On safety and real-time in embedded operating systems

using modern processor architectures in different safety-critical applications

OSPERT WS - Keynote - 2018-07-03

Michael Paulitsch

Legal Notices and Disclaimers

This presentation contains the general insights and opinions of Intel Corporation (“Intel”). The information in this presentation is provided for information only and is not to be relied upon for any other purpose than educational. Use at your own risk! Intel makes no representations or warranties regarding the accuracy or completeness of the information in this presentation. Intel accepts no duty to update this presentation based on more current information. Intel is not liable for any damages, direct or indirect, consequential or otherwise, that may arise, directly or indirectly, from the use or misuse of the information in this presentation.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

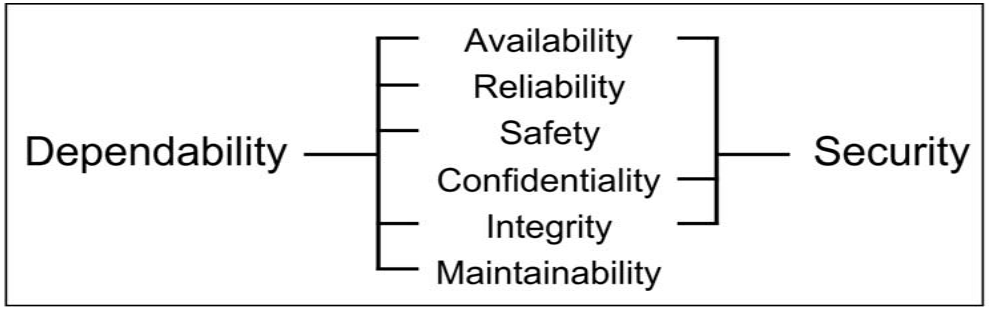
*Other names and brands may be claimed as the property of others.

AT INTEL

WE'RE POWERING THE FUTURE OF COMPUTING AND COMMUNICATIONS, DELIVERING EXPERIENCES ONCE THOUGHT TO BE IMPOSSIBLE.

What is Dependability & Security?

Dependability an integrating concept that encompasses the following **attributes**:

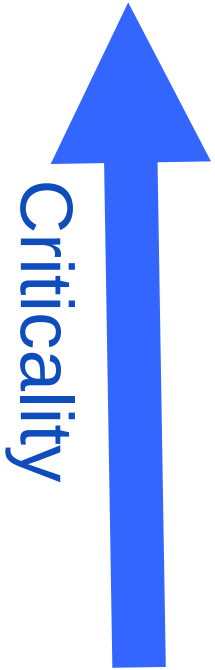


- Availability - readiness for correct service
- Reliability - continuity of correct service
- Safety - absence of catastrophic consequences on the user(s) and the environment
- Integrity - absence of improper system alteration
- Maintainability - ability for a process to undergo modifications and repairs

Security: composite of the attributes of confidentiality, integrity, and availability, requiring the concurrent existence of 1) availability for authorized actions only, 2) confidentiality, and 3) integrity with “improper” meaning “unauthorized”

Laprie et al 2004 :

Safety Assurance Levels in Aerospace and Railway (e.g. DO-178C/ED-12C, EN 50129, …)



Software/hardware whose anomalous behaviour would cause or contribute to a failure of system function resulting in a failure condition for the aircraft / railway system that is:

Level A - Catastrophic 10^{-9} failures/hour	**SIL 4** 10^{-8} failures/hour
Level B - Hazardous/Severe-Major	**SIL 3**
Level C - Major	**SIL 2**
Level D - Minor	**SIL 1**
Design Assurance Level E - No Effect	**SIL 0**
	Safety Integrity Level - SIL 0 (non-SIL)

Avionics

Electronics in Airplane

Trends in Aerospace

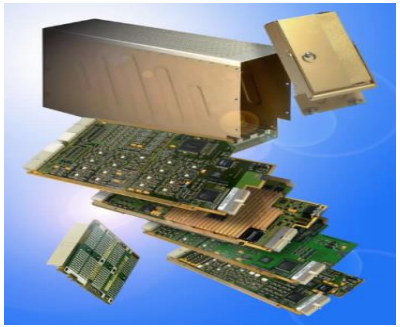
Trend towards new and additional IT-services and denser functional integration:

Demand for new and additional IT-services on aircraft itself and between aircraft and ground

- Integrate formerly physically separated functions onto one platform
- New failure modes and failures
- New threats and vulnerabilities (security, but affecting safety)

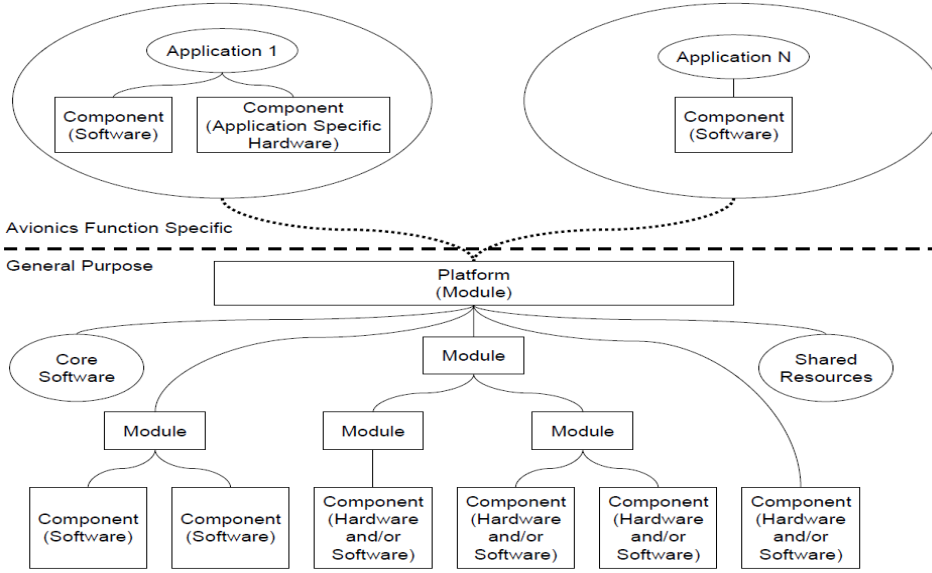
Trend Towards Integrated Modular Avionics (IMA)

Due to weight constraints integration of multiple aircraft functions (of possibly different criticality) onto common platforms is an ongoing architectural trend in aerospace



A380 IMA components

Source: Airbus © Airbus



Relationship of IMA applications and HW/SW Modules

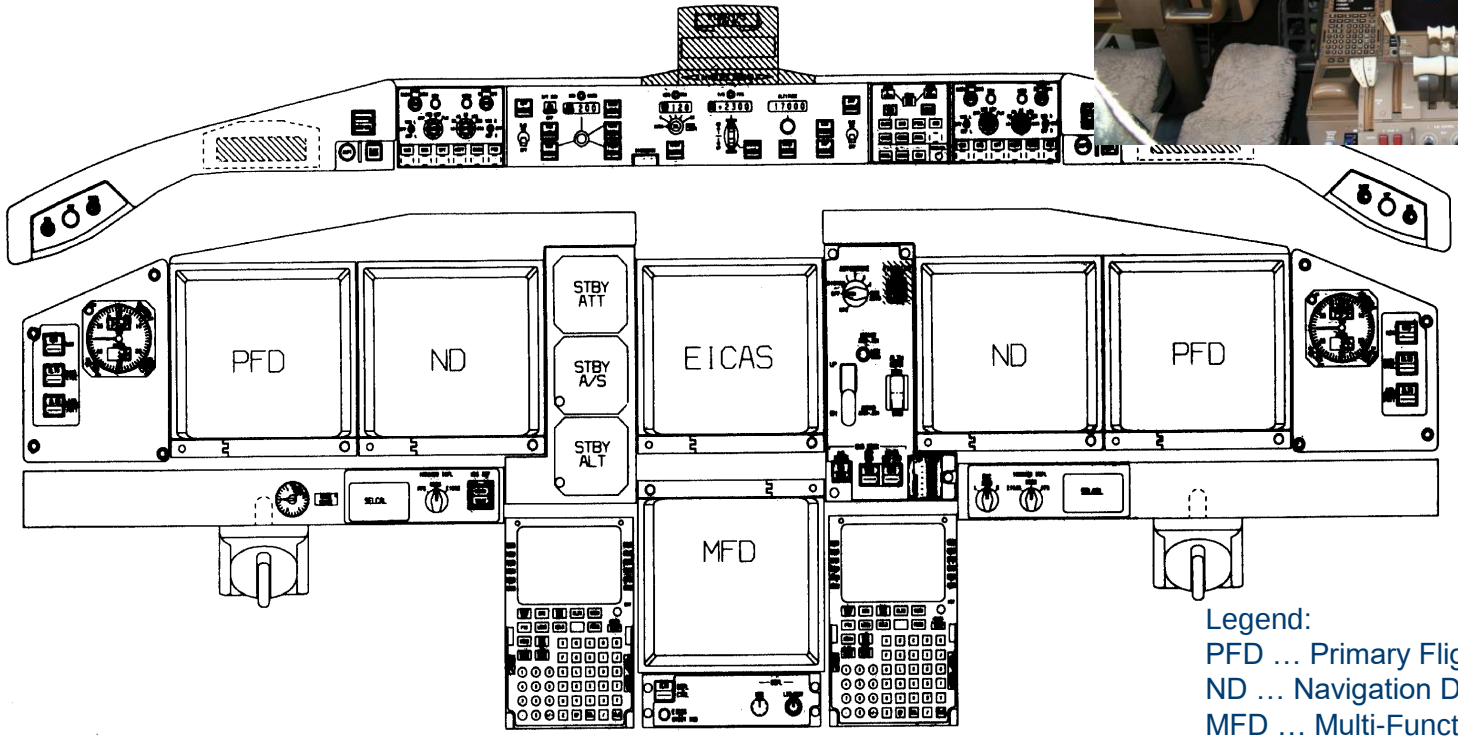
Source: ARINC297
© ARINC

Mixed-Criticality System in Industry – What's it?

