EMC² workshop @Artemis Technology Conference 2016

October 6th, 2016

1. Morning session

9:00-9.30 Overview of EMC² project – achievements and challenges (Werner Weber)

9.30-10:00 System Architectures (Andreas Eckel)

10:00-10.30 Multi-core Hardware Architectures and Concepts (Rolf Meyer replacing Rainer Buchty)

10.30-11:00 Coffee break

11:00-11.30 Executable Application Models and Design Tools for Mixed-Critical, Multi-Core Embedded Systems (Albert Cohen)

11:30-12.00 EMC² towards standardization (Erwin Schoitsch)

EMC² workshop @Artemis Technology Conference 2016 cont'd

2. Afternoon session

13.30-14:00 Demo by Volvo (Thomas Söderqvist and Atul Yadav)

14.00-14:30 Real Time Concept Prototyping and Validation of Functional Safety Certification Test Measures by FPGA platform (Zuhal Clarke)

14:30-14.50 Carrier Under Carrier Interference Detector (Javier Valera)

14.50-15.10 Industrial manufacturing and logistics (Juha Kuusela)

15:10-15.30 Internet of Things (Yudani Riobo)

ARTEMIS 2013 AIPP5

EMC²

A Platform Project on Embedded Microcontrollers in Applications of Mobility, Industry and the Internet of Things

Werner Weber Infineon Technologies AG
Werner.Weber@infineon.com
+49 89 234 48470

… in cooperation with Alfred Hoess, Jan van Deventer, Frank Oppenheimer, Rolf Ernst, Adam Kostrzewa, Philippe Doré, Thierry Goubier, Haris Isakovic, Norbert Druml, Egon Wuchner, Daniel Schneider, Erwin Schoitsch, Eric Armengaud, Thomas Söderqvist, Massimo Traversone, Sascha Uhrig, Elena Terradillos, Manuel Sanchez, Javier Valera Juan Carlos Pérez-Cortés, Sergio Saez, Benito Caracuel, Jose Luis Gutiérrez, Juha Kuusela, Mark van Helvoort, Xing Cai, Bjørn Nordmoen, Geir Yngve Paulsen, Hans Petter Dahle, Michael Geissel, Jürgen Salecker, and Peter Tummeltshammer

Project Overview
Numbers

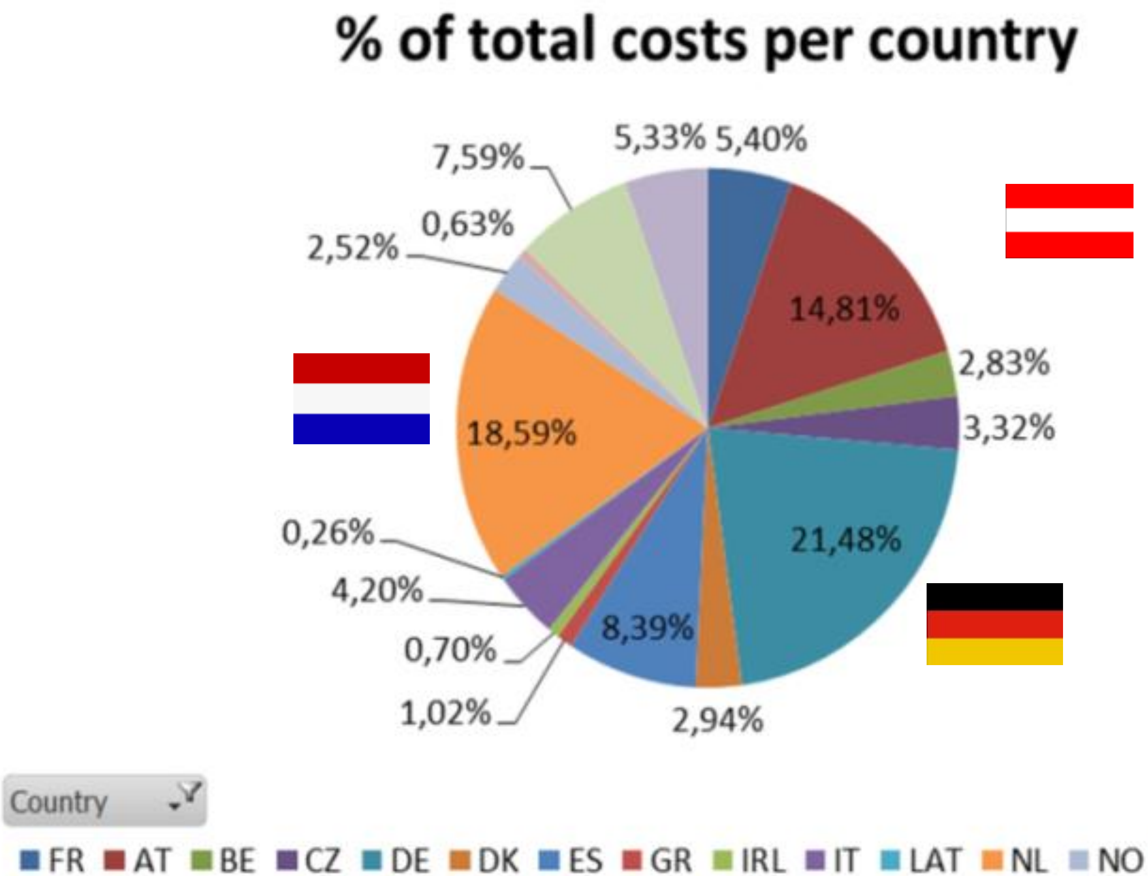
Embedded Multi-core Systems for Mixed-Criticality Applications in Dynamic and Changeable Real-Time Environments – EMC²

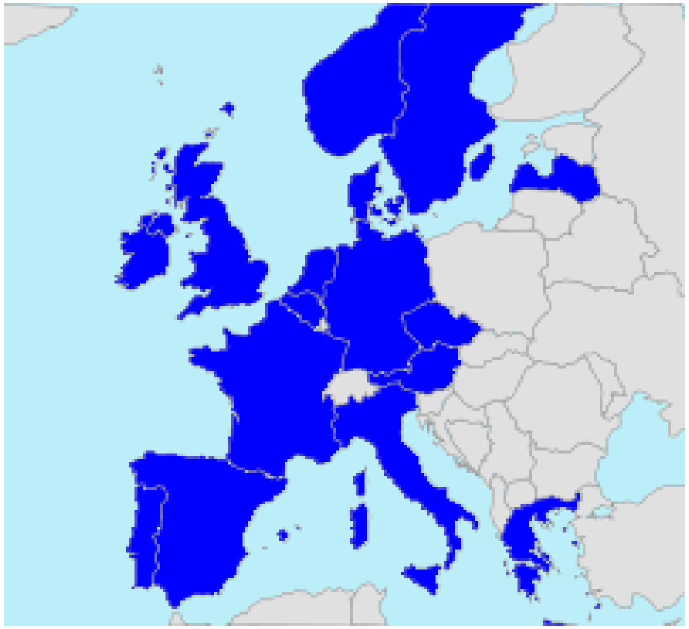
(Artemis Innovation Pilot Project (AIPP)

➢ AIPP 5:	Computing Platforms for Embedded Systems
➢ Budget:	93.9 M€
➢ Funding:	15.7 M€ EU funding (Artemis)
	26.7 M€ National funding
➢ Resources:	9636 person months (803 person years)
➢ Consortium:	101 Partners (plus 1 associate partner)
➢ From:	16 EU Countries

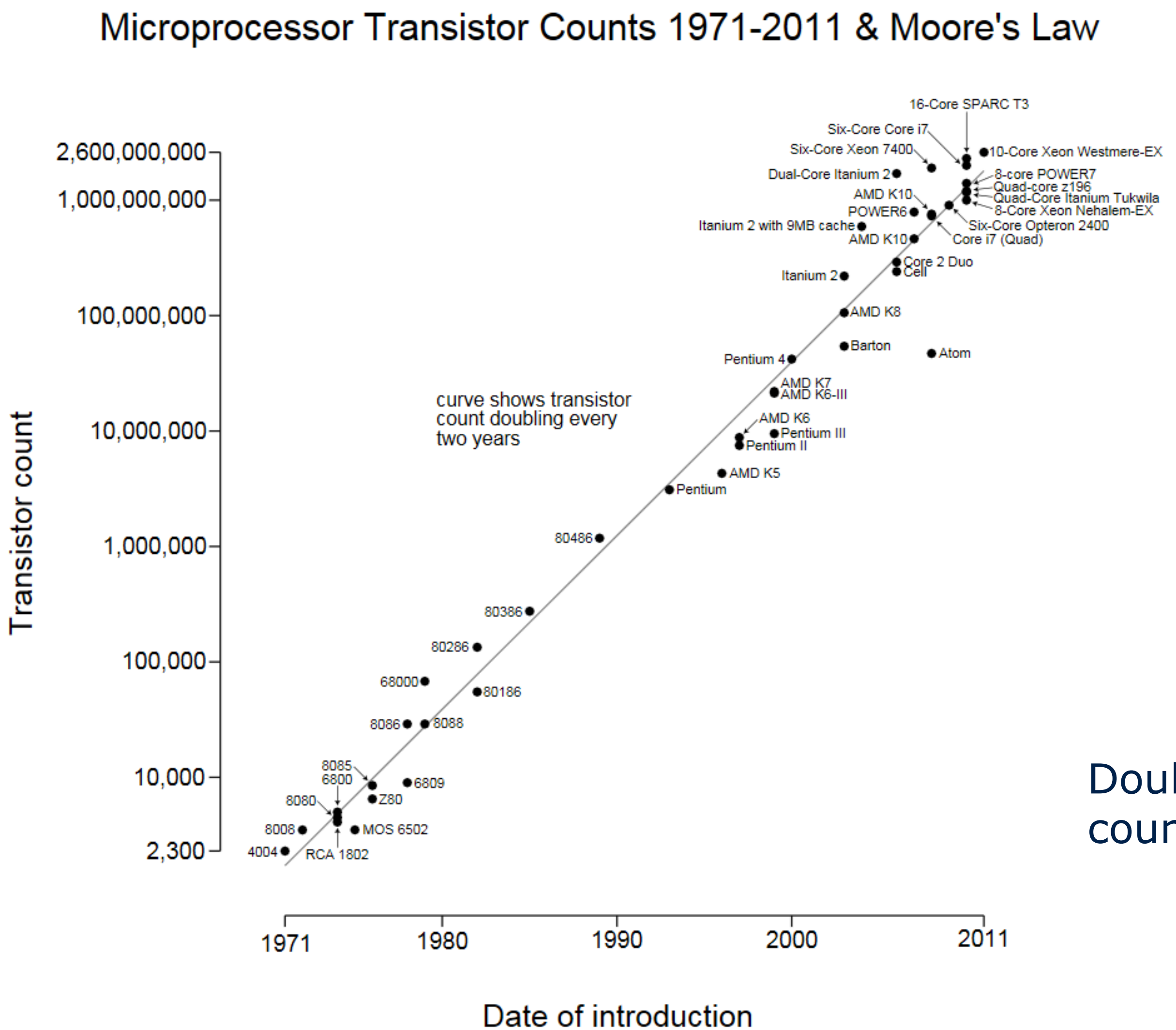
→ Largest ARTEMIS-JU project ever!
most relevant EU players on board

