Safety- en Security Certificatie van embedded software

Standaarden, keuzes en kostenbeheersing

Gerard Fianen
INDES-Integrated Development Solutions BV

Stand 22/23

V 1.2.1

INDES –
Integrated Development Solutions BV

Cross Compilers, Debuggers, IDE
RTOS, Middleware, Protocol stacks, Security, GUI, EFS
Debug & Trace probes, Emulators
Real-Time Trace, RTOS-Event Trace
Static Analysis, Timing Analysis, Stack Analysis
Unit Test, Code Coverage
(Production) Flash Programming

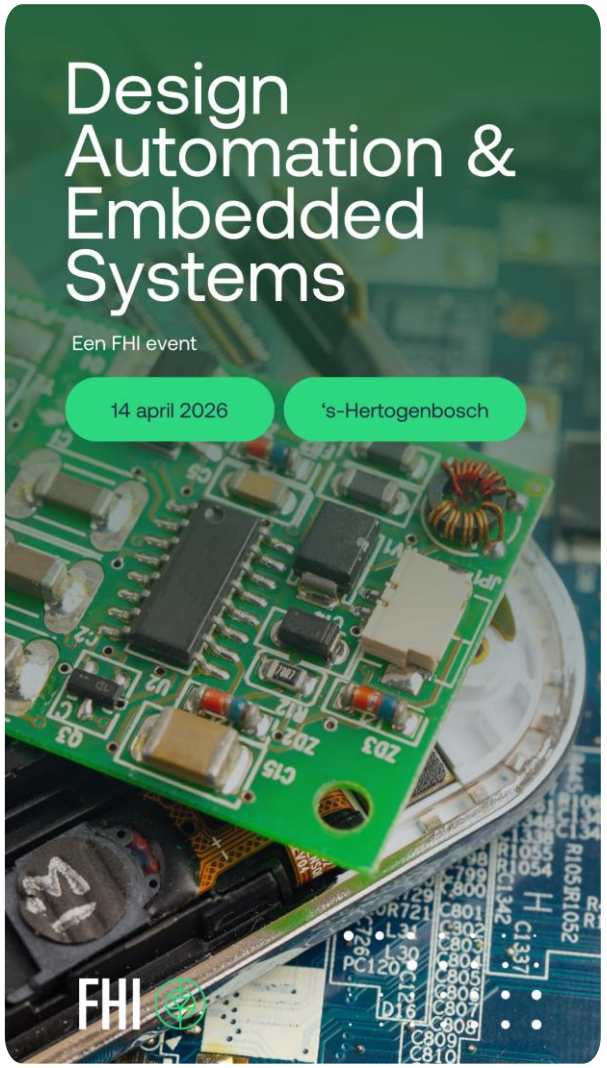
On-site support
Test-as-a-Service
Verification-as-a-Service

PoE conformance testing

www.indes.com info@indes.com
Tel: 0345 - 545.535

FHI Federatie van technologiebranches

indes The choice of professionals



Agenda

- Safety & Security standards
- How / where to start ?
- Tools : What does certification mean ?
- Some considerations for the selection of :
  - Hardware
  - Compiler, IDE
  - Test tools
  - RTOS
- Q & A

Safety standards – History

- DOD-2167a ; US Defence
- CMM ; Development Process

- VDE-V-0801, DIN-V-19250 ; German industrial standard, AK – Safety classes
- IEC 601-1-4 ; Medial standard

- DO178b ; Avionics & Space

1990's brought a need to harmonisation :

- IEC-(6)1508 : first international standard for functional safety

Security / Cyber Resiliency Act (CRA)

ISO 27001 framework → IEC 62443 Standard

- Security for Industrial Automation and Control Systems
- Security Level Sl1 - SL4