Multiple criticalities (residing) on same platform

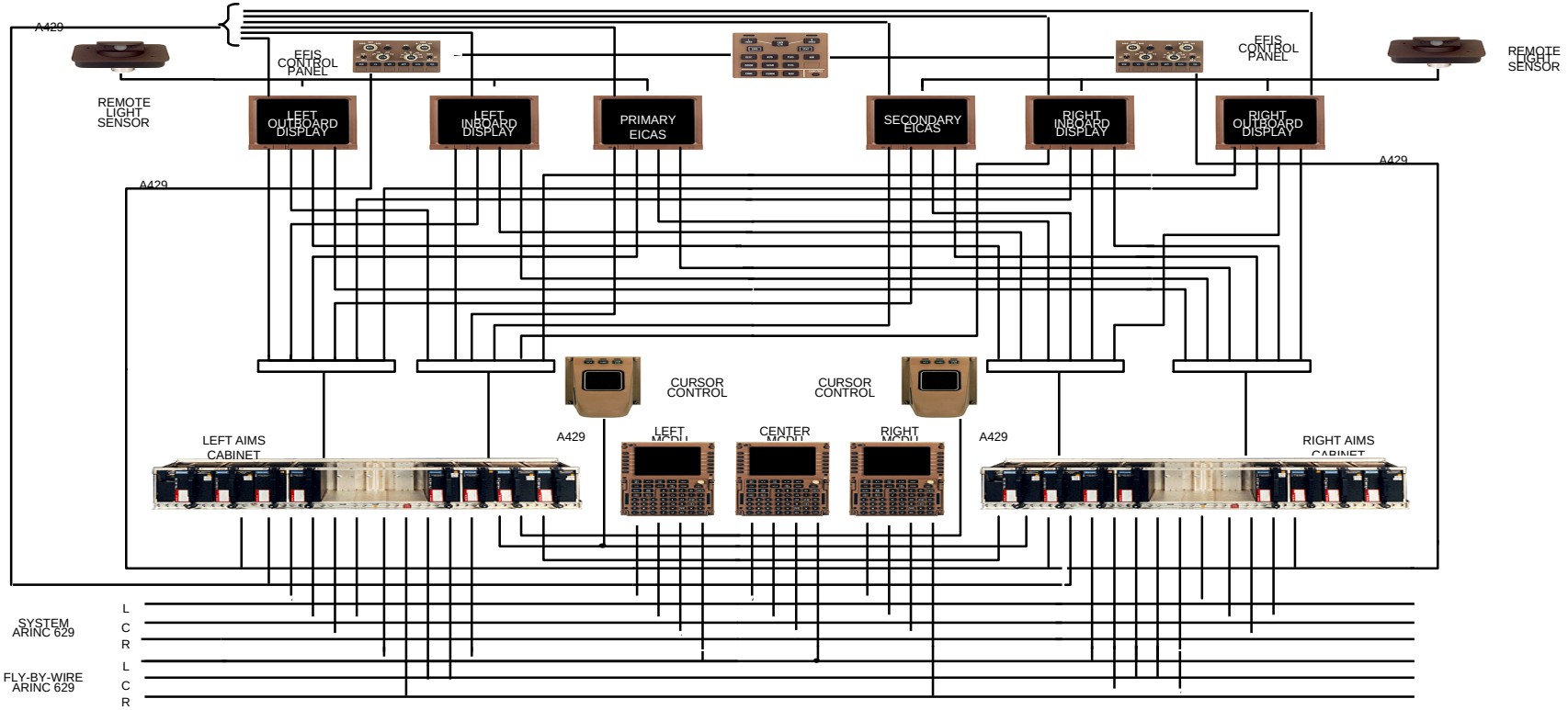
- Key requirement for platform: Platform needs to fulfill safety requirements at minimum of **highest safety** requirement of application. Security criticality requirements may be derived from safety requirements or from security data separation.
- Criticalities are **assigned by safety or security process** and typically don't change during operation
- Safety: Chosen independence between applications to minimize interaction between otherwise independent "safety chapters" (system level safety analysis extremely complicated w/o this requirement).
- Security: co-habitance of different security levels needed for cost reasons or because of inherent security function (gateway, firewall)
- Deployed for many years in aerospace (B777, B787, A380, A350, E170/175, E190/195, …) under the name Integrated Modular Avionic (IMA) systems

Aircraft Cockpit



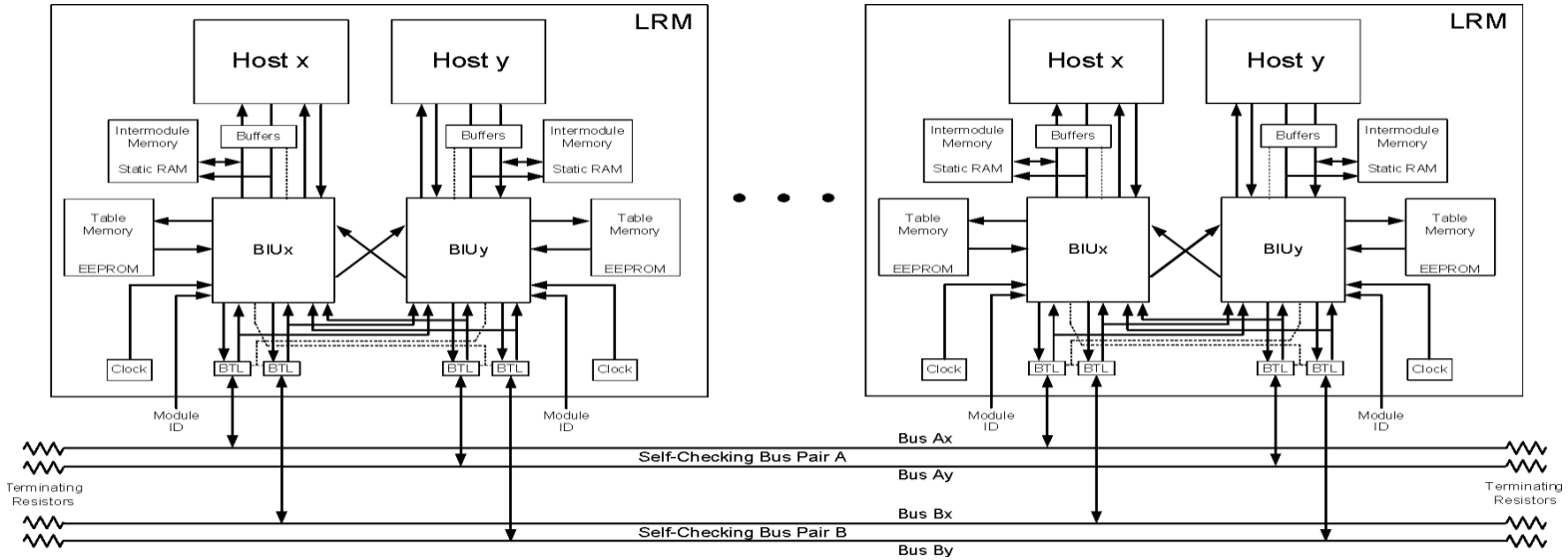
Legend:
PFD … Primary Flight Display
ND … Navigation Display
MFD … Multi-Function Display
EICAS … Engine Info & Crew Alert System

Boeing 777 Avionics Architecture
Real-Life Mixed Criticality System



Boeing 777 – Avionics – Computer Level

Avionics based on ARINC629 system bus and ARINC659 (SafeBus).

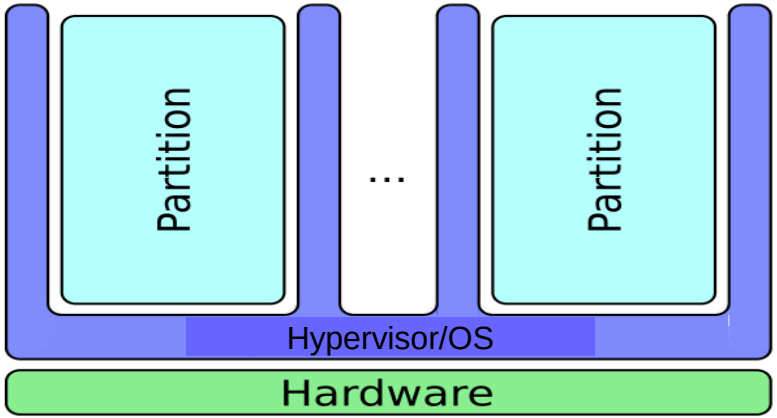


Deterministic relatively simple compute and network architecture

Partitioning

Is a concept for spatial and temporal separation/segregation of functionally independent components:

- Prevents interference between two components
- Incremental development



Types of partitioning

- Time partitioning: temporal aspect
- Space partitioning: memory aspect
- I/O partitioning: time and space partitioning for I/O

Implementation means

- Partition/process: independent segregated environment
- Separation kernel / Memory Management Unit: control instance
- Temporal partitioning: time slicing; dynamic (fair) scheduling policies

How to Achieve Availability and Integrity in a Mixed-Criticality System?

