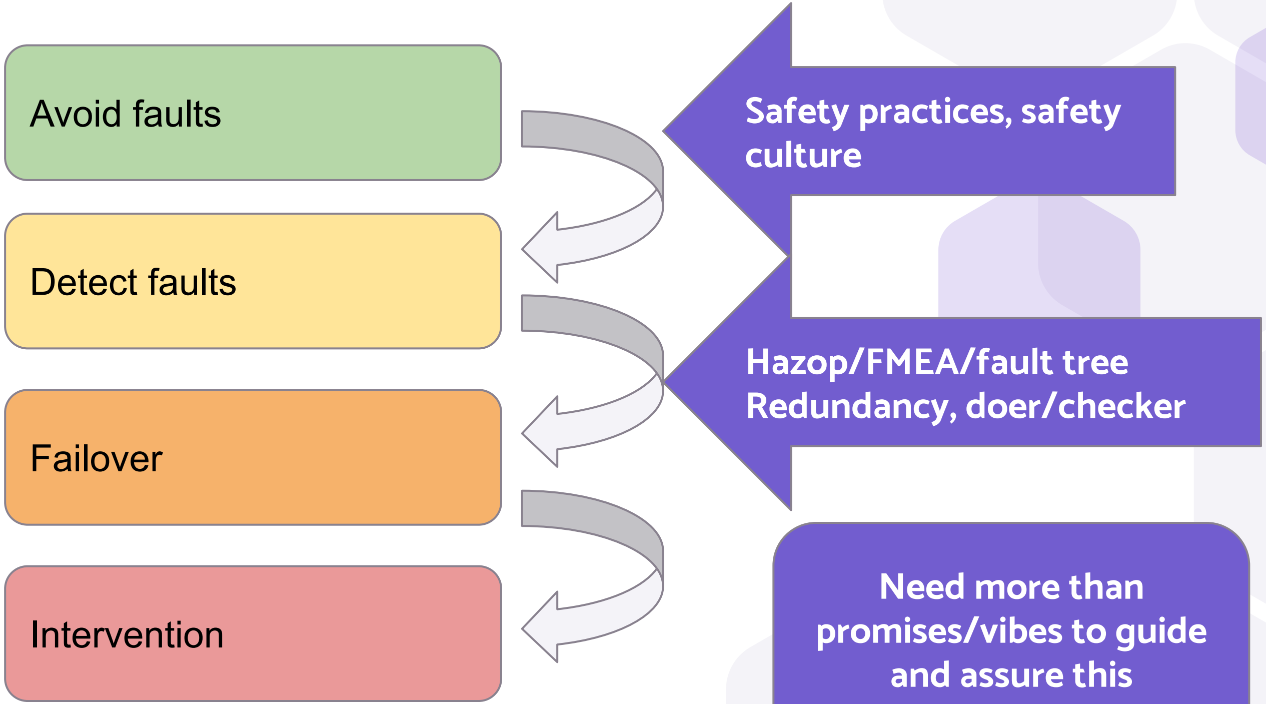
25/26: Safety standards and redundancy architectures

Review: escalation of safety



Avoid faults

Detect faults

Failover

Intervention

Safety practices, safety culture

Hazop/FMEA/fault tree
Redundancy, doer/checker

Need more than promises/vibes to guide and assure this
→ Safety standards

Safety standards

Guide how to engineer for safety

- How to assess risk
- What SW processes/code standards to use
- Which tools/components can be used
- How much/what kinds of testing and formal verification

Auditable by 3rd party (with concrete process for approval!)

Different standards for different domains

- Progression for automotive: MISRA -> IEC 61508 ⟶ ISO 26262 ⟶SOTIF/ISO21448 (⟶UL 4600?)

Risk Matrices

A way of reasoning about the amount of risk of a hazardous event

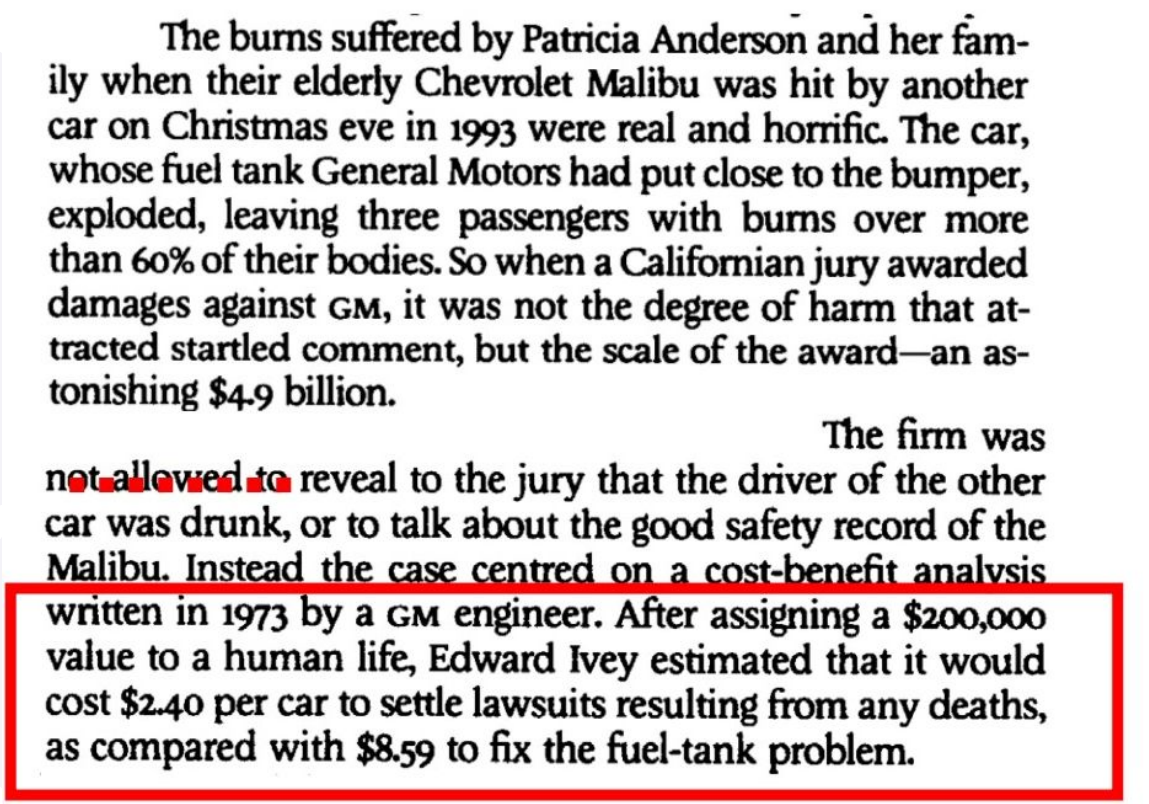
IEC 61508		Consequence			
Likelihood (failures per year)		Catastrophic	Critical	Marginal	Negligible
		Multiple loss of life	Single loss of life	Major injuries	Minor injuries at worst
Frequent	$> 10^{-3}$	[illegible]	[illegible]	**I**	**II**
Probable	10^{-3} - 10^{-4}			**II**	**III**
Occasional	10^{-4}-10^{-5}	**I**		**III**	**III**
Remote	10^{-5}-10^{-6}	**II**	[illegible]		**IV**
Improbable	10^{-6}-10^{-7}	**III**	**III**	**IV**	**IV**
Incredible	$< 10^{-7}$	**III**	**IV**	**IV**	

