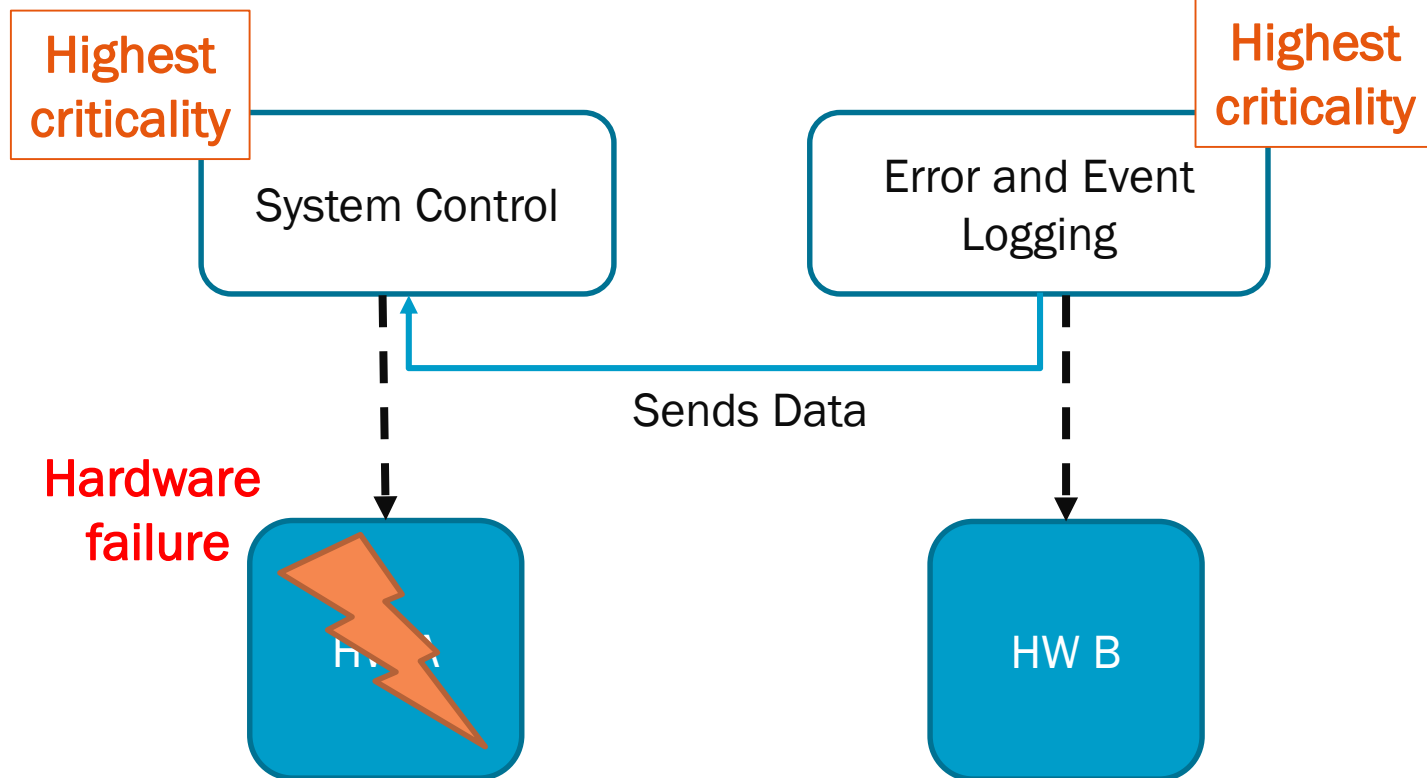
# Confusion Between the Notions of Criticality and Importance