Project Overview
European Dimension





Technological Productivity

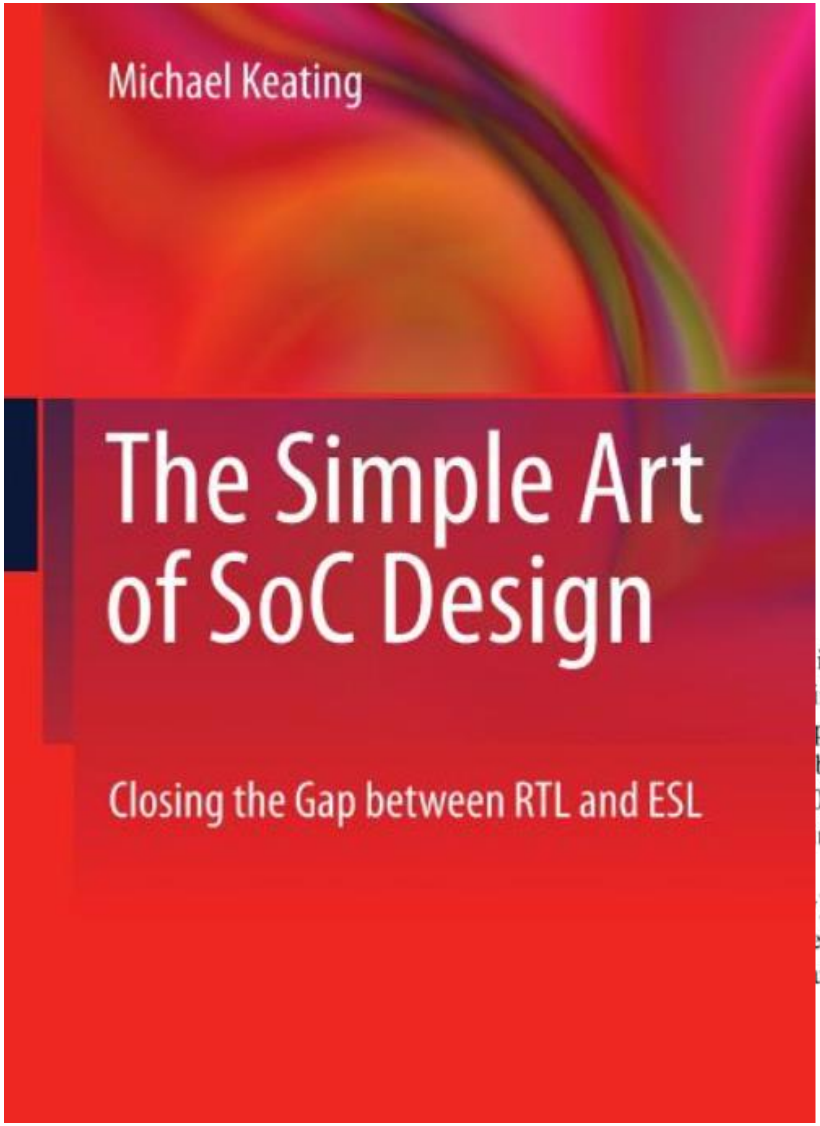


Doubling of transistor count every two years

Software Productivity
Michael Keating, Synopsis Fellow

„We live in a world where function (hardware and software) is described in code. But code does not scale. **Individual coders cannot code more lines of code than they could decades ago.** And as the projects get bigger, productivity actually decreases: One engineer can code about 10,000 lines of (debugged, production-ready) code in a year. But 100 Engineers cannot code 1,000,000 lines of (debugged, production-ready) code in a year“…

„The functionality of the 30 lines of code (per engineer-day) has increased very slowly over time, and most of this increase has been due to the use of libraries and IP. The introduction of C++ in 1983 also gave software productivity a small, one time gain.“



Motivation for EMC2

- Very fast technological advances of µ-electronics in past decades
- Amazing capabilities at lowered cost levels
- Today primarily exploited in consumer-oriented products
- Systems quickly put together since the next technology generation is already waiting around the corner
- Errors may be tolerated and a new execution attempt started
- This (and similar) way(s) of handling errors acceptable for consumer products

Application innovation

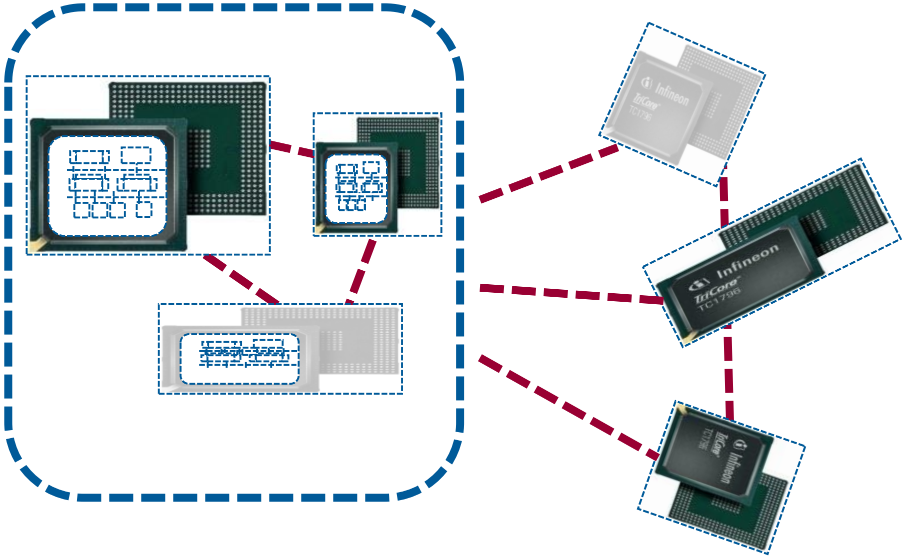
- In professional areas the consumer approach is not feasible: **Automotive, Avionics, Space, Industry, Health care, Infrastructure**

- Need much higher level of operational reliability
- Higher HW/SW complexity
- Have to fulfill real-time safety requirements

- Dynamic reconfiguration during runtime
- Prime task of EMC² to bring two worlds together
  - Consumer world: use of advanced µC systems
  - Professional world: reliability, complexity, real-time

Technological innovation

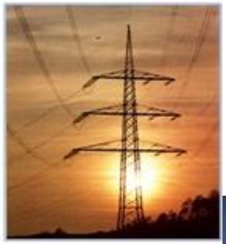
- Mixed Criticality
  - Handle applications with different priorities
- Dynamic Re-configuration
  - Full range of dynamic changes on application level
- Hardware Complexity
  - Variable number of control units at runtime



Application innovation

- EMC² - Embedded Multi-core Systems for Mixed-Criticality Applications in Dynamic and Changeable Real-Time Environments
- Applications: Automotive, Avionics, Space, Industry, Health care; Infrastructure
- Improve performance, lower cost
- Improve energy efficiency







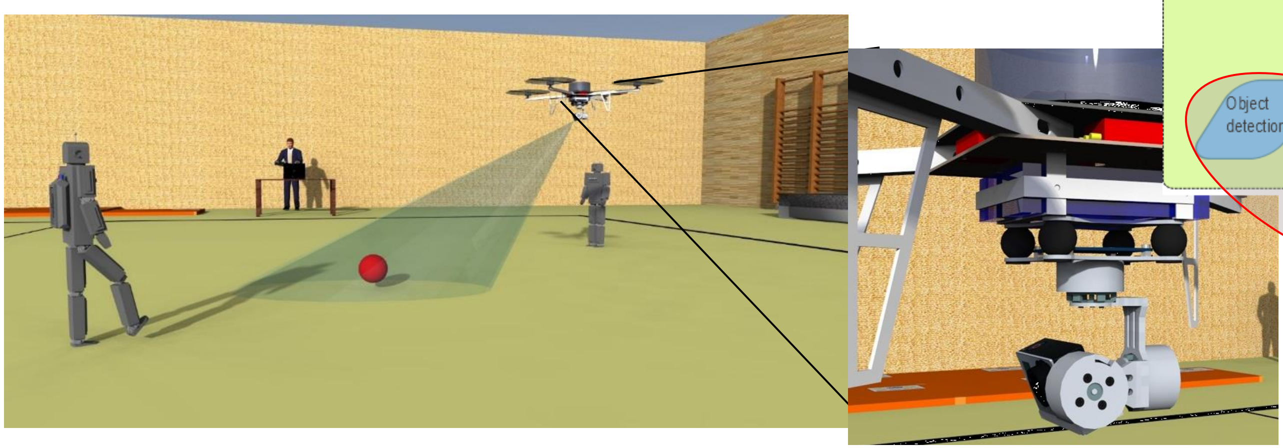


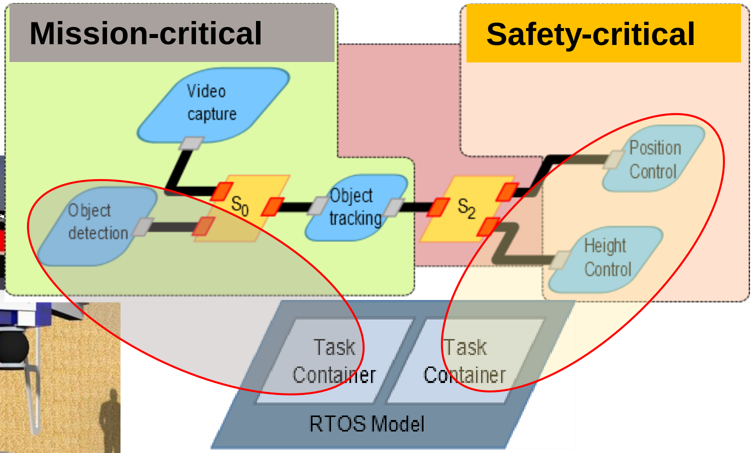

Model based Design for MC Systems

Goal: Safe optimization of QoS in Mixed-Critical Applications

Use-case Avionic Control and Payload Platform for Multi-Rotor Systems

- Safety critical System
  - 3 parallel Flight Control Tasks (2 ms)
  - 6 Sensor Channels (2-30ms)
  - 3 Sensor Compute Tasks (2 ms)
  - Small violations accumulate to crash
- High Throughput Video application
  - Mission critical object detection
  - Minimal 6 frames/second
  - Demand for high data throughput

