EMC^2

Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments

- an ARTEMIS 2013 innovation pilot project (AIPP5)

Vienna, Austria, January 21-22, 2014

at workshop:
‘Integration of Mixed-Criticality Subsystems on Multi-Core and Many-Core Processors’
hosted by the HiPEAC 2014 conference

Knut Hufeld
Infineon Technologies AG
Knut.hufeld@infineon.com
+49 89 234 52653

Bernd Koppenhöfer
Airbus Defence and Space
Bernd.Koppenhoefer@cassidian.com
+49 731 392 5354

General Overview

Embedded Multi-Core Systems for Mixed Criticality Applications in dynamic and changeable Real-time Environments

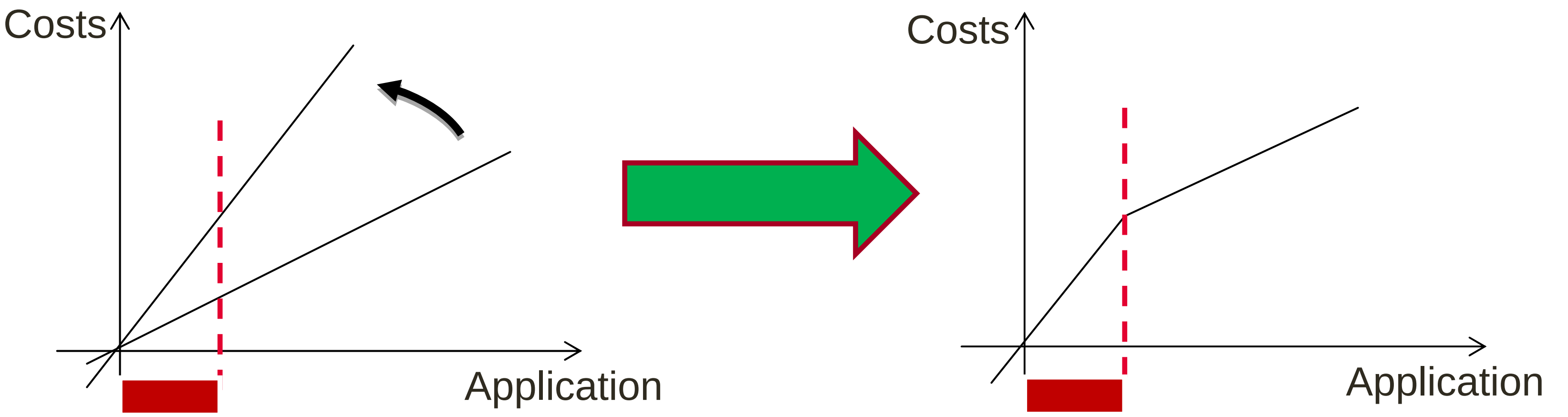
Duration:	3 years , start: April 1st 2014
Total budget:	100 Mio €
Total resources:	800 person years
Co-ordination:	Infineon Technologies AG
Consortium size:	98 from 16 European countries + Israel
Largest ARTEMIS-Project	
Contact:	Knut Hufeld, knut.hufeld@infineon.com, +49 (0)89 234 52653

Problem Overview

Costs are increasing due to ***critical / secure / power / reliability*** requirements more than it will be economically feasible

The number of ***ECUs*** in a car cannot (economically) ***grow*** -> need ***economically viable mixed criticality*** (properties) systems

System properties ***do not scale*** well – need of "intelligent" mixed criticality scenarios