Unacceptable

Undesirable

Tolerable (cost tradeoff)

Acceptable



The burns suffered by Patricia Anderson and her family when their elderly Chevrolet Malibu was hit by another car on Christmas eve in 1993 were real and horrific. The car, whose fuel tank General Motors had put close to the bumper, exploded, leaving three passengers with burns over more than 60% of their bodies. So when a Californian jury awarded damages against GM, it was not the degree of harm that attracted startled comment, but the scale of the award—an astonishing $4.9 billion.

The firm was not allowed to reveal to the jury that the driver of the other car was drunk, or to talk about the good safety record of the Malibu. Instead the case centred on a cost-benefit analysis written in 1973 by a GM engineer. After assigning a $200,000 value to a human life, Edward Ivey estimated that it would cost $2.40 per car to settle lawsuits resulting from any deaths, as compared with $8.59 to fix the fuel-tank problem.

Article source: Economist, July 17 1999

Image source

Safety Integrity Levels

A (standards-based) target to attain for each safety function

Named SIL levels (IEC 61508 has SIL-1, SIL-2, SIL-3, SIL-4)

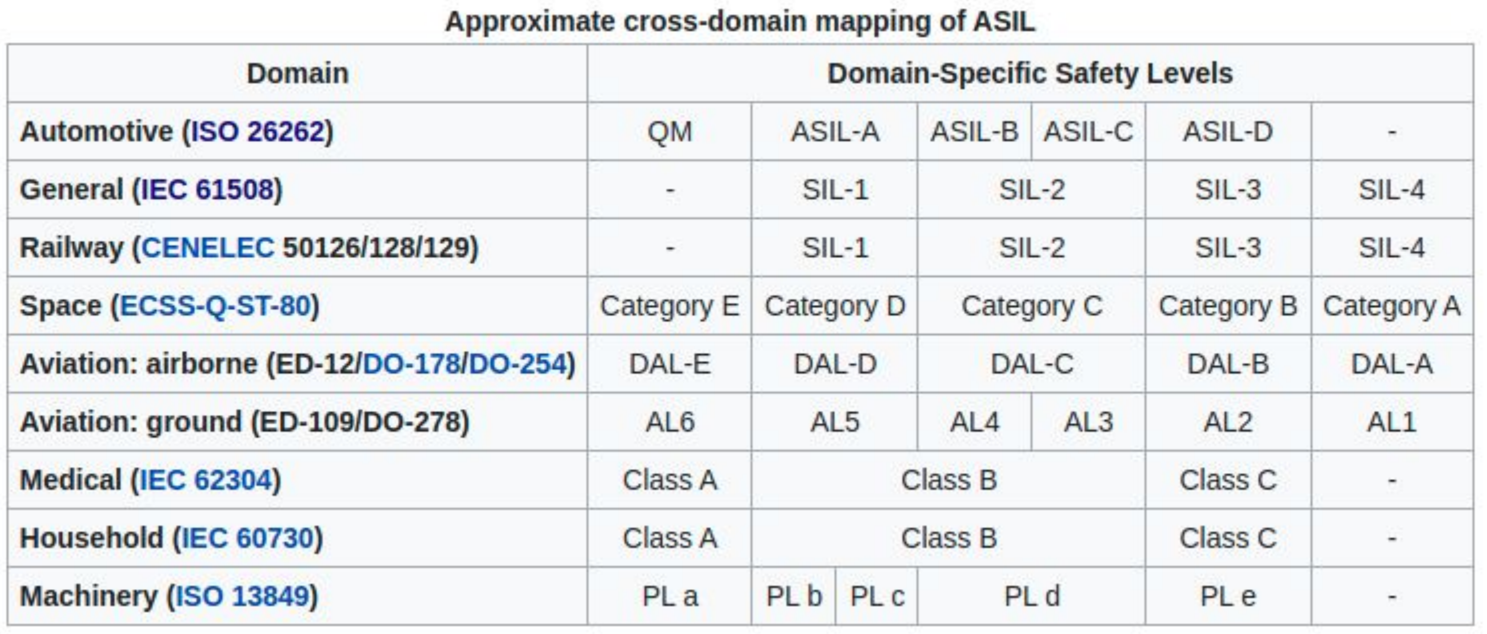
- "high SIL" (4/D) means smallest acceptable failure rate (in ISO26262, $< 10^{-9}$ per hour)
- *confusingly, aviation flips this ("high SIL" is analogous to DAL-A)*

Each SIL may require:

- Maximum accepted risk of failure
- Minimum accepted software quality
- Minimum accepted redundancy architecture
- All hardware to be certified at or above that level