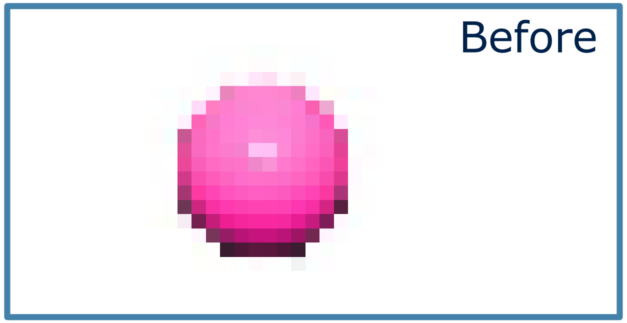


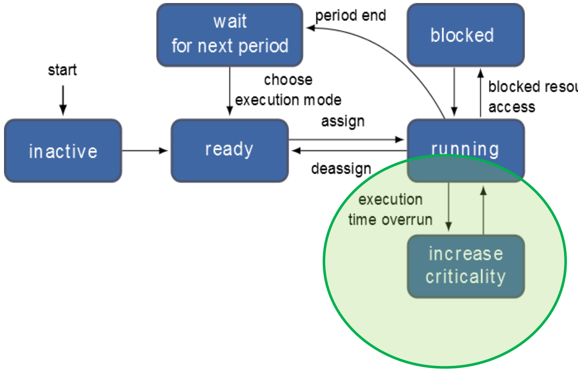


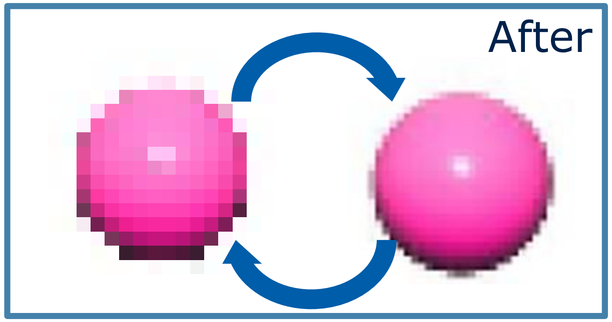
Optimized QoS in Mixed-Critical Applications with Dynamic Criticalities

Static schedule (WCET based)

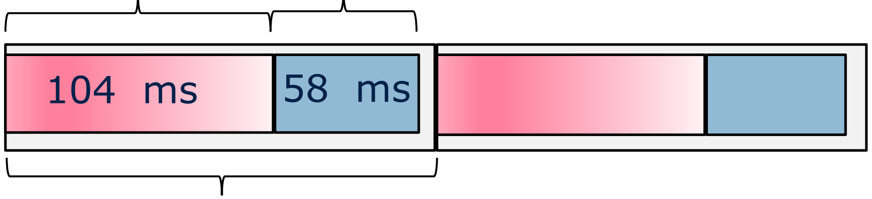
Dynamic Criticality Modes





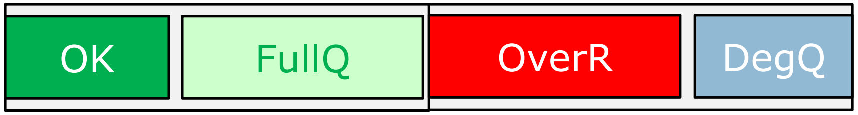


Flight CS Video Proc: 300x200 px.



167 ms frame

95% (typical case) Flight CS: 64 ms
Leaves: 103 ms or 460x320 px.



Criticality Policy	# Degraded	# Full Quality	Av. Throughput
Static	30	0	1055 Kib/sec
Dynamic	13 (±3)	17(±3)	1923 Kib/sec (182%)

Dynamic runtime environments and services

- Various advanced mechanisms (~80) enabling
  - handling high congestion in networked systems
  - e.g. in cooperative intelligent transportation system (ITS) with wireless sensor networks
- mixing different criticality domains in networks for performance and high integration
  - automotive Ethernet networks
  - networks-on-chip

dynamic network reconfiguration

Dynamic runtime environments and services

- Isolation between safety critical parts and user domain
  - Improving security
  - Inter-core communication
  - Reduced Power Consumption

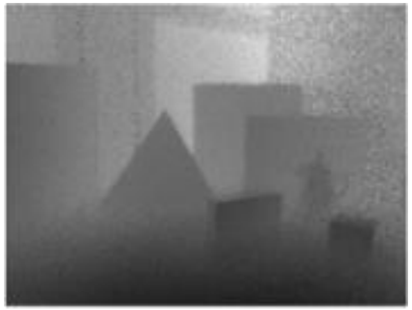
Advanced Virtualization and Isolation Mechanisms

Source: Lars Völker, BMW AG

HW Architectures & Concepts Time-of-Flight 3D Imaging

- Objective: Exploration of novel Time-of-Flight (ToF) 3D imaging concepts targeting multi-cores and mixed-criticality
- Key achievements
  - ToF / RGB sensor fusion
    - First time high-performance sensor fusion solution for embedded systems achieved
    - Upscaled resolution, increased sharpness, less noise, less motion artifacts, high Frames per second
  - HW-accel. ToF processing
    - Novel Zynq-based system solution for mixed-critical app.

ToF 3D camera

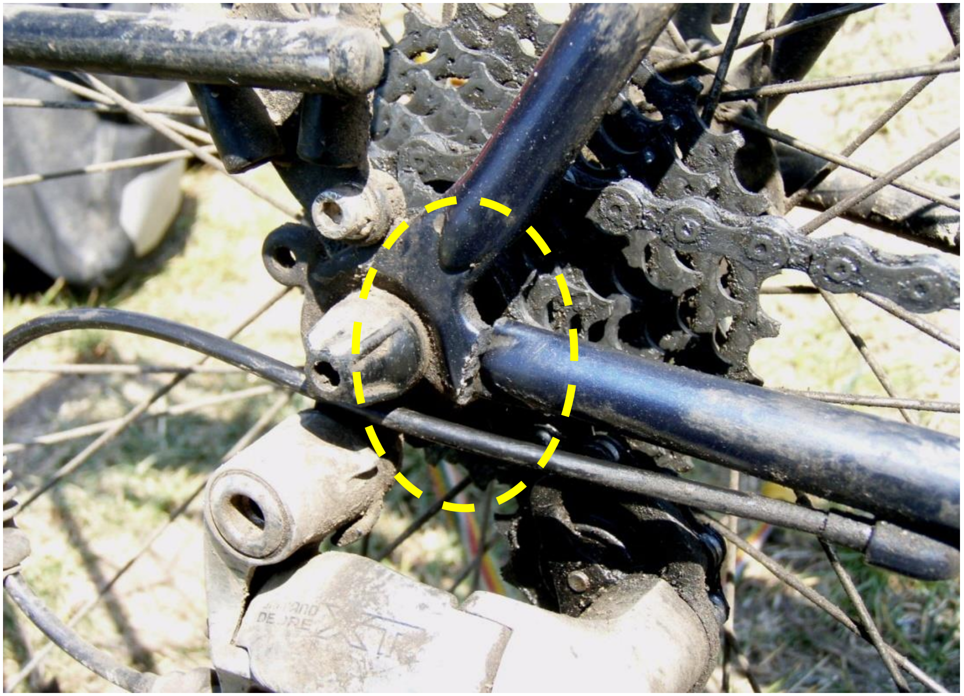


Low-res. ToF image

High-res. RGB image

ToF/RGB fused 3D image

Digitalization of Software Engineering, Why?



BUG

PATCH

Digitalization of Software Engineering, Why?

BUGFIXING with high CRITICALITY, because the “Bugfix” destroyed the derailer

BUG FIXED at least to some degree

Digitalization of Software Engineering, Why?

The result of the bugfixing was:

One problem is fixed, another one is created.

This looks like software engineering…

[Herman Veldhuizen, during its way from Norway to Tibet]

Source:
http://www.hermanveldhuizen.com/wp/?p=141

Digitalization of Software Engineering

Context

- Software development with a high focus on time-to-market
- Time is therefore critical and the test-team is always overloaded

Problem

- How to verify 258 bug fixes provided with the last software version?
- There is not enough time to re-verify all of them.
- Which bug fixes could be ignored safely?

Solution

- Use the big-data approach and calculate a ***criticality-factor*** for every bug fix which reflects the complexity of every bug fix.
- The higher the ***criticality-factor*** the higher is the probability that a new bug might have been introduced.
- Bug fixes with relatively low ***criticality-factors*** could be ignored, i.e. they do not need to be re-tested by the test team.

Details of the Solution

1. Analyze the complete Software data base under development, by:
   1. Count the number of **change operations** (commit/check-in) for every single bug-fix
      <**c**>
   2. Count the number of **different developers** involved
      <**n**>
   3. Count the number of **different software modules** which had been changed
      <**s**>
2. Multiply all this numbers listed above, like:

criticality-factor = c * n * s

Recursively through the complete software structure, which could be several million lines of code

Qualification and Certification

- **Cyber Physical Systems**
  - Important trend across application domains
  - Huge potential for new applications and business

- But there are significant challenges ahead that might be show stoppers:
  - Safety
  - Security (for Safety!)
  - Adaptability, dynamic system evolvement and re-configurability over lifetime

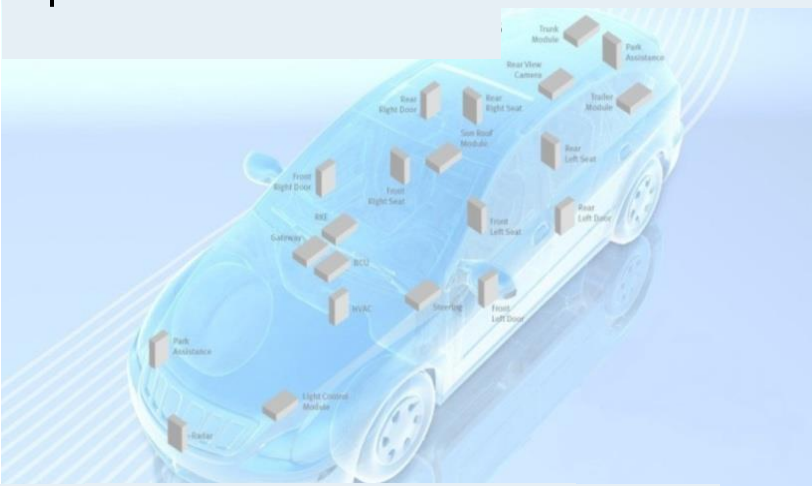
Cf. "Umsetzungsempfehlungen Zukunftsprojekt Industrie 4.0"

Application Topics in EMC2

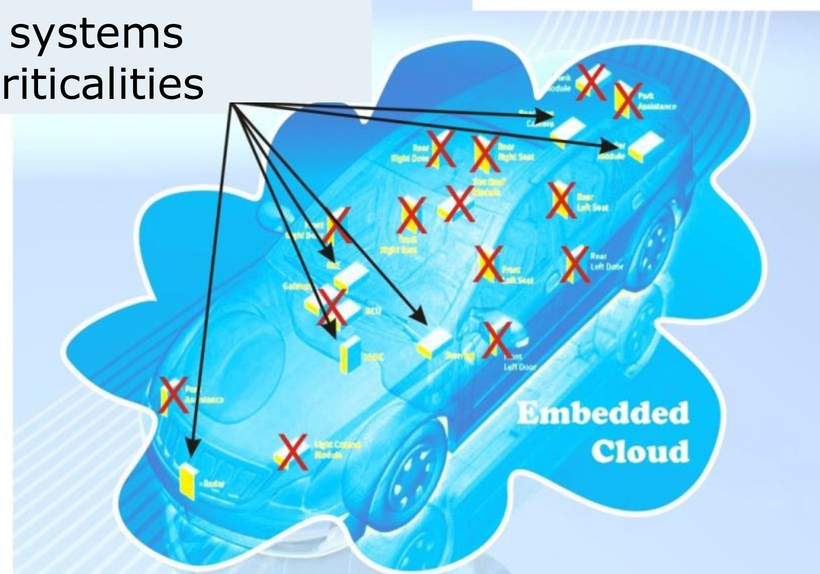
- Automotive
- Avionics
- Space
- Industrial manufacturing
- Logistics
- IT-infrastructure (‘Internet of Things’)
- Healthcare
- Railway
- Seismic surveying