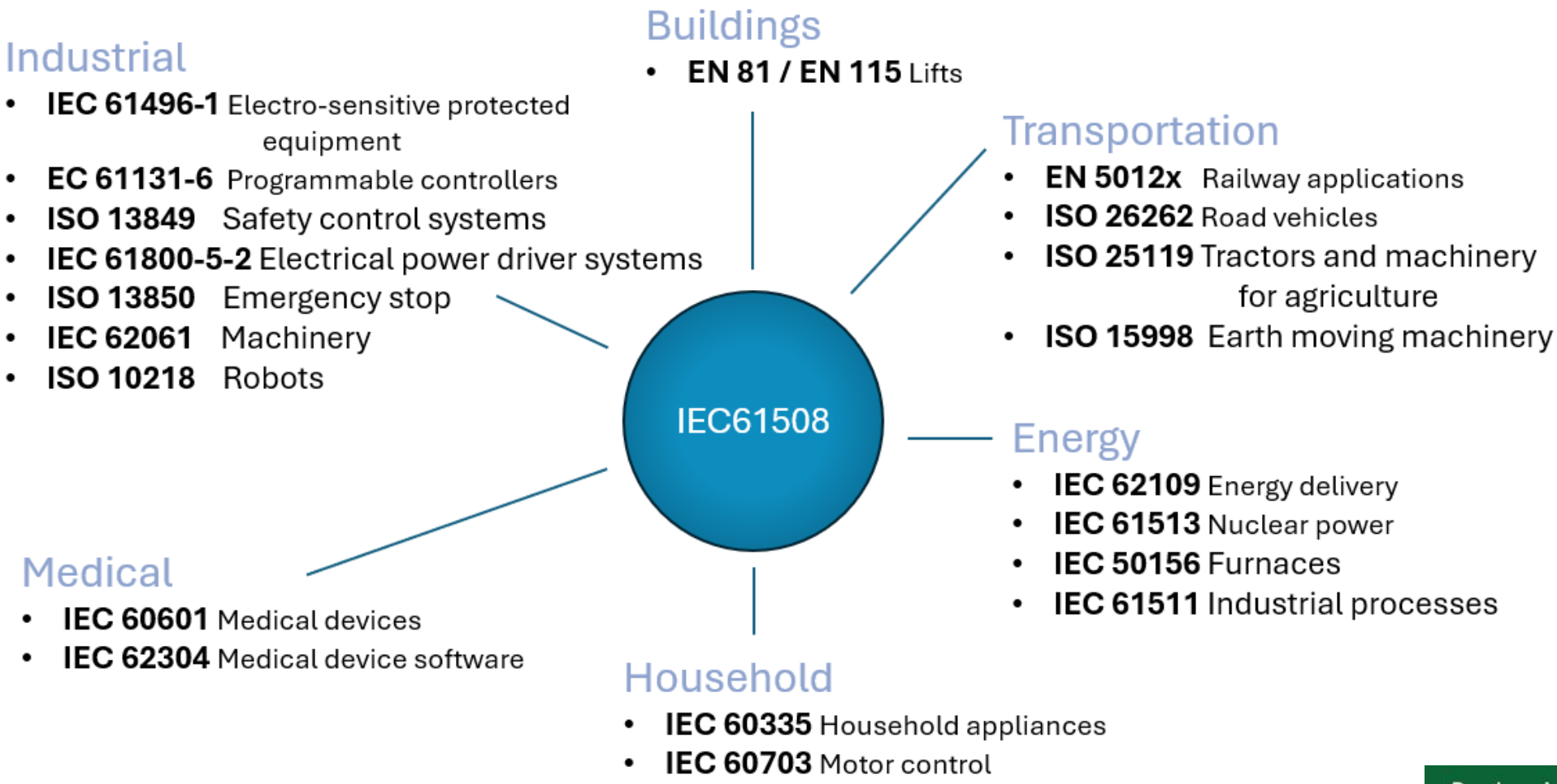
Embedded Software Development

- Security Middleware (SSL / SSH, Secure Bootloader, RSA encryption)
- Embedded Development Tools

PQC = Post Quantum Cryptography

- 2035 ---> 2031 ---> 2029 ---> 2027 ?
- RSA2048, ML-KEM-768

IEC 61508


IEC61508
Industrial
• IEC 61496-1 Electro-sensitive protected equipment
• EC 61131-6 Programmable controllers
• ISO 13849 Safety control systems
• IEC 61800-5-2 Electrical power driver systems
• ISO 13850 Emergency stop
• IEC 62061 Machinery
• ISO 10218 Robots
Buildings
• EN 81 / EN 115 Lifts
Transportation
• EN 5012x Railway applications
• ISO 26262 Road vehicles
• ISO 25119 Tractors and machinery for agriculture
• ISO 15998 Earth moving machinery
Energy
• IEC 62109 Energy delivery
• IEC 61513 Nuclear power
• IEC 50156 Furnaces
• IEC 61511 Industrial processes
Medical
• IEC 60601 Medical devices
• IEC 62304 Medical device software
Household
• IEC 60335 Household appliances
• IEC 60703 Motor control

Pragmatic guidelines (example RTOS)

IEC 61508 HR Items for RTOS From Part 3 – Software Requirements

Timing & Scheduling

- HR1: The RTOS scheduler must be deterministic and bounded in execution time
- HR2: Worst-case execution time (WCET) must be analyzed for all tasks
- HR3: Task preemption and priority inversion must be controlled (e.g., priority ceiling or inheritance protocols)
- HR4: Interrupt latency must be bounded and verified

Memory Management

- HR5: Memory partitioning must prevent tasks from corrupting each other's data (spatial isolation)
- HR6: Dynamic memory allocation should be avoided or tightly controlled in safety-critical tasks
- HR7: Stack overflow detection/protection must be implemented

Inter-Task Communication

- HR8: Shared resource access must be controlled via semaphores/mutexes with defined timeout behavior
- HR9: Message queues and buffers must have defined behavior on overflow/underflow