Analysis and mitigation techniques

Different standards for different domains



Approximate cross-domain mapping of ASIL

Domain	Domain-Specific Safety Levels						
Automotive (ISO 26262)	QM	ASIL-A		ASIL-B	ASIL-C	ASIL-D	-
General (IEC 61508)	-	SIL-1		SIL-2		SIL-3	SIL-4
Railway (CENELEC 50126/128/129)	-	SIL-1		SIL-2		SIL-3	SIL-4
Space (ECSS-Q-ST-80)	Category E	Category D		Category C		Category B	Category A
Aviation: airborne (ED-12/DO-178/DO-254)	DAL-E	DAL-D		DAL-C		DAL-B	DAL-A
Aviation: ground (ED-109/DO-278)	AL6	AL5		AL4	AL3	AL2	AL1
Medical (IEC 62304)	Class A	Class B				Class C	-
Household (IEC 60730)	Class A	Class B				Class C	-
Machinery (ISO 13849)	PL a	PL b	PL c	PL d		PL e	-

Image source

Standards inform practice

ISO 26262

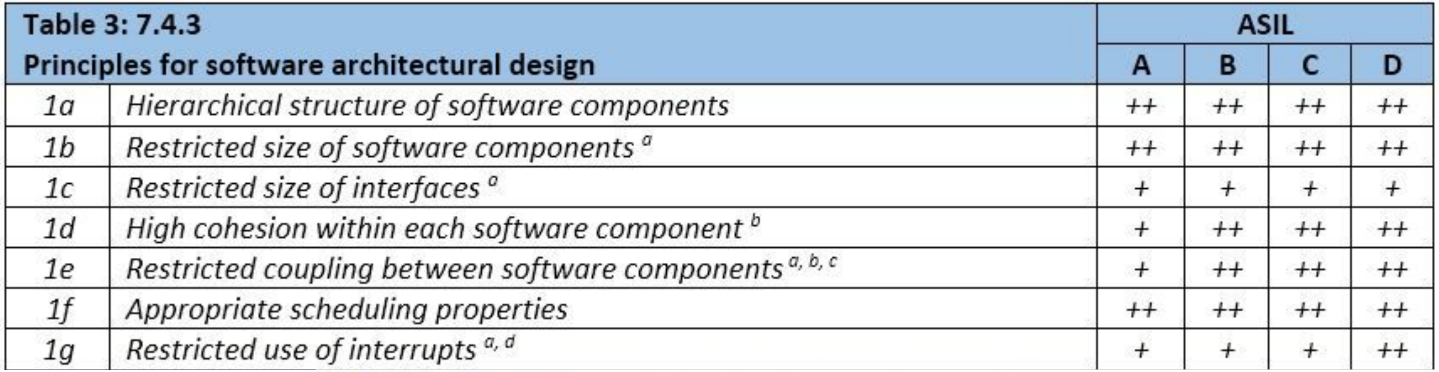


Table 3: 7.4.3		ASIL			
Principles for software architectural design		**A**	**B**	**C**	**D**
1a	*Hierarchical structure of software components*	++	++	++	++
1b	*Restricted size of software components* [a]	++	++	++	++
1c	*Restricted size of interfaces* [a]	+	+	+	+
1d	*High cohesion within each software component* [b]	+	++	++	++
1e	*Restricted coupling between software components* [a, b, c]	+	++	++	++
1f	*Appropriate scheduling properties*	++	++	++	++
1g	*Restricted use of interrupts* [a, d]	+	+	+	++