Correctness of implementation important for safety and availability

Examples of High-Assurance Requirements

- Domains need to fulfill **separation** requirements despite possible integration on same hardware to ensure proper item integrity and availability
- **Controlled information flow**: Communication between domains need to fulfill rules to ensure proper protection of functions – stronger focus on
  - Integrity and availability of functions
  - Authorized flow definition

Orion

Multi-Purpose Crew Vehicle

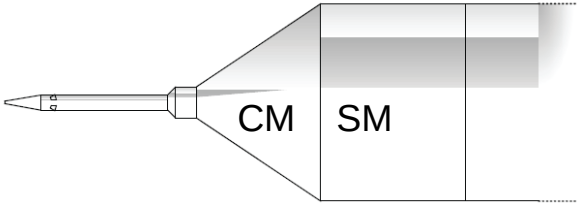
Next generation U.S. spacecraft

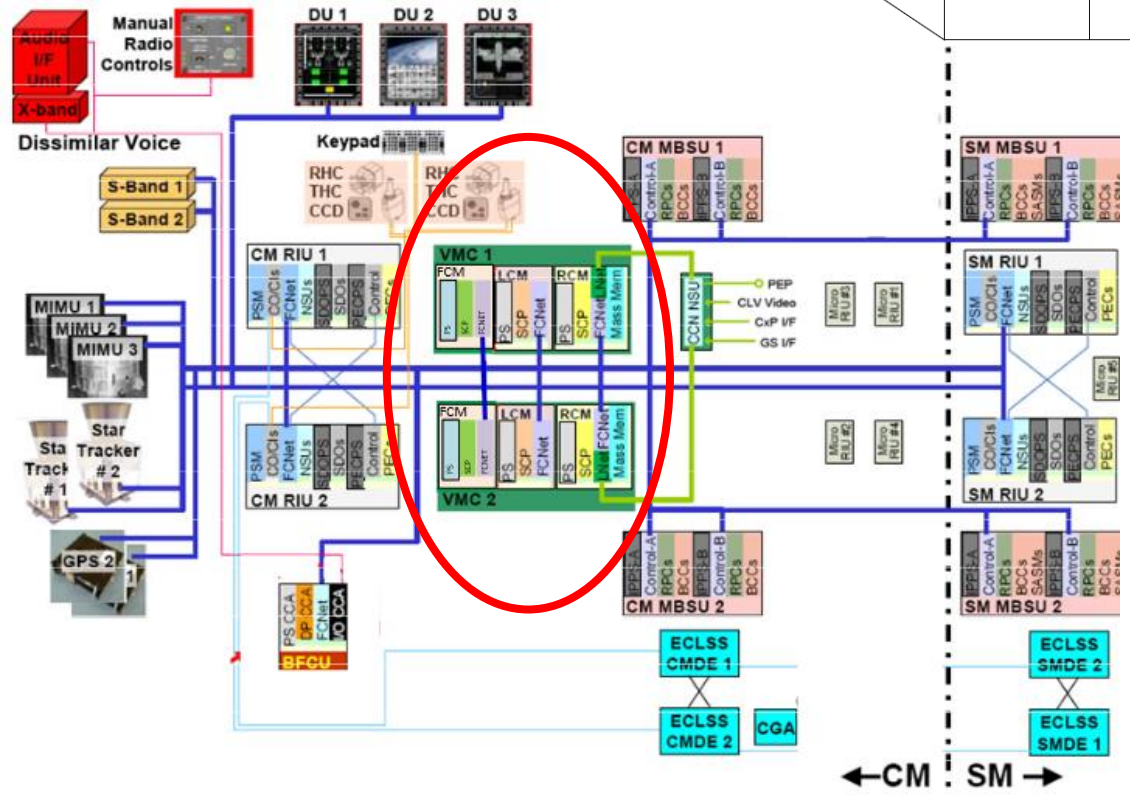
Long mission times (weeks to 6 months)

Inside View – Cockpit Orion

Avionics (Snapshot)

- Time-triggered network
- High-integrity compute
- System-level redundancy management

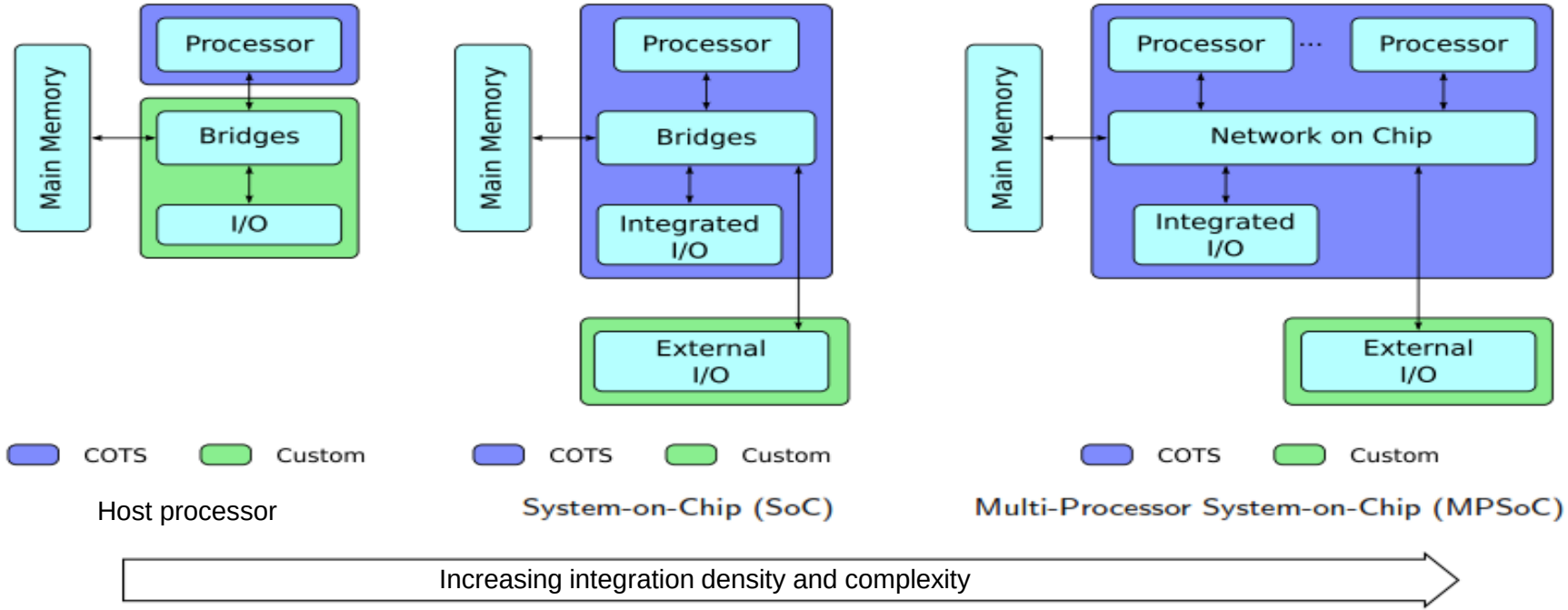




Multi-core

Time Partitioning

Chip Evolution



View of Aerospace **Multi-Core** Certification Body Related to Timing

Only selective view of publicly available FAA CAST-32 paper

(Functional) interference channels of multi-core processors

- Concerns: there may be software or hardware channels through which the MCP cores or the software hosted on those cores could interfere with each other

Shared resources like Memory / Cache

- Concerns: Memory or cache memory that are shared between the processing cores
- … can lead to problems such as the worst-case execution times (WCETs) of the software applications hosted on cores increasing greatly due to repeated cache accesses by the processes hosted on the other core, leading to repeated cache misses.

Planning and Verification of Resource Usage

- Concern: Interconnect Fabrics / Interconnect Modules as source of non-deterministic behavior, fear of resource capacity violation, …

Multi-core: General Possible Undesired Effects (Temporal)

Other possible undesired effects affecting temporal determinism

- How does current hardware affect mixed criticality and especially interference?
- What can be done about it (analysis, improvement, inclusion in processes) especially in current commercial off the shelf (COTS) architectures.

Details in papers

- O. Kotaba, J. Nowotsch, M. Paulitsch, S. Petters, H. Theiling. Multicore In Real-Time Systems - Temporal Isolation Challenges Due To Shared Resources. WICERT workshop as part of DATE 2013.
- D. Dasari, B. Akesson, V. Nelis, M.A. Awan, S.M. Petters. Identifying the Sources of Unpredictability in COTS-based Multicore Systems. SIES conf. 2013.

Shared resource	Mechanism
System bus	Contention by multiple cores Contention by other device - IO, DMA, etc. Contention by coherency mechanism traffic
Bridges	Contention by other connected busses
Memory bus and controller	Concurrent access
Memory (DRAM)	Interleaved access by multiple cores causes address set-up delay Delay by memory refresh
Shared cache	Cache line eviction Contention due to concurrent access Coherency: Read delayed due to invalidated entry Coherency: Delay due to contention by coherency mechanism read requested by lower level cache Coherency: Contention by coherency mechanism on this level
Local cache	Coherency: Read delayed due to invalidated entry Coherency: Contention by coherency mechanism read
TLBs	Coherency overhead
Addressable devices	Overhead of locking mechanism accessing the memory I/O Device state altered by other thread/application Interrupt routing overhead Contention on the addressable device - e.g. DMA, Interrupt controller, etc. Synchronous access of other bus by the addressable device (e.g. DMA)
Pipeline stages	Contention by parallel hyperthreads
Logical units	Contention by parallel applications
	Other platform-specific effects, e.g. BIOS Handlers, Automated task migration, Cache stashing, etc.