Fault Detection

- HR10: The RTOS must support detection of task overrun (deadline monitoring)
- HR11: Watchdog timer support must be provided and used
- HR12: Error handling for OS service call failures must be defined

Where to start …

- Requirements, Requirements, Requirements …

  Functional Requirements
  Test Requirements
  → Requirements Traceability Matrix

- Designated auditor (as early as possible)
- Standard, SIL Level (now & future)
- Training, who does what, assign certification roles & items
- Hardware selection
- Tool planning / selection

Safety Integrity Level

FMEA (Failure Mode and Effects Analysis / HAZOP (Hazard and Operability Analysis

Safety standards generally require you to perform a hazard and operability analysis (HAZOP analysis) of potential tool failures. See IEC 61508-3, section 7.4.4.5 and IEC 61508-7, section C.6.2.

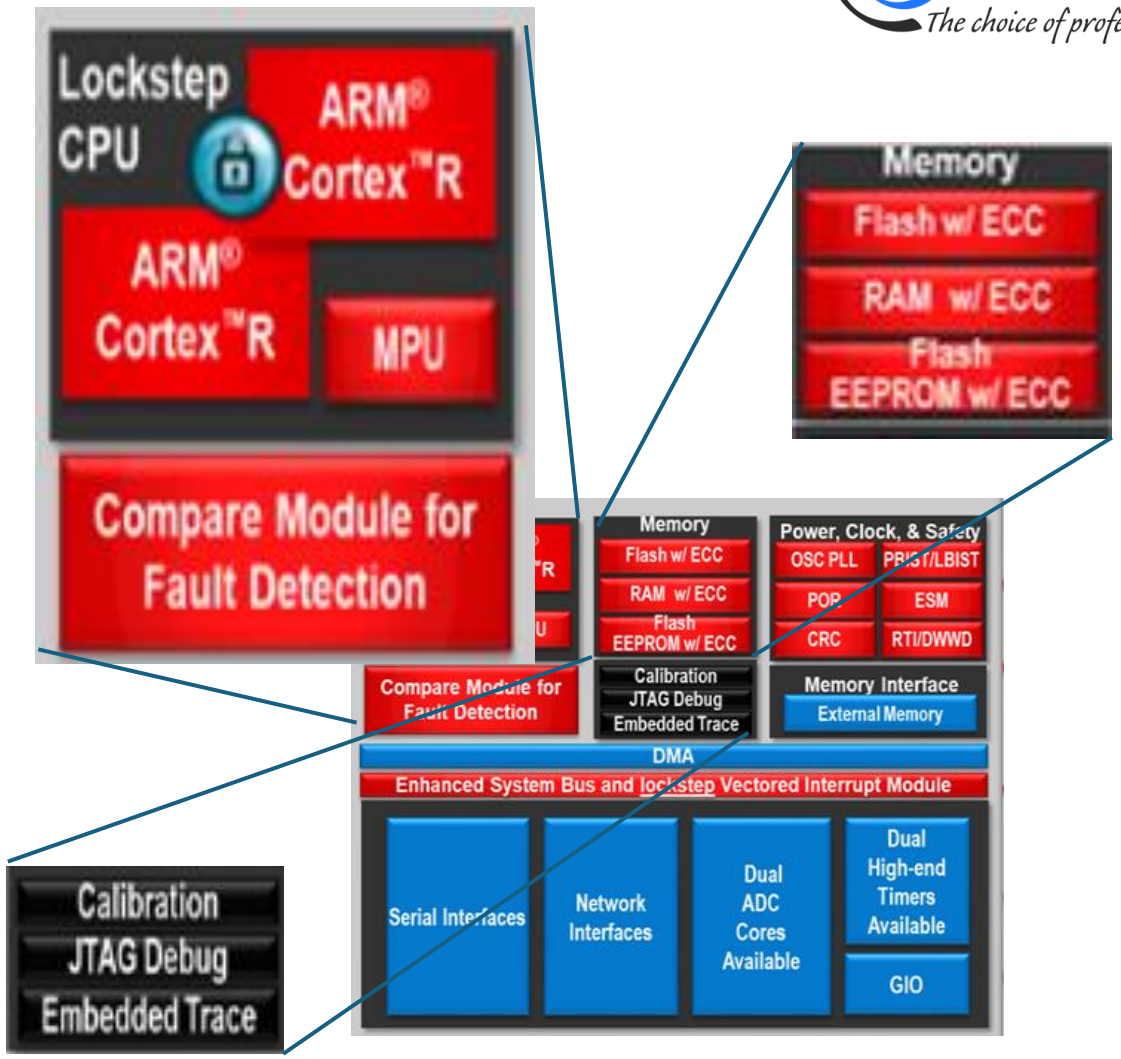
Safety Integrity Level (SIL)	corresponding German Requirements Class (AK)	Probability of Failure-on-Demand
1	2 - 3	$>=10^{-2}$ to $<10^{-1}$
2	4	$>=10^{-3}$ to $<10^{-2}$
3	5 - 6	$>=10^{-4}$ to $<10^{-3}$
4	7	$>=10^{-5}$ to $<10^{-4}$

