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, Juan Carlos Pérez-Cortés, Sergio Saez, 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

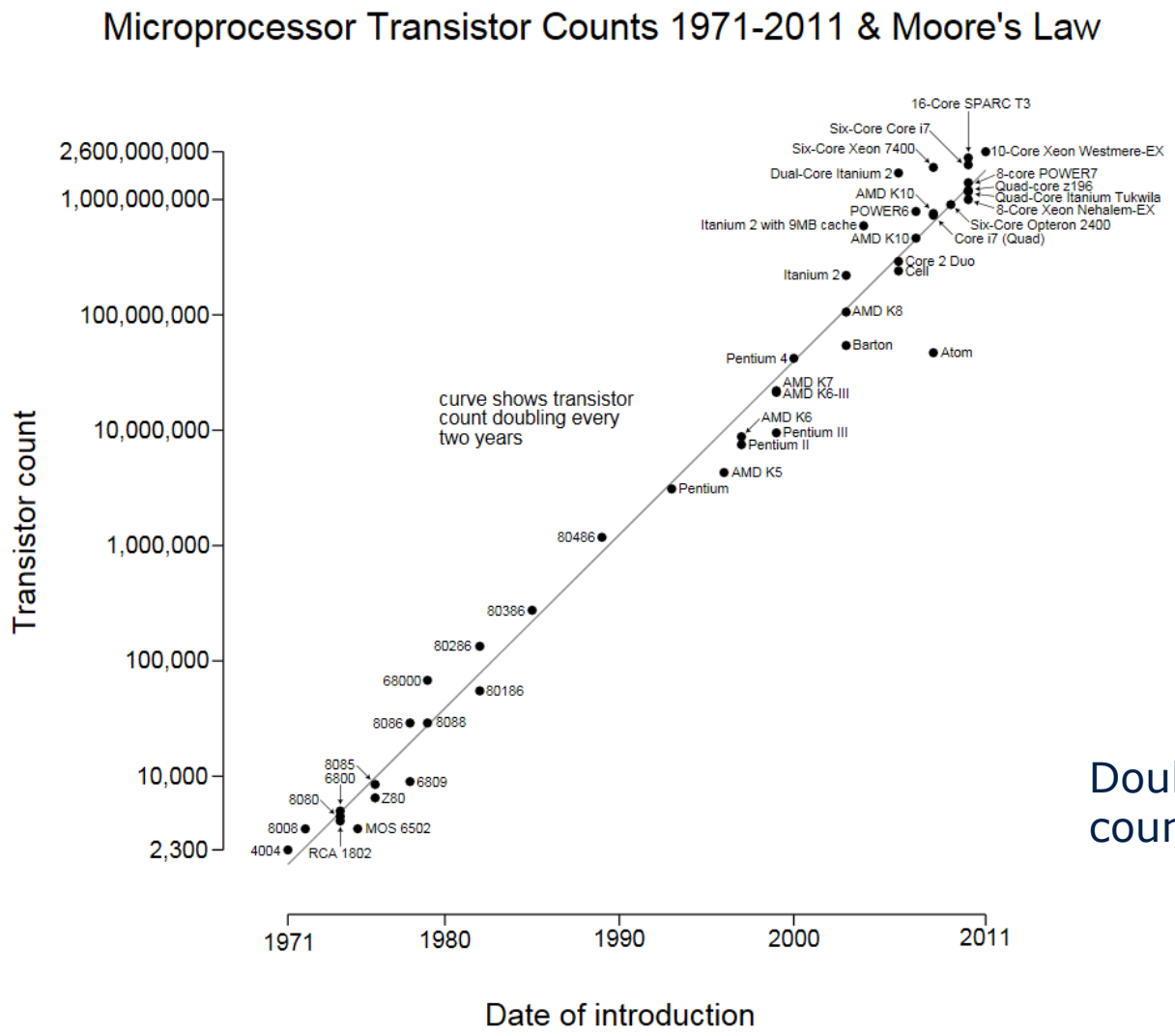
Embedded Multi-core Systems for Mixed-Criticality Applications in Dynamic and Changeable Real-Time Environments – EMC²

(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

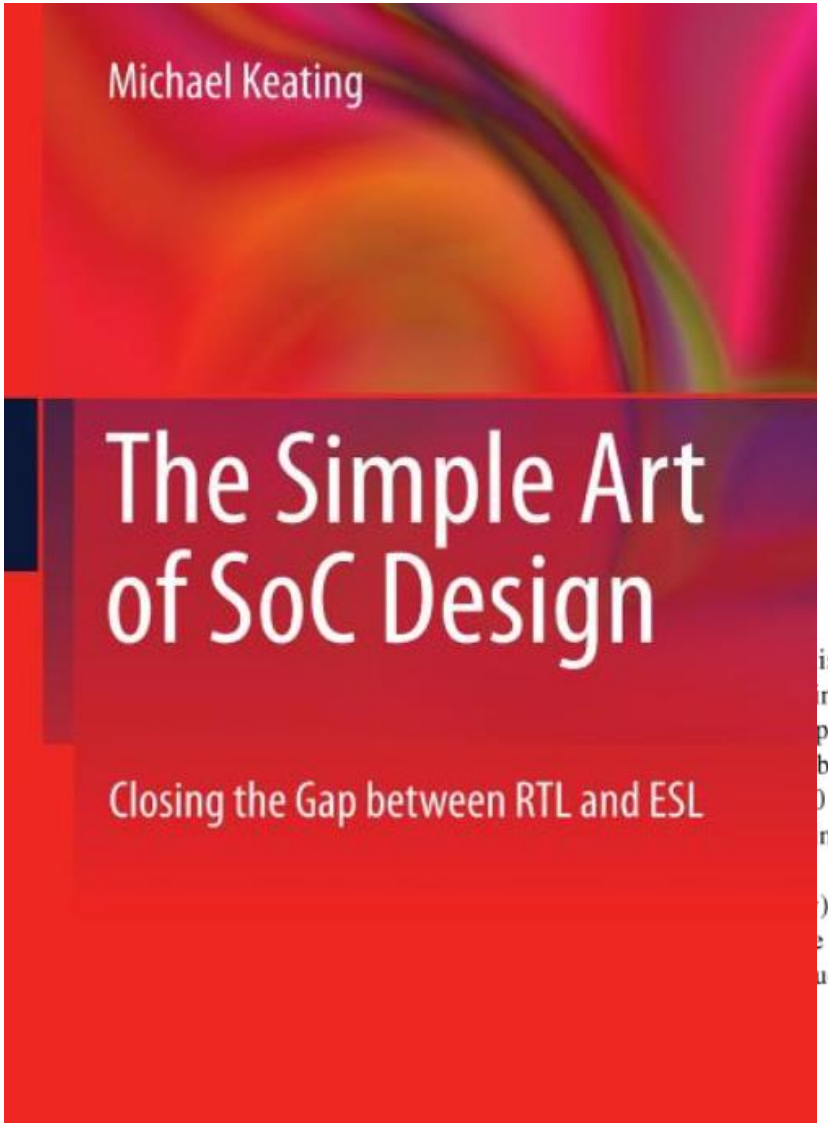
Technological Productivity



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.

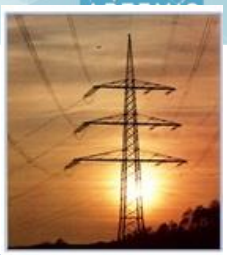


Motivation for EMC²

- Very fast technological advances of µ-electronics in past decades
- Amazing capabilities at lowered cost levels
- Systems quickly put together since the next technology generation is already waiting around the corner
- Today primarily exploited in consumer-oriented products
- 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





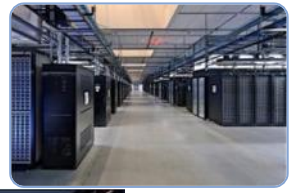





Application innovation

- EMC^2 - 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





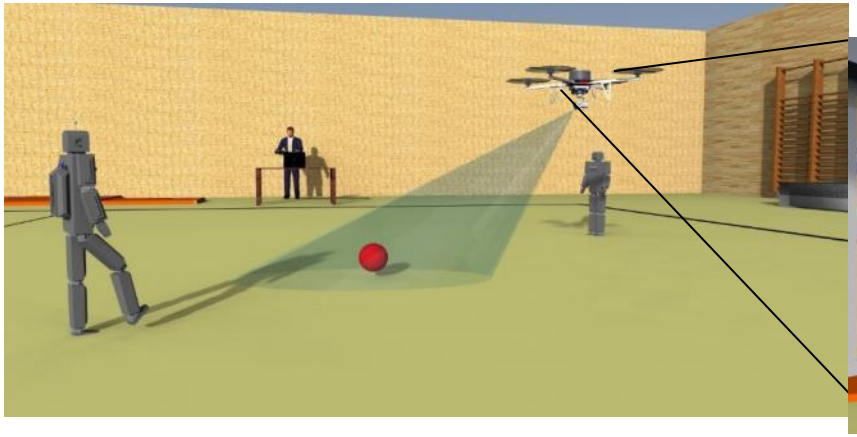


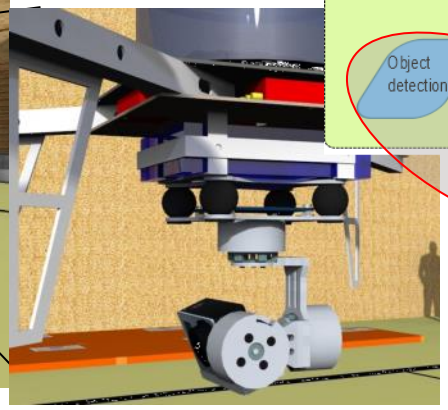
Model based Design for MC Systems

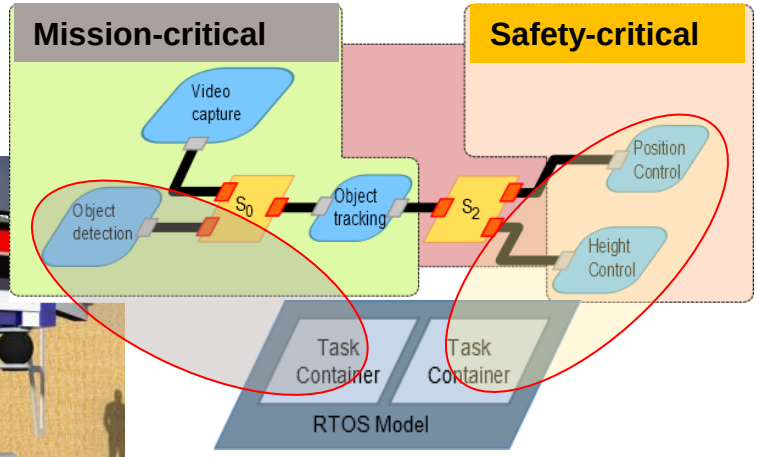
Goal: Safe optimization of QoS in Mixed-Critical Applications

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

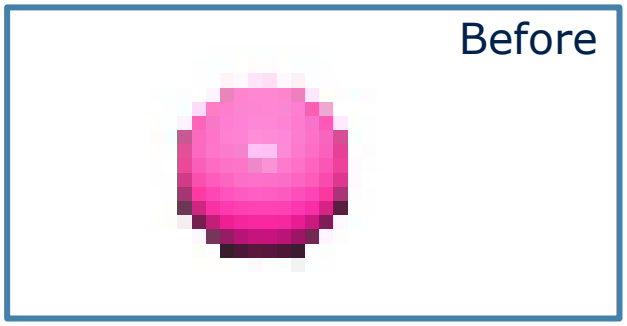




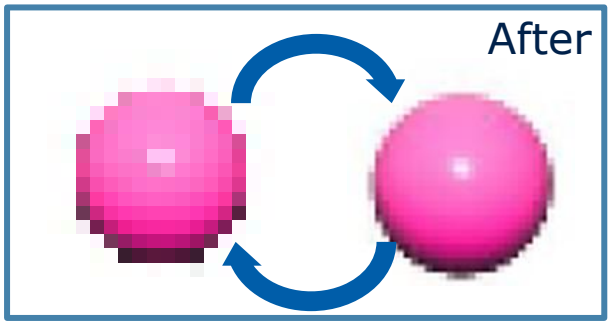


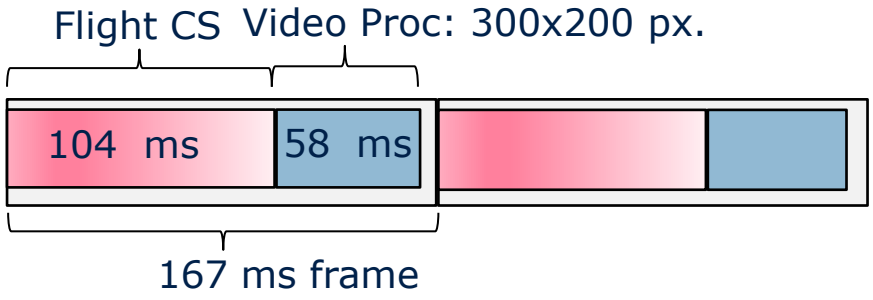
Optimized QoS in Mixed-Critical Applications with Dynamic Criticalities

Static schedule (WCET based)

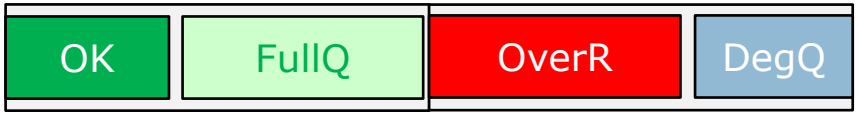


Dynamic Criticality Modes





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%)

Multi-Core Hardware Architectures

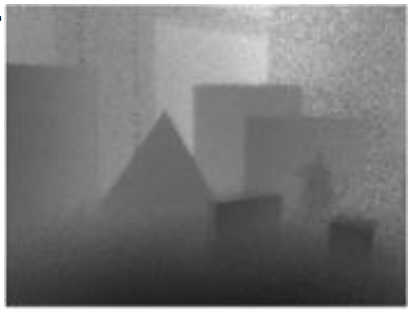
- Various advanced mechanisms (~80) enabling
  - handling high congestion in networked systems
  - e.g. in cooperative intelligent transportation system (ITS) with wireless sensor networks
- mixing different criticality domains in networks for performance and high integration
  - automotive Ethernet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

dynamic network reconfiguration

HW Architectures & Concepts
Use case: 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 FPS
  - HW-accel. ToF processing
    - Novel Zynq-based system solution for mixed-critical app.

ToF 3D camera

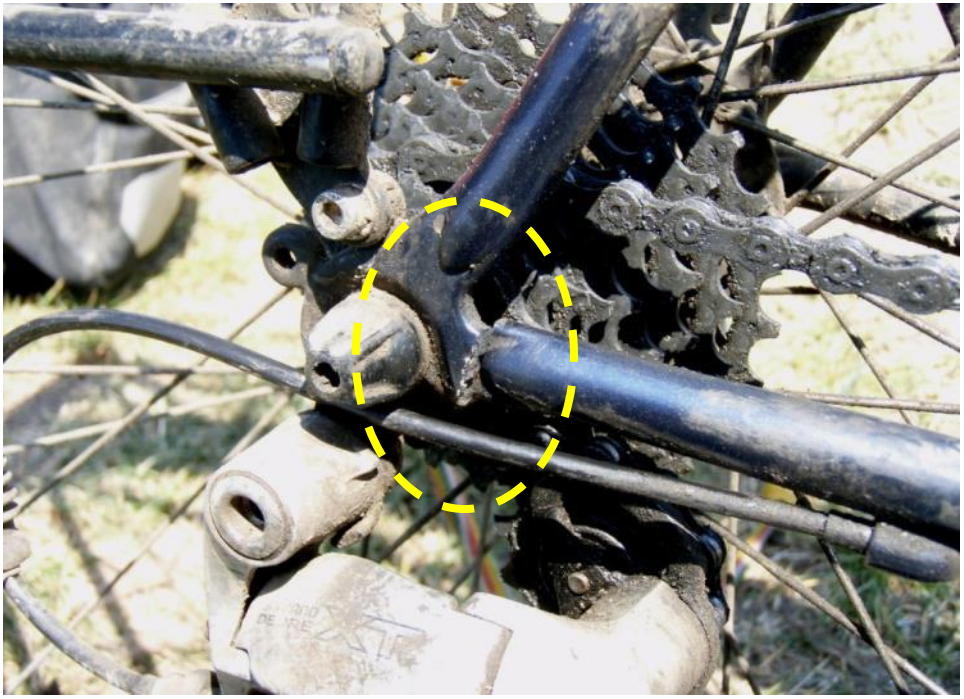


Low-res. ToF image

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.