Assessment of Multi-Core Worst-Case Execution Behavior - Overview

Motivation:

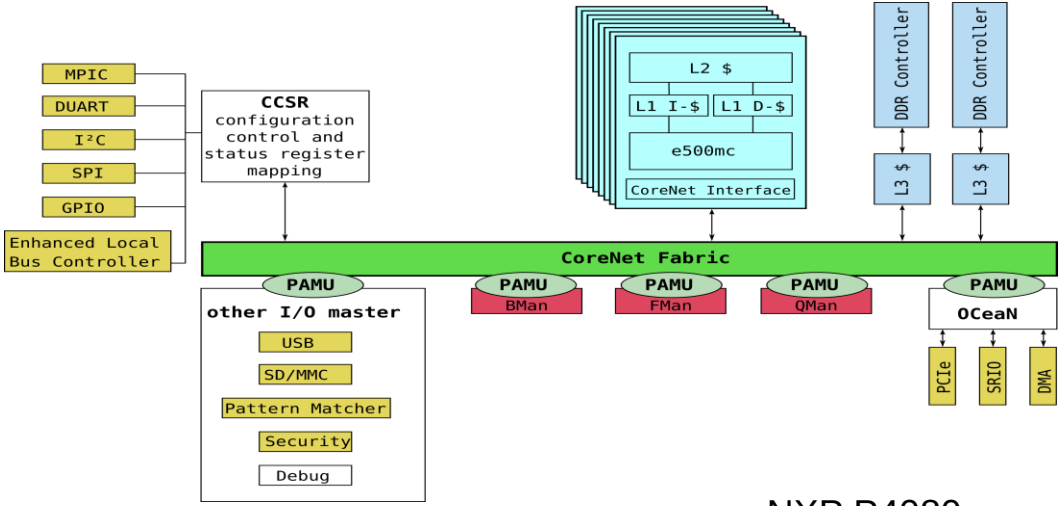
- Integration leads to common use of shared resources. Partitioning impact needs to be evaluated for safety-critical applications, such as IMA

Goal:

- Analysis of partitioning features of modern multi-core computer in context of use in IMA
- Impact of integration on worst-case timing (WCET) of application

Approach

- memory-intensive tests



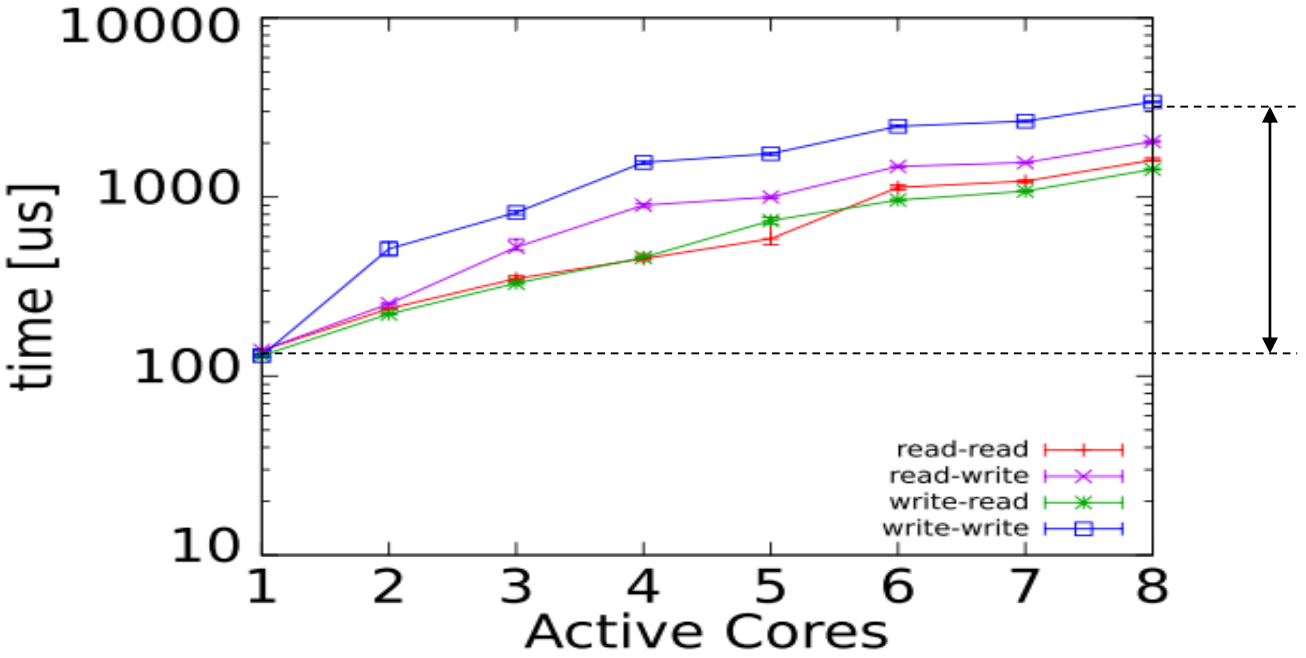
NXP P4080

Focus of work:

- Network on Chip (limited data available); some memory access performance tests

Details of work published at EDCC2012 (J. Nowotsch, M. Paulitsch)

Assessment of Multi-Core WCET
Memory (DDR) Accesses (8 Cores)



1/30 about 4 times slower overall (8*1/32)

Worst-case access time increases over-proportionally with more cores.

Some Measured Values for NXP P4080
Interference Between Single-Core and 8-Core Systems

Worst-case influence (for 8 core multi-core system)

Worst case observed versus worst-case analysis → some conclusions can be drawn for average case (slack between average and worst case)

	single-core			multi-core		
bmark	max. OET [ms]	upper bound [ms]	bound deviation [%]	max. OET [ms]	upper bound [ms]	bound deviation [%]
cacheb	619	705	13.9	1934	9378	384.9
iirflt	745	951	27.7	2476	12497	404.8
rspeed	963	1418	47.3	2327	19021	717.3
a2time	121	251	107.3	334	2971	790.9
bitmnp	2300	3504	52.4	5781	49170	750.5
tblook	2699	4556	68.8	7684	61156	695.9
matrix	464	8075	1642.0	1212	98075	7993.5
aifftr	188	1217	547.4	489	159313	32513.9

>> 8 times greater

Difference greater for multi-core (more “slack”)

Context info: EEMBC benchmark; OET … Observed Execution Time; bound … analyzed using AbsInt AiT

WCET for Multi-Core Computer Combined with Monitoring

Basic idea to benchmark/analyze hardware and include access interference and monitor memory accesses (RTNS 2013 paper, ECRTS 2014 paper)

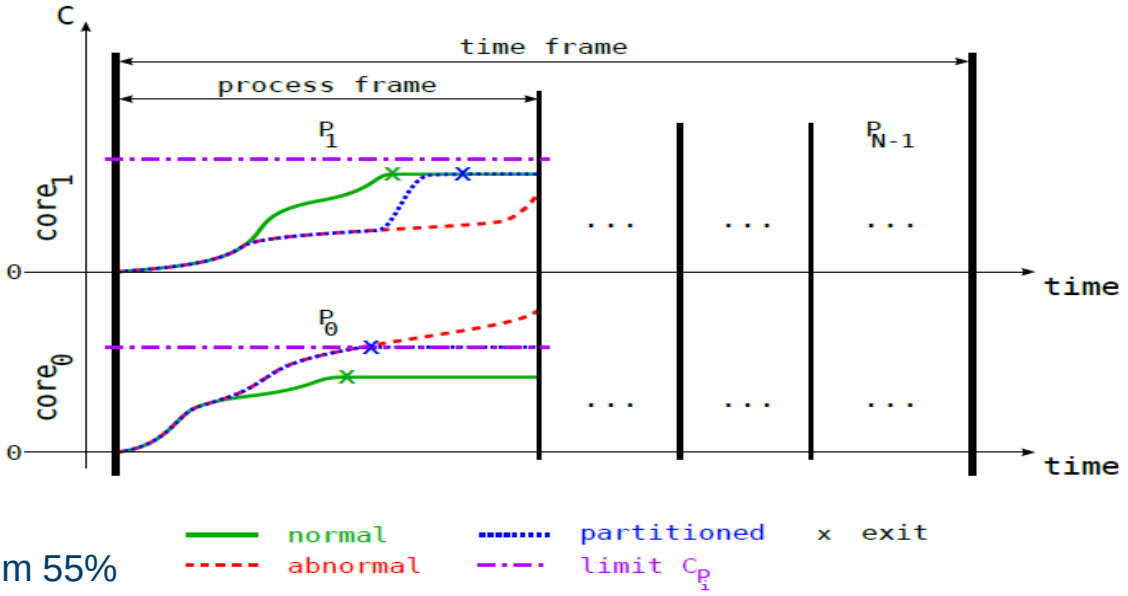
- Extension of timing analysis
- Applied to AbsInt's aiT – commercial static WCET framework (extension memory accesses)
- Runtime Monitoring
- Applied to bare-metal OS layer
- Applied to SYSGO's PikeOS
- Applied to Windriver VxWorks