Reduce Number of Control Units
Save cost and increase performance

Many heterogeneous single-core systems,
specialized for the individual criticality levels



Multi-core systems
for mixed criticalities



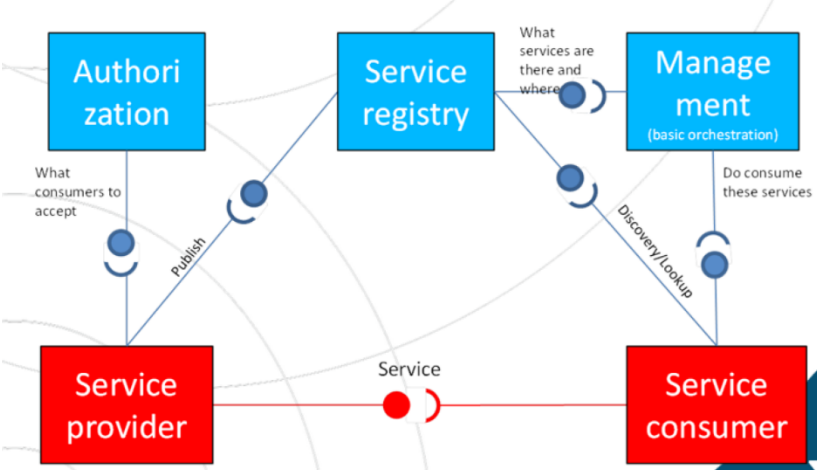
Vision

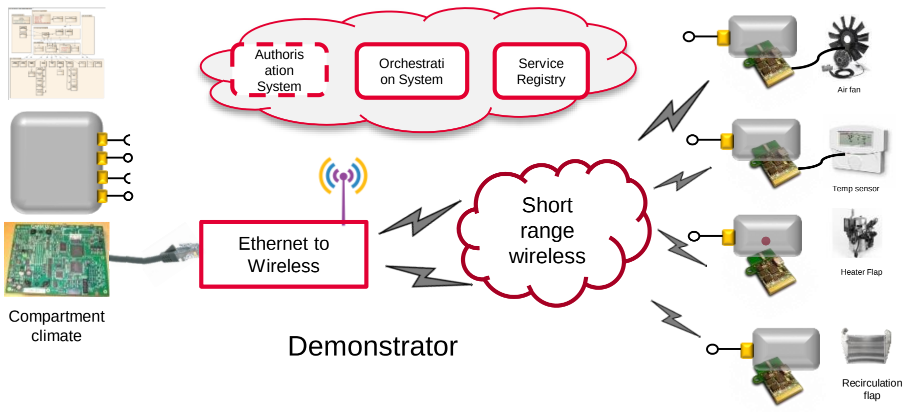
Aggregate resources
In multi/many cores,
ECU networks

Offer system proper-
ties as services and
not as independent
systems

Service-oriented Architecture for embedded truck architecture

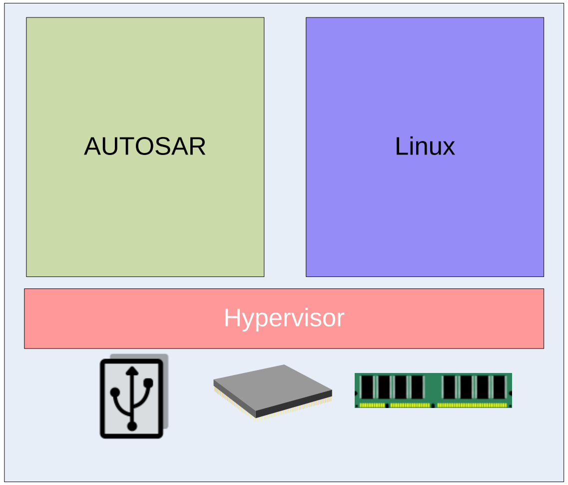
- **Objective**: SoA for embedded truck architecture
  - Vehicle as a service in larger application domain, or Multi service provider: each potential in-vehicle software element as a service
  - Functionalities in form of services orchestrated at design/runtime
  - Resource aware services for realtime systems



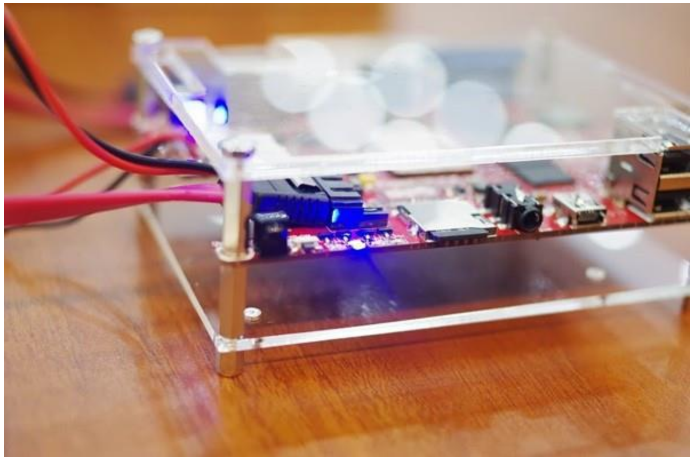


Multiple OS support and virtualization

- **Objective**: Multiple OS support and virtualization
  - merge multiple ECUs and reduce hardware costs
  - separate software components of different criticality, e.g. safety or security
  - better utilization of hardware resources



- **Key achievement**
  - Prototype of Embedded Linux and AUTOSAR running within the same MCU using the open source Xen Hypervisor (HW Dual Core ARM A7)



Multicore Processors for Avionics

- **Objective:**

Since certification authorities have concerns on mixed-critical applications on MCPs, the goal is to implement a safety net for the MCP, mitigating unforeseen or undesirable MCP operation *using SW Hypervisor* that monitor the MCP. Based on this information the monitor will decide if the MCP operates in normal conditions. In case of abnormal behaviour of the MCP the monitor can warn the pilot that the system is no longer operational

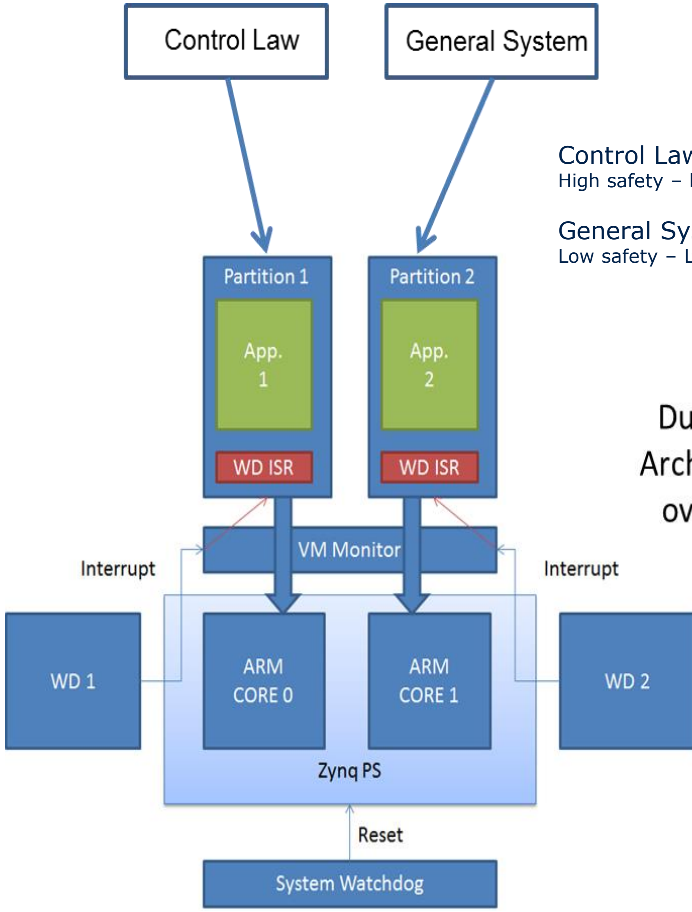
- **Results:**

No functional interference between partitions;

Bounded temporal interference (<4%) of one faulty partition on the other fault-free one.

Good separation thanks to HW watchdog + virtualization.

Hardware features to guarantee bounded quality of services are needed.



Control Law :
High safety – high iteration rate

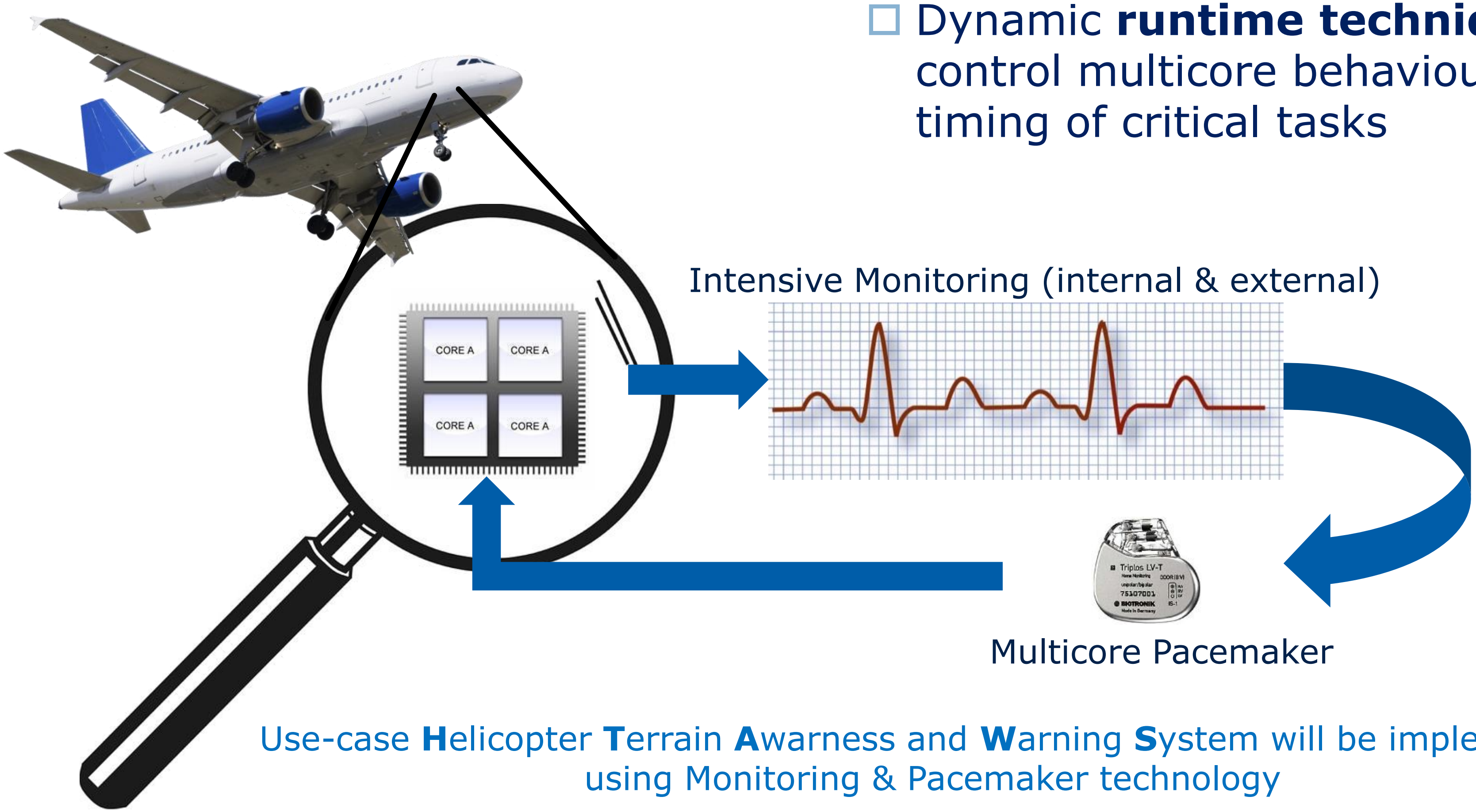
General System :
Low safety – Low iteration rate