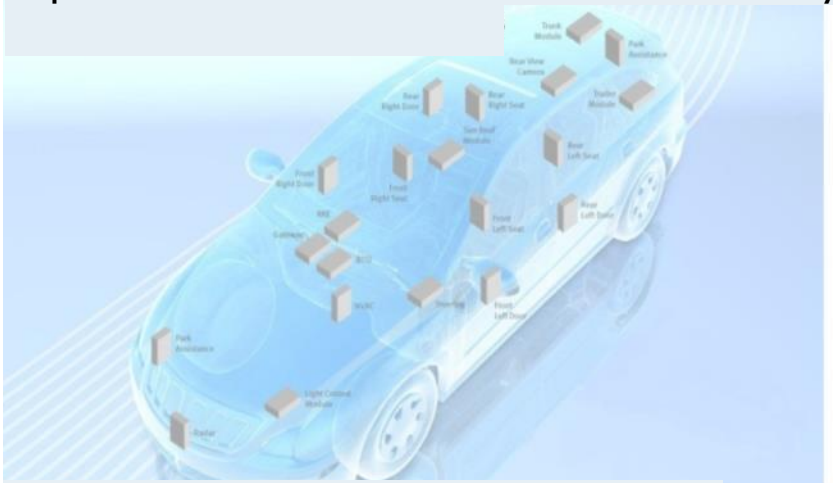
Qualification and Certification

- **Objective:** Enable new applications and business through enabling Safety-Security Assurance and Certification in EMC²-Systems
- **Key achievements**
  - Integration of Safety and Security Engineering to handle the impact of security on safety
  - Conditional runtime certification enabling safety checks of dynamic system compositions
  - High impact on Standardization Consideration of cybersecurity in upcoming editions of functional safety standards

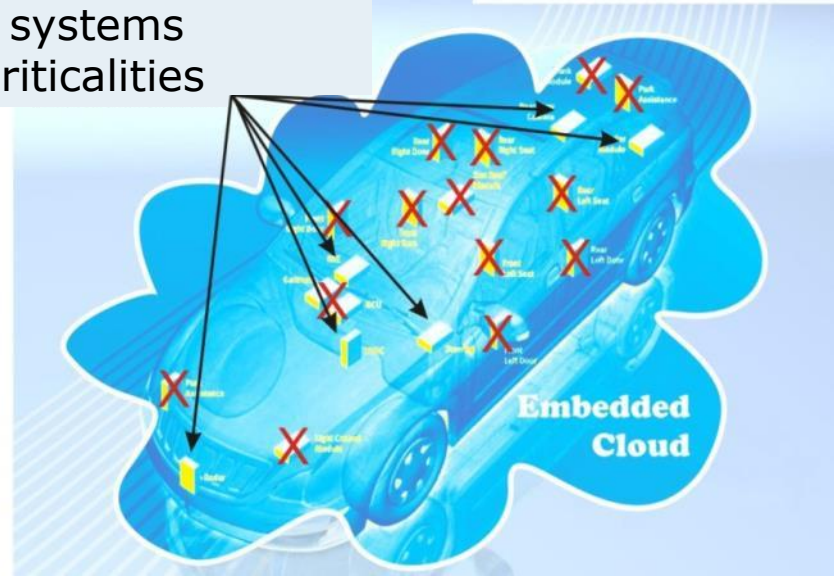
Cf. "Umsetzungsempfehlungen Zukunftsprojekt Industrie 4.0"

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



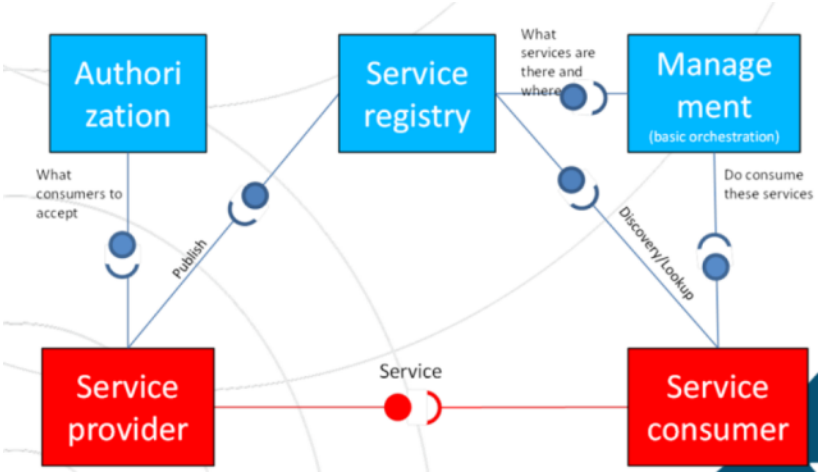
Vision

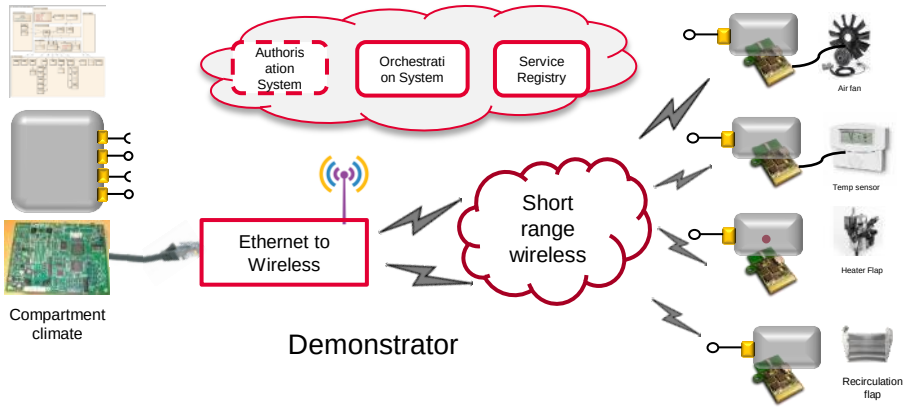
Aggregate resources
In multi/many cores,
ECU networks

Offer system properties 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

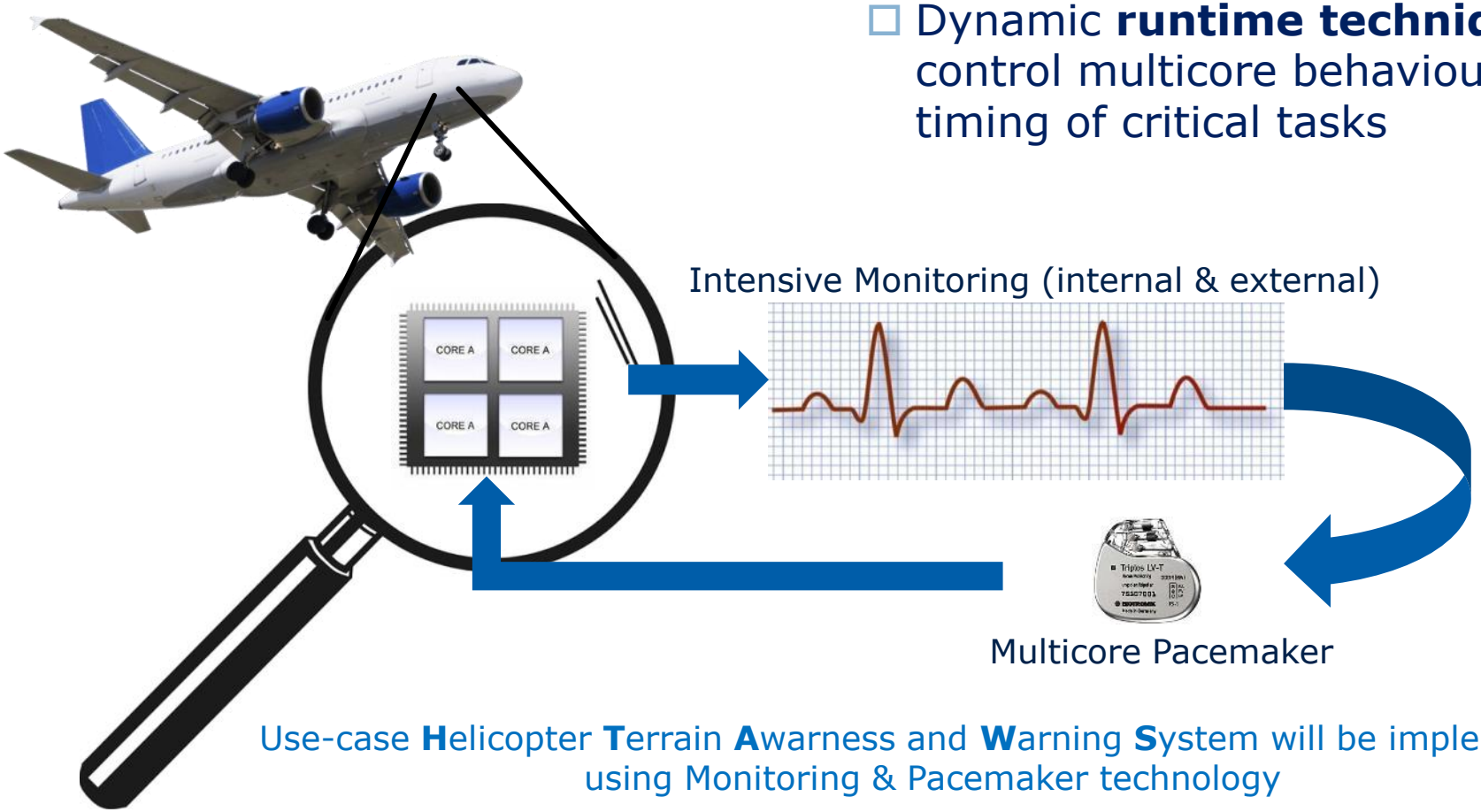




Avionics Use-Case: Technical Results

- **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

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.

Seismic processing

Real-time processing on sea:

300 Mbit/sec per streamer

Seismic volume

Real-time signal processing

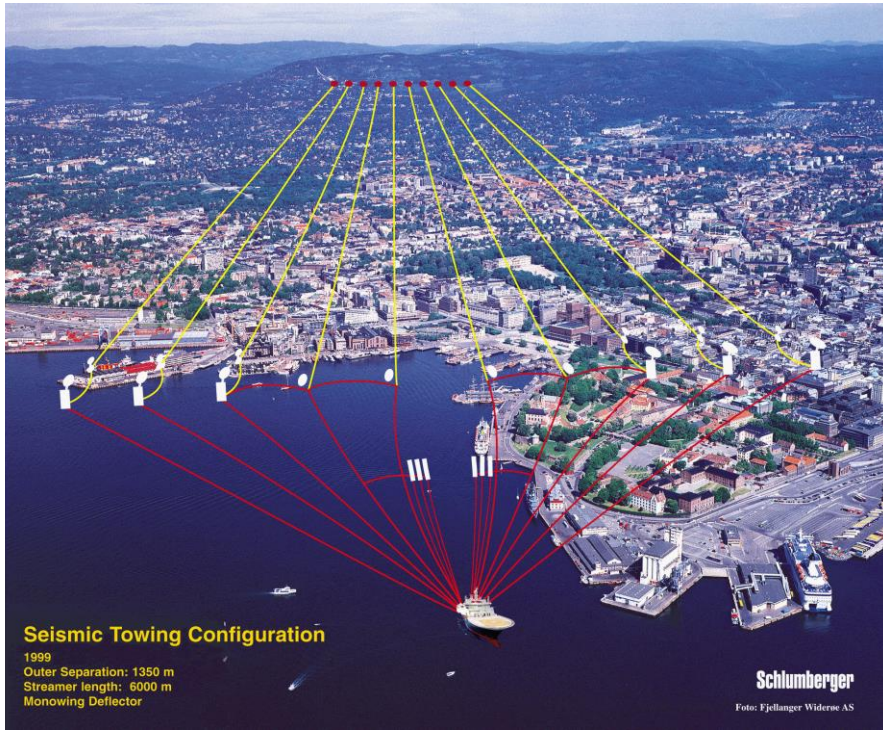
300 Mbit/sec per streamer

On ship:

Further seismic processing

On land:

Further seismic processing



200 computers with 4 000 cores

8-14 streamers behind ship

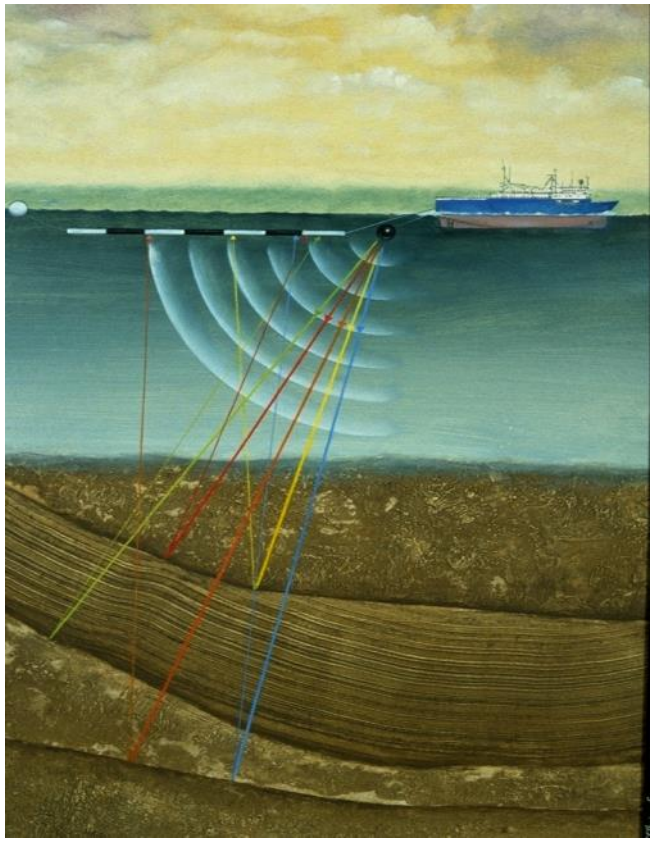
Streamer length 10km - 14 km

100 - 200 computers per streamer

200 000 sensors per streamer

Potential impacts of Use Case on Seismic Processing at sea and on land

- Generated C++ code runs 5 times faster than MATLAB code
- **Reduced engineering time**: New algorithms exploiting multi-cores can be implemented much faster.
- **Reduced execution time**: Reduced execution time tran-slates 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

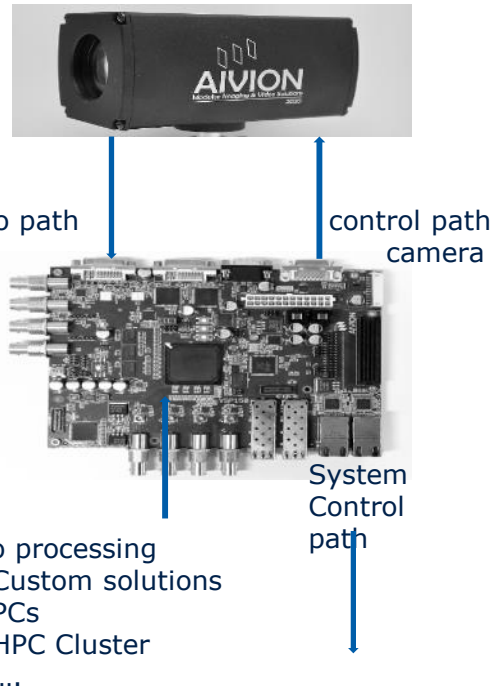
- Surveillance
- Medical applications
- Automated driving
- Quality control in production
- Automatic access control
- …. just a few examples

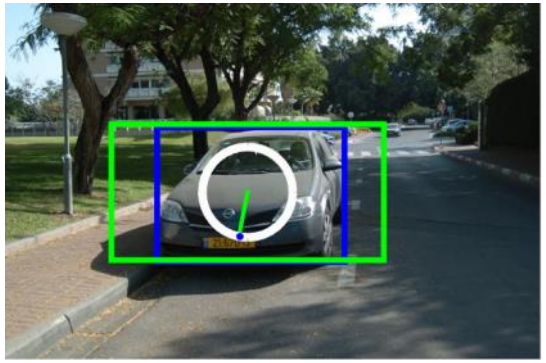
Objective: Acceleration of an object (face) detection algorithm by using multi-core or FPGA architectures.