Average-Case Extension

- Applied to bare-metal OS layer

Evaluation

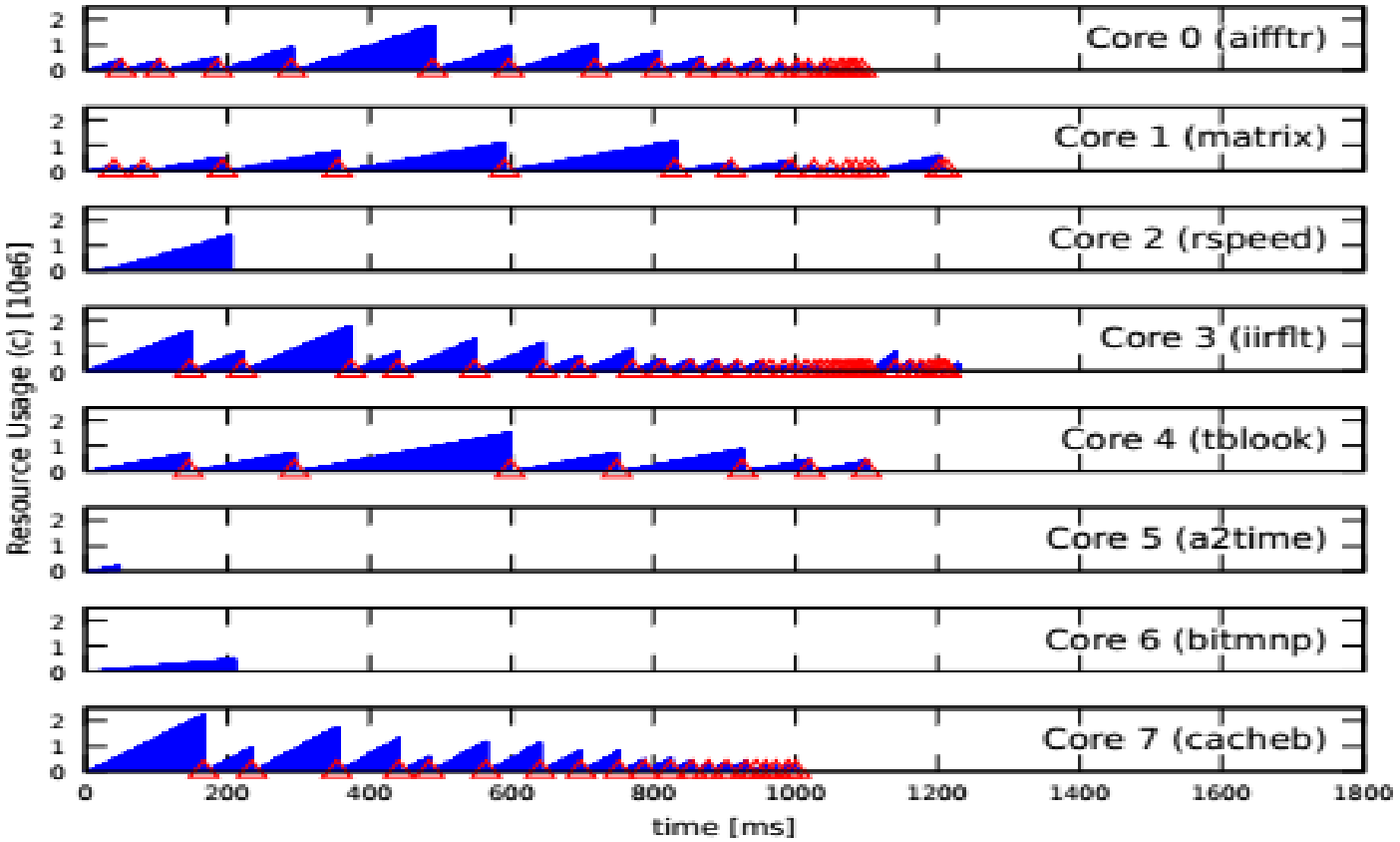
- Based on Freescale's P4080, other processors evaluated
- Benchmarks deduced from EEMBC Autobench benchmark suite



WCET reduction:

- Utilisation increase: core 98.9%, system 55%
- Additional accesses: 2 to 70 times the accesses that were statically assigned (Nowotsch et al, 2014+15)

Evaluation – Runtime Analysis



Complexity Is Increasing …

How would such a approach scale with “more” complex systems?

What about new memory architectures?

Memory accesses are not an optimal measure of progress: are there other metrics achieving better WCET and performance?

What about more DMA channels?

I/O?

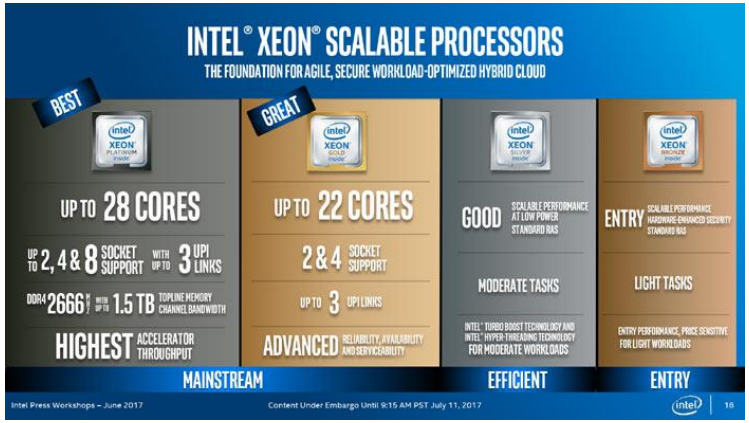
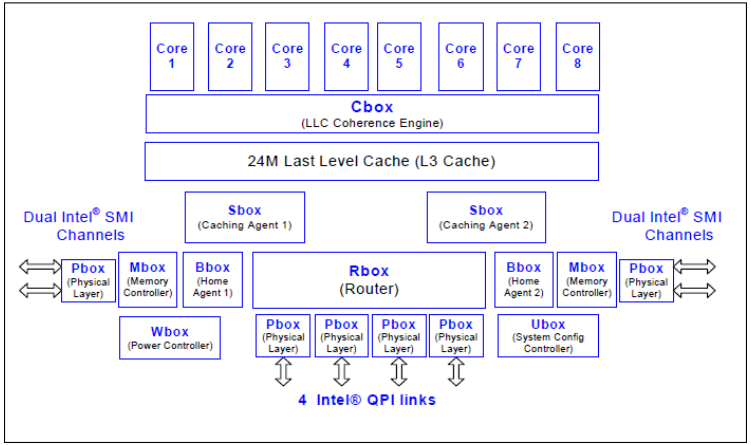


Figure 2-2. Intel® Xeon® Processor 7500 Series Block Diagram



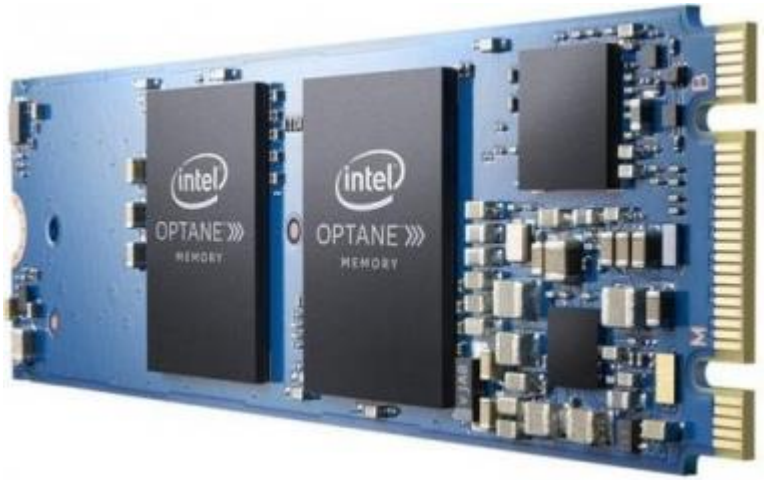
New Memory Architectures/Properties

Are there different OS structures with different memory properties?

E.g. Optane memory

- Persistence
- Quick access

Can we leverage this for improved guaranteed performance?



Time-Coordinated Computing (TCC) & Time-Sensitive Networks (TSN)

TCC … coordination of pheripherals and across SOC

TSN (802.1Q) … Ethernet timing sync, path control and reservation, …

Is there a new system optimium? Are we back in the “old days”?

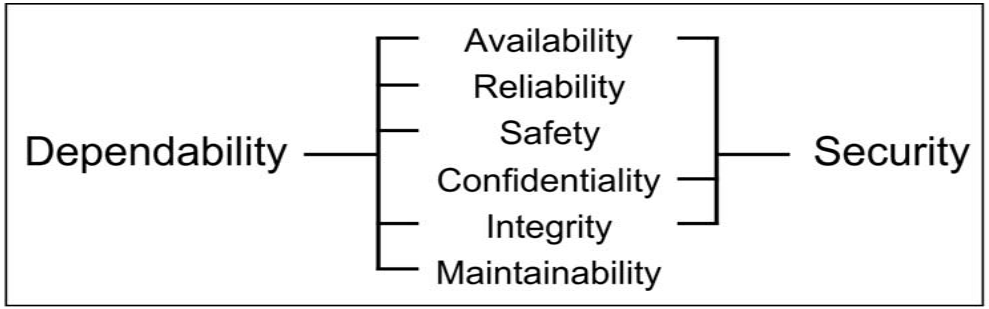
OS support in critical systems?

The next level of determinism

Most importantly for industrial applications, the new Intel Atom processor enables Intel® TCC Technology, which coordinates and synchronizes peripherals and networks of connected devices, such as PLCs and gateways. By synchronizing clocks inside the SoC and across the network, Intel TCC Technology can achieve network accuracy to within a microsecond, which can lead to extremely low cycle times that improve efficiency, reliability, and productivity in complex manufacturing and industrial processes.