ISO 26262 Automotive : ASIL A – D
IEC 62304 Medical : Class A - C

DO178B/C : Level D - A

Hardware

- Single core / Multi Core
- Safety MCU’s
- MPU / MMU
- MCU Availability (e.g. 10 years)
- Certified / Qualified Tools availability
  - Compiler / IDE, Test Tools
  - Debug & Tracing



FHI Federatie van technologiebranches

indes The choice of professionals

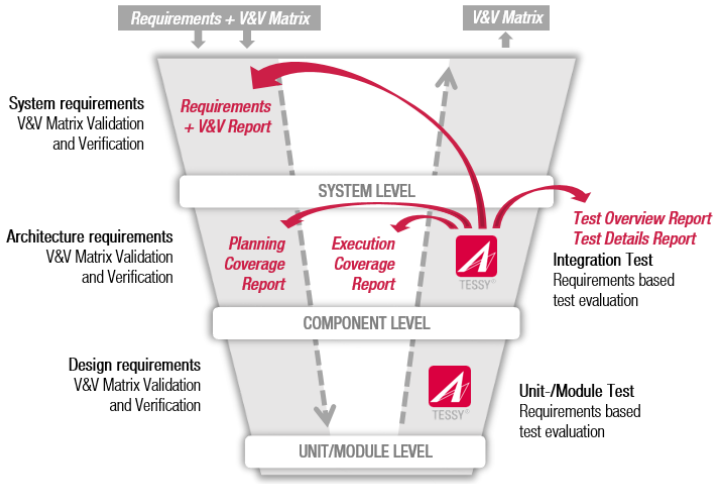
Compiler / IDE

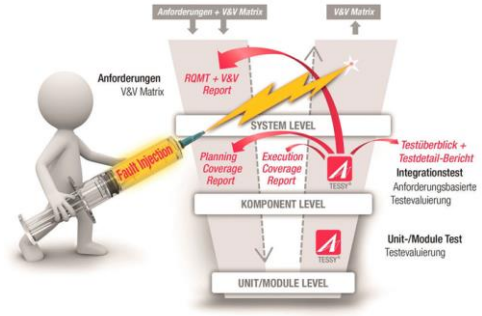
- GCC or not-to-GCC
- Relevant for certification :
  - Who developed & tested it ?
  - Who is responsible for support ?
  - How was the compiler tested ?
  - Proven in use ?
  - Safety manual (See our stand for examples)
- Improve Software Development Live Cycle (SDLC) with CI / CD
  (Continuous Integration / Continuous Deployment)
- Archiving / Make sure to have access to older version
- MUST I use a Certified edition ?
  or will a standard edition do ?
- Upgrade to Certified, buy 1certified edition, multiple standard ....
- Very high safety : Mathematically proven correct compiler

FHI Federatie van technologiebranches

indes The choice of professionals

Test Tools





Static Analysis is effectively mandatory

Very high level : formal Static Analysis of code, timing and memory usage

Make sure it covers everything, no “mandatory options”