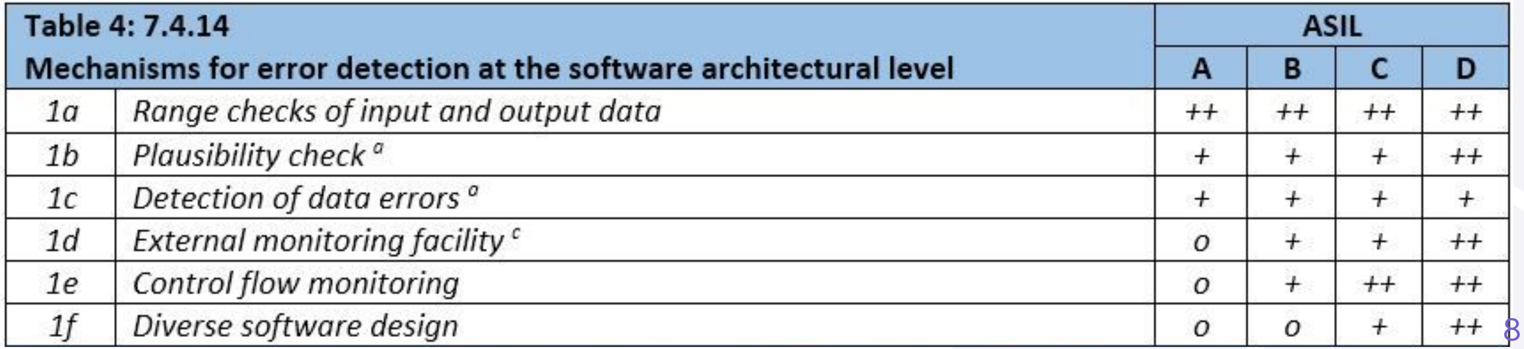
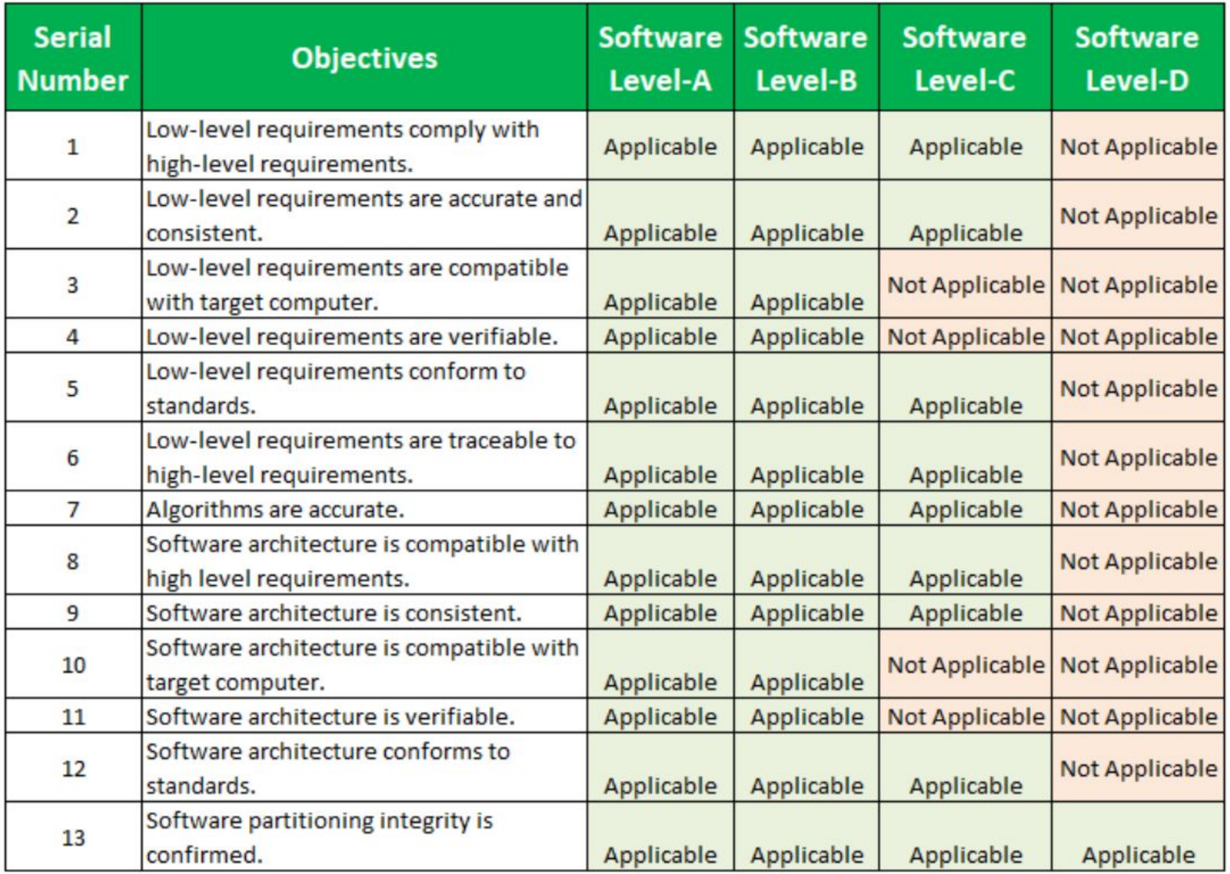


Table 4: 7.4.14		ASIL			
Mechanisms for error detection at the software architectural level		**A**	**B**	**C**	**D**
1a	*Range checks of input and output data*	++	++	++	++
1b	*Plausibility check* [a]	+	+	+	++
1c	*Detection of data errors* [a]	+	+	+	+
1d	*External monitoring facility* [c]	o	+	+	++
1e	*Control flow monitoring*	o	+	++	++
1f	*Diverse software design*	o	o	+	++