In the end …

Simplicity needed for timing guarantees (availability) affecting safety



Integrity is a must

More diverse computing requirements (safety-criticality / real-time) expected

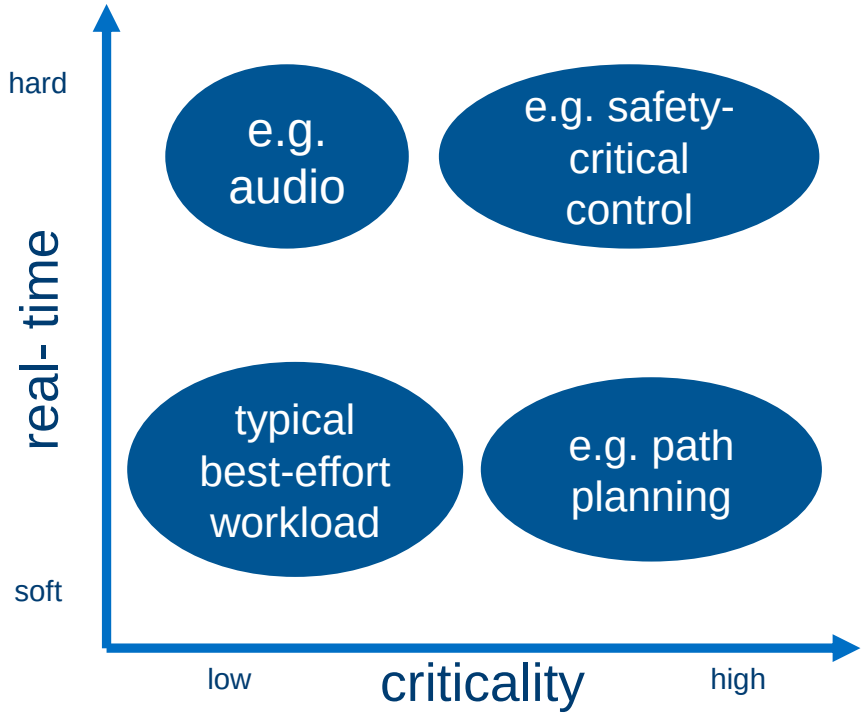
Criticality / Real-Time Application Requirements

Differentiate workload

- Tight timing requirements
- Criticality

Possible consequences

- Intelligent load management
- Slicing of computing & networking with guarantees



Virtualization is Key

ACRN™

Current
Data Center Hypervisors

- Too large for embedded IoT development
- No safety-critical workload considerations
- Requires too much overhead for embedded development

Current
Embedded Hypervisors

- Highly dependent on closed source proprietary solutions
- Expensive
- Makes product longevity difficult
- Hard partition, no ability to share resources

No Open Source Hypervisor solution currently exists that is **optimized for embedded IoT** development

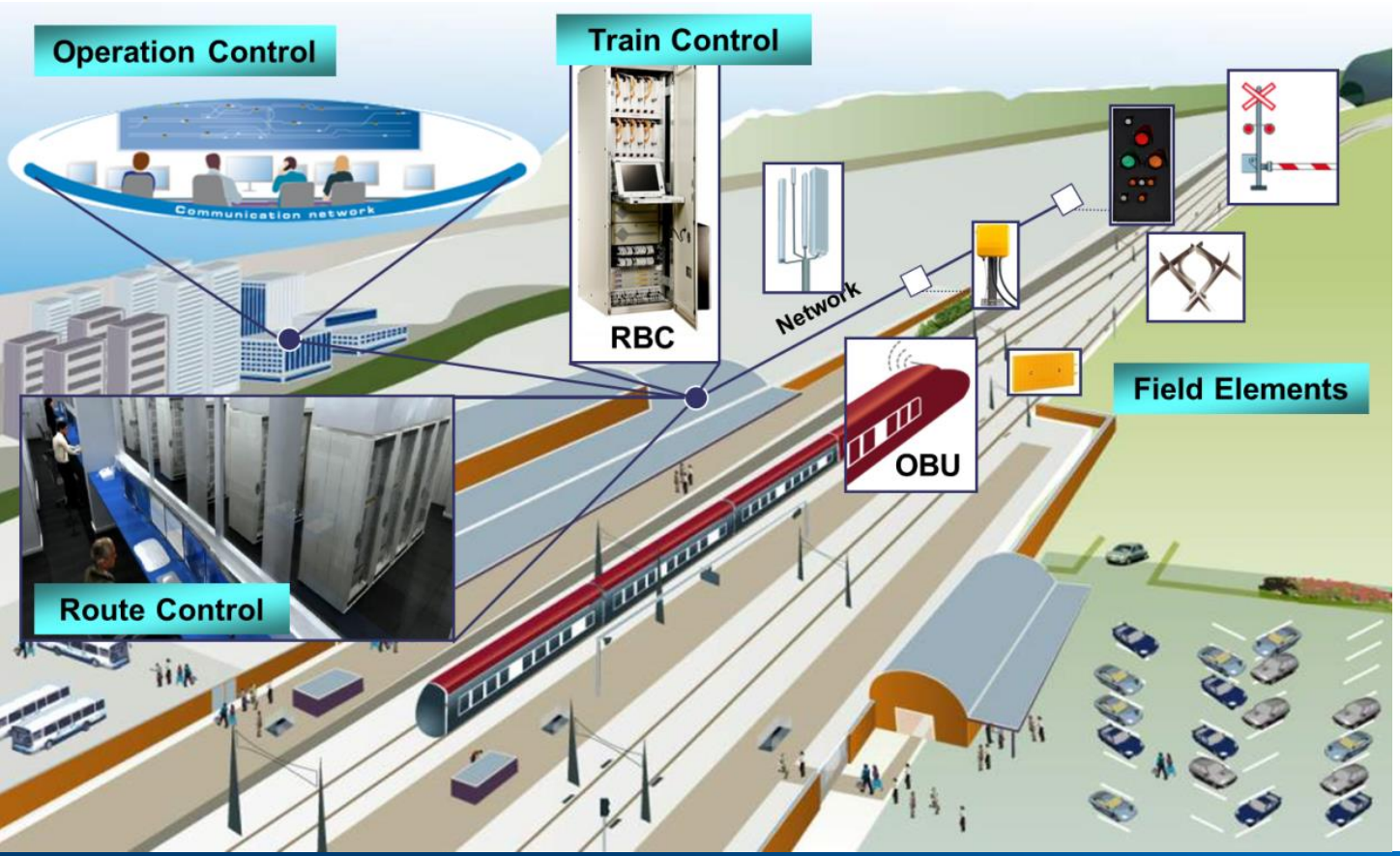
Project ACRN™ Pillars

ACRN™ is a flexible, lightweight reference hypervisor, built with real-time and safety-criticality in mind, optimized to streamline embedded development through an open source platform

Small footprint	Built with Real Time in Mind	Built for Embedded IoT	Safety Criticality	Adaptability	Truly Open Source
• Optimized for resource constrained devices • Few lines of code: Approx. only 25K vs. <156K for datacenter-centric hypervisors	• Low latency • Enables faster boot time • Improves overall responsiveness with hardware communication	• Virtualization beyond the “basics” • Virtualization of Embedded IoT dev functions included • Rich set of I/O mediators to share devices across multiple VMs	• Safety critical workloads have priority • Isolates safety critical workloads • Project is built with safety critical workload considerations in mind	• Multi-OS support for guest operating systems like Linux and Android • Applicable across many use cases	• Scalable support • Significant R&D and development cost savings • Code transparency • SW development with industry leaders • Permissive BSD licensing

Railway

Overview Railway – Signal Control



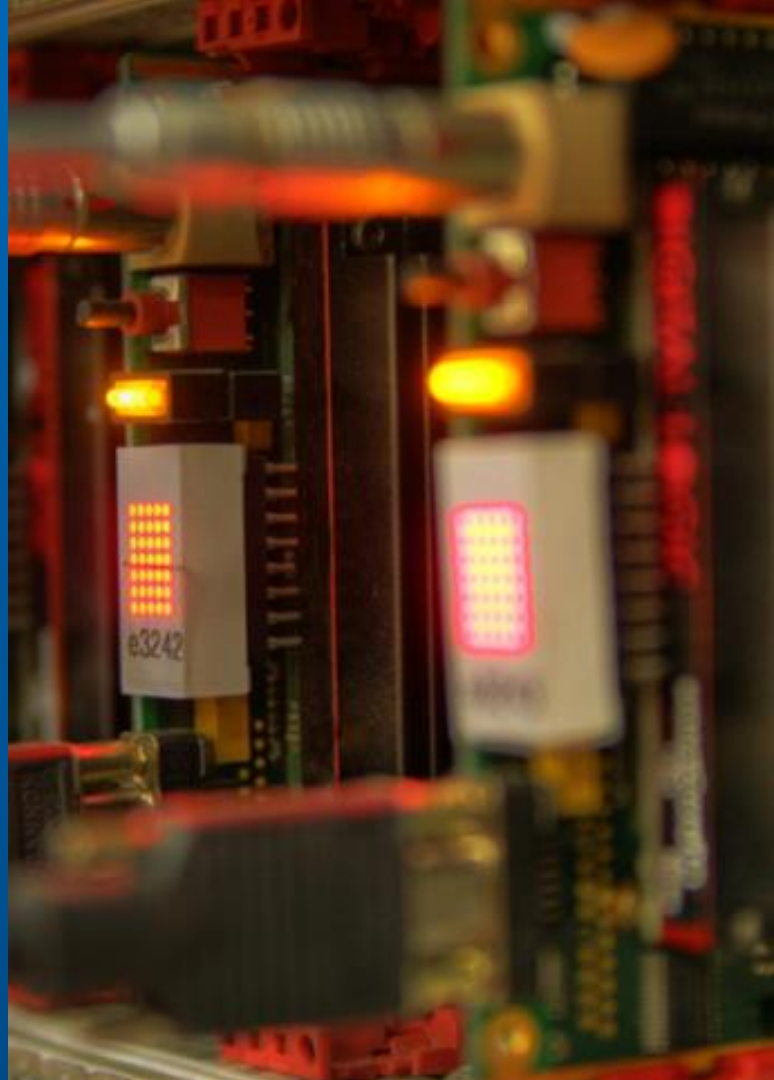
Trends

- Removal of some field elements (signals, …)
- Remote moving authority
- Central operation centers
- Autonomous operation