Dual Core Architecture overview

Avionics

- **Objective:** Enable Multicores for use in safety critical avionics applications

- Key achievements
  - **External & Internal Monitoring** of **MC Activities**
  - Dynamic **runtime techniques** to control multicore behaviour and timing of critical tasks

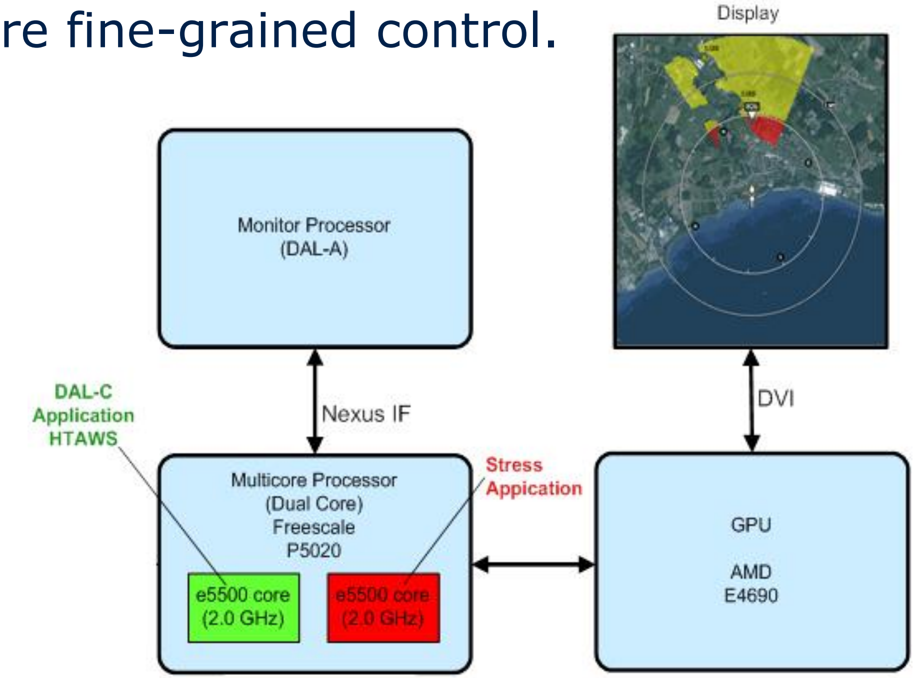


Use-case **H**elicopter **T**errain **A**warness and **W**arning **S**ystem will be implemented using Monitoring & Pacemaker technology

Avionics

- Two **Monitoring & Pacemaker** approaches implemented:
  - **External monitoring and control** by a separate hardware for highly critical avionics systems. The external hardware monitors the main multicore system and provides extra functionality in case of a complete failure of the main multicore.
  - **Internal monitoring and control** by enhanced system software support. The internal approach saves the extra hardware (costs, weight and space) and allows more fine-grained control.

Both approaches show advantages and disadvantages which are evaluated

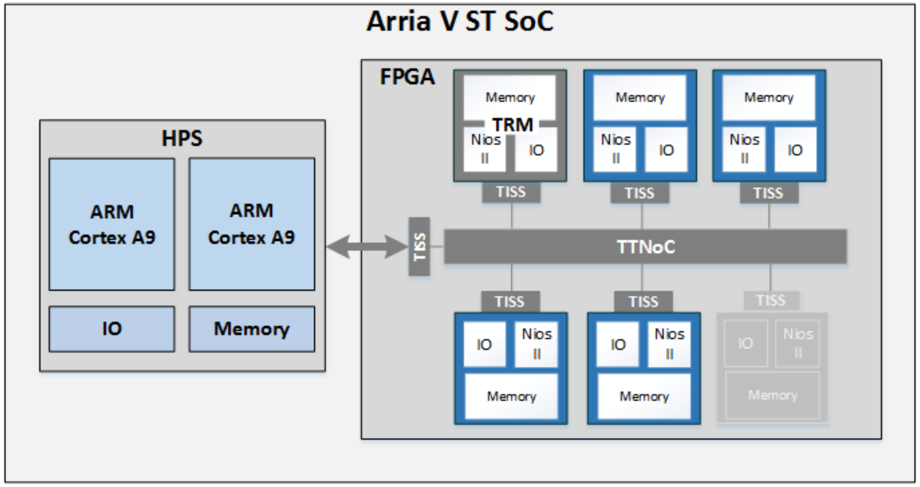


WP8

MPSoC Hardware for Space

- **Objective:** Heterogeneous time-triggered architecture implemented on a hybrid MPSoC platform

- **Key achievements:** Architecture hardware definition
  - Time-triggered Network-On-Chip
  - Trusted Interface Subsystem
  - Trusted Resource Manager



Payload Applications for Space

■**Objective:** MPSoC image processor based on CCSDS 122 & 352 standards

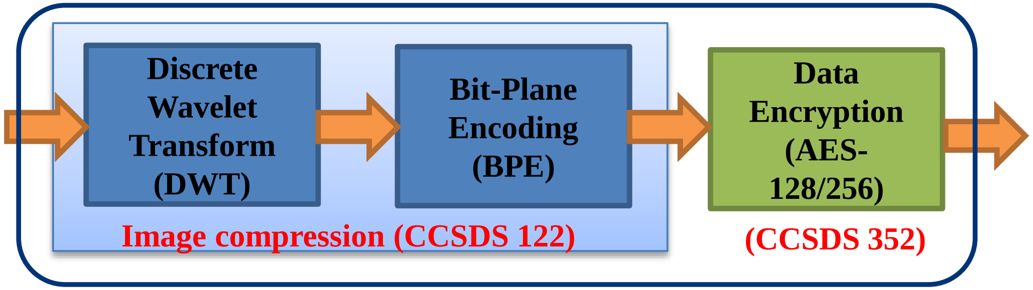
■**Key achievements:**

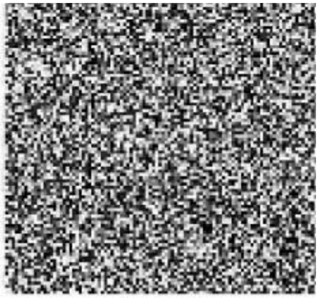
Implementation of CCSDS 352 standard for data encryption based on OpenMP paradigm

Partial Implementation of CCSDS 122 standard for image compression based on OpenMP paradigm

Discrete Wavelet Transform

Validation on a Multicore architecture (ARM Quad-Core)





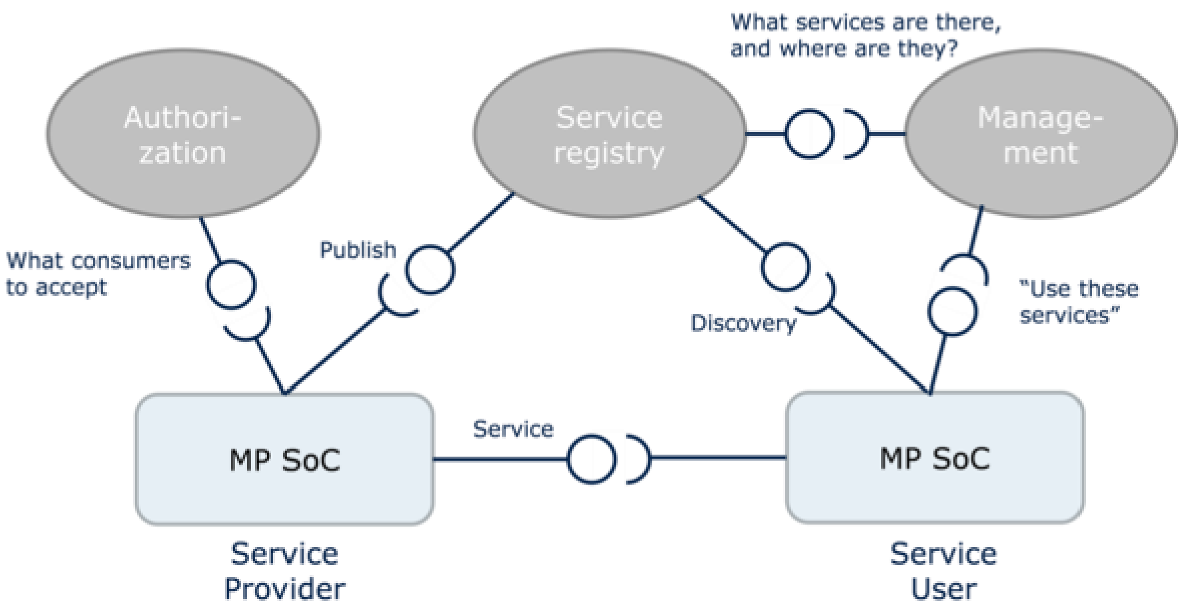
Quality Control by 3D Inspection

■**Objective**:
Comparison between sequential and parallel models for a task of 3D object reconstruction. Object reconstruction used to distinguish different objects and to find surface defects based on texture comparison.

■**Key achievements:**
Increased overall inspection performance by 300%: With OpenMP parallelization and an execution platform composed of 2 processors, 16 cores and multithreading capabilities a reduction of computation time from 24.563 milliseconds to 7.996 milliseconds is achieved by exploiting coarse parallelism and thus decreasing latency.

Service Oriented Architecture on Multicore Embedded Systems

- **Objective:**
  - Design and implement an accurate, fault tolerance and reliable timing system distribution for usecase ‘Synchronized low-latency deterministic networks ‘
- **Key achievements**
  - Synchronization accuracy ***<1ns*** based on enhanced PTP protocol.
  - Ethernet traffic with low latency and high determinism



Synchronized low-latency deterministic networks

Key achievements

- Timing Scalability (Fig. A)

Signal propagation jitter is always under *250ps* in all nodes

- Dependability features (Fig. B)

Low-cost redundant implementation

Single point of failure avoidance

Able to recover from a failure with ~zero-recovery time.