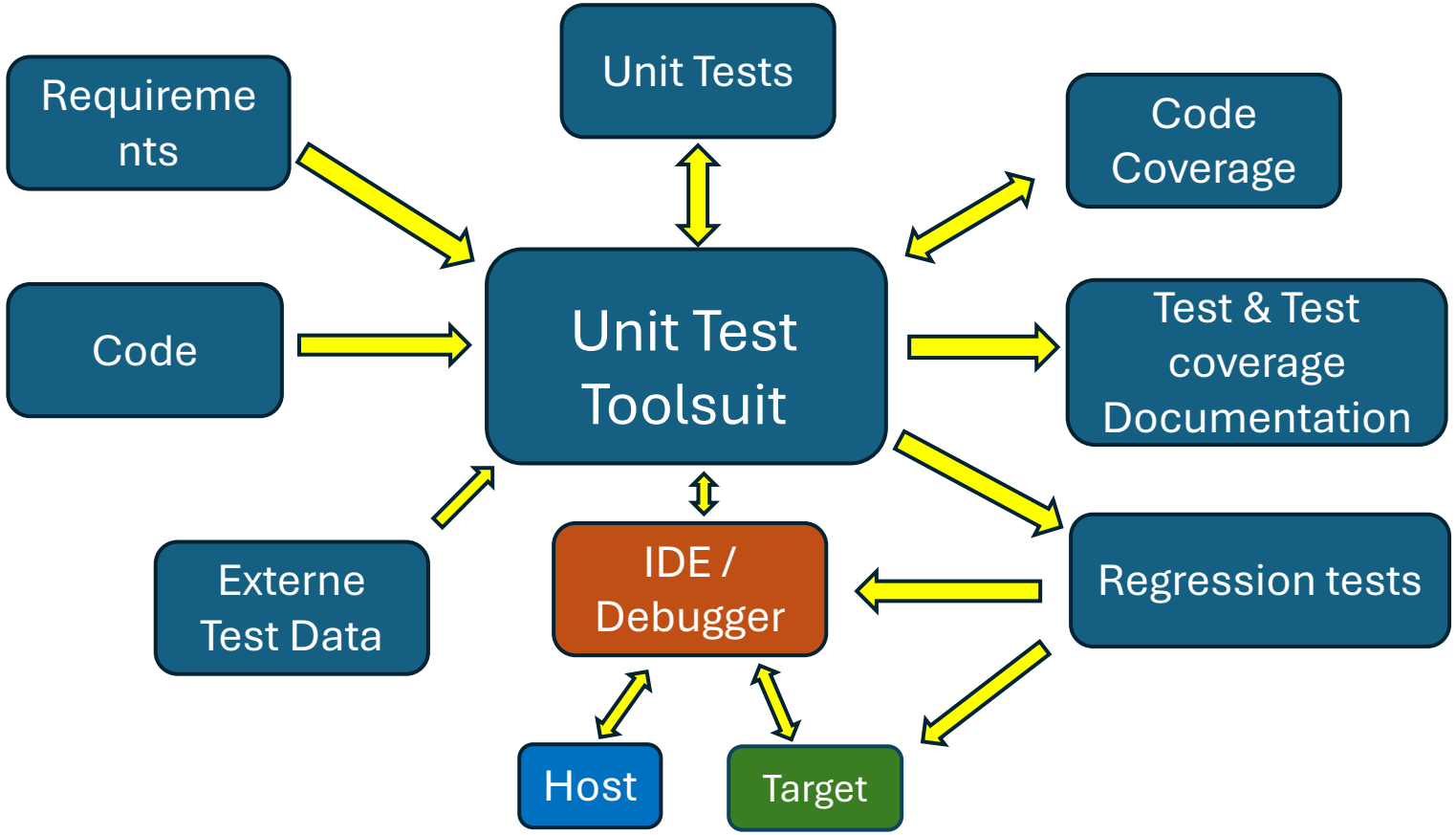
- Requirements import
- Code Coverage / Unit Test
- On – Target testing
- Integrated with IDE / Compiler
- Regression testing
- Documentation & reporting
- For highest version : Assembly level test and MCDC coverage
- Certified edition, Safety manual

DEMO on Stand 22/23

Nice to have :

- Fault injection, Variant support
- Interfacing with System level test
- Platform support (e.g. Autosar)

Unit Test ecosystem



Considerations when selecting Unit Test toolsuite

Is it suitable for Embedded Software ?

- IDE / Compiler integration
- CI / CD Build tools integration
- On Target testing

What is included ? (extra options for many tool-suites !)

- Requirements in- and export
- Target support (one specific or all ?)
- Regression testing
- Documentation generation
- Interfacing for System testing
- Certification for your standard

Price (for Floating Network edition)

FHI Federatie van technologiebranches

indes The choice of professionals

Considerations when selecting an RTOS for your Safety project

Should I use an RTOS ?

Retrofitted Safety or Architectural safety by design

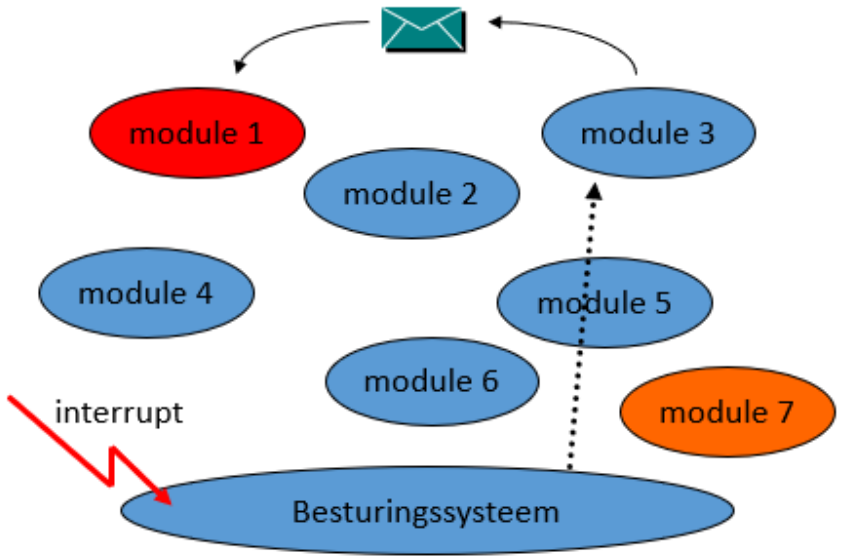
Certified Device drivers !