Results: 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

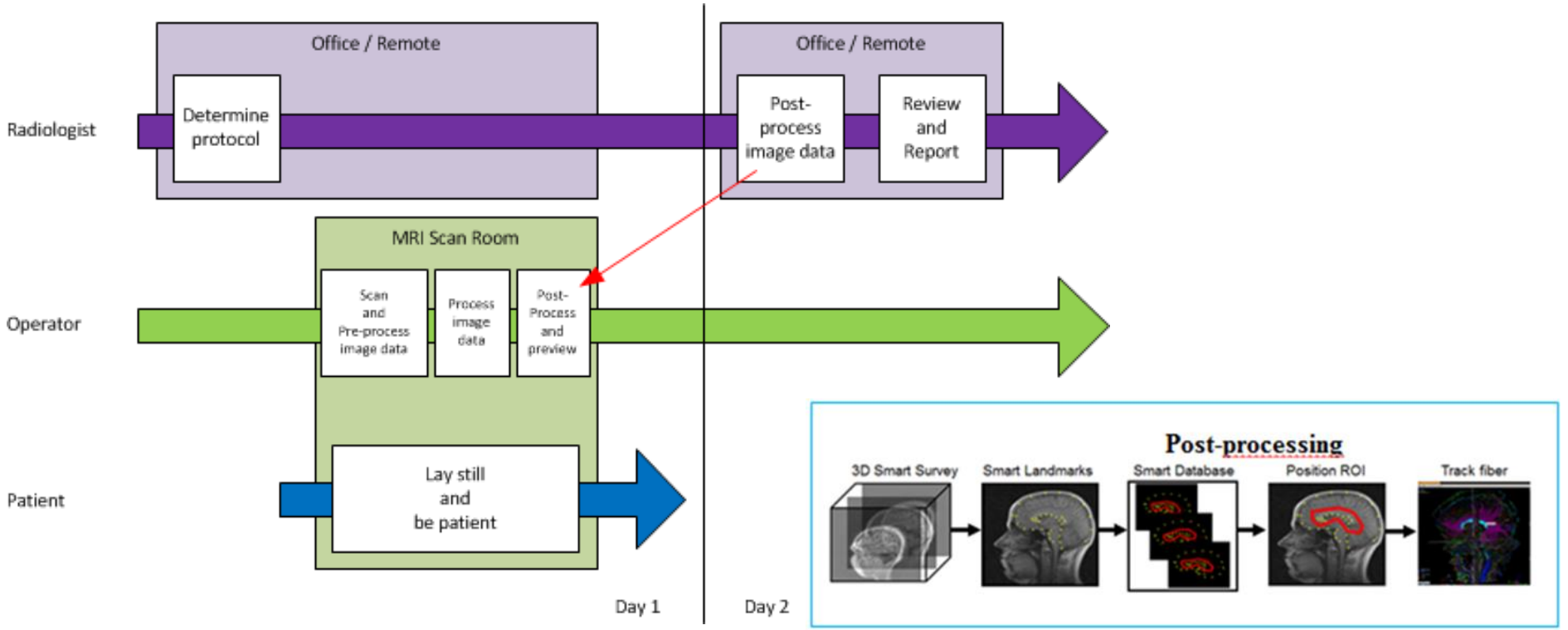




Random forest vehicle detector

Medical imaging

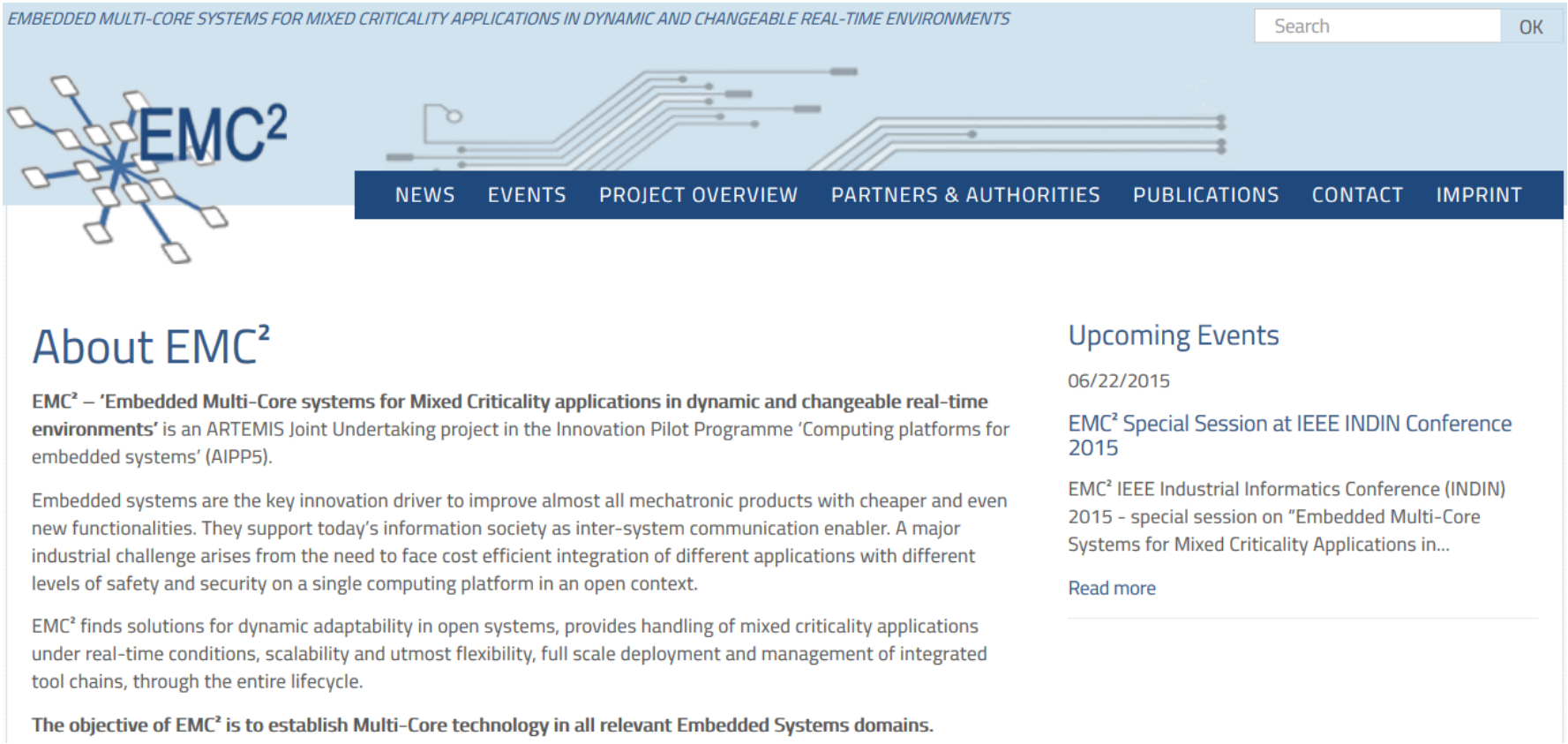
- **Challenge:**
  Prevent patient call back for complex diagnostic procedures
- Go from separate tasks deployed on separate systems to a single system solution



Workflow after EMC² (innovation)

Public project website

- First version online at project start: www.emc2-project.eu
- Website is updated whenever news, events and other information for publication becomes available



EMC² at MCC Workshop

Agenda EMC2 project

- Werner Weber: Introduction to the EMC2 project (10 min)
- Mladen Berekovic : Multicore-Hardware architectures and concepts (20 min)
- Vittoriano Muttillo: A Survey of Mixed-Criticality System Implementation Techniques (20 min)
- Elías Pérez Carrera: Internet of Things and Multimedia Applications (20 min)

ARTEMIS 2013 AIPP5

EMC²

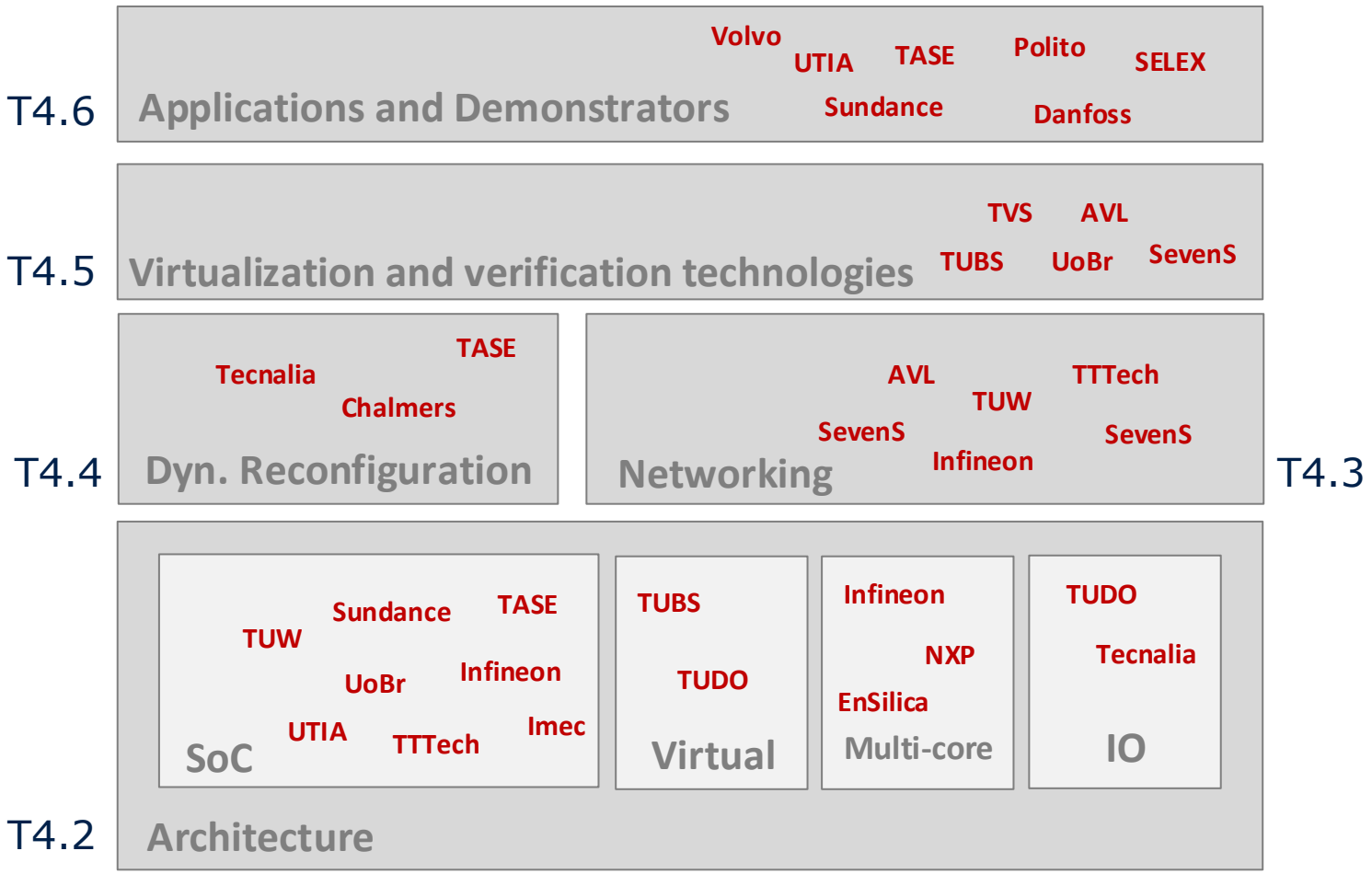
Mixed Criticality Workshop
Barcelona, November 22, 2016

WP4 HW Architectures & Concepts

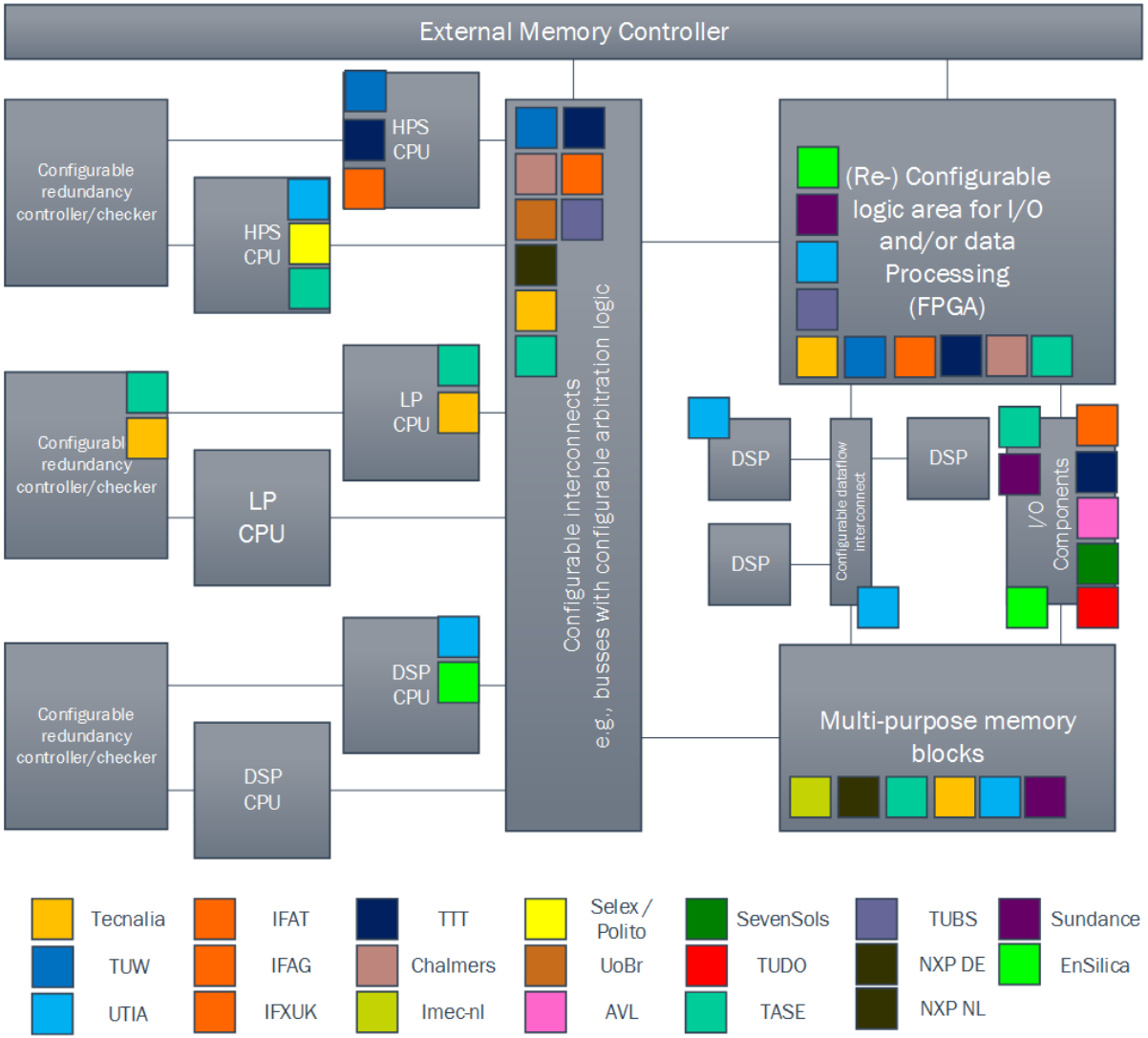
Alexander Lipautz , Infineon AT
e-mail: Alexander.Lipautz@infineon.com , (phone)
Mladen Berekovic, TU Braunschweig
e-mail: berekovic@c3e.cs.tu-bs.de, (phone)
Haris Isakovic, TU Wien
e-mail: haris@vmars.tuwien.ac.at, (phone)

Mixed Criticality Workshop
EMC2-WP4 – Technical Overview

Activities bundled across 5 Technology Lanes:

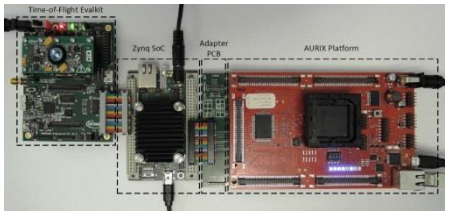


Mixed Criticality Workshop
EMC2-WP4 – Technical Overview



Mixed Criticality Workshop
EMC2-WP4 – Objectives / Highlights

- USPs/Highlights of WP4
  - Architecture and Hardware support for mixed-criticality applications on multi-core platform along the 5 technology lanes
    - EMC2-DP platform – *Reconfiguration (T4.4)*
    - Software Defined **Asymmetric Multiprocessing** on EMC2-DP ZYNQ platform – *Architecture(T4.2)*
    - Heterogeneous TTNoC many-core architecture – *Networking (T4.3)*
    - High-Accurate Distributed Control Systems – *Networking (T4.3)*
    - **Analog-Mixed-Signal Power System** for MC Multi-Core - *Architecture*



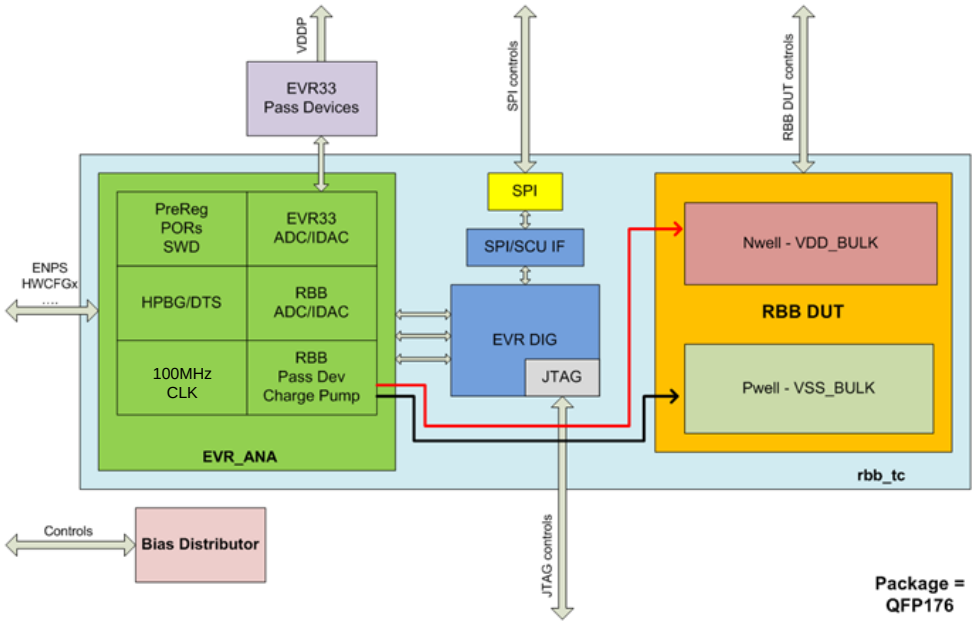
  - System optimisation for MCMC applications
    - **Time-of-Flight 3D Imaging** – *Application & Demonstration (T4.6)*
    - Application-specific Exploration and Optimization MCMC Hardware with **Virtual Hardware platform** – *Virtualization*

- Status at start of period 3:
  - Hardware platforms and tools applied **to internal use-cases**
  - Ongoing **evaluation** and **knowledge transfer** of the proposed technologies and tools with other WPs and Living Labs

WP4 Achievements per technology

Analog-Mixed-Signal Power System (Infineon)

- First full functional safety compliant Analog-Mixed-Signal Power System for multi core and mixed critical systems including standby controller, robust concept implementation, out of operating range functionality is now fully available for next generation of products.
- It introduce highest flexibility and have very efficient regulators with SMPS power optimization features, revers back biasing and dynamic voltage scaling.