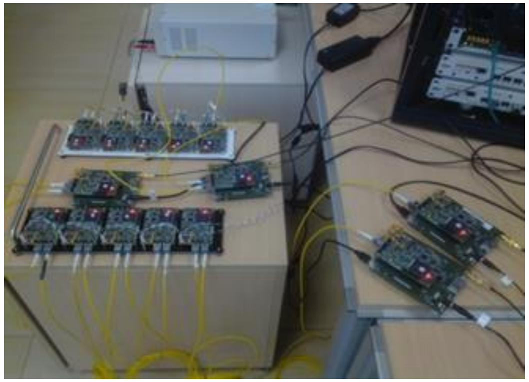


Fig A: Timing scalability jitter and QoS test using a daisy-chain setup with White Rabbit

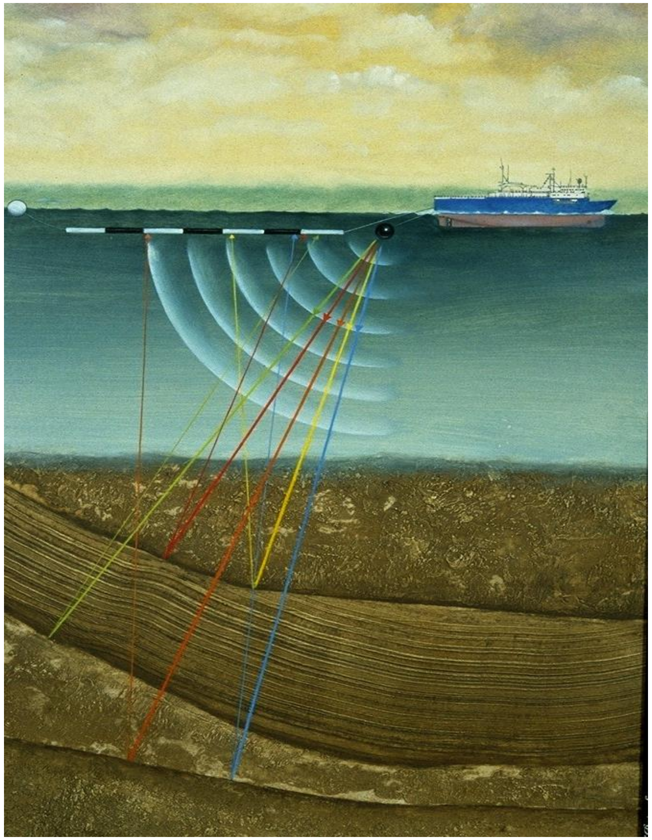
Fig B: Heterogeneous Time distribution with White-Rabbit and IRIG-B in a daisy-chain setup and a redundant HSR ring

Seismic processing

Purpose: Produce images of geological features and their structure below the surface of the earth

On sea:

- Networked computers
  - In the streamers > 2 000 computers
  - Onboard the ship > 200 computers
- Compute power > 2 Tflops
- Number of sensors > 200 000
- Huge Data rate 1-3 Gbit/s
- Disk capacity > 100 Tbytes



Seismic processing

Real-time processing on sea:

300 Mbit/sec per streamer

Seismic volume

Real-time signal processing

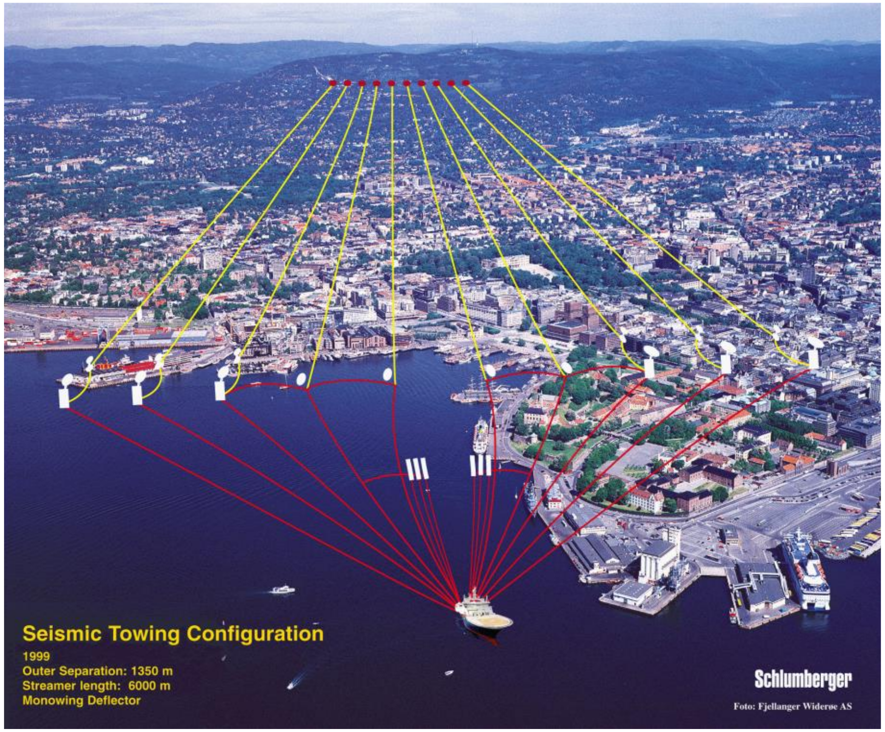
300 Mbit/sec per streamer

On ship:

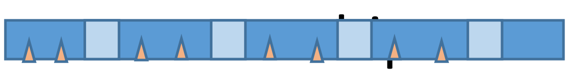
Further seismic processing

On land:

Further seismic processing



200 computers with 4 000 cores



8-14 streamers behind ship

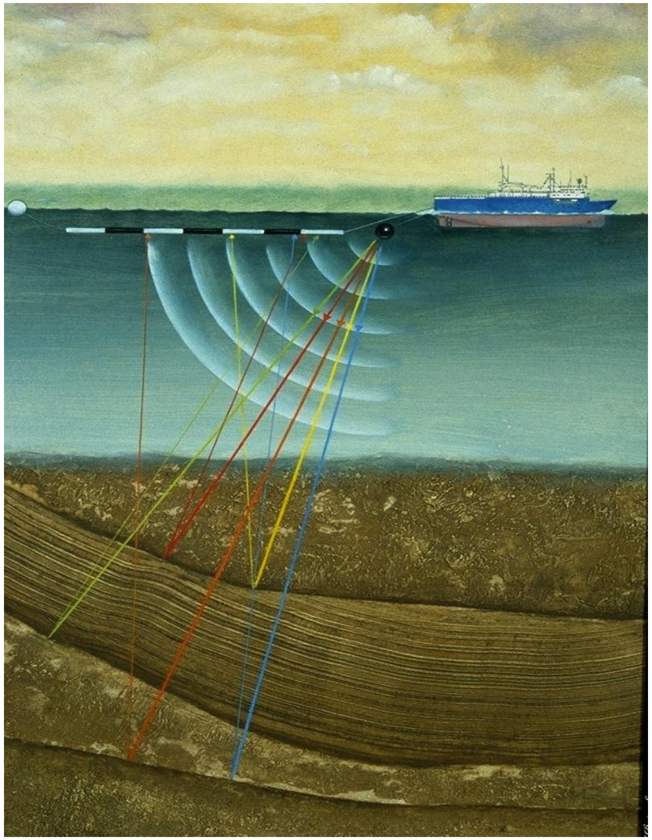
Streamer length 10km - 14 km

100 - 200 computers per streamer

200 000 sensors per streamer

Potential impacts on Seismic Processing at sea and on land

- **Reduced engineering time**: New algorithms exploiting multi-cores can be implemented much faster.
- **Reduced execution time**: Reduced execution time translates into **reduced costs** for seismic processing.
- **Achievement 2016 Q1**: For the first prototype, the generated C++ code runs **2-4 as fast** as the MATLAB code.



[simula . research laboratory]

WesternGeco

FORNEBU CONSULTING A PART OF DEVOTEAM

Video surveillance for critical infrastructure

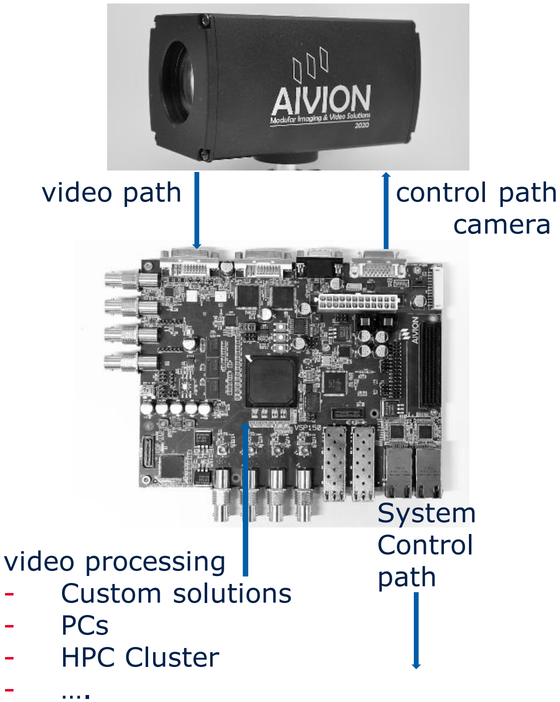
- **Video applications** are entering into more and more markets such as
  - Surveillance
  - Medical applications
  - Automated driving
  - Quality control in production
  - Automatic access control

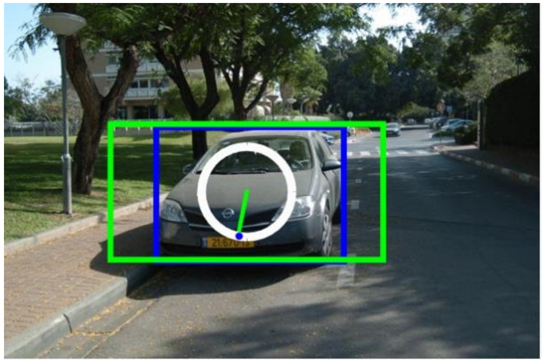
Objective: Acceleration of an object (face) detection algorithm by using multi-core or FPGA architectures.