Example from IEC61508



# Confusion Between the Notions of Criticality and Importance

Example from IEC61508



# Confusion Between the Notions of Criticality and Importance

Example from IEC61508

Highest  
criticality

System Control

STOP

Highest  
criticality

Error and Event  
Logging

Hardware  
failure

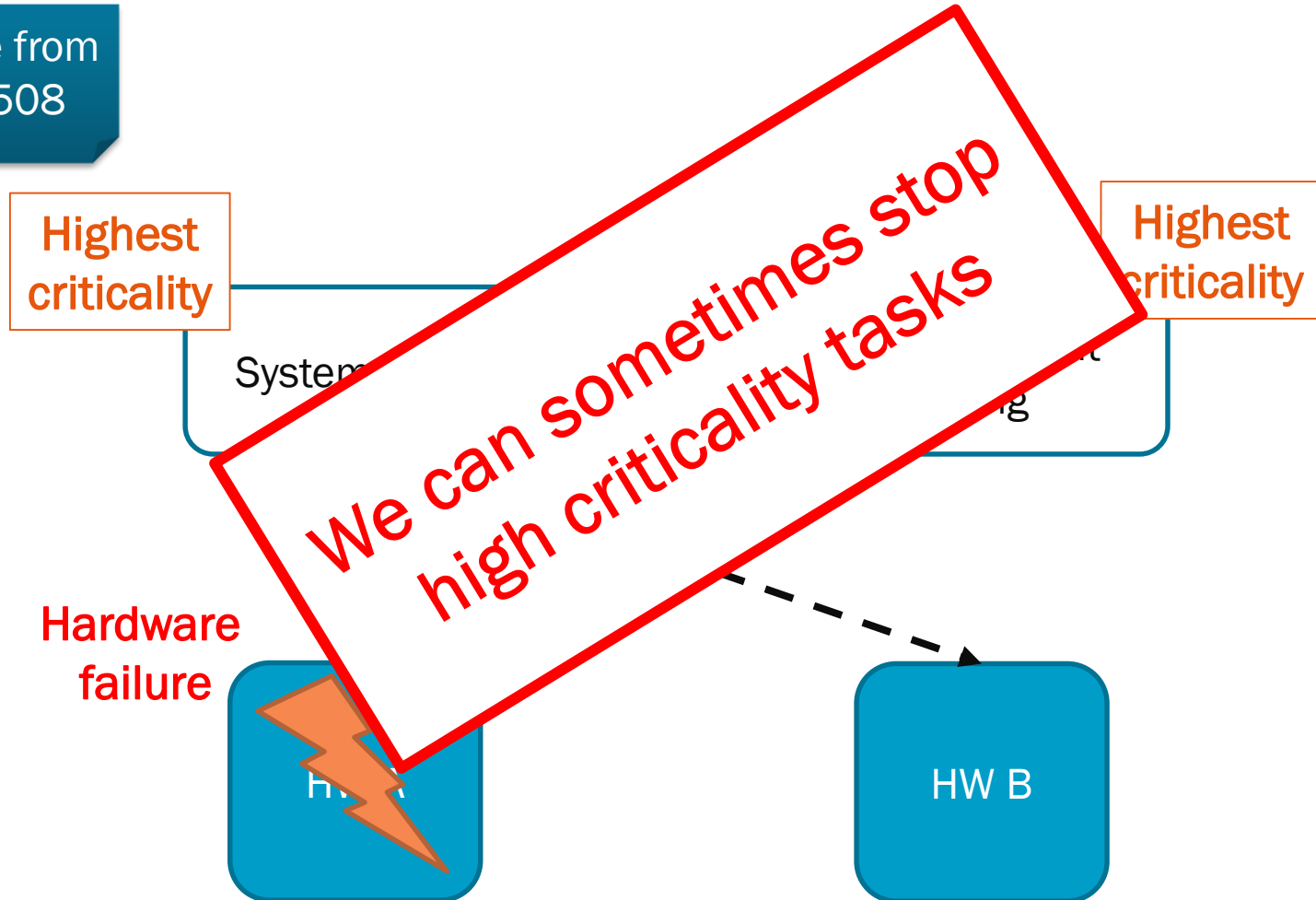


HW B



# Confusion Between the Notions of Criticality and Importance

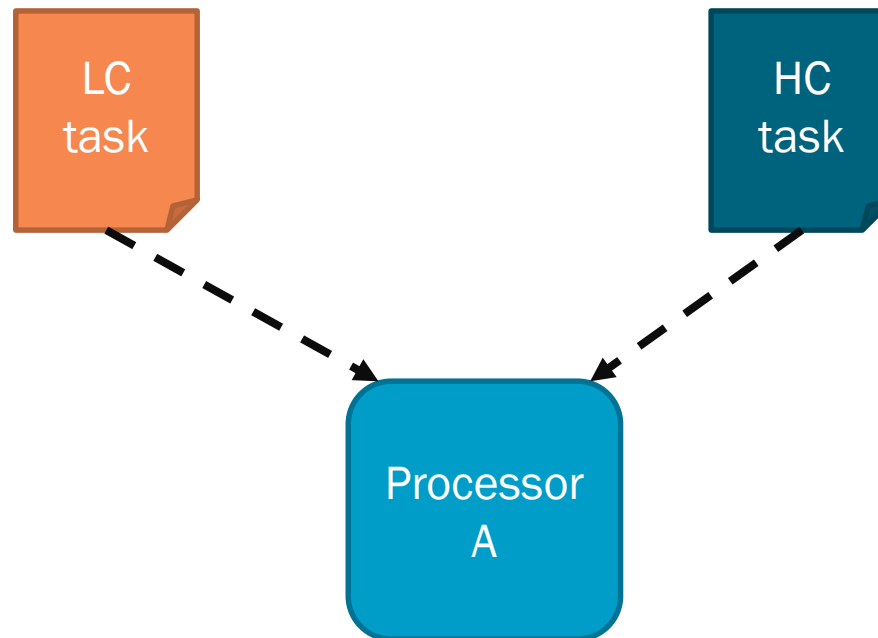
Example from IEC61508



# Vestal's Model and Isolation

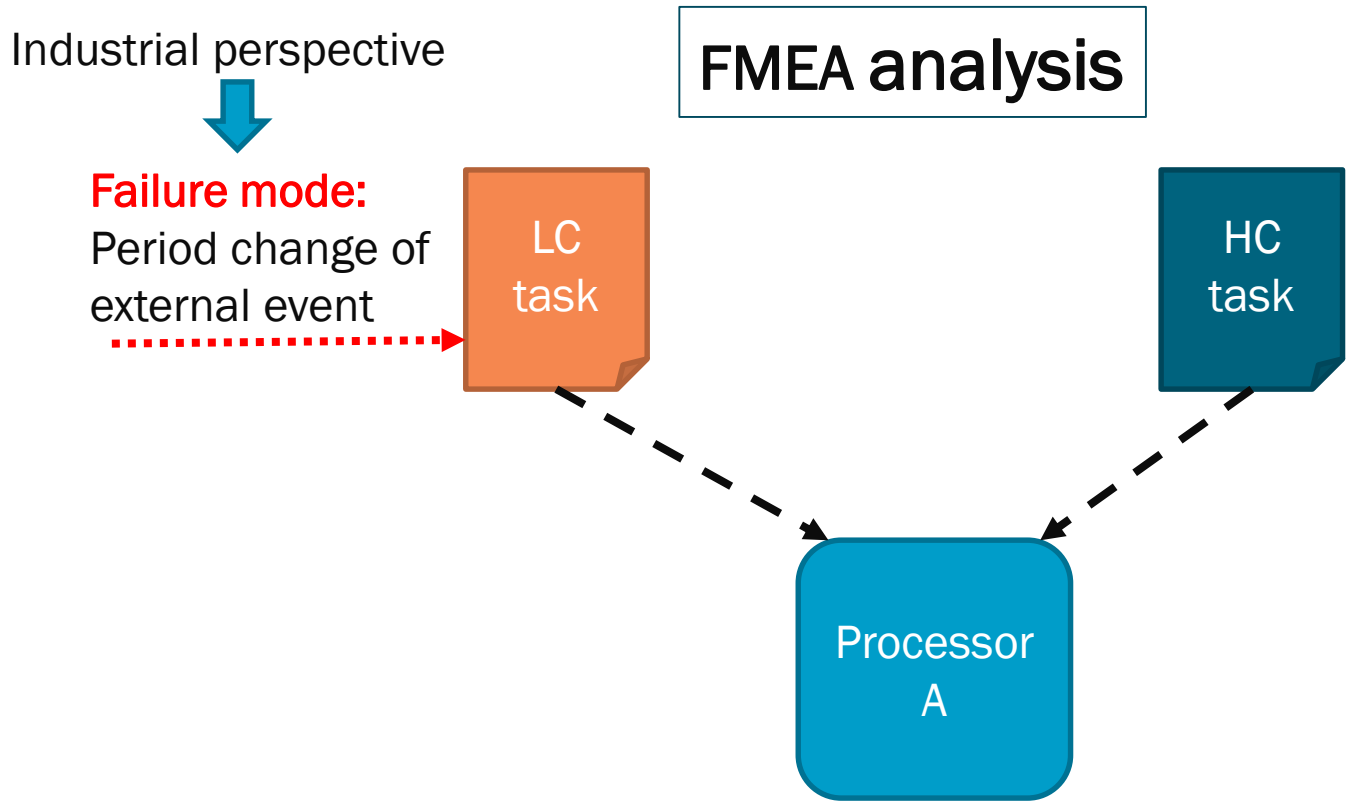