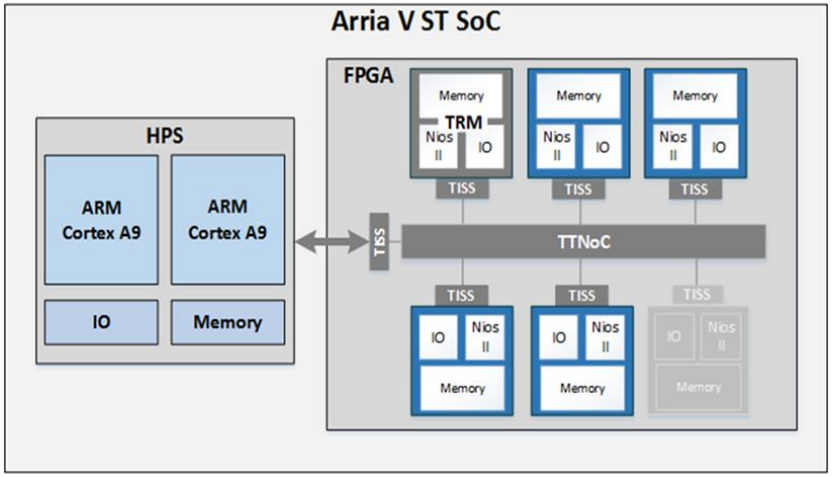


Block diagram of the concept

Mixed Criticality Workshop
EMC2-WP4 – Highlights: T4.3

Heterogeneous TTNoC many-core architecture (TU Wien)

- Objective: Building a deterministic heterogeneous architecture on Altera Arria V SoC
- Key achievements:
  - Integration of TTNoC on the Arria V SoC platform.
  - The architecture combines 4 Nios 2 components and an ARM Cortex A9 component.
  - Full time and space isolation of individual components, designed for mixed-criticality applications.



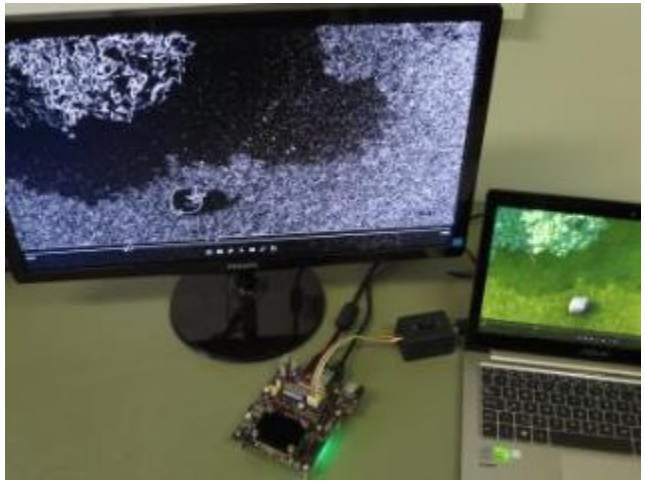
Block diagram of the architecture

Mixed Criticality Workshop
EMC2-WP4 – Highlights, T4.4/4.6

Software Defined Asymmetric Multiprocessing on EMC2-DP ZYNQ platform

- EMC2-DP is Sundance HW platform enabling to use System on Module Component with ZYNQ.
- EMC2-DP is compatible with the Xilinx Software Defined System on Chip (SDSoC 2015.4) flow. UTIA & Sundance designed support for SDSoC.
- EMC2-DP parameters: MicroBlaze and UTIA EdkDSP floating point accelerator deliver:
  - Adaptive LMS filter: 776 MFLOP/s
- This is 2.3x faster than 666 MHz ARM A9 CPU optimized SW with NEON vector proc. unit.
- EMC2-DP HW image processing accelerators are generated by the SDSoC from C.
  See figure: Edge detection on Full HD Video:
  - EMC2-DP ARM + SDSoC HW: 60.0 Frames/s
  - EMC2-DP ARM + platform SW: 5.3 Frames/s

EMC2-DP



Demonstration

Mixed Criticality Workshop
EMC2-WP4 – Highlights: T4.3/T4.6

High-Accurate Distributed Control Systems (SevenS)

- Development of new White Rabbit nodes for the project to improve scalability and stability of the distributed frequency:
  - White Rabbit ZEN
  - White Rabbit LEN
- In terms of time in distributed networks considered as critical data, we have adapted White-Rabbit (able to provide time with <1ns accuracy) and also using our own devices, two distribute time over 14 hops maintaining synchronization capabilities with a jitter in 1-PPS signal under 250ps
- A new clock distribution mechanism (Peer-to-Peer and PeerDelay) to provide White-Rabbit with the possibility of being adapted to industrial Ethernet networks, which opens doors to the development of redundancy protocols such as High-availability Seamless Redundancy (HSR).

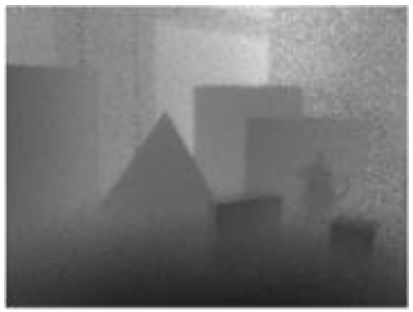
WR-ZEN

WR-LEN

Time-of-Flight 3D Imaging (Infineon)

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

ToF 3D camera



Low-res. ToF image

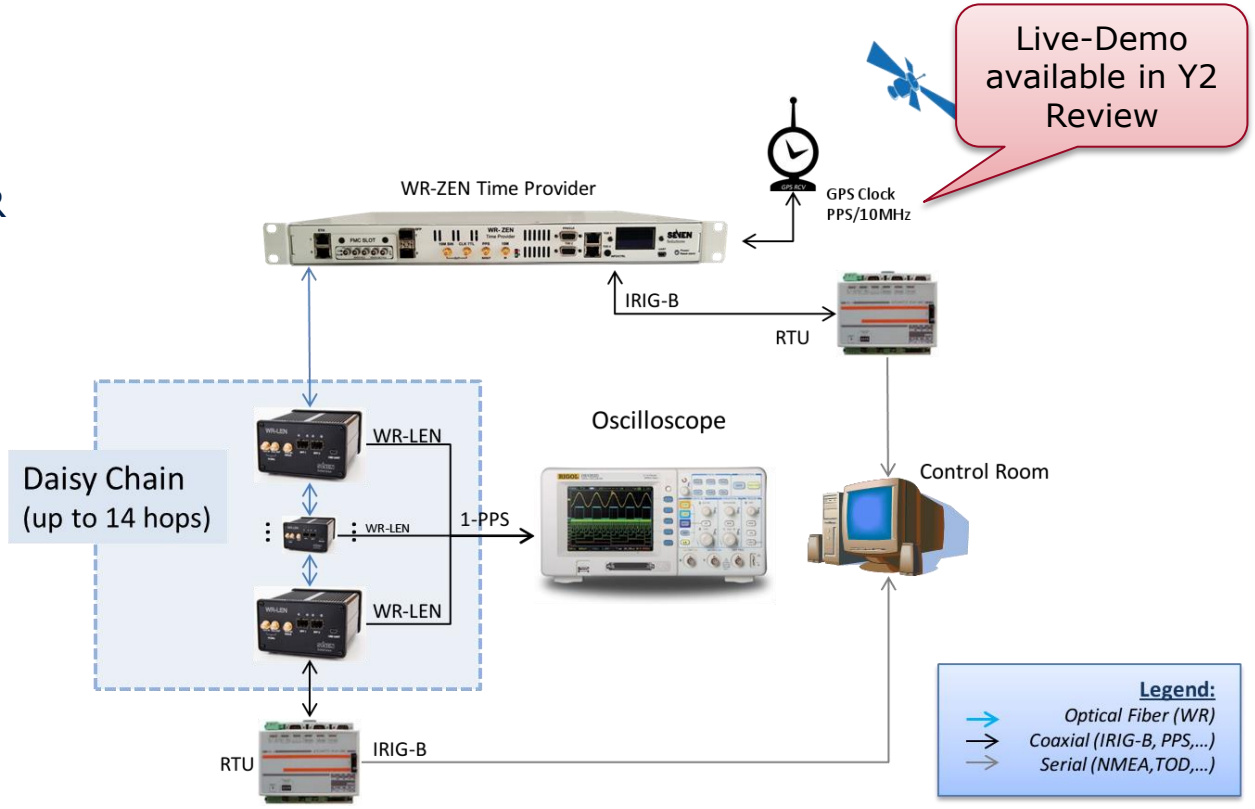
High-res. RGB image

ToF/RGB fused 3D image

Prototype A: <1ns accuracy deterministic time distribution system

Main features:

- **WR-ZEN** board as main time provider (better oscillator)
- **WR-LEN**, a double-port WR node
- **Accuracy <1ns** synchronization
- 1 Pulse per Second and 10MHz outputs
- Daisy-chain configurations with less than **250ps of jitter**
- **Scalable up to 12 nodes** in cascade
- Time distribution using **WR-PTP and IRIG-B**.
- Remote Time Units (RTU) connected to WR devices using IRIG-B



Prototype A

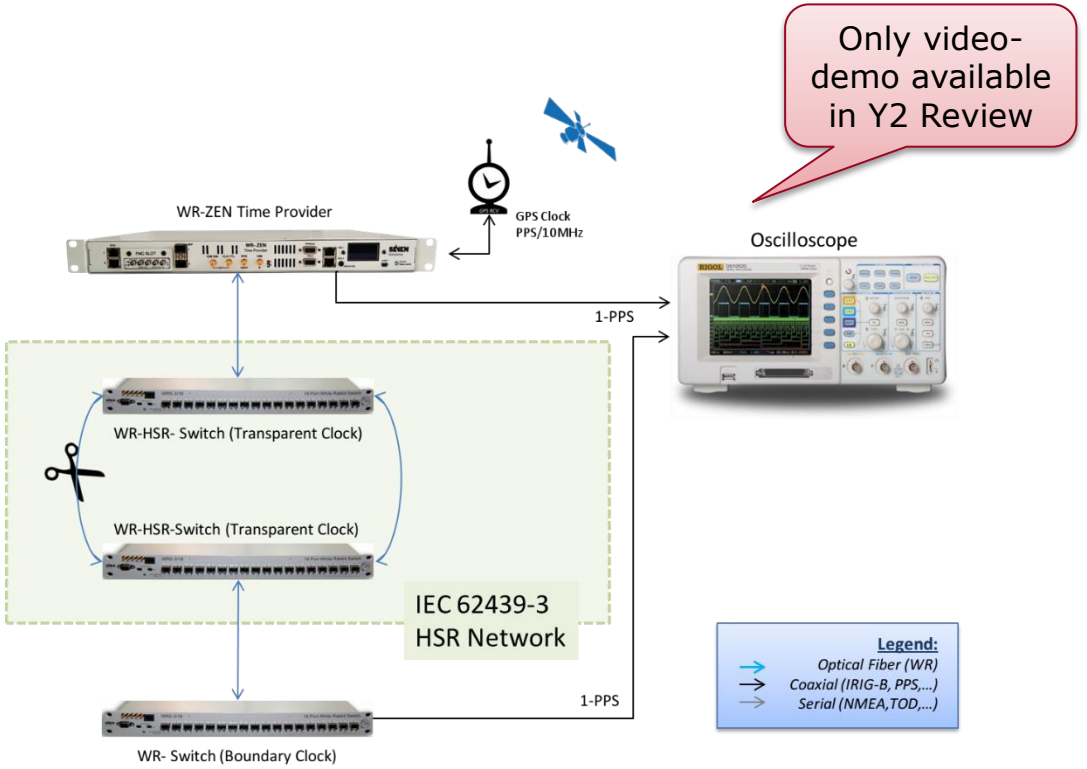
- Setup to demonstrate the scalability of the timing solution and also the utilization of different timing protocols in the same network.

Mladen Berekovic, Alexander Lipautz, Haris Isakovic

Prototype B: Single point of failure avoidance using Transparent Clocks (Redundancy Protocols)

Main features:

- <1ns synchronization
- 1 Pulse per Second (1-PPS) and 10MHz outputs
- **Redundancy Capabilities** (HSR implementation for timing)
- Able to **recover** from a link failure **in ~zero-time**.
- 1-PPS skew improvement in terms of **~50ps**



Prototype B

- Implementation of the HSR redundancy protocol using Transparent Clocks to recover from a system failure in ~zero-time.
- Demo will consist in how two devices connected to a HSR ring are able to remain 1-ns synchronized even after the main time reference is lost (link down).

Mixed Criticality Workshop
EMC2-WP 4 – Technology Transfer

			WP7						WP8		WP9					WP10				WP11					WP12				
			UC_ADAS and C2X	UC_Highly automated driving	UC_Design and validation of next generation hybrid powertrain / E-Drive	UC_Modelling and functional safety analysis of an architecture for ACC system	UC_Infotainment and eCall Application Multi-Critical Application	UC_Next Generation Electronic Architecture for Commercial Vehicles	UC_Multi Domain Avionic Architecture	UC_Hybrid Avionics Integrated Architecture	UC_MPSoC Hardware for Space	UC_MPSoC Software and Tools for Space	UC_Optical Payload Applications	UC_Platform Applications	UC_Radar Payload Application	UC_Drives and electric motors in industrial applications	UC_Identification and authentication	UC_Tracking	UC_Manufacturing quality control by 3D inspection	UC_Multimedia communication WEBRTC	UC_Open deterministic networks	UC_Autonomic home networking	UC_Ultralowpower high datarate communi-cation	UC_Synchronized low-latency deterministic networks	UC_Seismic surveying by ship	UC_Video surveillance for critical infra-structure	UC_Medical imaging	UC_Control applications for critical infra-structure	UC_Railway applications
	No.	Technology title	T7.1	T7.2	T7.3	T7.4	T7.5	T7.6	T8.1	T8.2	T9.1	T9.2	T9.3	T9.4	T9.5	T10.1	T10.2	T10.3	T10.4	T11.1	T11.2	T11.3	T11.4	T11.5	T12.1	T12.2	T12.3	T12.4	T12.5
WP4	4.1	Heterogenous Multiprocessor SoC architectures (T4.1)	2					2			2		1				4												
	4.2	Dynamic reconfiguration on HW accelerators and reconfigurable logic (T4.2)			1						1			1															
	4.3	Networking (T4.3)	2		1												4					3	3			2			
	4.4	Verification and Validation Techniques (T4.4)			1						3						4												

Technology transfer phases
Definition
Evaluation
Development
In Transfer
Transfer complete

Scope 1: Complete implementation into use cases in the project
Scope 2: Partial implementation into use case
Scope 3: Will be transferred to LL (WP7-12) but not implemented into use case
Scope 4: Topic for future applications; technology transfer subsequent to EMC2

A Survey of Mixed-Criticality Systems Implementation Techniques

Authors:

V. Muttillo[1], L. Pomante[1], G. Valente[1]

[1] *Center of Excellence DEWS, University of L'Aquila, Italy*

EMC2 at Mixed-Criticality Cluster Workshop, Barcelona, 22 Nov

Summary

1. Introduction
2. Safety-Related Standards
3. Mixed Criticality Systems Analysis
4. Mixed-Criticality Classification
5. EMC² Examples
6. Conclusion and Future Works

1.

Introduction

"Brief Introduction to Mixed Criticality and Cyber Physical Systems"

Cyber-Physical Systems

- A cyber-physical system (CPS) is an integration of computation with physical processes whose behavior is defined by both cyber and physical parts of the system.

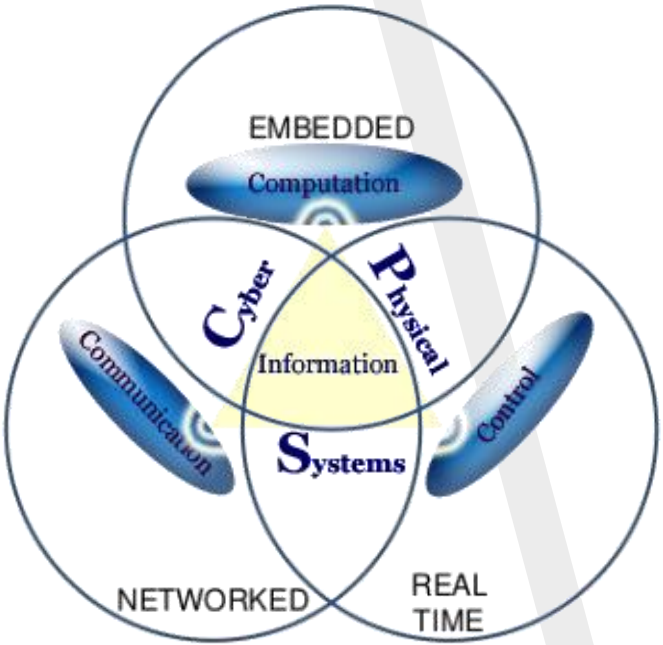
- Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa.