Reduce costs & risk by :

- Start with Standard RTOS, upgrade to Certified
  - Some Vendors allow this, some within a year.

- Delay fixing MCU choice until committed to certify

RTOS



Should I use an RTOS ?

When use an RTOS

- Multitasking needed
- Combine with non-certifiable code / libraries

- Improve architecture for better / safer code
  - Determinism
  - Inter-Process communication
  - Error handling
  - Improved performance
  - Maintainability / readability

IEC 61508 HR Items for RTOS From Part 3 – Software Requirements

Timing & Scheduling

•HR1: The RTOS scheduler must be deterministic and bounded in execution time
•HR2: Worst-case execution time (WCET) must be analyzed for all tasks
•HR3: Task preemption and priority inversion must be controlled (e.g., priority ceiling or inheritance protocols)
•HR4: Interrupt latency must be bounded and verified

Memory Management

•HR5: Memory partitioning must prevent tasks from corrupting each other's data (spatial isolation)
•HR6: Dynamic memory allocation should be avoided or tightly controlled in safety-critical tasks
•HR7: Stack overflow detection/protection must be implemented

Inter-Task Communication

•HR8: Shared resource access must be controlled via semaphores/mutexes with defined timeout behavior
•HR9: Message queues and buffers must have defined behavior on overflow/underflow

Fault Detection

•HR10: The RTOS must support detection of task overrun (deadline monitoring)
•HR11: Watchdog timer support must be provided and used
•HR12: Error handling for OS service call failures must be defined

Retrofitted versus Architectural safety

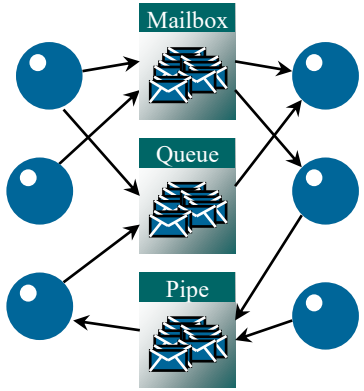
(Source : Claude)

Criterion	FreeRTOS (SAFERTOS)	SCIOPTA
Certification status	SAFERTOS SIL3 (separate product)	SIL3/SIL4/ASIL D, TÜV SÜD certified
Safety architecture	Retrofitted	Native
Freedom of interference	Application responsibility	Kernel-enforced
IPC safety	Shared memory + semaphores	Direct message passing
Safety Manual provided	Yes (SAFERTOS)	Yes
Codebase auditability	Moderate (larger)	High (small, focused)
Mixed-criticality support	Limited	Strong (Protektor)
Toolchain qualification	You arrange separately	IAR certified alongside
Community/ecosystem	Very large	Niche/industrial

Safe by Design / Architecture versus “retrofitted”