**Vestal's model:**

High and low criticality tasks run on the same processor and scheduler



# Vestal's Model and Isolation

**Vestal's model:**  
High and low criticality tasks run on the same processor and scheduler



# Vestal's Model and Isolation

## Vestal's model:

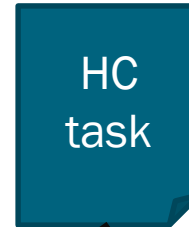
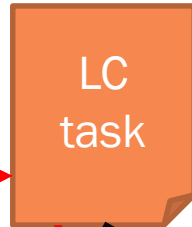
High and low criticality tasks run on the same processor and scheduler

Industrial perspective

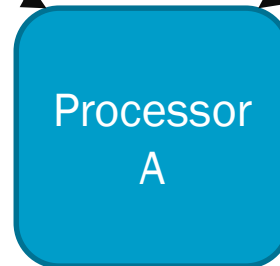
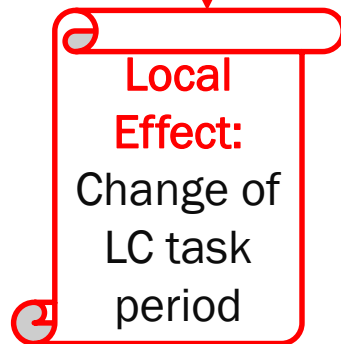


### Failure mode:

Period change of external event