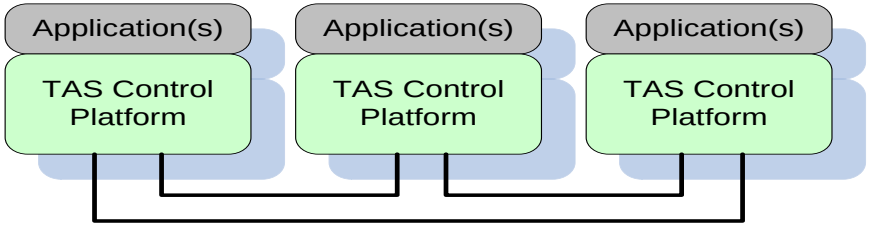
RBC … remote block center
OBU … on-board unit

Thales - TAS Platform

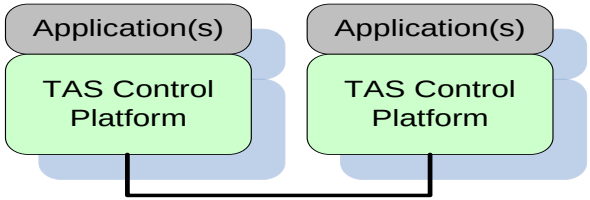
- Vital Hardware & Software Platform, common for all signalling applications in Ground Transportation Systems (GTS)
- Enables hardware independent signalling applications



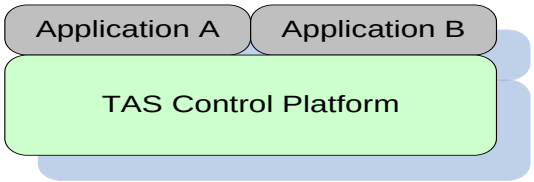
TAS Control Platform: Supported Redundancy Architectures



2oo3

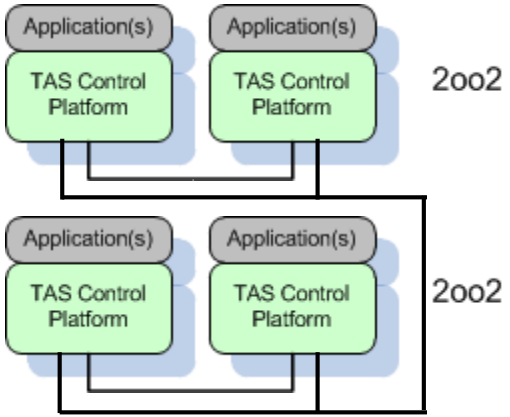


2oo2



1oo1

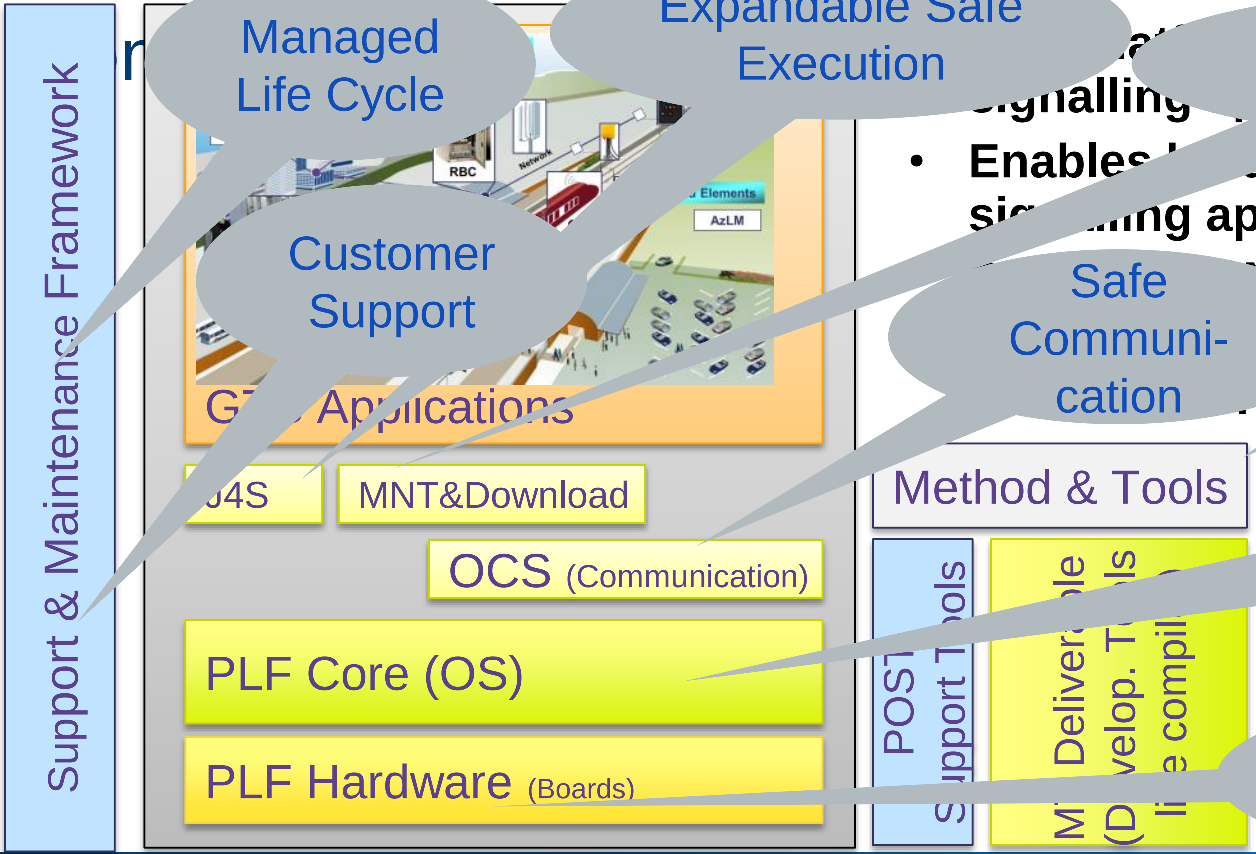
2x2oo2



2oo2

2oo2

TAS Platform – Safe C… tation and …



…signalling…

- Enables h…ware independent sig…ling applicat…

…N501…

…ct l… deployed all…

…nd

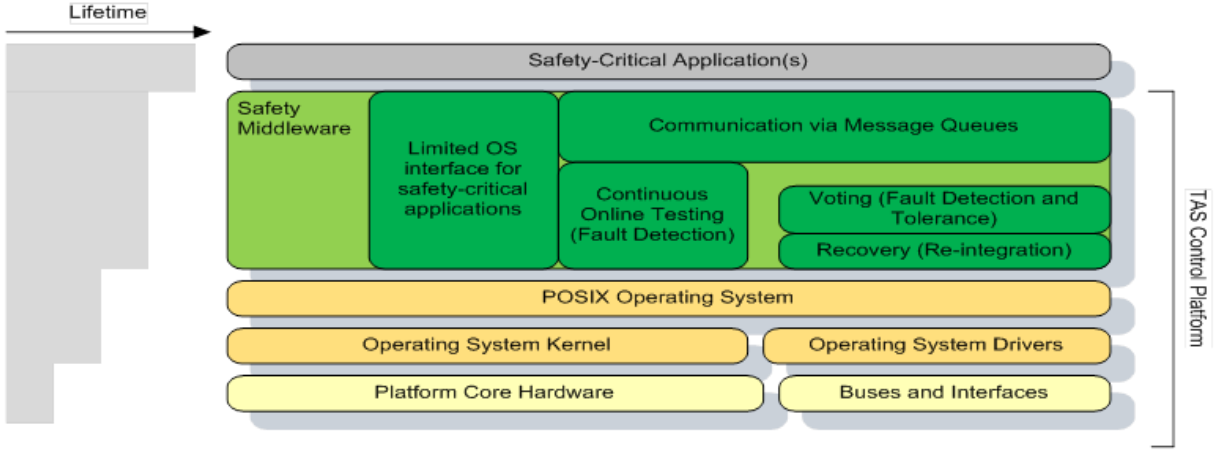
TAS Platform is Based on Linux

In addition to safety layer and functional services (communication)

Integrity of SIL4 is essential!

Supervision of timing

Use existing COTS security packages of Linux possible

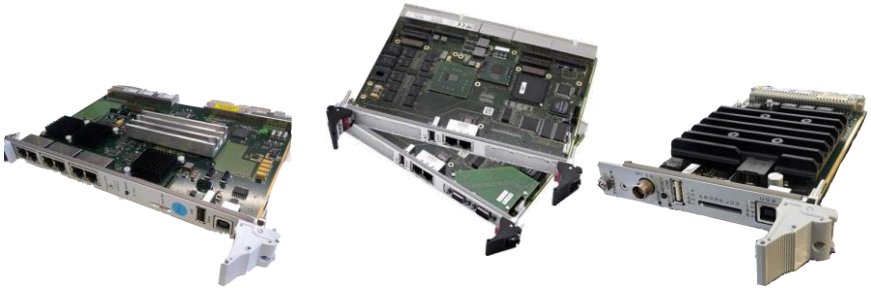


Layered safety approach allows integration of security and implement safety functions