Image source

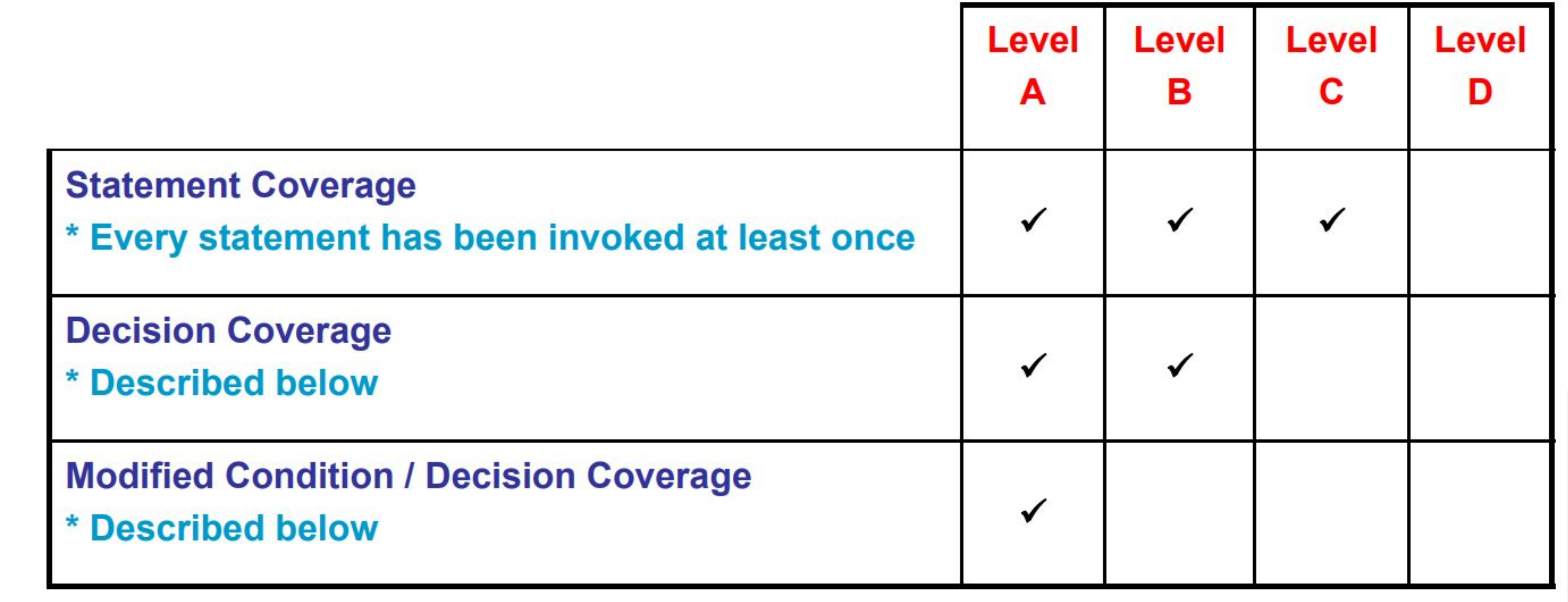
DO 178C



Serial Number	Objectives	Software Level-A	Software Level-B	Software Level-C	Software Level-D
1	Low-level requirements comply with high-level requirements.	Applicable	Applicable	Applicable	Not Applicable
2	Low-level requirements are accurate and consistent.	Applicable	Applicable	Applicable	Not Applicable
3	Low-level requirements are compatible with target computer.	Applicable	Applicable	Not Applicable	Not Applicable
4	Low-level requirements are verifiable.	Applicable	Applicable	Not Applicable	Not Applicable
5	Low-level requirements conform to standards.	Applicable	Applicable	Applicable	Not Applicable
6	Low-level requirements are traceable to high-level requirements.	Applicable	Applicable	Applicable	Not Applicable
7	Algorithms are accurate.	Applicable	Applicable	Applicable	Not Applicable
8	Software architecture is compatible with high level requirements.	Applicable	Applicable	Applicable	Not Applicable
9	Software architecture is consistent.	Applicable	Applicable	Applicable	Not Applicable
10	Software architecture is compatible with target computer.	Applicable	Applicable	Not Applicable	Not Applicable
11	Software architecture is verifiable.	Applicable	Applicable	Not Applicable	Not Applicable
12	Software architecture conforms to standards.	Applicable	Applicable	Applicable	Not Applicable
13	Software partitioning integrity is confirmed.	Applicable	Applicable	Applicable	Applicable

DO 178C Table A-4: Verification of Outputs of Software Design Process

Image source



	Level A	Level B	Level C	Level D
Statement Coverage * Every statement has been invoked at least once	✓	✓	✓	
Decision Coverage * Described below	✓	✓		
Modified Condition / Decision Coverage * Described below	✓			

https://apps.dtic.mil/sti/tr/pdf/ADA558107.pdf

Traceability

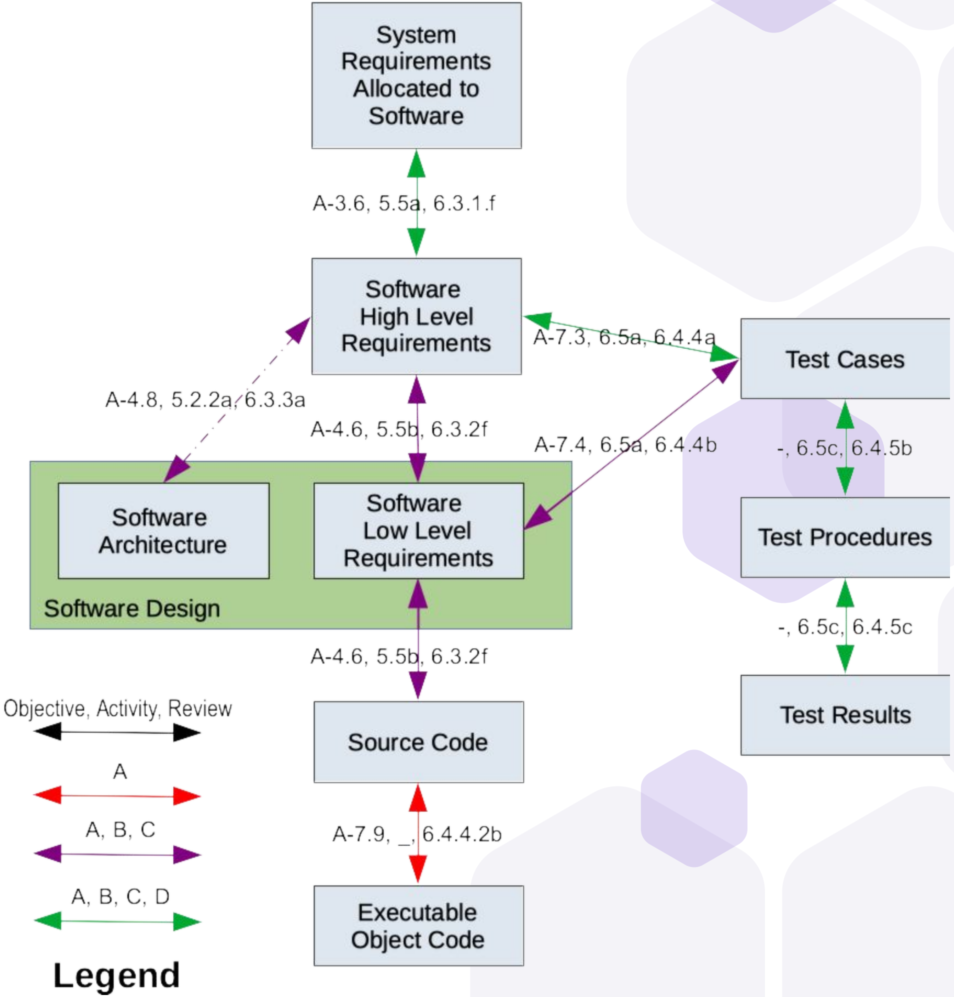


Image source

Standards in the wild

Marketing/application

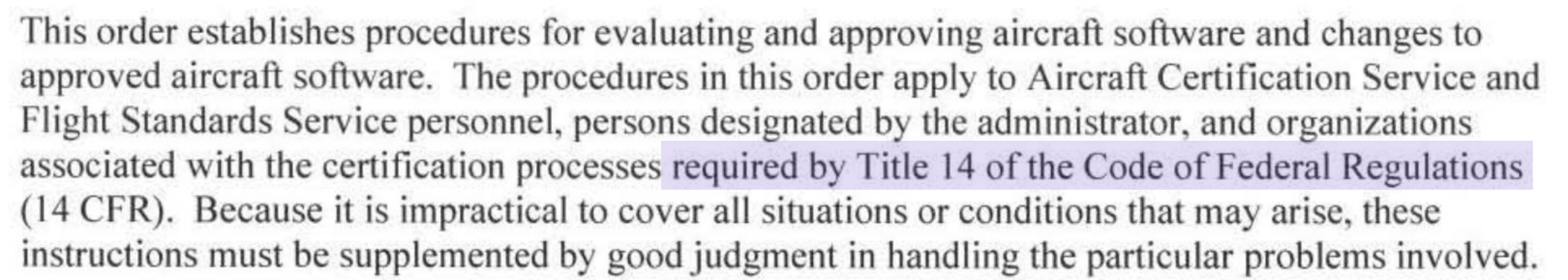
FPGAs (IEC 61508): Microchip functional safety page