## FMEA analysis



# Vestal's Model and Isolation

## Vestal's model:

High and low criticality tasks run on the same processor and scheduler

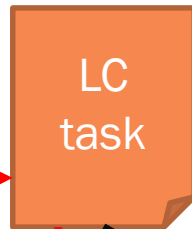
Industrial perspective



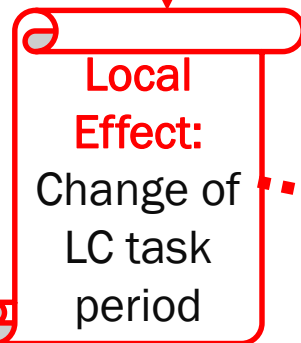
## FMEA analysis

### Failure mode:

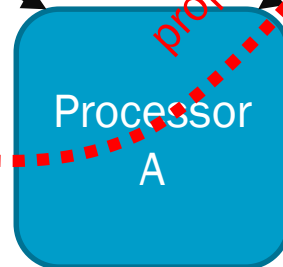
Period change of external event



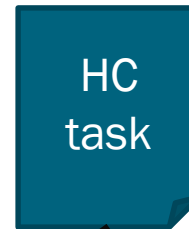
LC task



Local Effect:  
Change of LC task period



Processor A



HC task

propagates

# Vestal's Model and Isolation

## Vestal's model:

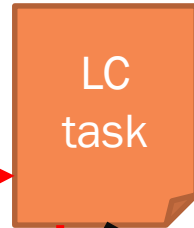
High and low criticality tasks run on the same processor and scheduler