- RTOS families

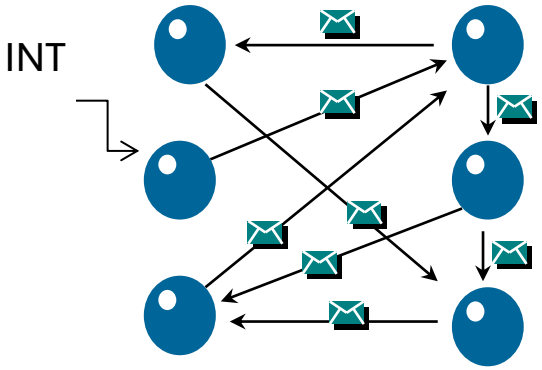
Classic / mailbox



FreeRTOS
embOS
VxWorx
ThreadX
GreenHills

Queues, Semaphors, Mailboxes, pipes etc, etc

(Asynchronous) Direct Message passing

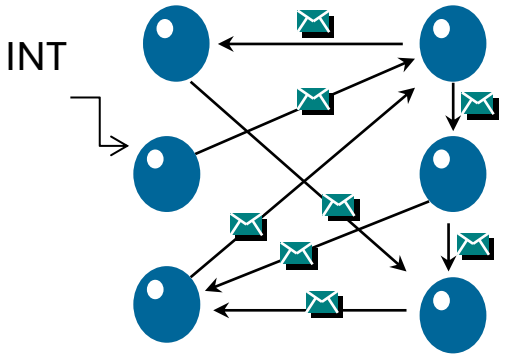


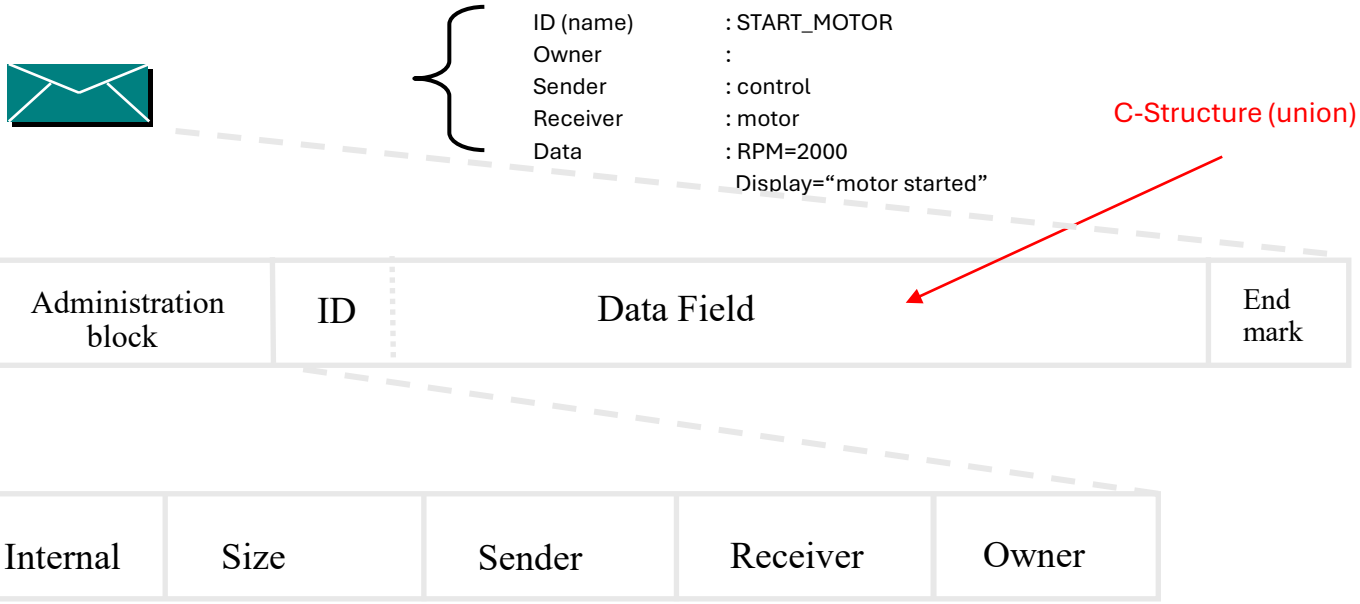
QNX
Sciopta
OSE

Home Brew

Messages (C structures)