Achievements: Implemented object/license plate detector in Xilinx Zynq

Experiments with High Dynamic Range detection of license plates

Further experiments with Random Forests for object detection

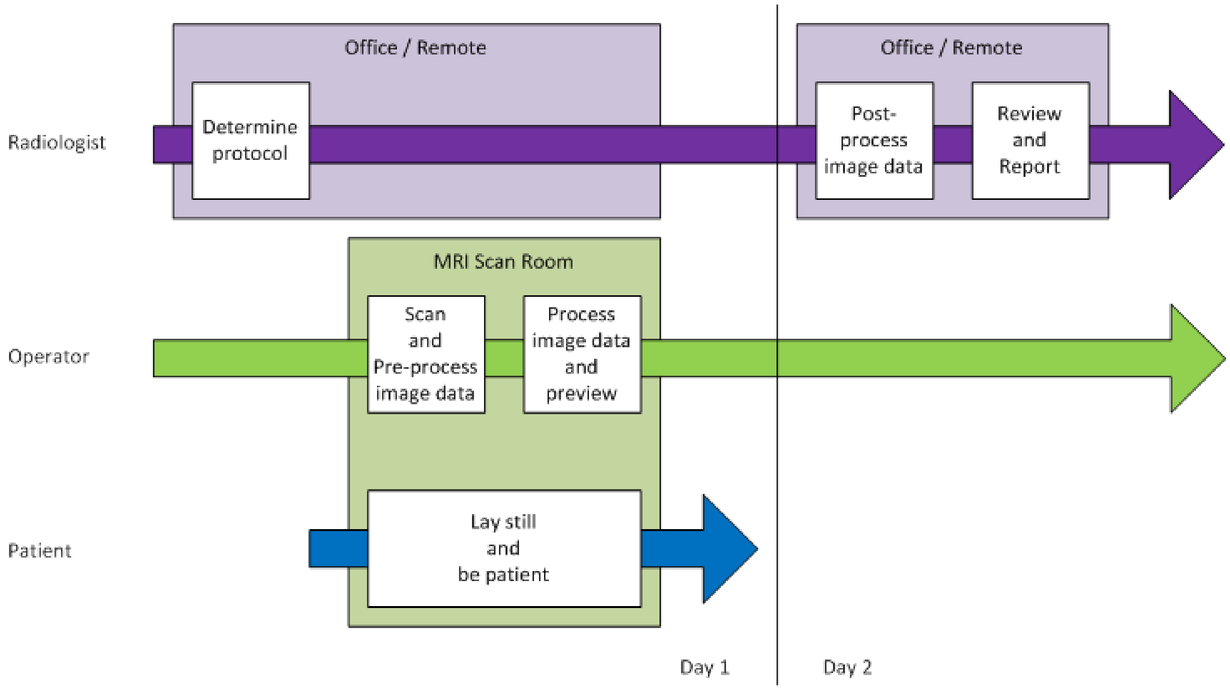




Random forest vehicle detector

Medical imaging

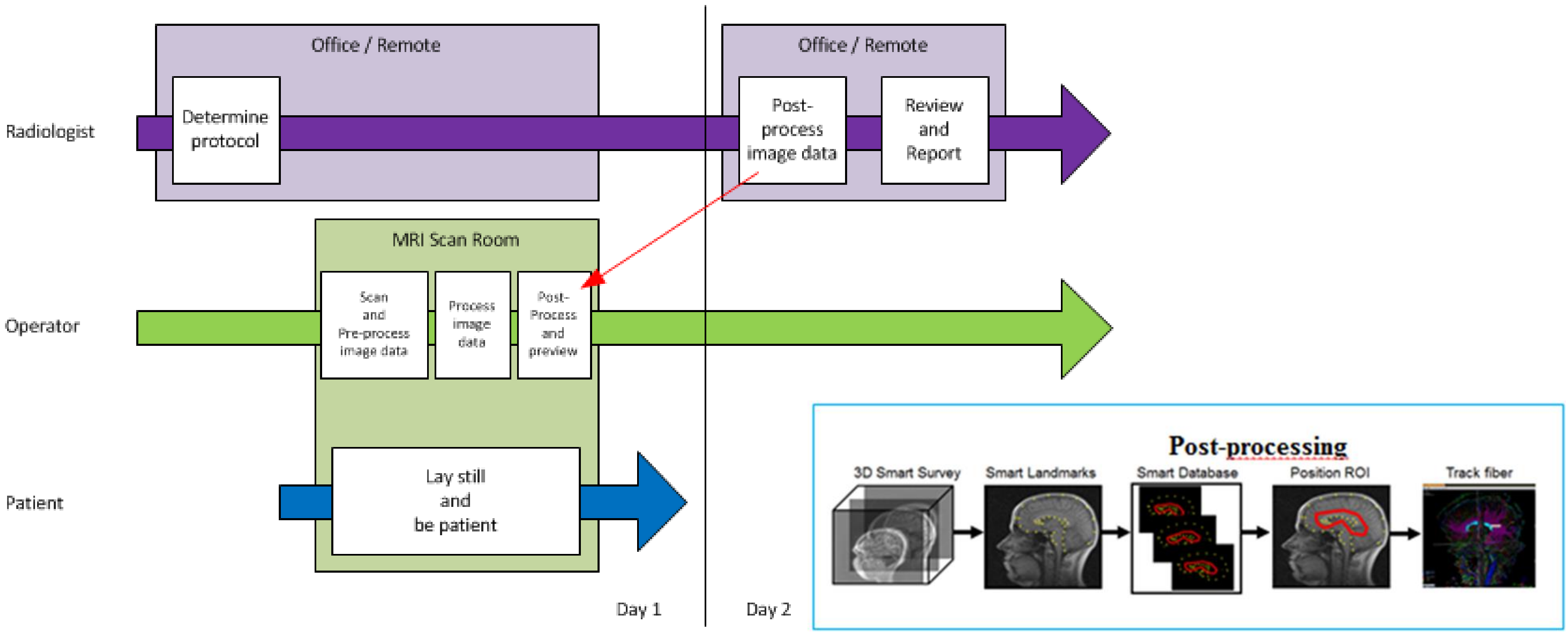
- **Purpose:** Advanced diagnostic MR Imaging
- **Challenge:**
  Prevent patient call back for complex diagnostic procedures



Workflow before EMC² (state-of-the-art)

Medical imaging

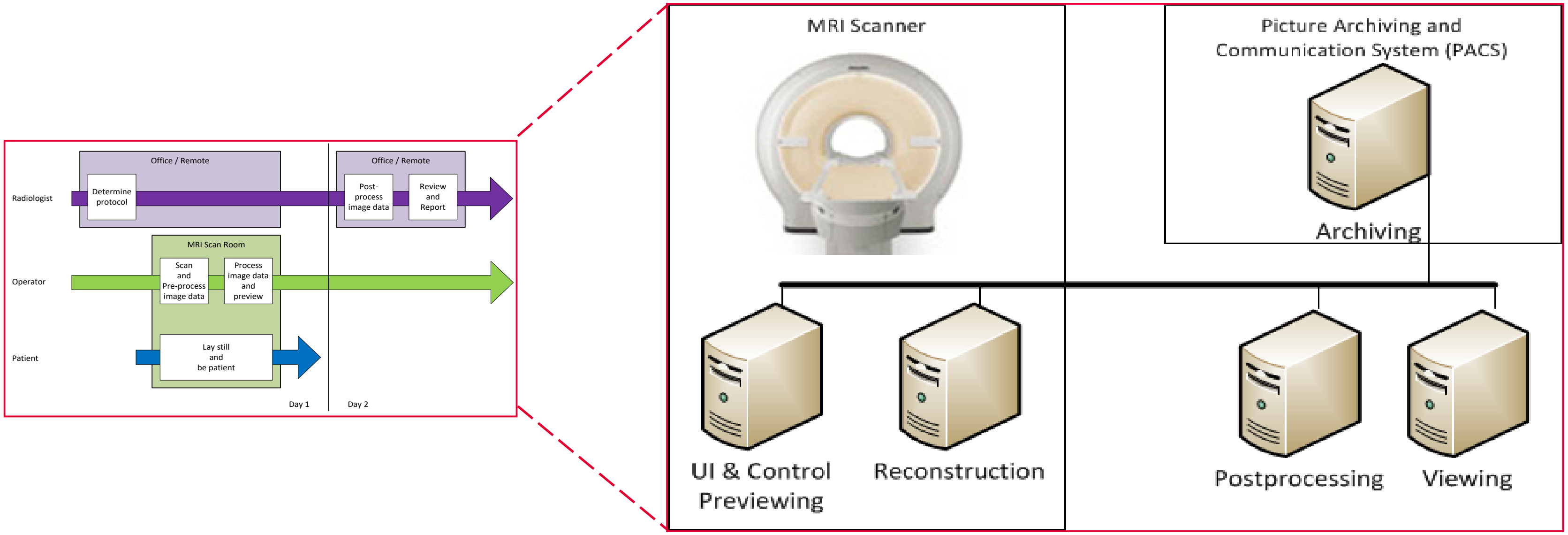
- **Challenge:**
  Prevent patient call back for complex diagnostic procedures



Workflow after EMC² (innovation)

Medical imaging

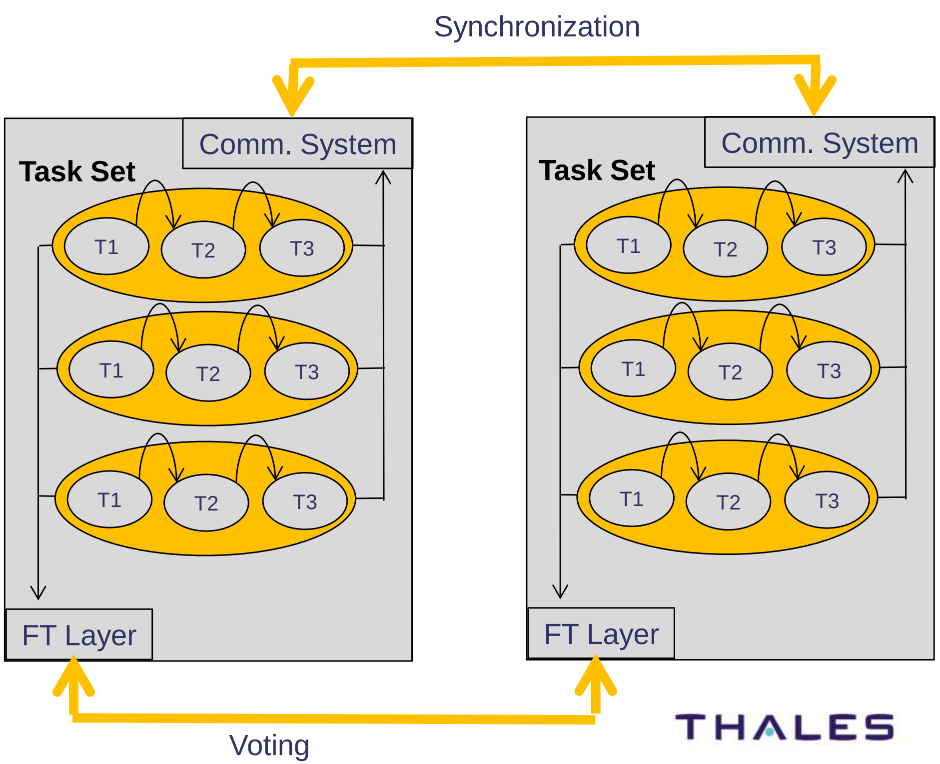
- **Objective:** Go from separate tasks deployed on separate systems to a single system solution