Objectives

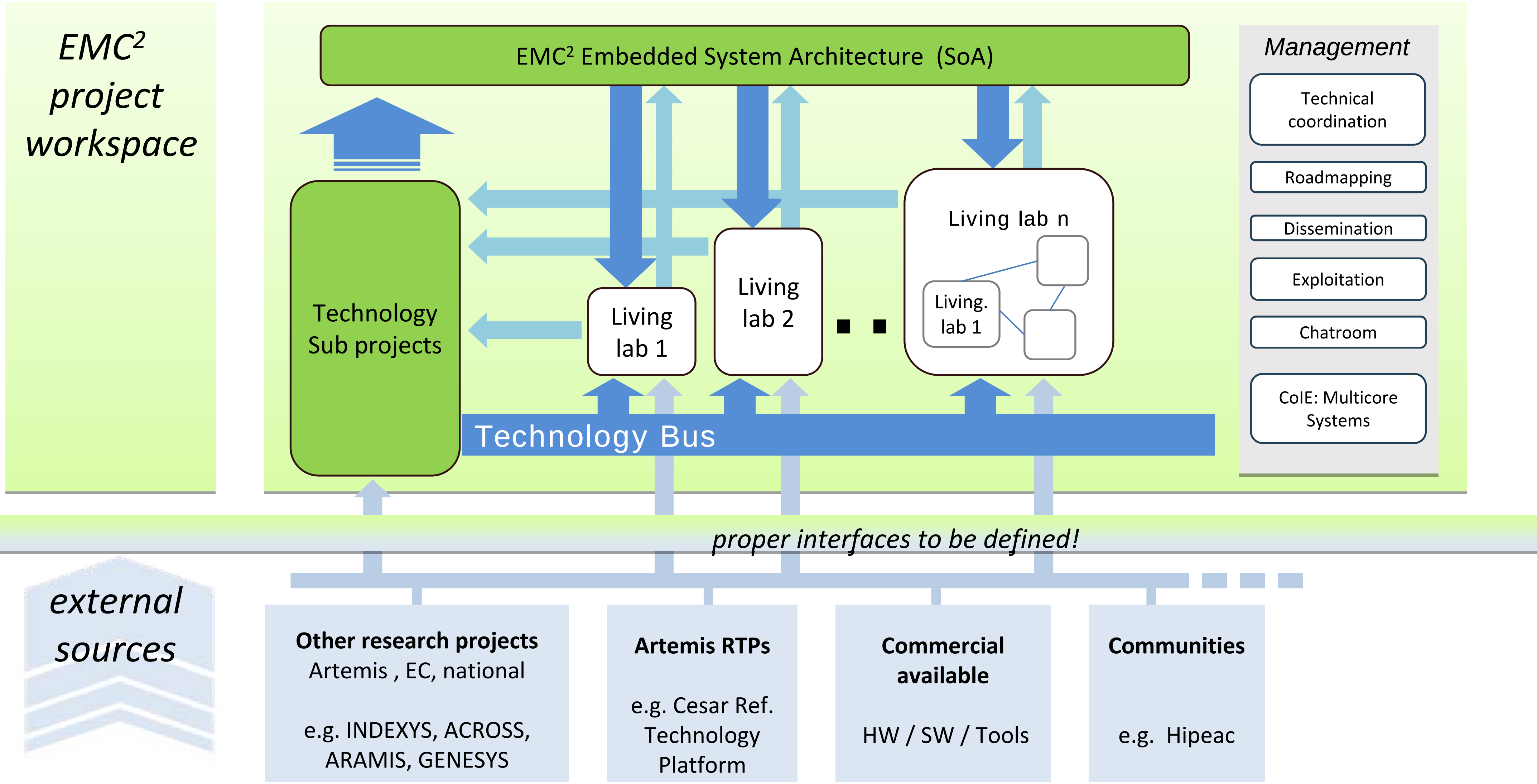
EMC^2 objectives:

- Innovative and sustainable Service-oriented Architecture
- Dynamic Adaptability in Open Systems
- Qualification and certification of Multi-core Systems
- Scalability and Utmost flexibility

  → cross-domain deployment in almost all ES-domains
- Integrated tool chains, through the entire lifecycle

..to foster embedded multi-core-based systems for mixed criticality applications..

..a look at the project structure..



EMC² project workspace

EMC² Embedded System Architecture (SoA)

Technology Sub projects

Living lab 1

Living lab 2

. .