Certification

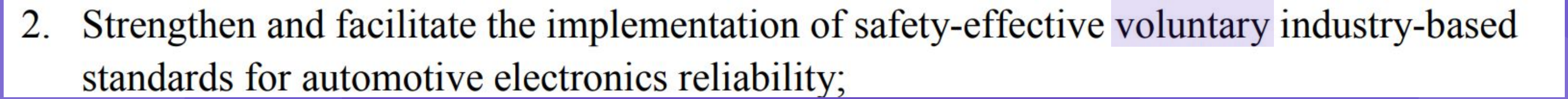
Airplanes (DO-178C): FAA software approval guidelines

Development/recommendation

Automotive (ISO26262 + others): NHSTA study of safety standards



This order establishes procedures for evaluating and approving aircraft software and changes to approved aircraft software. The procedures in this order apply to Aircraft Certification Service and Flight Standards Service personnel, persons designated by the administrator, and organizations associated with the certification processes required by Title 14 of the Code of Federal Regulations (14 CFR). Because it is impractical to cover all situations or conditions that may arise, these instructions must be supplemented by good judgment in handling the particular problems involved.

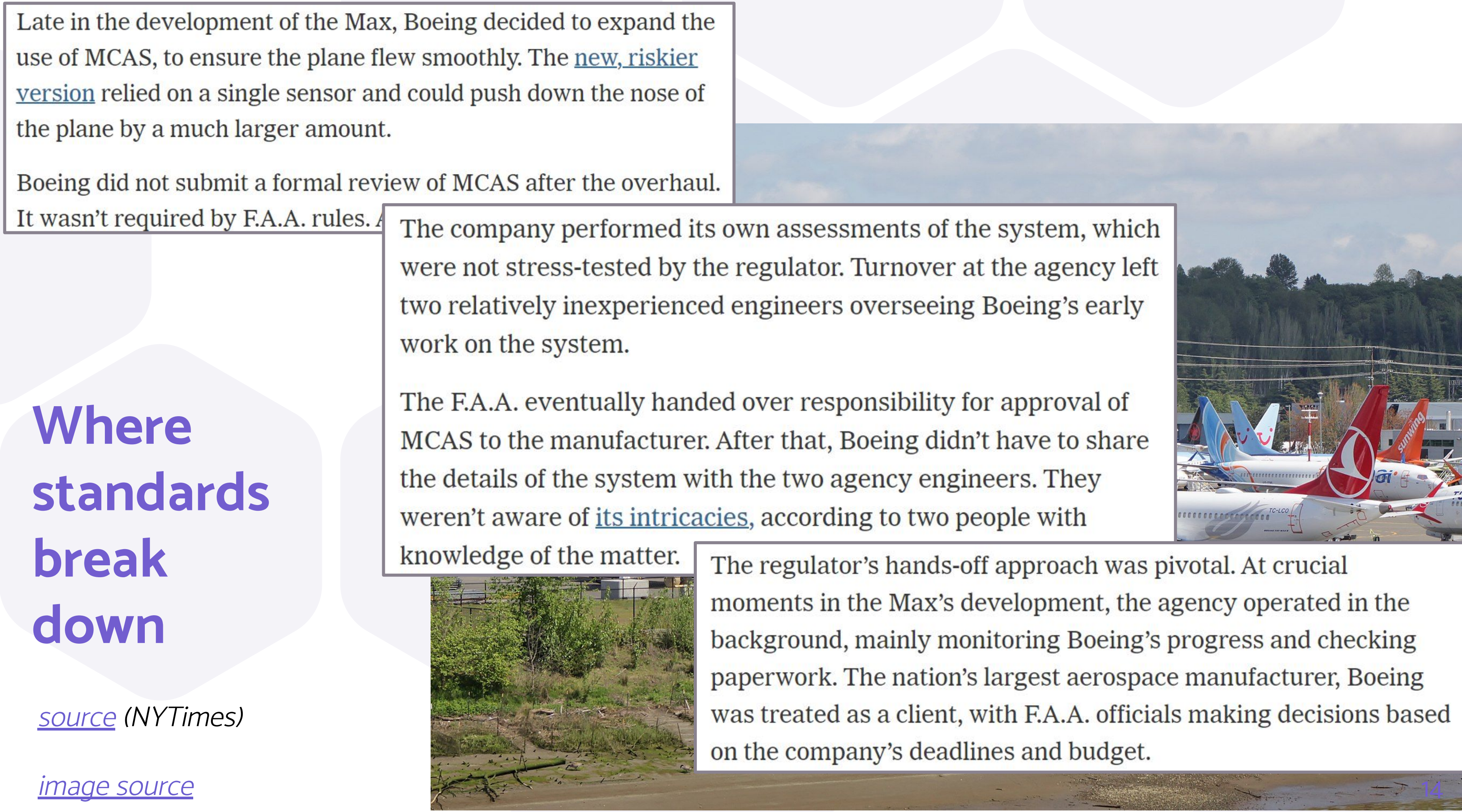


2. Strengthen and facilitate the implementation of safety-effective voluntary industry-based standards for automotive electronics reliability;

Should cars be engineered to the same safety standards as planes?