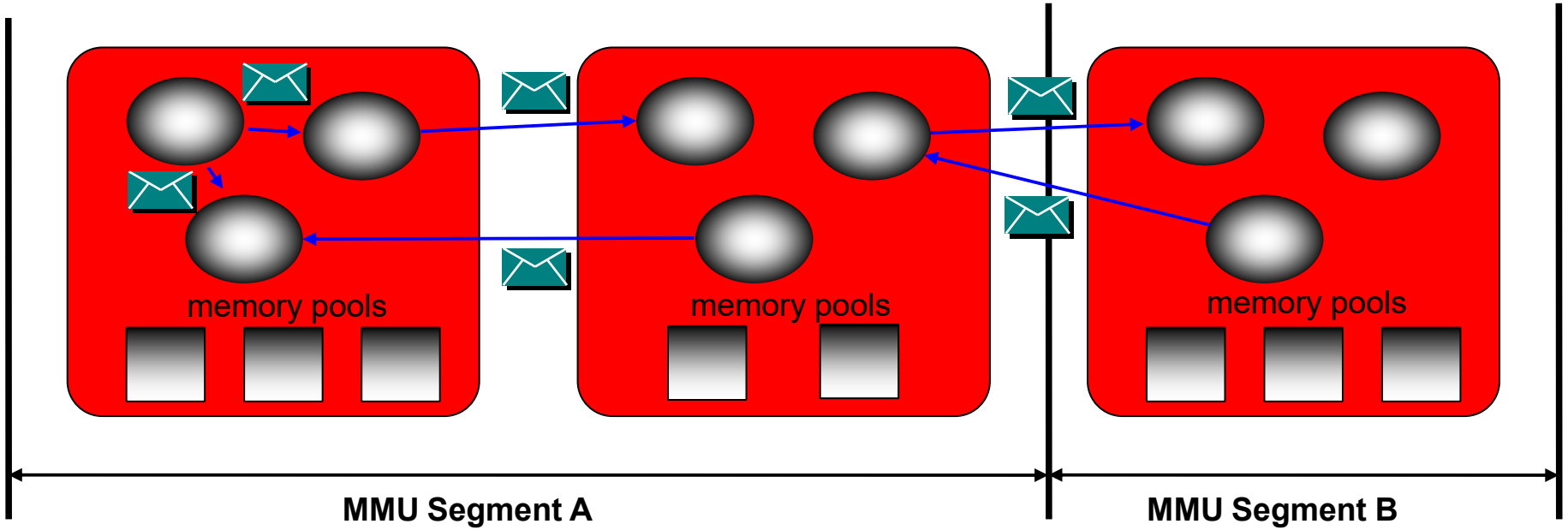
(Asynchronous) Direct Message passing





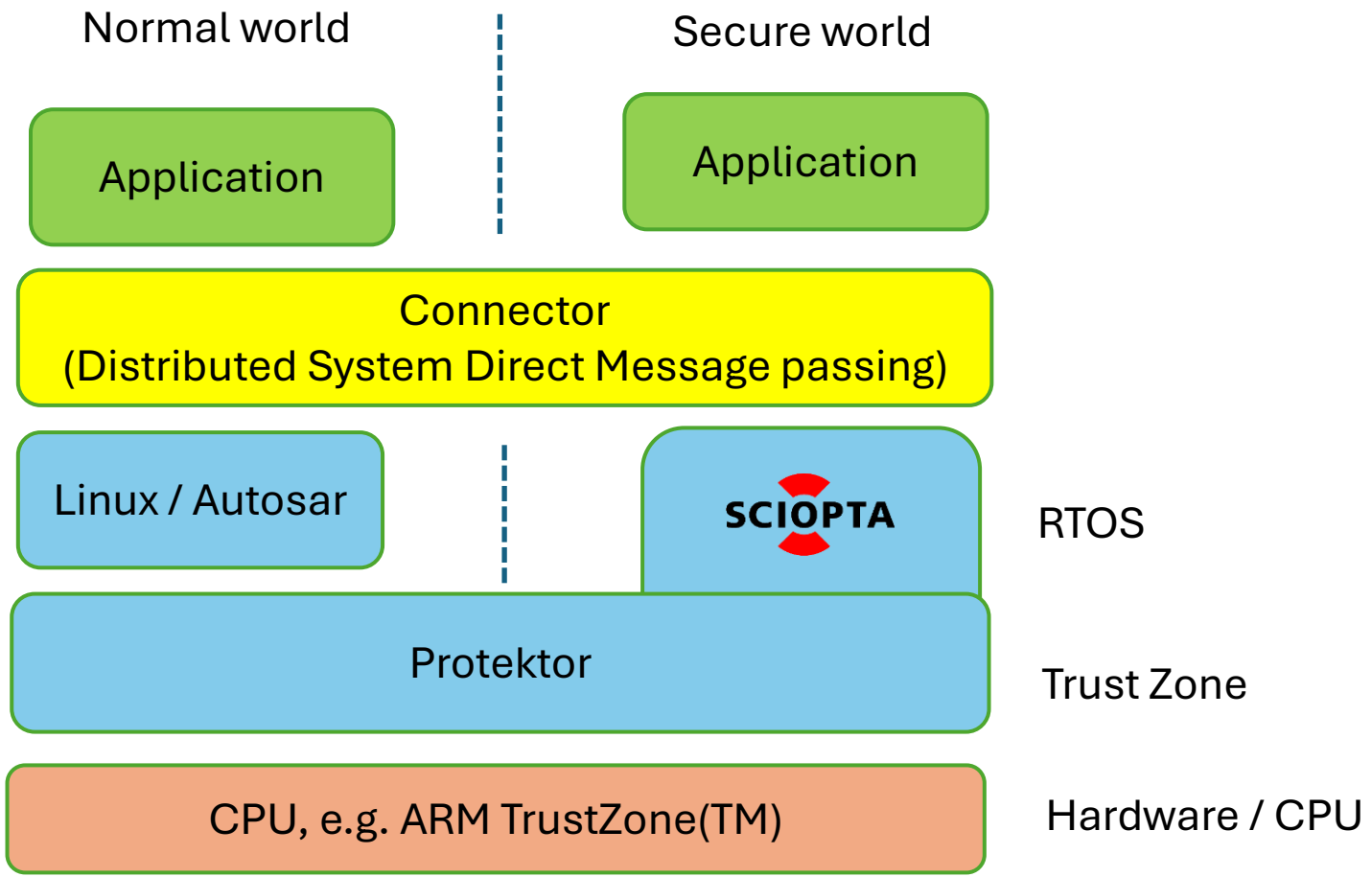
- No fixed format for data
- No requirement for shared memory
- Can be traced and monitored
- Select needed message Asynchronous
- Top – Down / Black Box
- Easy to read & understand
- Safe

Combine safe and 'unsafe' code – Memory Protection



- Automatic Message ~~copy~~ transfer between modules
- No message copy in same segment
- System protection (MMU / MPU)

Combine safe and ‘unsafe’ Software Platforms - Hypervisor



FHI Federatie van technologiebranches

indes The choice of professionals

Thank you for your time

- Welcome on stand 22/23

- Various Safety Manuals
- Demonstration Tessy Unit Test environment