Cyber-Physical Systems

- A cyber-physical system (CPS) is an integration of computation with physical processes whose behavior is defined by both cyber and physical parts of the system.

- Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa.

- As an intellectual challenge, CPS is about the intersection, not the union, of the physical and the cyber*.



** Lee, E. A., Seshia, S. A.,: Introduction to Embedded Systems, a Cyber-Physical Systems approach, Second Edition, LeeSeshia.org, 2015*

Embedded Systems

- In contrast to a generic reprogrammable general purpose computer, an embedded system is composed of a set of tasks already known during the development. This make possible to identify a hardware/software combination specifically designed for such an application.

- Hardware can be reduced to a minimum in order to reduce area, consumption, processing times and manufacture cost, while considering F/NF requirements.

- Many embedded systems are also real-time systems, in which "the correctness of the system behavior depends not only on the logical results of the computations, but also on the time when these results are produced"

embedded/**real-time**

Mixed-Criticality Systems

- A mixed criticality system is "**an integrated suite of HW, OS, middleware services and application software that supports the concurrent execution of safety-critical, mission-critical, and non-critical software within a single, secure computing platform**", i.e. a system containing computer hardware and software that executes concurrently several applications of different criticality (such as safety-critical and non-safety critical).

- Different criticality applications are engineered to provide different levels of assurance, with high criticality applications being the most costly to design and verify.

- Mixed-Criticality systems are typically embedded in more complex systems such as an aircraft whose safety must be ensured.

Embedded/real-time/ safety-critical/**mixed-critical**

2.

Safety-Related Standards

“Criticality is a designation of the level of assurance against failure needed for a system component”

Safety-Related Standards

- Most of the MCS-related researches published in the state-of-the-art cite the **safety-related standards** associated to each application domain (e.g. aeronautics, space, railway, automotive) to justify their methods and results. However, those standards are not, in most cases, freely available, and do not always clearly and explicitly specify the requirements for mixed-criticality

- New MC task model is in essence the result of combining the **standard hard real-time requirements** (studied by the real-time research community since the 70's) with the **notion of "criticality" of execution**. When transposed into the industrial world, the applications that better to such a MC model and its combined requirements are those in which a part of the core functionality is delivered by safety-critical components.

Safety-Related Standards

- **GENERAL** (IEC-61508) based on **SIL (Safety Integrity Level)**: Functional safety standards (of electrical, electronic, and programmable electronic)
  - **AUTOMOTIVE** (ISO26262) based on **ASIL (Automotive Safety Integrity Level)** (Road vehicles - Functional safety)
  - **NUCLEAR POWER** (IEC 60880-2)
  - **MEDICAL ELECTRIC** (IEC 60601-1)
  - **PROCESS INDUSTRIES** (IEC 61511)
  - **RAILWAY** (CENELEC EN 50126/128/129])
  - **MACHINERY** (IEC 62061)
- **AVIONIC** based on **DAL (Development Assurance Level)** related to ARP4761 and ARP4754
  - DO-178B (Software Considerations in Airborne Systems and Equipment Certification)
  - DO-178C (Software Considerations in Airborne Systems and Equipment Certification, replace DO-178B)
  - DO-254 (Airborne - Design), similar to DO-178B, but for hardware
  - DO-160F (Airborne - Test)
- **MEDICAL DEVICE**
  - FDA-21 CFR
  - IEC-62304

3.

Mixed Criticality Systems Analysis

"The more confidence one needs in a task execution time bound (the less tolerant one is of missed deadlines), the larger and more conservative that bound tends to become in practice"

MCS State-Of-The-Art Model

- Almost 200 papers treating of the scheduling of MCS have been referenced in Burns and Davis* paper, and tens of related papers are still published every year. Most of the works about MCS published by the real-time scheduling research community are based on a **model proposed by Vestal*** paper.

- This model assumes that the system has several modes of execution, say **modes {1, 2, … , L}**. The application system is a **set of real-time tasks**, where each task τ_i is characterized by a period T_i and a deadline D_i (as in the usual real-time task model), an assurance level li and a set of **worst-case computational estimates** $\{C_{i,1}, C_{i,2}, \dots, C_{i,l_i}\}$, under the assumption that $C_{i,1} \leq C_{i,2} \leq \dots \leq C_{i,l_i}$

- The different WCET estimates are meant to model estimations of the **WCET at different assurance levels**. The worst time observed during tests of **normal operational scenarios** might be used as $C_{i,1}$ whereas at **each higher assurance level** the subsequent estimates $\{C_{i,2}, \dots, C_{i,l_i}\}$ are assumed to be obtained by more conservative **WCET analysis techniques**.

** Burns, A, Davis, R.I.: "Mixed Criticality Systems - A Review", University of York, 4 March 2016.*
*** S. Vestal, "Preemptive Scheduling of Multi-criticality Systems with Varying Degrees of Execution Time Assurance," Real-Time Systems Symposium (RTSS) 28th IEEE International on, Tucson, AZ, 2007, pp. 239-243.*

MCS Behavioral Model

- The system starts its execution in **mode 1** and **all tasks are scheduled** to execute on the core[s]. Then at runtime, if the system is running in **mode k** then each time the **execution budget $C_{i,k}$ of a task τ_i is overshot**, the **system switches to mode k+1**. It results from this transition from mode k to mode k+1 that **all the tasks of criticality not greater than k** (i.e., $l_i \geq k$) are suspended. Mechanisms have also been proposed to eventually re-activate the dropped tasks at some later points in time*.

- It must be noted that one of the simplifications of this model is the **Vestal's model** with **only two modes**, usually referred to as **LO** and **HI** modes (which stand for **Low- and High-criticality modes**). Multiple variations of that scheduling scheme exist (please refer to [3] for a comprehensive survey); some for single-core, others for multicore architectures. In the case of multicore, both global and partitioned scheduling techniques have been studied. Solutions for **fixed priority scheduling (RM), Earliest Deadline First (EDF)** and **time triggered scheduling** have been proposed. Note that some works also propose to **change the priorities or the periods of the tasks during a mode change** rather than simply stopping the less critical ones.

* *F. Santy, G. Raravi, G. Nelissen, V. Nelis, P. Kumar, J. Goossens, and E. Tovar. Two protocols to reduce the criticality level of multiprocessor mixed-criticality systems. In RTNS 2013, RTNS '13, pages 183–192. ACM, 2013.*

4.

Mixed-Criticality Classification

“A major industrial challenge arises from the need to face cost efficient integration of different applications with different levels of safety and security on a single computing platform in an open context”

MCS Architectures

Separation technique:

- **Timing separation**: scheduling policy, temporal partitioning with HVP, NoC
- **Spatial separation**: one task per core, one task on HW ad hoc (DSP, FPGA), spatial partition with HVP, NoC, MMU, MPU etc.

➢HW:

- **Temporal isolation**: Scheduling HW
- **Spatial isolation**: separated Task on dedicated components (HW ad hoc, FPGA etc.)

➢Single core:

- **Temporal isolation**: Scheduling policy with SO o RTOS, Scheduling policy with HVP
- **Spatial isolation** : MMU, MPU, HVP Partitioning

➢Multi-core

- **Architecture**: shared memory systems, Uniform Memory Architecture, UMA (SMP), Not Uniform Memory Architecture, NUMA, distributed systems, NoC
- **Temporal isolation**: Scheduling policy with SO o RTOS, Scheduling policy with HVP
- **Spatial isolation**: MMU, MPU, HVP partitioning

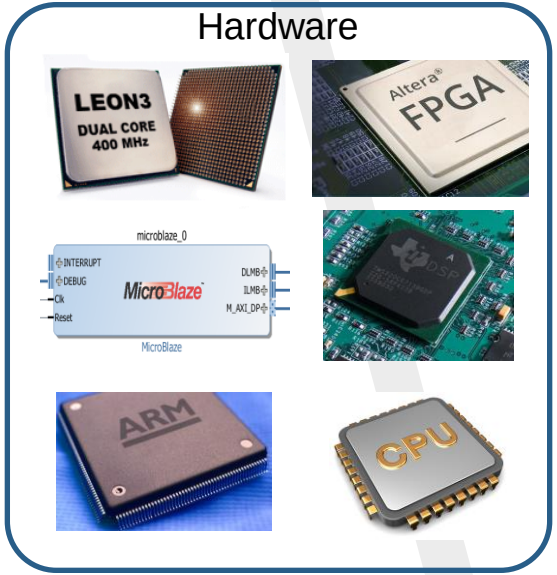
➢Many-core

- **Work in progress**

MCS Technologies

Tecnologies:

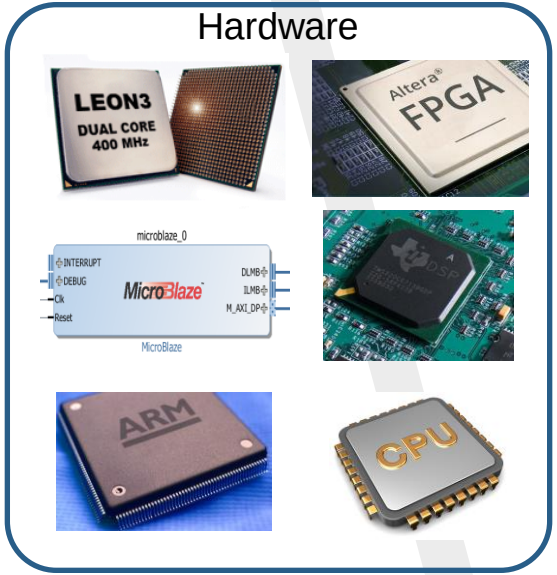
- **Hardware**: HW ad hoc, FPGA, DSP, Processor
  - **Processor**: LEON3, ARM, MICROBLAZE etc.



MCS Technologies

Tecnologies:

- **Hardware**: HW ad hoc, FPGA, DSP, Processor
  - **Processor**: LEON3, ARM, MICROBLAZE etc.
- **Software**: Bare-metal, OS, RTOS, HVP
  - **OS**: Linux etc.



MCS Technologies

Tecnologies:

- **Hardware**: HW ad hoc, FPGA, DSP, Processor
  - **Processor**: LEON3, ARM, MICROBLAZE etc.
- **Software**: Bare-metal, OS, RTOS, HVP
  - **OS**: Linux etc.
  - **RTOS**: eCos, RTEMS, FreeRTOS, Threadx, VxWorks, EriKa etc.

RTOS

THREADX

eCos

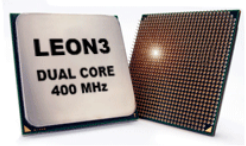
VxWorks

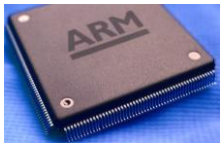
RTEMS

ERIKA ENTERPRISE

freeRTOS

Hardware





OS

MCS Technologies

Tecnologies:

- **Hardware**: HW ad hoc, FPGA, DSP, Processor
  - **Processor**: LEON3, ARM, MICROBLAZE etc.
- **Software**: Bare-metal, OS, RTOS, HVP
  - **OS**: Linux etc.
  - **RTOS**: eCos, RTEMS, FreeRTOS, Threadx, VxWorks, EriKa etc.
  - **HVP**: PikeOS, Xtratum, Xen etc.

HVP

PikeOS

XtratuM

Xen

RTOS

THREADX

eCos

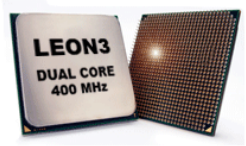
VxWorks

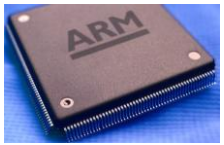
RTEMS

ERIKA ENTERPRISE

freeRTOS

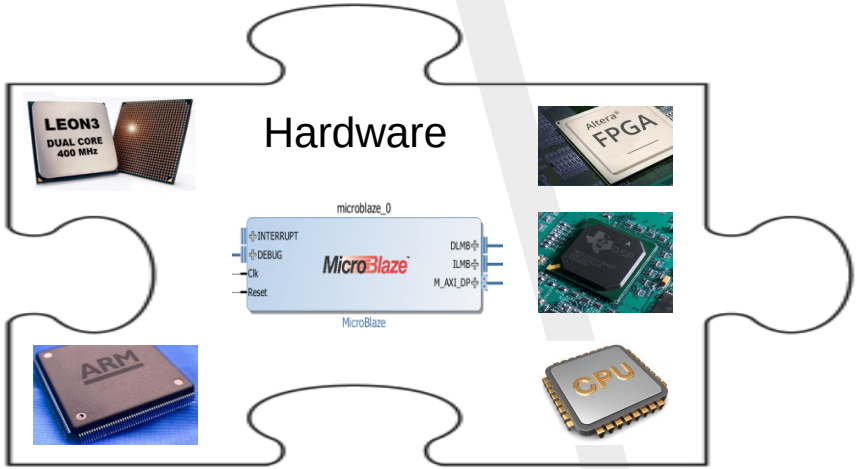
Hardware





OS

MCS Implementations



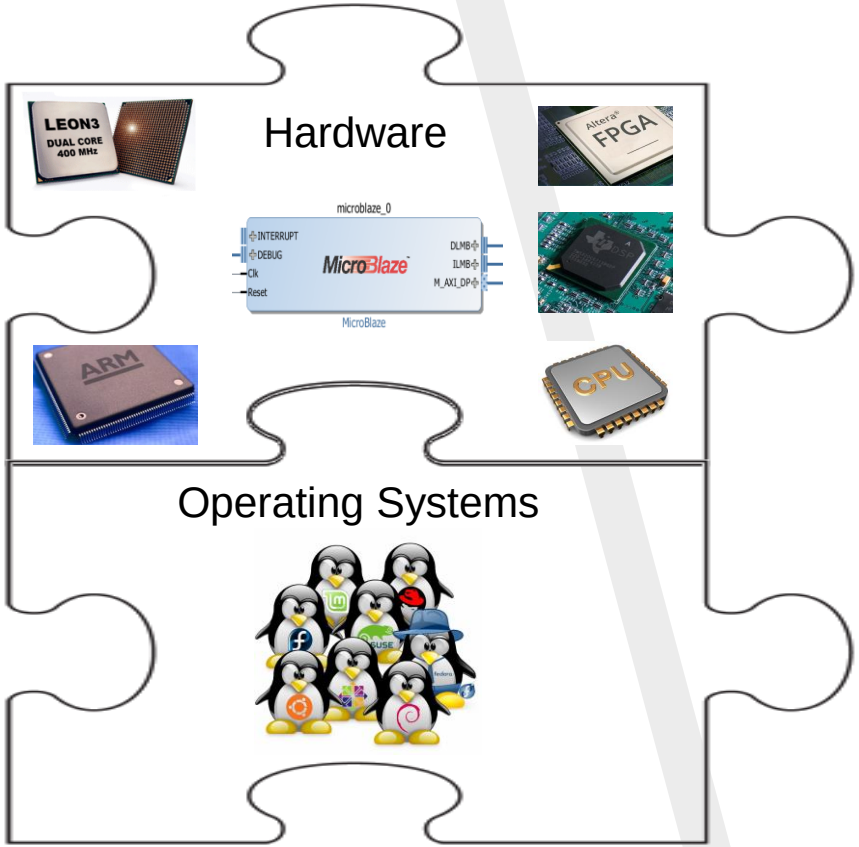
Scheduling:

- **0-Level**

MCS Implementations

Scheduling:

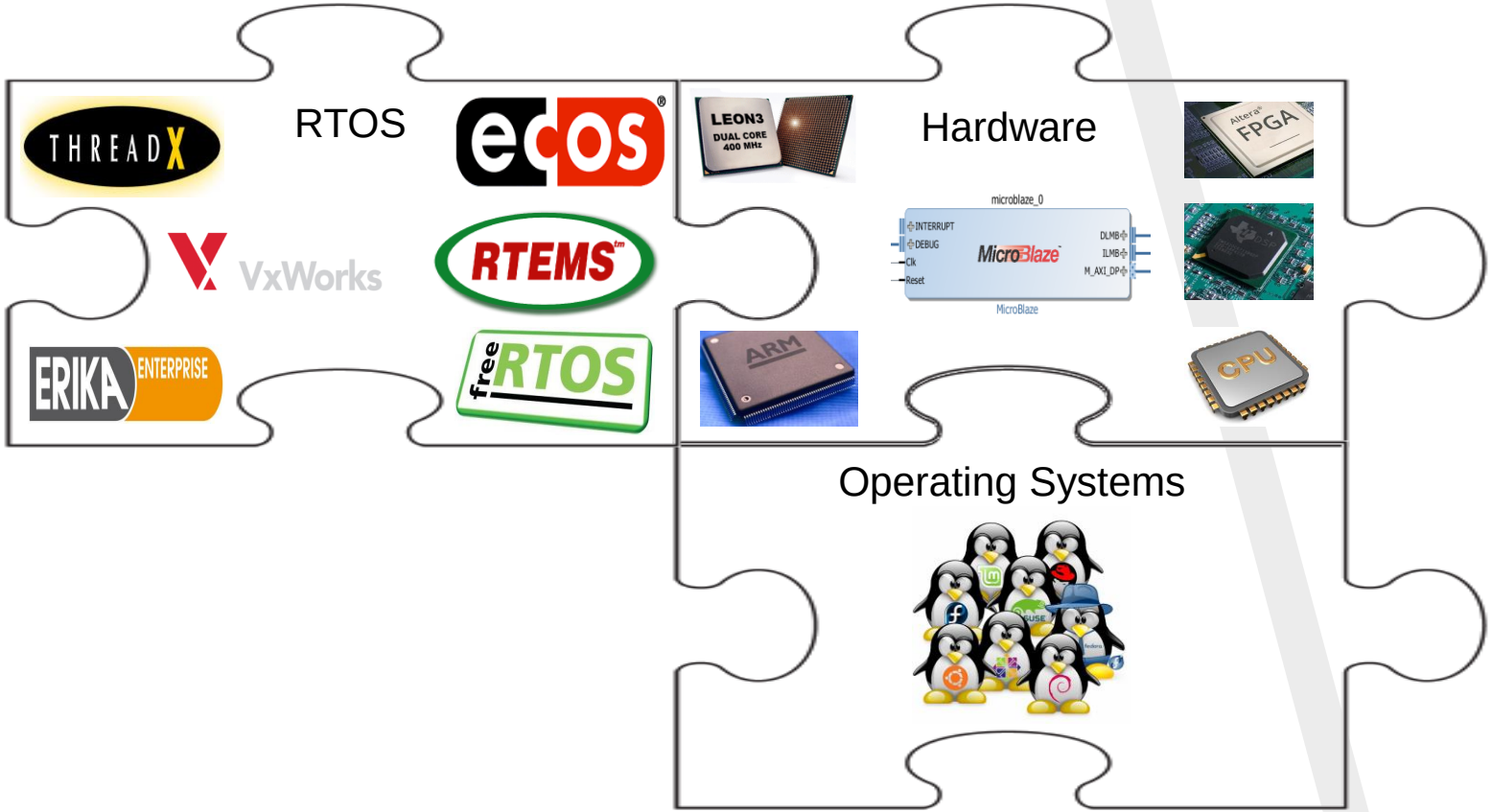
- 0-Level
- 1-Level



MCS Implementations

Scheduling:

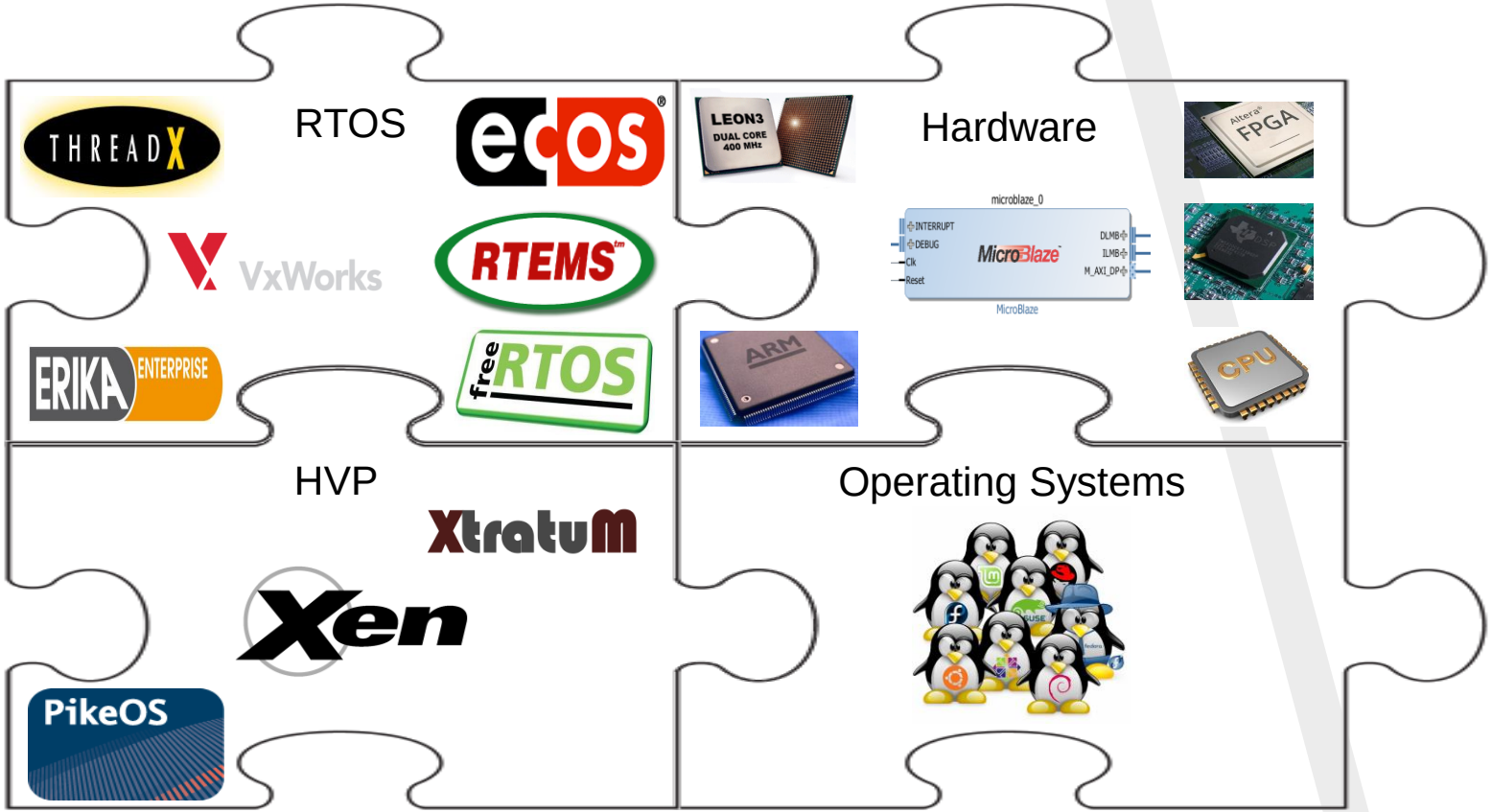
- 0-Level
- 1-Level



MCS Implementations

Scheduling:

- 0-Level
- 1-Level
- 2-Level



MCS Classification

Separation Technique	HW	Single core	Multi-core
Spatial	0-level scheduling [10]	0-level scheduling [11][16]	0-level scheduling [15]
		1-level scheduling [2][5][10][13][16]	1-level scheduling [4][9][15][16]
		2-level scheduling [6][11]	2-level scheduling [3][4][6][7][8][9][14]
Temporal	0-level scheduling [10]	0-level scheduling [11][16]	0-level scheduling [15][16]
		1-level scheduling [1][2][10][13][16]	1-level scheduling [4][9][12][15][16]
		2-level scheduling [6][11][16]	2-level scheduling [1][4][6][7][8][9][14]

Reference

[1] S. Vestal, “Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance,” in Proc. Int’l Real-Time Systems Symp. (RTSS), 2007

[2] A. Gerstinger, H. Kantz, C. Scherrer: ”TAS Control Platform: A Platform for Safety-Critical Railway Applications”, ERCIM NEWS 75, Oct. 2008

[3] D. Muench, O. Isfort, K. Mueller, M. Paulitsch, A. Herkersdorf: "Hardware-Based I/O Virtualization for Mixed Criticality Real-Time Systems Using PCIe SR-IOV," 2013 IEEE 16th International Conference on Computational Science and Engineering, Sydney, NSW, 2013, pp. 706-713

[4] M. Paulitsch, O. M. Duarte, H. Karray, K. Mueller, D. Muench, J. Nowotsch: "Mixed-Criticality Embedded Systems -- A Balance Ensuring Partitioning and Performance" Digital System Design (DSD), 2015 Euromicro Conference on, Funchal, 2015, pp. 453-461

[5] M. G. Hill, T. W. Lake: "Non-interference analysis for mixed criticality code in avionics systems," Automated Software Engineering, 2000. Proceedings ASE 2000. The Fifteenth IEEE International Conference on, Grenoble, France, 2000, pp. 257-260.

[6] J. E. Kim, M. K. Yoon, S. Im, R. Bradford, L. Sha: “Optimized Scheduling of Multi-IMA Partitions with Exclusive Region for Synchronized Real-Time Multi-Core System” in Proceedings of the 16th ACM/IEEE Design, Automation, and Test in Europe (DATE 2013), Mar. 2013.

[7] J. E. Kim, M. K. Yoon, R. Bradford, L. Sha, "Integrated Modular Avionics (IMA) Partition Scheduling with Conflict-Free I/O for Multicore Avionics Systems," to appear in Proceedings of the 38th IEEE Computer Software and Application Conference (COMPSAC 2014), Jul. 2014.

[8] F. Federici, V. Muttillo, L. Pomante, G. Valente, D. Andreetti, D. Pascucci: “Implementing mixed-critical applications on next generation multicore aerospace platforms”, CPS Week 2016, EMC² Summit, Vienna, Austria

[9] B. Huber, C. El Salloum, and R. Obermaisser. A resource management framework for mixed-criticality embedded systems. In 34th IEEE IECON, pages 2425–2431, 2008

Reference

[10] R. Pellizzoni, P. Meredith, M. Y. Nam, M. Sun, M. Caccamo, L. Sha,: “Handling Mixed-criticality in SoC-based Real-time Embedded Systems”, Proceedings of the Seventh ACM International Conference on Embedded Software, 2009

[11] M. Zimmer, D. Broman, C. Shaver and E. A. Lee, "FlexPRET: A processor platform for mixed-criticality systems" 2014 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS), Berlin, 2014, pp. 101-110.

[12] M. Mollison, J. Erickson, J. Anderson, S. Baruah, J. Scoredos: “Mixed-criticality real-time scheduling for multicore systems,” in Proc. of the 10th IEEE International Conference on Computer and Information Technology (CIT), 2010, pp. 1864–1871.

[13] K. Goossens, A. Azevedo, K. Chandrasekar, M. D. Gomony, S. Goossens, M. Koedam, Y. Li, D. Mirzoyan, A. Molnos, A. B. Nejad, A. Nelson, S. Sinha: Virtual execution platforms for mixed-time-criticality systems: the CompSOC architecture and design flow. SIGBED Rev. 10, 3 (October 2013), 23-34.

[14] G. Heiser, B. Leslie: “The OKL4 microvisor: convergence point of microkernels and hypervisors”, In: Proceedings of the first ACM asia-pacific workshop on Workshop on systems (APSys '10). ACM, New York, NY, USA, 19-24

[15] M. Schoeberl, S. Abbaspour, B. Akesson, N. Audsley, R. Capasso, J. Garside, K. Goossens, S. Goossens, S. Hansen, R. Heckmann, S. Hepp, B. Huber, A. Jordan, E. Kasapaki, J. Knoop, Y. Li, D. Prokesch, W. Puffitsch, P. Puschner, A. Rocha, C. Silva, J. Sparsø, A. Tocchi: “T-CREST: Time-predictable multi-core architecture for embedded systems, Journal of Systems Architecture”, Volume 61, Issue 9, October 2015, Pages 449-471

[16] W, Weber, A. Hoess, J. van Deventer, F. Oppenheimer, R. Ernst, A. Kostrzewa, P. Dorè, T. Goubier, H. Isakovic, N. Druml, and others: “The EMC2 Project on Embedded Microcontrollers Technical Progress after Two Years”. Digital System Design (DSD), Euromicro Conference on. Pp. 524-531

5.

EMC² Examples

“Multi-core and many-core computing platforms have to significantly improve system (and application) integration, efficiency and performance”

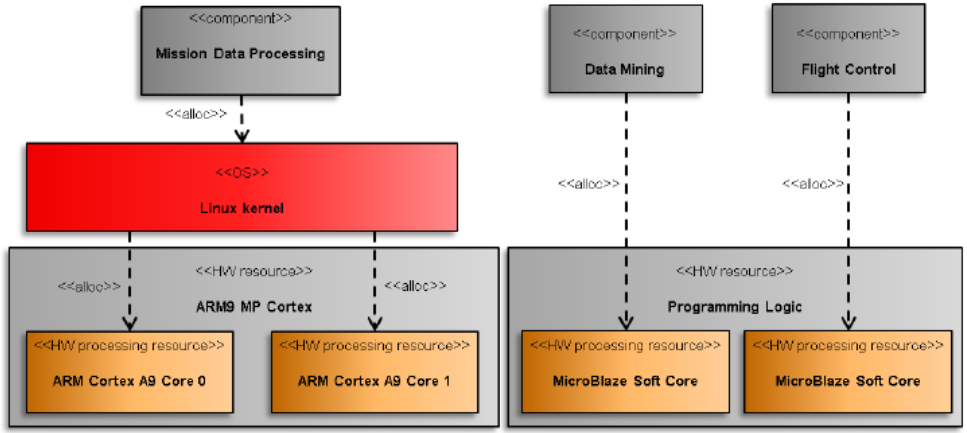
EMC² Examples

Separation Technique	HW	Single core	Multi-core
Spatial	0-level scheduling [10]	0-level scheduling [11][16]	0-level scheduling [15] [16]
		1-level scheduling [2][5][10][13][16]	1-level scheduling [4][9][15] [16]
		2-level scheduling [6][11]	2-level scheduling [3][4][6][7][8][9][14]
Temporal	0-level scheduling [10]	0-level scheduling [11][16]	0-level scheduling [15][16]
		1-level scheduling [1][2][10][13][16]	1-level scheduling [4][9][12][15][16]
		2-level scheduling [6][11]	2-level scheduling [1][4][6][7][8][9][14]

Multi-core Implementation

EMC² WP2 - *4-Copter Demonstrator* [16]

- Flight and Position control
  - Execution on Soft-Cores in FPGA
  - Bare metal, no OS support
  - Interfaces for I2C, PPM and GPIO used
- Object tracking
  - Execution on Dual ARM-Core
  - Needs Linux as OS
  - Multimedia Libraries
  - Needs interfaces USB und Network



Xilinx Zynq 7020:

- ARM dual-core Cortex-A9 (866MHz)
- Artix-7 FPGA (85k Logik Zellen)

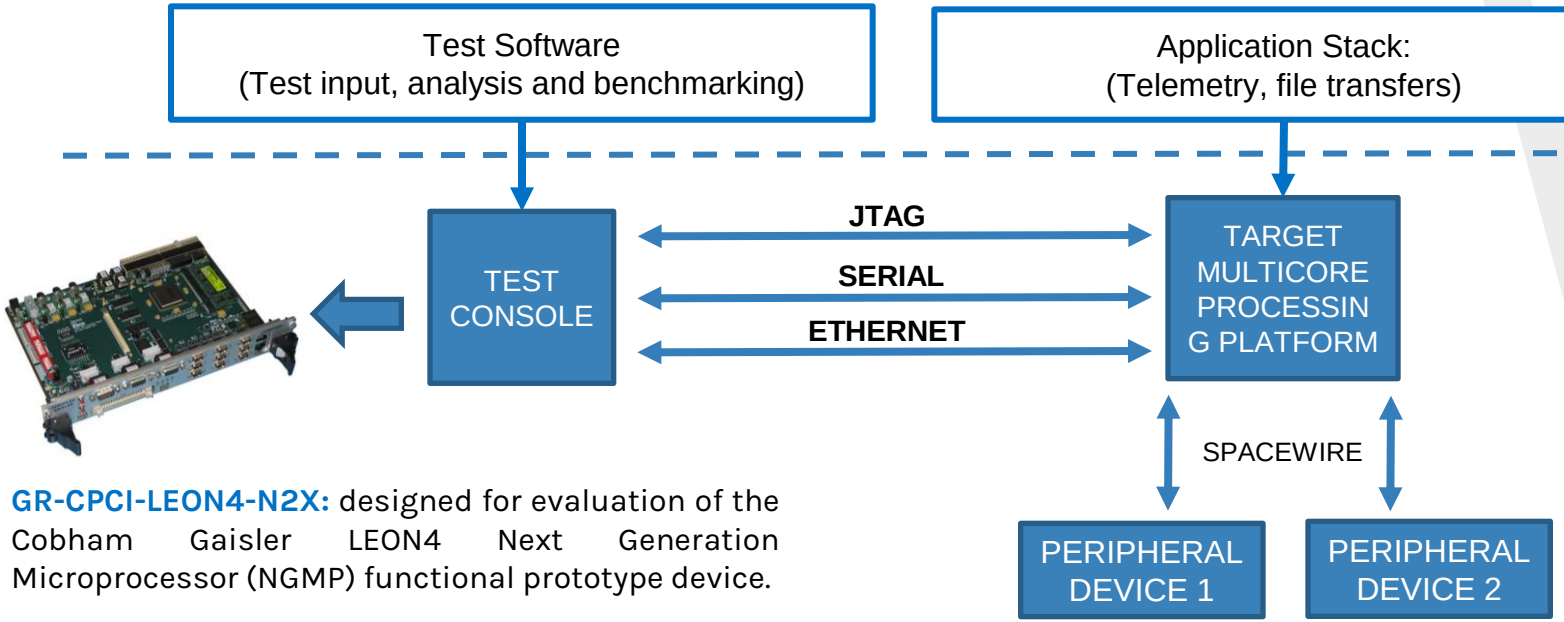
Safety critical tasks: All tasks which are needed for a stable and safety flight of the multi-rotor system, e.g. the flight and navigation controllers. An error, like missing a deadline, will cause a crash-landing!