- *Same rigor?*
- *Legal requirement?*

Where standards break down



Late in the development of the Max, Boeing decided to expand the use of MCAS, to ensure the plane flew smoothly. The new, riskier version relied on a single sensor and could push down the nose of the plane by a much larger amount.

Boeing did not submit a formal review of MCAS after the overhaul. It wasn't required by F.A.A. rules.

The company performed its own assessments of the system, which were not stress-tested by the regulator. Turnover at the agency left two relatively inexperienced engineers overseeing Boeing's early work on the system.

The F.A.A. eventually handed over responsibility for approval of MCAS to the manufacturer. After that, Boeing didn't have to share the details of the system with the two agency engineers. They weren't aware of its intricacies, according to two people with knowledge of the matter.

The regulator's hands-off approach was pivotal. At crucial moments in the Max's development, the agency operated in the background, mainly monitoring Boeing's progress and checking paperwork. The nation's largest aerospace manufacturer, Boeing was treated as a client, with F.A.A. officials making decisions based on the company's deadlines and budget.

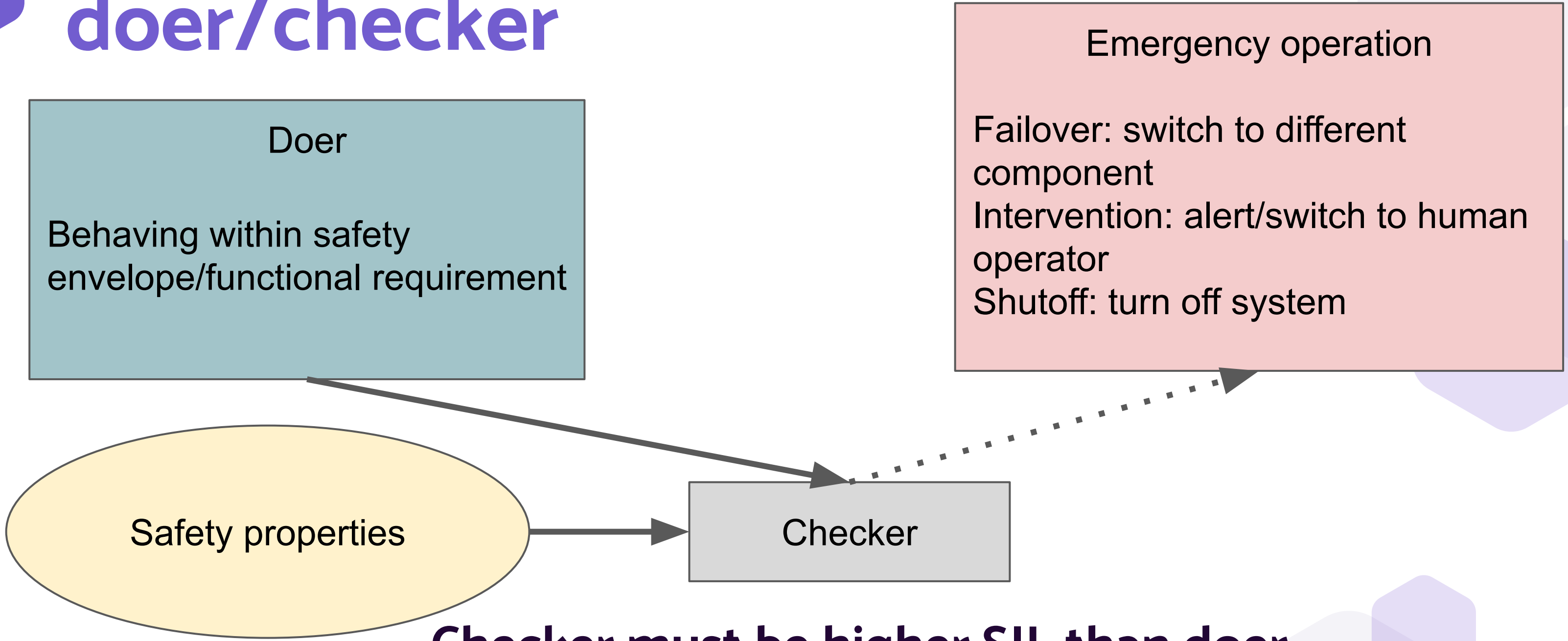
source (NYTimes)

image source

Why would you want to mix low-SIL and high-SIL software/components in the same system?

How would you do it?

Possible idea: mixed-SIL doer/checker



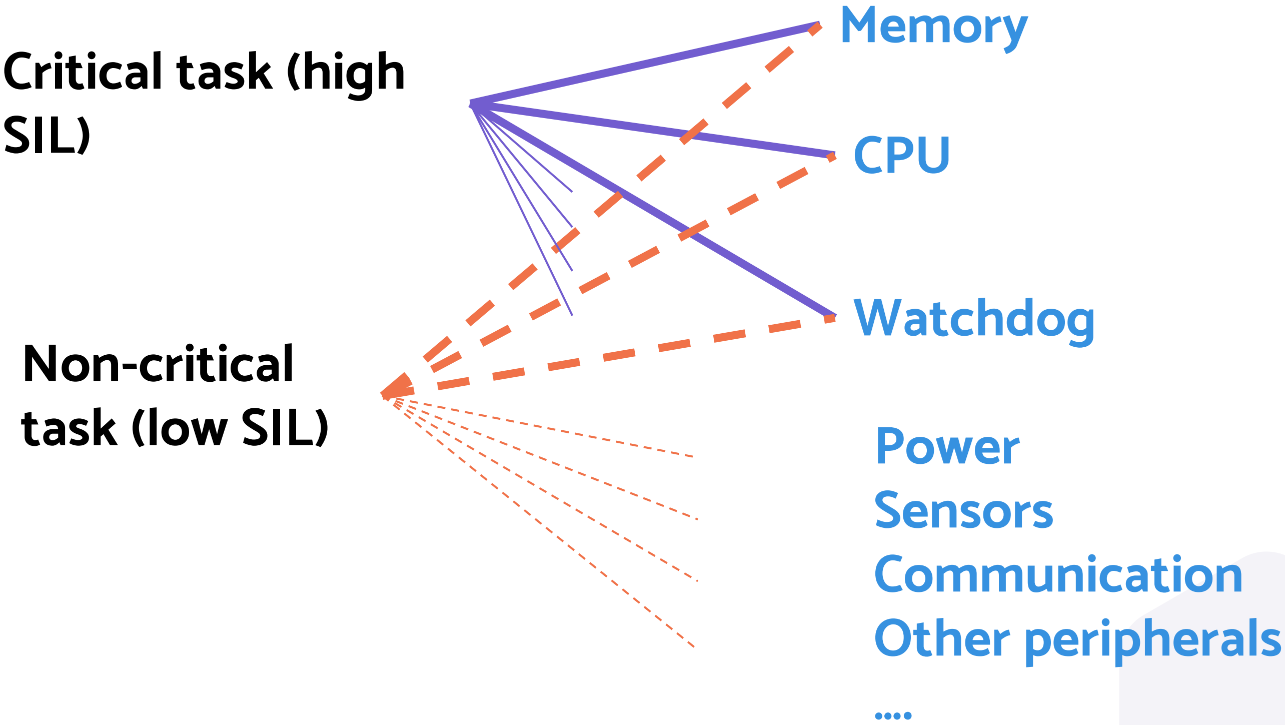
Doer
Behaving within safety envelope/functional requirement
Safety properties
Checker
Emergency operation
Failover: switch to different component
Intervention: alert/switch to human operator
Shutoff: turn off system