Industrial perspective

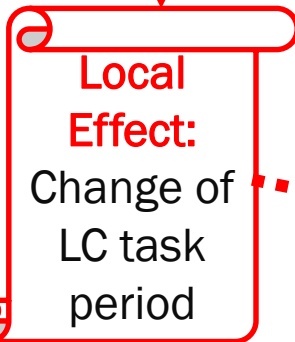


### Failure mode:

Period change of external event



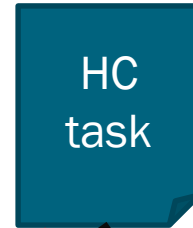
LC task



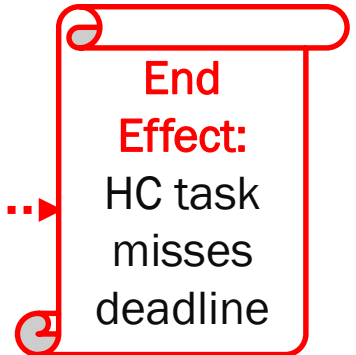
Local Effect:  
Change of LC task period

## FMEA analysis

Processor A



HC task



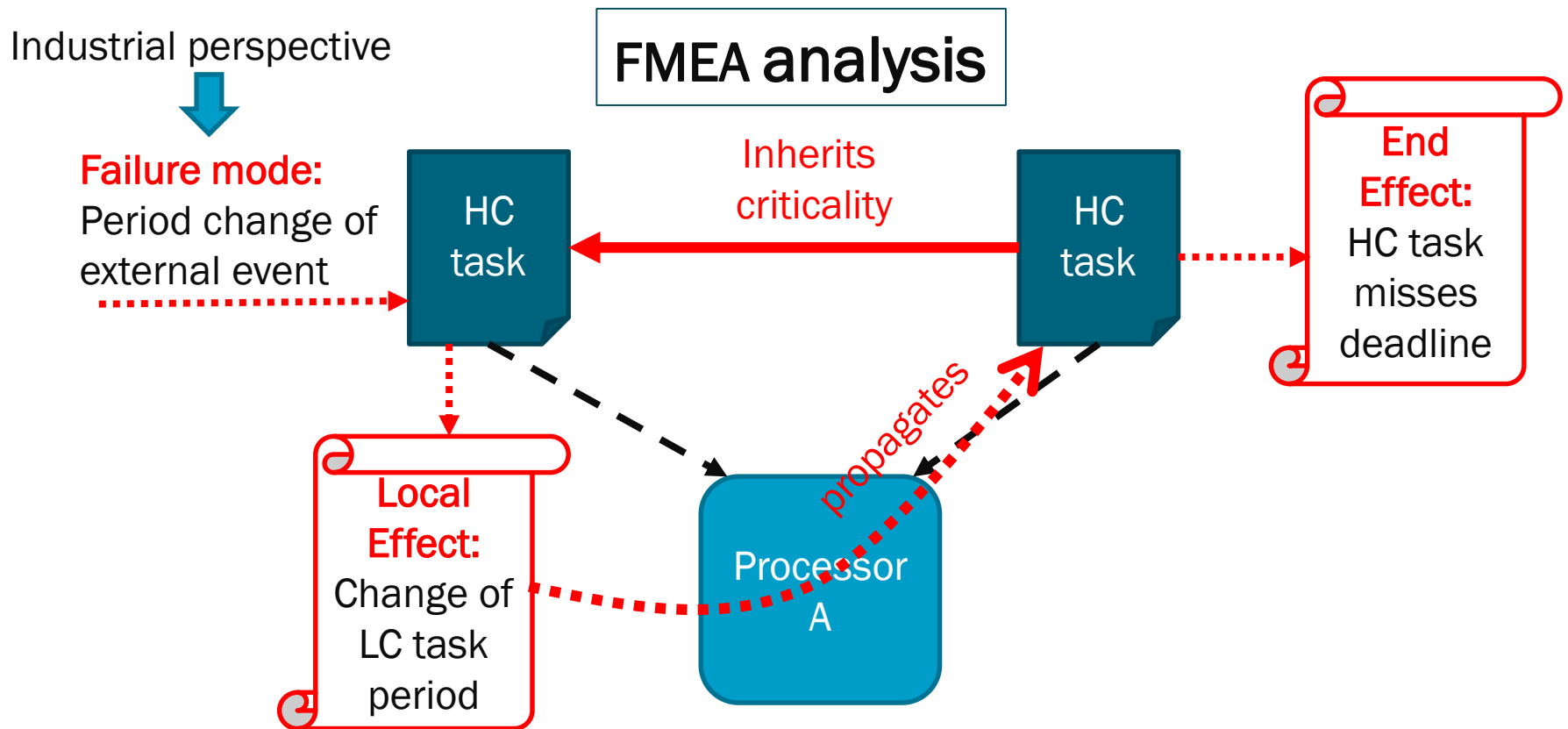
End Effect:  
HC task misses deadline

propagates

# Vestal's Model and Isolation

## Vestal's model:

High and low criticality tasks run on the same processor and scheduler



# Vestal's Model and Isolation

Vestal's model:

High and low criticality tasks run on the same processor and scheduler

Industrial perspective



**Failure mode:**

Period change of external event



FMEA analysis

**HC and LC tasks not isolated in time  
→ all tasks will have to be certified at HC level**

**Local Effect:**  
Change of LC task period



**End Effect:**  
HC task misses deadline



# Vestal's Model and Isolation

## Vestal's model:

High and low criticality tasks run on the same processor and scheduler

- Will never be able to convince a certification authority that the tasks are isolated in time

- The cost of the system would increase exponentially...

- We miss the initial goal of integrating a mixed-criticality system in the same platform to decrease costs



# WCET Estimation

## Vestal's model & Derivatives:

Assumption: Higher degree of assurance of a task → more pessimistic WCET estimation



- WCET upperbound → necessary but not sufficient condition to ensure safety

# WCET Estimation

## Vestal's model & Derivatives:

Assumption: Higher degree of assurance of a task → more pessimistic WCET estimation

- WCET upperbound → necessary but not sufficient condition to ensure safety

- Requires mechanisms to ensure that safety is not compromised in case of timing violation
  - E.g. time partitioning

Safety-  
Standards



# Probabilistic WCET vs. Software Reliability

## Probabilistic WCET

- Provides a probabilistic upper-bound on the execution time
- Aims at building a reliability model of the software



# Probabilistic WCET vs. Software Reliability

## Probabilistic WCET

- Provides a probabilistic upper-bound on the execution time
- Aims at building a reliability model of the software

Safety-  
Standards

## Software reliability models:

- Still under debate