Example: TAS Platform in Used in Applications

Interlocking

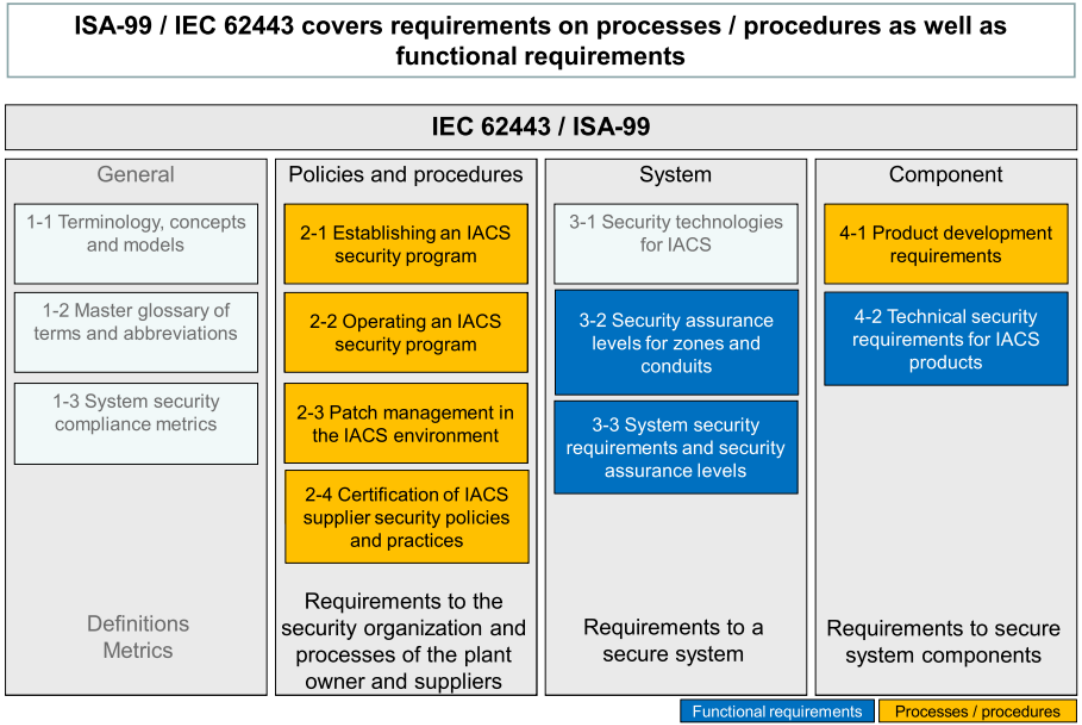
Onboard System (ETCS)



Exemplary boards

© Thales

IEC 62443 – An Applicable Security Standard Process is Key



Typical Security Management – Patch Management

Removal of zero-day vulnerabilities following standards: IEC 62443 2-3 for Patch Mgmt

Separate safety and security life-cycles

- Using suitable architectures and processes or physical separation of security and safety functions

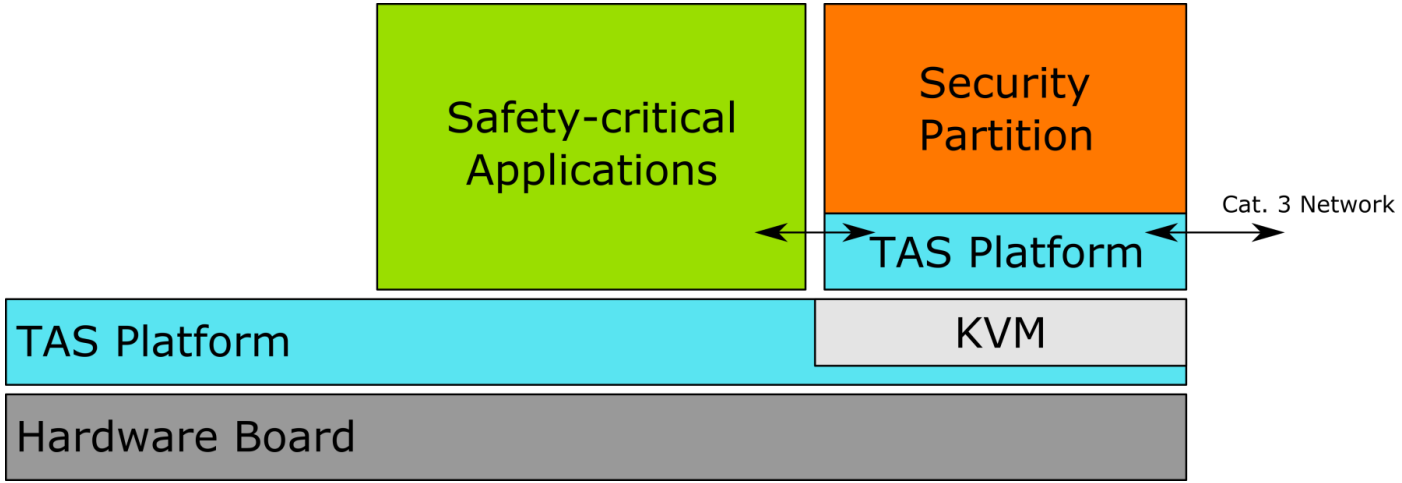
Comment in draft norm (prEN50129: 2016)

NOTE 3 Sometimes it can be necessary to balance between measures against systematic errors and measures against security threats. An example is the need for fast security updates of SW arising from security threats, whereas if such SW is safety related, it needs to be thoroughly developed, tested, validated and approved before any update.

Safety and Security Life Cycle is Different

TAS Platform Safe Security Approach

Virtualization for security and safety life cycle decoupling



- Integration of Safety and Security

Legend:
KVM … Kernel-based Virtual Machine

Automotive

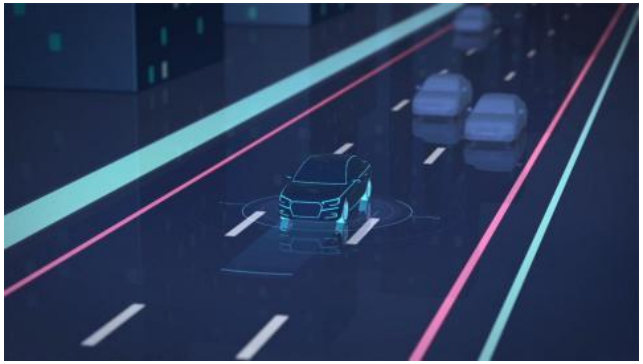
Automated / Autonomous Driving

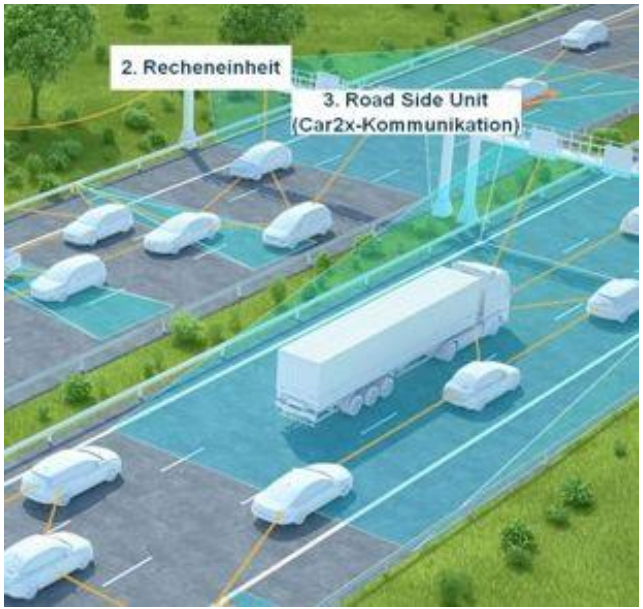
Operational requirements (mission/safety):

- Avionics: safety few hours; operational few hours
- Railway: 24/7 trackside; few hours onboard
- Space: mission and safety: days to months
- Autonomous car: mission time: 1-2 hours?; safety: 1 minute continued operation?
- Automated driving with infrastructure: 24/7?

What does this mean for assurance and temporal supervision/guarantees?

What about integrated compute platforms?





Summary & Conclusions

Re-Cap & Future (1)

ACRN™

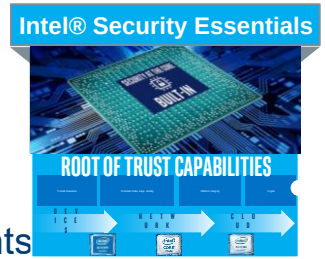
Diagnosis info and operational management approach key to current and future IoT lead to connectivity needs and potential vulnerabilities

- Affecting safety-critical systems (due to security vulnerability)
- Different workloads and criticalities coexist

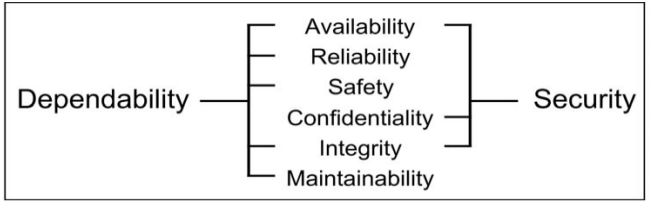
Updates will be the norm: Updates for security purposes (removal of zero-day vulnerabilities)

Application-level fault tolerance aspects often driving factor e.g. image processing: degree of correctness

- With learned behavior improvements for safety reasons safety update process changes
- SOTIF (Safety Of Intended Functionality)
  - NEW: updates to improve safety!!
- Leads possibly to “joint goal” of frequent updates due to safety and security improvements



Re-Cap & Future (2)

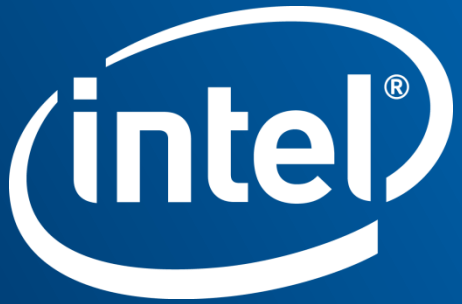


Safety goal: can be diverse for different criticality

- Real-time guarantees or guaranteed supervision
- Guaranteeing availability will be tough research questions e.g. with correctness of design (integrity is much easier)

Hard challenges:

- Balance between guarantees and performance
- Additional services required from computing platform (complexity)
- Virtualization: Hard challenge is guarantee of safety on top of virtualization (w/o hardware knowledge)
- Long-term guarantees of dependability: 10 to 15 years or more
- Automated safety approaches (automated verification and validation approaches)


intel®