Mission critical tasks: All tasks which are not needed for a safe flight, but may also have defined deadlines, e.g. tasks which are belonging to the payload processing, like video processing.

Uncritical tasks: All tasks which are not needed either for a safe flight or a correct execution of the mission task, e.g. control of the debug LEDs or transmission of telemetry data.

Multi-core Implementation

Univaq EMC² UC – Satellite Demo Platform (Hardware and Software) [8]



GR-CPCI-LEON4-N2X: designed for evaluation of the Cobham Gaisler LEON4 Next Generation Microprocessor (NGMP) functional prototype device.

Processor: Quad-Core 32-bit LEON4 SPARC V8 processor with MMU, IOMMU

XtratuM

pikeOS

- Migrate a typical aerospace application over a modern multicore platform
- Benchmarking hypervisors
- Compare different virtualization solutions

F. Federici, V. Muttillo, L. Pomante, G. Valente, D. Andreetti, D. Pascucci,: "Implementing mixed-critical applications on next generation multicore aerospace platforms", CPS Week 2016, EMC² Summit, Vienna, Austria

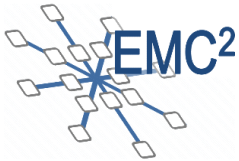
6.

Conclusion and Future Works

“Embedded systems are the key innovation driver to improve mechatronic products with cheaper and even new functionalities. They support today’s information society as inter-system communication enabler. Consequently, boundaries of application domains are alleviated and ad-hoc connections and interoperability play an increasing role”

Conclusions and Future Work

- This talk presents the MC/CPS domain, respect to implementation technologies and techniques
- During the work more than 200 papers related to the Mixed-Criticality Systems Implementation were analyzed (in this talk a subset of these papers has been presented)
- The work divide the whole set of the possible MC solution in different classes related to the architecture and to the specific technologies
- A survey related to this work will be submitted to journal and international conference in order to help designer in their design flow
- EMC^2 result will be used to improve this survey and to refine the implementation classes arising the whole ecosystem and works related to the MC scientific world



THANKS!

Any questions?

Backup Papers

MCS Classification

Separation Technique	HW	Single core	Multi-core
Spatial	0-level scheduling [10]	0-level scheduling [11][16]	0-level scheduling [15][16]
		1-level scheduling [2][5][10][13][16]	1-level scheduling [4][9][15][16]
		2-level scheduling [6][11]	2-level scheduling [3][4][6][7][8][9][14]
Temporal	0-level scheduling [10]	0-level scheduling [11][16]	0-level scheduling [15][16]
		1-level scheduling [1][2][10][13][16]	1-level scheduling [4][9][12][15][16]
		2-level scheduling [6][11]	2-level scheduling [1][4][6][7][8][9][14]

Hardware Implementation

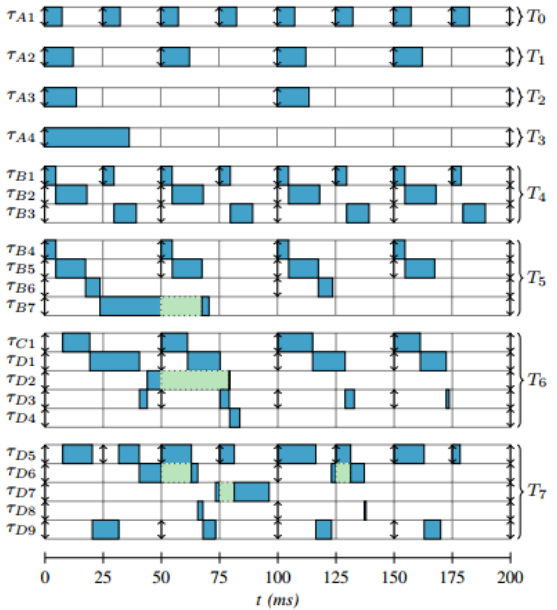
FlexPRET: Processor Platform for Mixed-Criticality Systems [11]

FlexPRET is a 32-bit, 5-stage, fine-grained multithreaded processor with software-controlled, flexible thread scheduling. It uses a classical RISC 5-stage pipeline: instruction fetch (F), decode (D), execute (E), memory access (M), and writeback (W). Predict not-taken branching and software-controlled local memories are used for fine-grained predictability. It also implements the RISC-V ISA [20], an ISA designed to support computer architecture research, that we extended to include timing instructions.

FlexPRET is implemented in Chisel [26], a hardware construction language that generates both Verilog code and a cycle-accurate C++-based simulator.

Task	Thread ID	Thread Mode	T_i, D_i (ms)	$E_{i,1}$ ($*10^5$)	$E_{i,1/2}$ ($*10^5$)	$E_{i,1/3+}$ ($*10^5$)
τ_{A1}	0	HA	25	1.10	1.00	0.95
τ_{A2}	1	HA	50	1.80	1.64	1.55
τ_{A3}	2	HA	100	2.00	1.82	1.72
τ_{A4}	3	HA	200	5.30	4.83	4.56
τ_{B1}	4	HA	25	1.40	1.27	1.20
τ_{B2}	4	HA	50	3.90	3.54	3.34
τ_{B3}	4	HA	50	2.80	2.54	2.40
τ_{B4}	5	HA	50	1.40	1.28	1.21
τ_{B5}	5	HA	50	3.70	3.37	3.19
τ_{B6}	5	HA	100	1.80	1.64	1.55
τ_{B7}	5	HA	200	8.50	7.75	7.32
τ_{C1}	6	SA	50	1.90	1.77	1.63
τ_{D1}	6	SA	50	5.40	5.03	4.65
τ_{D2}	6	SA	200	2.40	2.33	2.28
τ_{D3}	6	SA	50	1.30	1.26	1.23
τ_{D4}	6	SA	200	1.50	1.45	1.42
τ_{D5}	7	SA	25	2.30	2.14	1.98
τ_{D6}	7	SA	100	4.80	4.65	4.30
τ_{D7}	7	SA	200	13.00	12.70	12.44
τ_{D8}	7	SA	100	0.60	0.57	0.56
τ_{D9}	7	SA	50	2.40	2.33	2.28

A mixed-criticality avionics case study

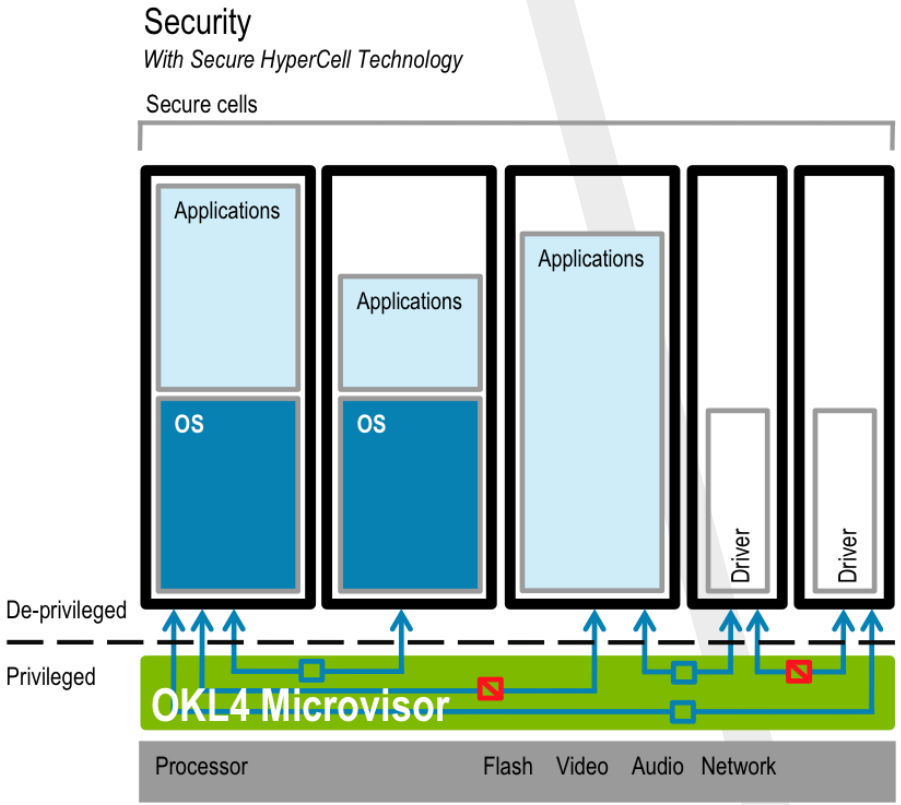


FlexPRET-8T (8 physical number of threads available) executing a mixed-criticality avionics case study

Single-core Implementation

OKL4 Microvisor [14]

- The **OKL4** Microvisor is an advanced secure type-1 hypervisor developed by General Dynamics C4 Systems and supports all ARM processors with MMU hardware
- Supporting virtualization with the lowest possible overhead, the microvisor's abstractions are designed with:
- the microvisor's execution abstraction is that of a virtual machine with one or more virtual CPUs (vCPUs), on which the guest OS can schedule activities;
- the memory abstraction is that of a virtual MMU (vMMU), which the guest OS uses to map virtual to (guest) physical memory;
- the I/O abstraction consists of memory-mapped virtual device registers and virtual interrupts (vIRQs);
- communication is abstracted as vIRQs (for synchronisation) and channels. The latter are bi-directional FIFOs with a fixed (configurable per channel) buffer allocated in user space (run also TCP/IP on a channel).



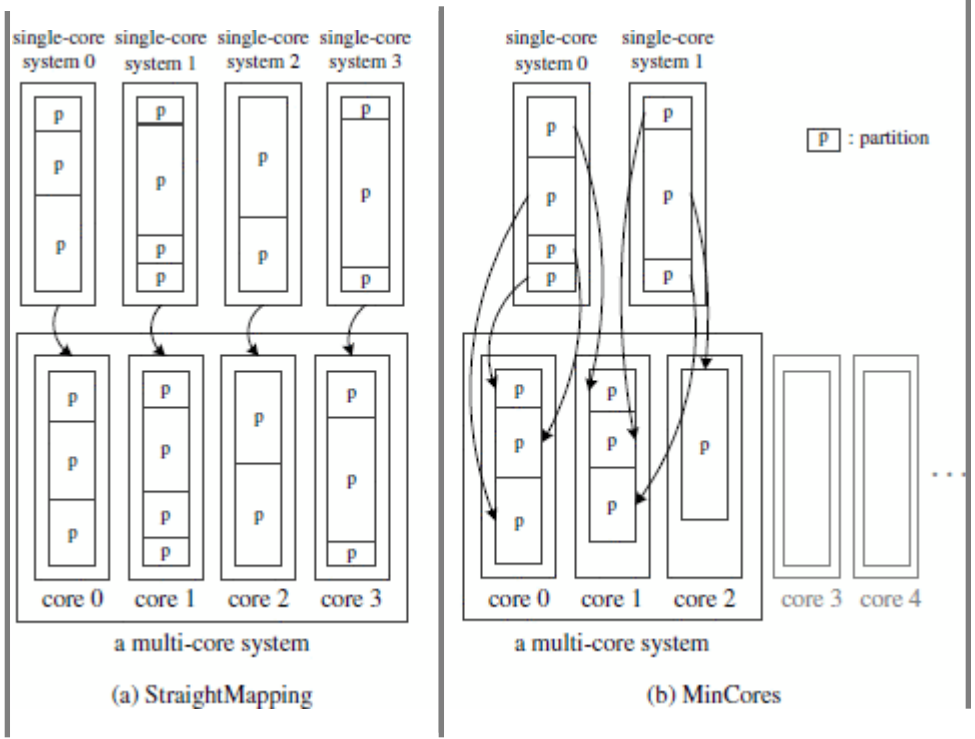
Multi-core Implementation

Multi-IMA Partitioning [6]

Given a set of:

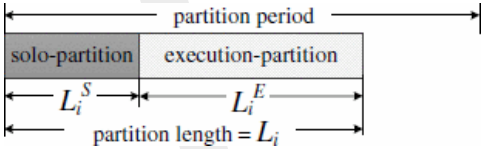
- Single-core IMA systems
- Partitions
- Multi-core system

All partitions from a single core must be scheduled on the same core of the multi-core system (they can be rescheduled within the same core)



Multi-IMA partition scheduling optimization where a partition consists of two logical regions:

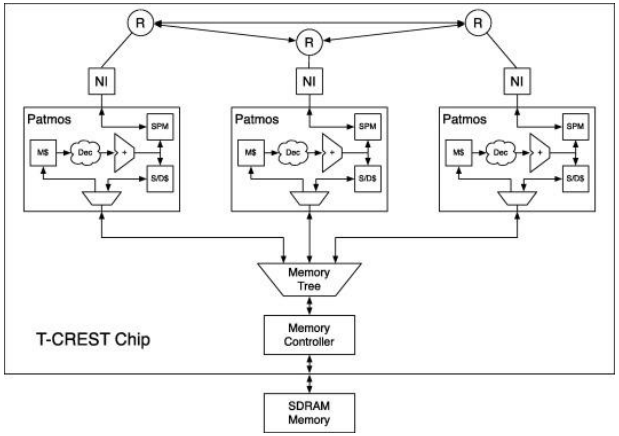
- solo- partition (in avionics systems performing I/O transactions)
- execution-partition

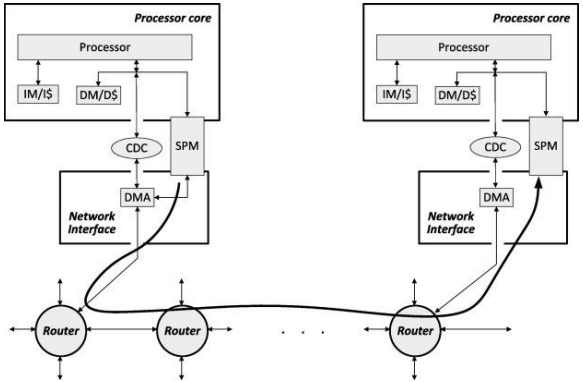


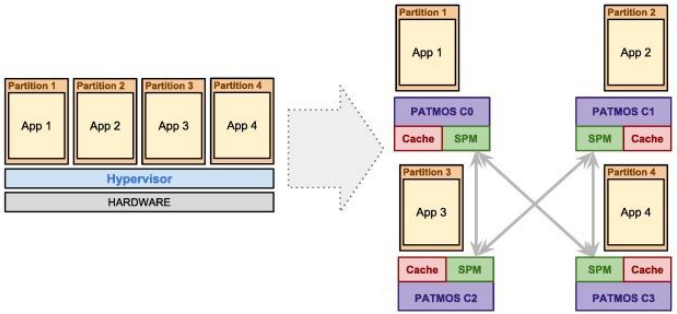
Partition can be scheduled on a core if it doesn't interfere with the solo-partitions of other partitions and doesn't overlap with other partitions assigned to the same core. Each partition is strictly periodic and non-preemptive. This supports **temporal** and **spatial** isolation among partitions.

Many-core Implementation

T-CREST Multi-core Architecture [15]





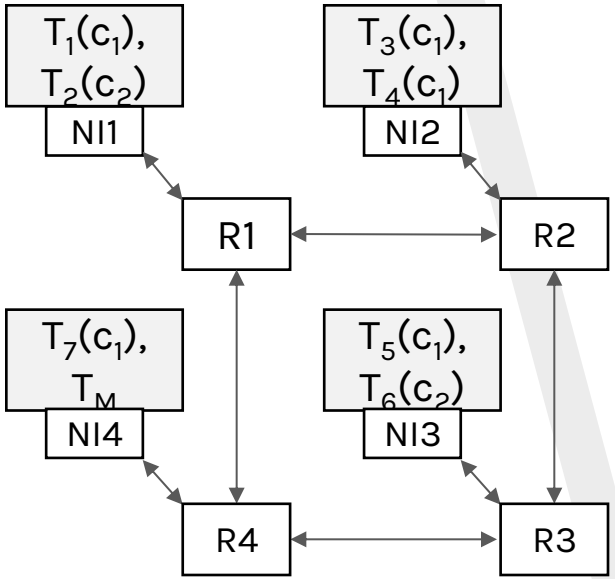


- The ***T-CREST*** platform consisting of Patmos processor nodes that are connected via an on-chip network for message passing communication and a memory tree to a memory controller for shared memory access
- Data transfer in the **T-CREST** core-to-core message passing NoC.
- Mapping of **ARINC 653** partitions to cores onto the **T-CREST** platform.

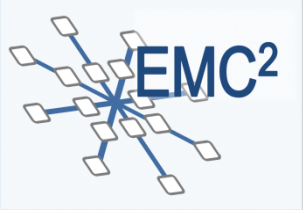
Many-core Implementation

University of L'Aquila CRAFTERS Case Study

- **Hardware mechanisms to support isolation in a Network-on-Chip**
  - Isolation of different application classes on NoC architectures
  - Hardware mechanisms supporting isolation to be introduced into existing network interfaces
  - Support for the execution of multiple applications with different criticality levels
  - Strategy: message exchange supervision



L. Pomante, C. Tieri, F. Federici, M. Colizza, M. Faccio, R. Cardinali, B. Iorio. "HW Mechanisms to Support Isolation in Mixed-Criticality NoC". Euromicro Conference on Digital System Design - WIP Session, Verona, August 2014.