PHILIPS

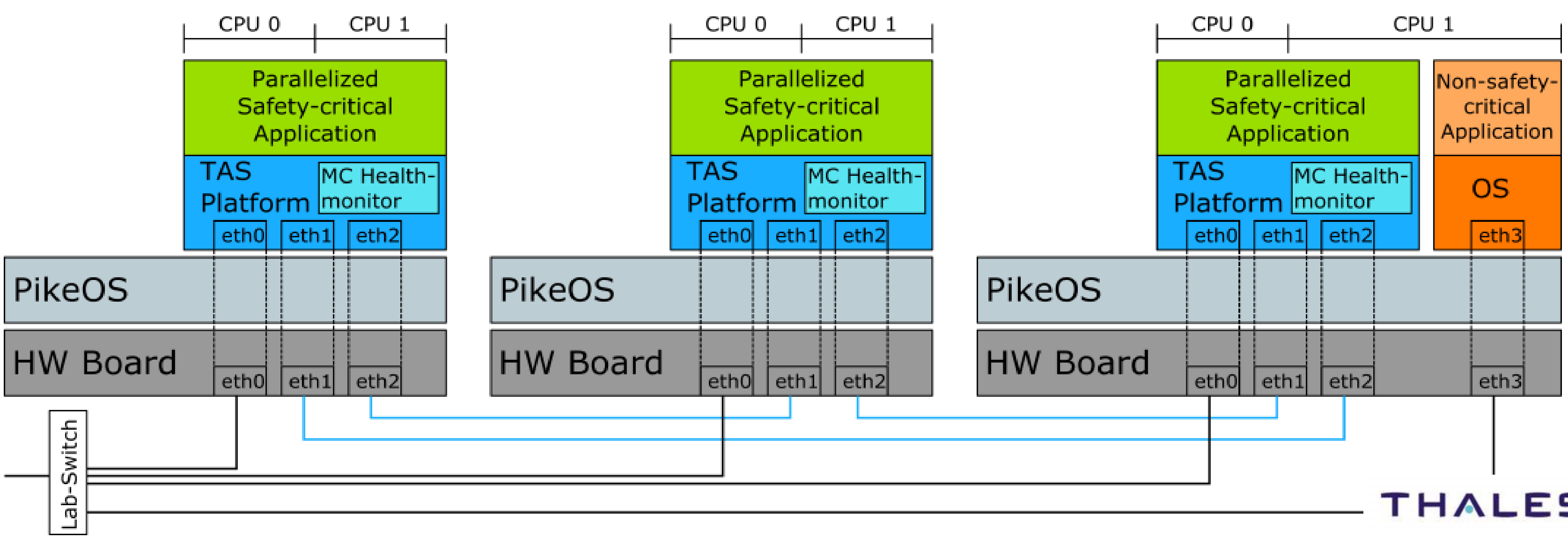
Railway applications

- **Fault tolerant platform, a common base for railway applications (mainline & urban)**
- **<u>Objectives</u>:**
  - Extend programming models to better exploit multicore resources
  - Extend hardware health monitoring for multicore
  - Use TAS Platform in a virtual environment Pike OS Hypervisor



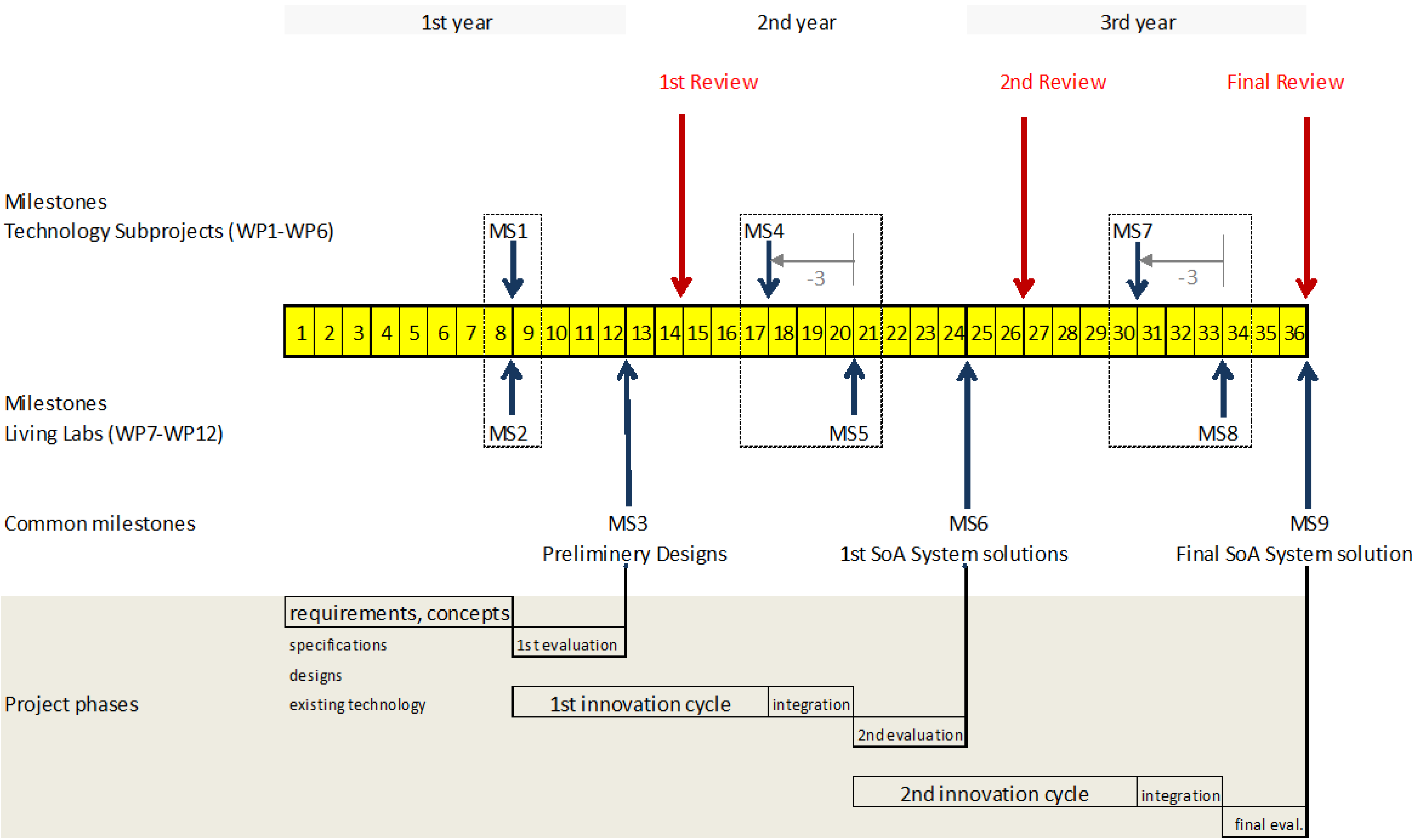
Railway applications

- **Key achievements:**
  - Classification of parallelization techniques for safety-critical applications
  - First implementation for health monitoring (memory testing)
  - First prototype running on virtualized environment (PikeOS)



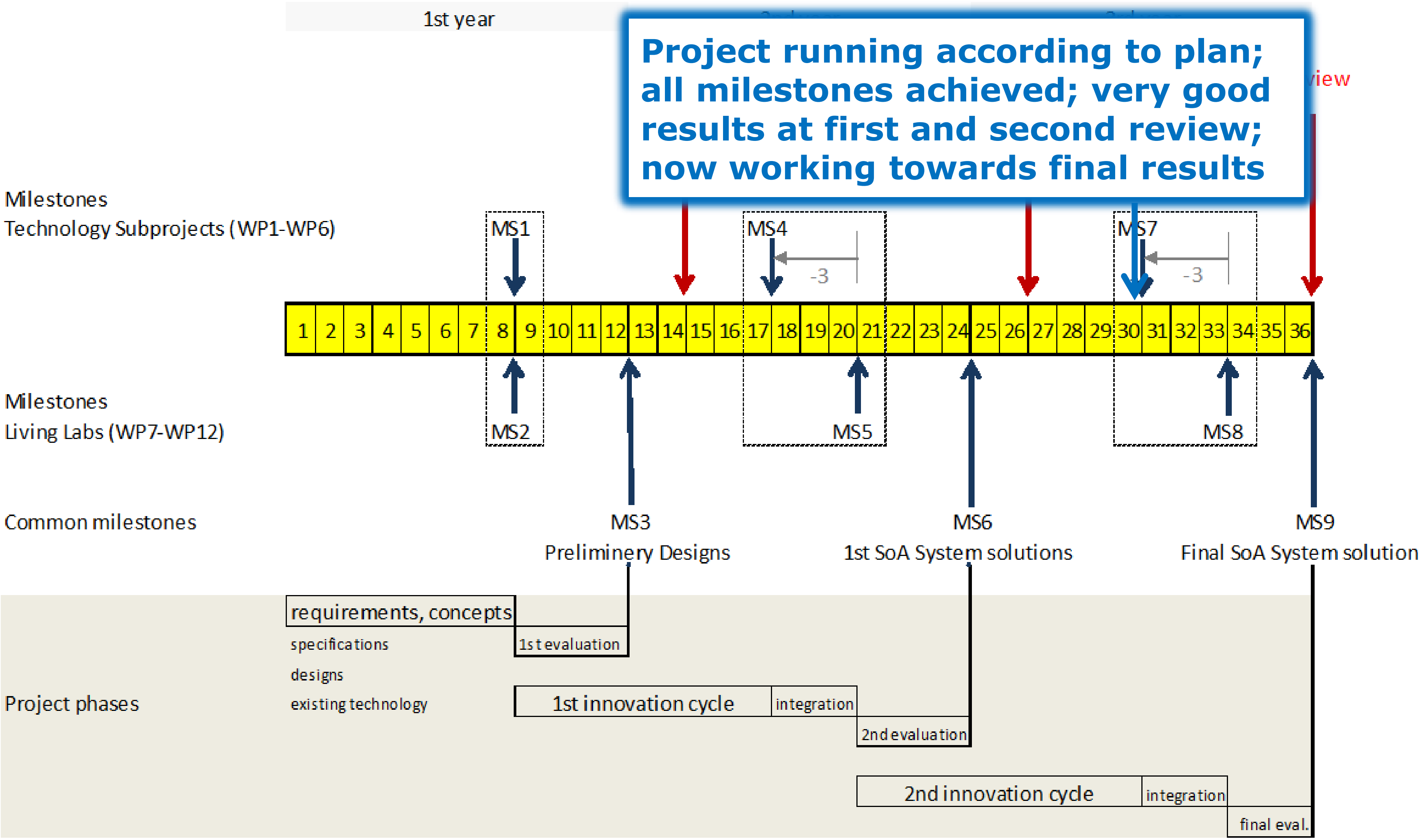
THALES

Work Plan


1st year
2nd year
3rd year
1st Review
2nd Review
Final Review
Milestones
Technology Subprojects (WP1-WP6)
MS1
MS4
MS7
-3
-3
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
Milestones
Living Labs (WP7-WP12)
MS2
MS5
MS8
Common milestones
MS3
Preliminery Designs
MS6
1st SoA System solutions
MS9
Final SoA System solution
Project phases
requirements, concepts
specifications
designs
existing technology
1st evaluation
1st innovation cycle
integration
2nd evaluation
2nd innovation cycle
integration
final eval.

Project monitoring: Milestones MS1, MS2, MS3 achieved

Project running according to plan; all milestones achieved; very good results at first and second review; now working towards final results



Public project website

- First version online at project start: www.emc2-project.eu
- New and significantly extended version online since beginning of July 2014: www.artemis-emc2.eu
- Website is updated whenever news, events and other information for publication becomes available (latest update after finalization of the 2nd EMC² Newsletter)