- Confidence cannot be placed in such models



# Probabilistic WCET vs. Software Reliability

## Probabilistic WCET

- Provides a probabilistic upper-bound on the execution time
- Aims at building a reliability model of the software

Safety-  
Standards

### Software reliability models:

- Still under debate
- Confidence cannot be placed in such models

- Important research direction
- But... cannot assume that they will ever be used in industrial systems to prove software safety

# Probabilistic WCET vs. Software Reliability

## Probabilistic WCET

- Provides a probabilistic upper-bound on the execution time
- Aims at building a reliability model of the software

Safety-  
Standards

### Software reliability models:

- Still under debate
- Confidence cannot be placed in such models

- Important research direction
- But... cannot assume that they will ever be used in industrial systems to prove software safety

- Need to work on the safety argumentation



# Probabilistic WCET vs. Software Reliability

## Probabilistic WCET

- Provides a probabilistic upper-bound on the execution time
- Aimed at building a reliability model of the software





# Probabilistic WCET vs. Software Reliability

## Probabilistic WCET

- Provides a probabilistic upper-bound on the execution time
- Aimed at building a reliability model of the software

Certification  
Authority



Typical question: would you fly an airplane designed with probabilistic software reliability models?

# Probabilistic WCET vs. Software Reliability

## Probabilistic WCET

- Provides a probabilistic upper-bound on the execution time
- Aimed at building a reliability model of the software

Certification  
Authority

Typical question: would you fly an airplane designed with probabilistic software reliability models?



# Conclusion

- Clear gap between some of the guidelines provided in safety-related standards and their interpretation by the academic community




- Misalignment of terminology leads to misunderstanding of each other's work



- Confusion between the notions of criticality and importance



- Ensuring safety in terms of timing isolation goes beyond accurate WCET estimates



- Probabilistic WCET estimates: in case that direction is followed → need to work on the argumentation

# Question & Answers