Checker must be higher SIL than doer

Must be confident detection/emergency operation won't fail

What are some downsides to the mixed-SIL doer/checker architecture?

Mixed-SIL Interference



Critical task (high SIL)

Non-critical task (low SIL)

Memory

CPU

Watchdog

Power

Sensors

Communication

Other peripherals

....

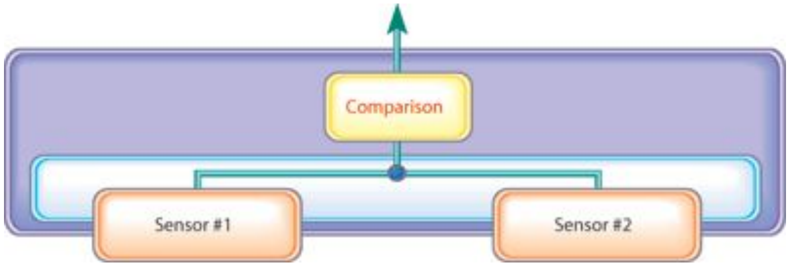
Mixed-criticality systems: an active area

Examples:

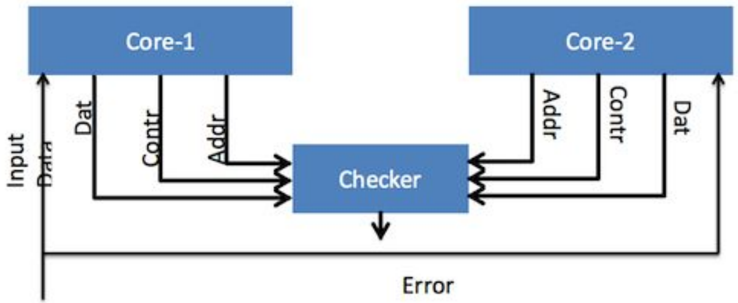
- AUTOSAR article
- WITTENSTEIN article

What criteria would you use to select an industry solution/read past marketing hype?

Additional ways to introduce redundancy

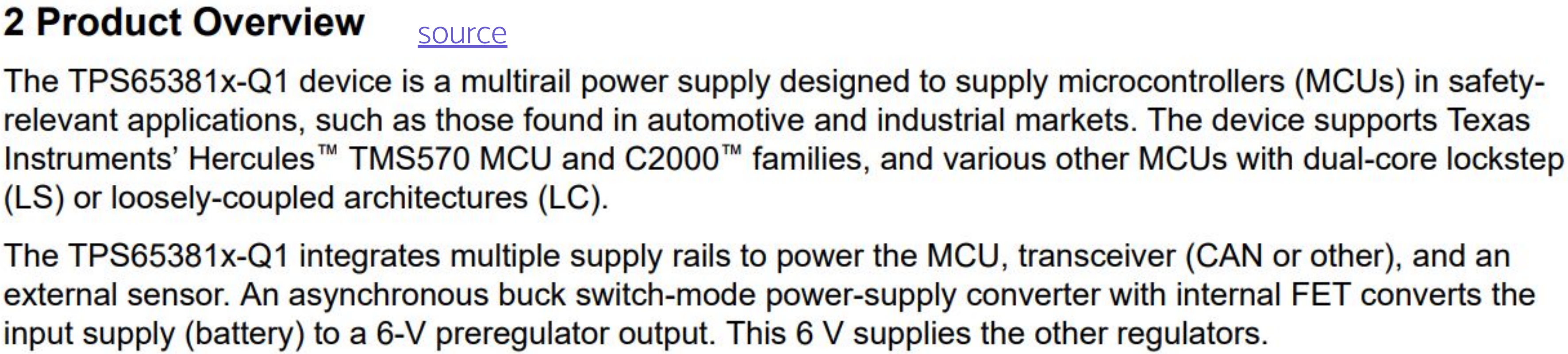


detect sensor failure with comparison (*image source*)



detect CPU fault with lockstep execution (image source)

Hardware support for power



2 Product Overview

source

The TPS65381x-Q1 device is a multirail power supply designed to supply microcontrollers (MCUs) in safety-relevant applications, such as those found in automotive and industrial markets. The device supports Texas Instruments' Hercules™ TMS570 MCU and C2000™ families, and various other MCUs with dual-core lockstep (LS) or loosely-coupled architectures (LC).

The TPS65381x-Q1 integrates multiple supply rails to power the MCU, transceiver (CAN or other), and an external sensor. An asynchronous buck switch-mode power-supply converter with internal FET converts the input supply (battery) to a 6-V preregulator output. This 6 V supplies the other regulators.