ARTEMIS 2013 AIPP5

EMC²

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

Internet of Things and Multimedia Applications

Mixed-Criticality Cluster Workshop

Barcelona, November 22, 2016

Elías Pérez

Quobis Networks

ABOUT QUOBIS

- Founded in 2006 as a VoIP system integrator.
- No VCs, privately held
- Addressing the software service provider market.
- HQ in Spain, worldwide sales through partners.
- Small size (~25 engineers).

Quobis® leading your ITinerary

ABOUT WebRTC

WebRTC

... is an opensource project that makes possible to manage multimedia communications in the web browsers, using simples API's in Javascript, in a native way.

Opensystems, with no propietary implementations

¡No plugins!

Multi-platform... and multi-device!

Quobis
leading your ITinerary

QUOBIS and WebRTC

We play a key-role in WebRTC industry, working on topics like standardization and dissemination in different groups and events:

Co-authoring different standards and drafts, like the RFC7118 standard for SIP over Websockets, SIPoWS

Quobis is member of the ATIS DSI initative, which is leading the ORCA.js API to be exposed by telcos

Quobis' is co-chairing the SIP Forum WebRTC Task Group, whose objective is to enable of WebRTC for SIP-based domains

Authors of QoffeeSIP, an opensource Javascript stack for WebRTC

QUOBIS proposal for WebRTC

- **We're focused on reducing the complexity of the deployment of WebRTC applications and clients by telcos and enterprises**
- **Our solutions interops & complement the offering of leading vendors in the telecom space**

WEBRTC APPLICATIONS

Web collaboration, click to call, net Apps connectors, ad-hoc applications, etc.

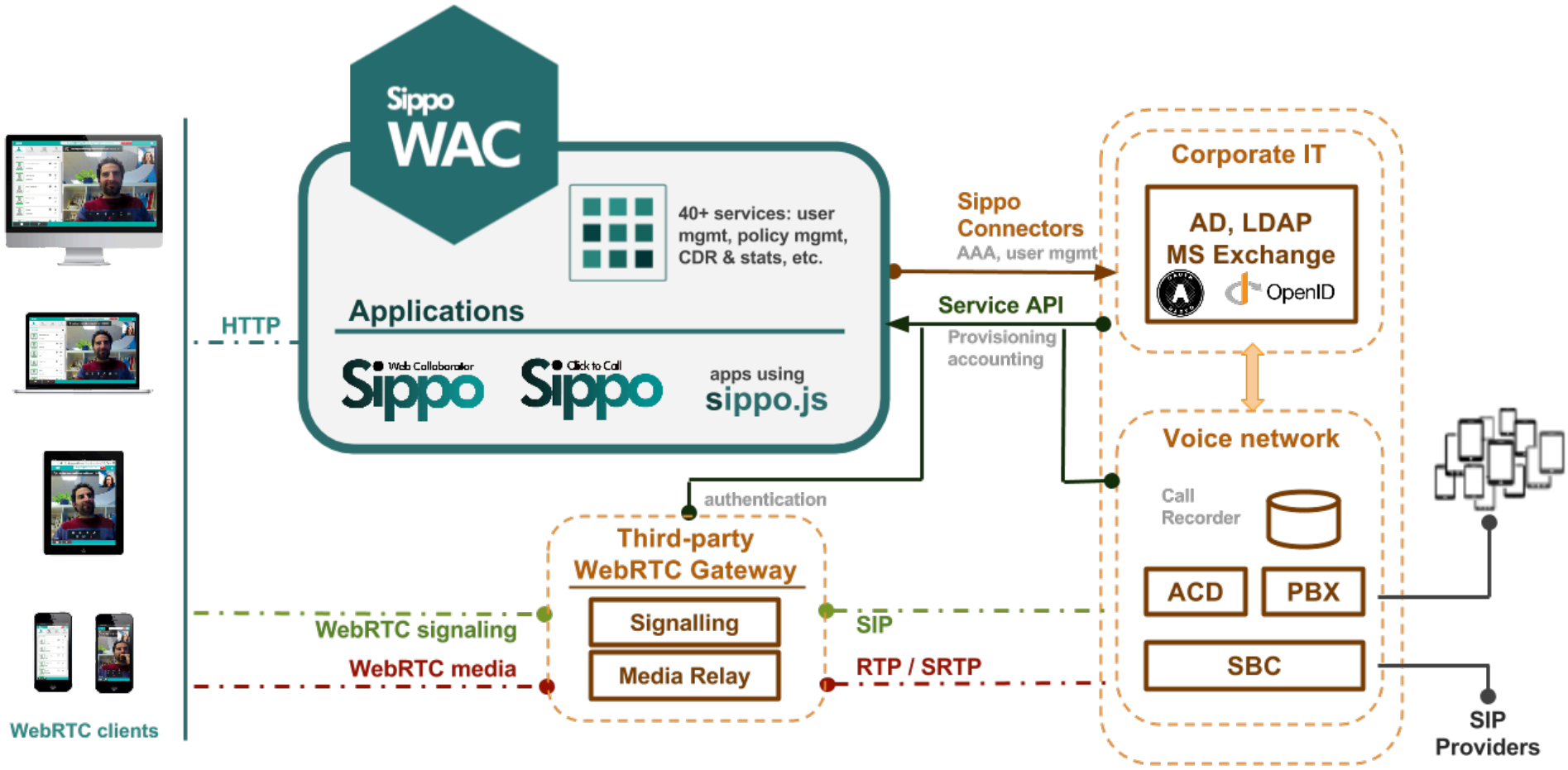
Sippo

WEBRTC APPLICATION CONTROLLER

Software based solution to abstract interconnection complexity, provide a complete set of API to develop applications fully interoperable with legacy architecture.

Quobis leading your ITinerary

Sippo WebRTC Application Controller



The role of Sippo WebRTC Application Controller

Sippo

The WAC enables the integration of browser-based real-time services with existing IMS/NGN or UC networks

1. Hides complexity of different implementations of WebRTC by **browsers**, including those that need a plugin to support WebRTC. Provides hybrid applications for smartphones like **Android and iOS**

2. Manages different **signaling** protocols (SIPoWS, JSON, propietary APIs, etc) to being able to use **any industry WebRTC gateway**

3. As a host of WebRTC applications, provides **security mechanisms** to avoid traditional VoIP attacks and pure web and **WebRTC threads**

Quobis leading your ITinerary

The role of Sippo WebRTC Application Controller

4

WebRTC applications are developed on top of a **orca.js compatible API** called sippo.js, available for 3rd parties that want to create applications.

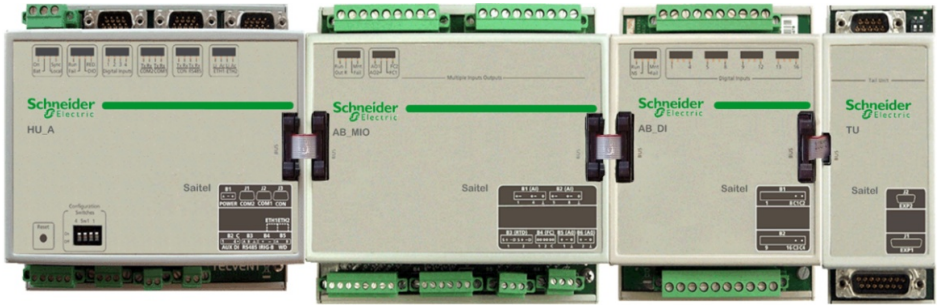
5

Manages **interconnection with existing systems** for user management (authentication, privileges, accounting, policies, etc) via a Service API and different Sippo connectors with well-known solutions like LDAP, MS Exchange, leading HSS, etc.

6

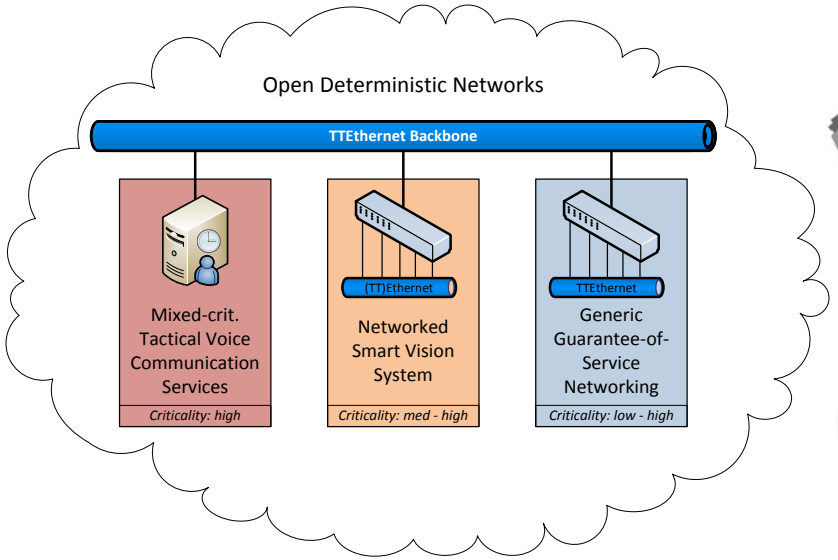
Makes multi-tenancy a reality, exposing different applications to corporate or residential customers of service providers. Includes **statistics**, easy to adopt **management tools** and **customization** functionalities,

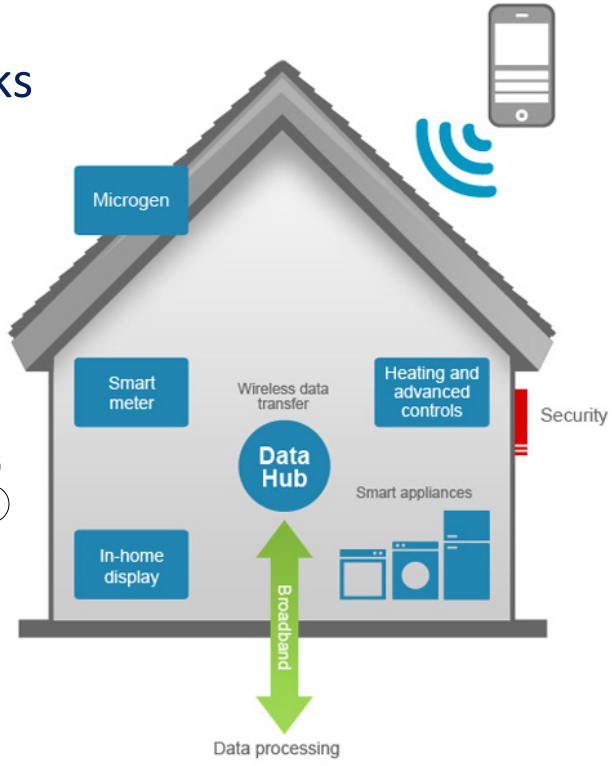
Internet of Things & IT infrastructure Motivation in EMC2



Living Lab Internet of Things

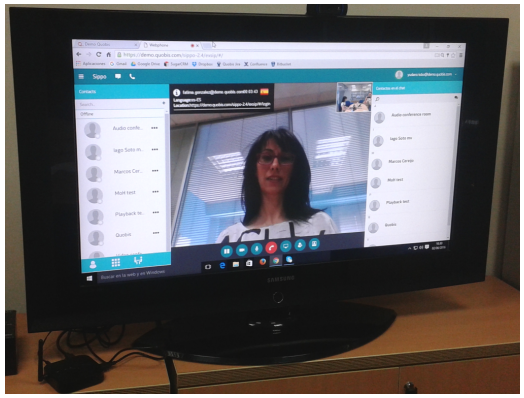
- Multimedia communications
- Open deterministic networks
- Autonomic home networking
- Ultralowpower high datarate communication
- Synchronized low-latency deterministic Networks





Internet of Things & IT infrastructure
Multimedia communication

- Address large-scale application of UC Services web-based on Embedded Systems.
- Main goal is to enable audio or video communication, images, files and data transfer through web-based applications on any type of small embedded systems, to have the possibility to adapt these systems to the new paradigm where the web browser is going to be the player.
- **Multimedia processes** distribution over multicore CPUs



Quobis
leading your ITinerary

MINIX NEO X7 Mini

- Released in September 2013.
- This element is part of the family of Android TVs (linked with elements like HDMI dongles, AppleTV or ChromeCasts).
- It runs an Android 4.2.2
- HDMI interface with 1080p HD video.
- Supports mouse, keyboard, camera and microphone.
- Video processing capacities to deal with video contents over WebRTC.

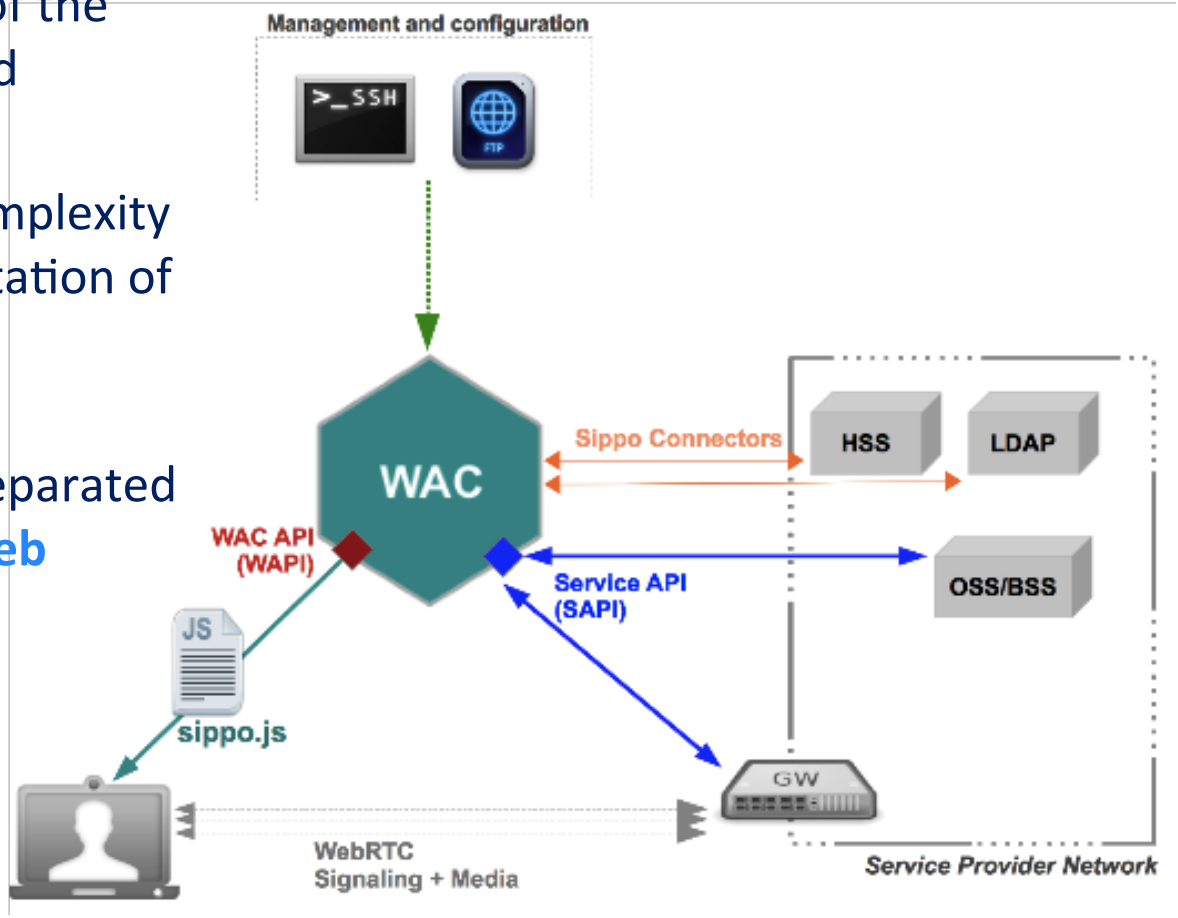
MINIX MEO X7 Mini features

Processor	Quad-Core Cortex A9 Processor
GPU	Quad-Core Mali 400
Memory	2GB DDR3
Internal Storage	8GB NAND Flash
Wireless Connectivity	802.11n Wi-Fi, Bluetooth 4.0
OS	Android™ Jelly Bean 4.2.2

Quobis
leading your ITinerary

Use case architecture

- The WAC solves part of the complexity of a real field implementation.
- The WAC hides the complexity of the existing fragmentation of devices, browsers and interconnection.
- Media processing is separated in parallel sources -> **Web Workers**



Web workers

- Defined by the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).
- **Web Workers** are scripts that are not interrupted by user-interface scripts (scripts that respond to user interactions).
- Web workers are able to utilize **multi-core CPUs** more effectively in the multimedia domain.
- Keeping such workers from being interrupted by user activities allow our use case to remain responsive to audio and video from users at the same time as it is running **critically data tasks**.
- The W3C and the WHATWG are currently in the process of developing a definition for an API for web workers.

Internet of Things & IT infrastructure
Multimedia communication

A real application: eHEALTH

- Communications between hospitals, emergency vehicles and patient portals at home.

- Interoperable collection of information from devices

- Wireless 3G/4G communication

- Sources: audio-video data, point-of-care device data and patient medical history data.

Many thanks!

Elías Pérez, Quobis
elias.perez@quobis.com
www.quobis.com

Quobis
leading your ITinerary