Living lab n

Living. lab 1

Technology Bus

Management

- Technical coordination
- Roadmapping
- Dissemination
- Exploitation
- Chatroom
- CoIE: Multicore Systems

proper interfaces to be defined!

external sources

Other research projects
Artemis , EC, national

e.g. INDEXYS, ACROSS, ARAMIS, GENESYS

Artemis RTPs

e.g. Cesar Ref. Technology Platform

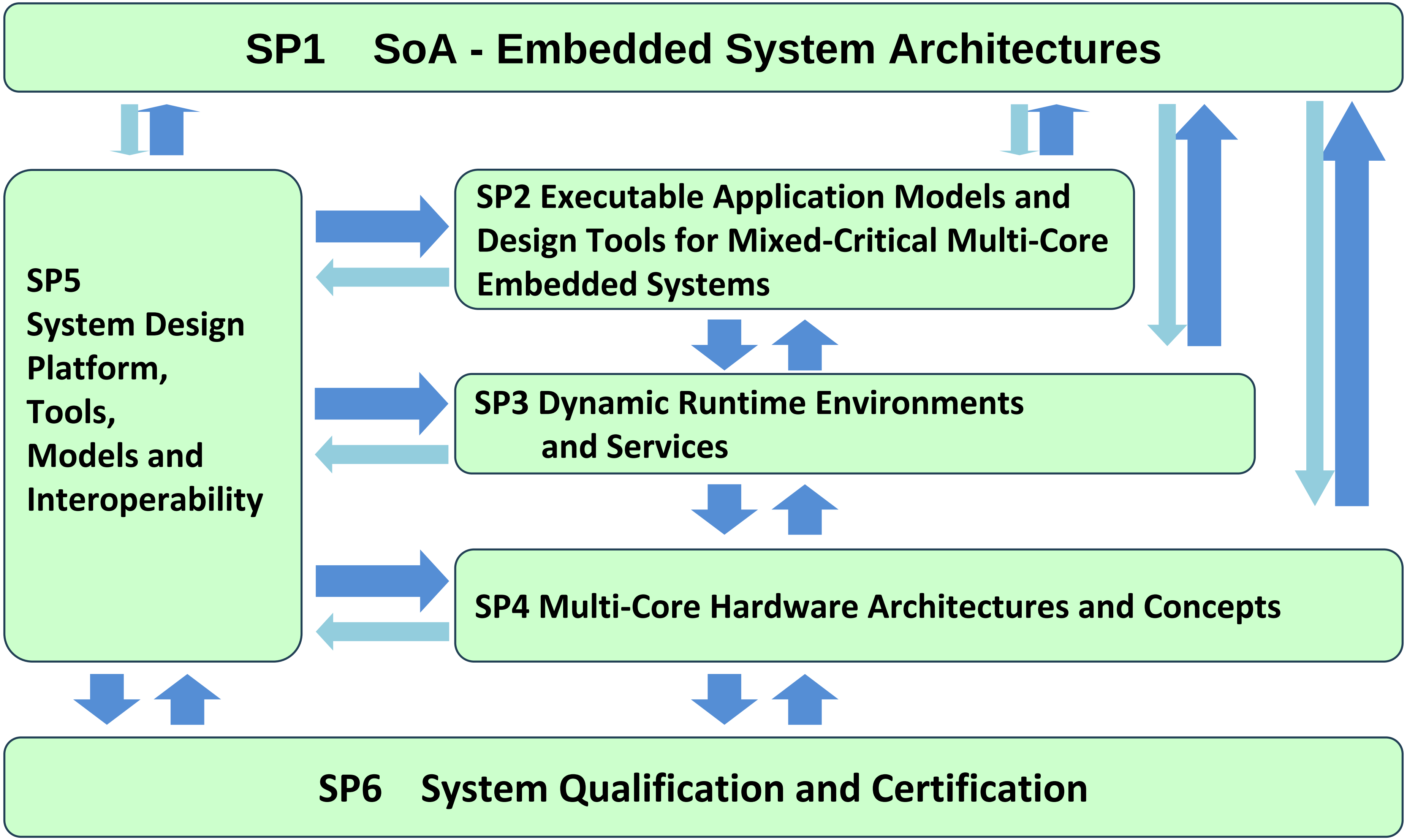
Commercial available

HW / SW / Tools

Communities

e.g. Hipeac

Project Structure
Technology Subprojects (SP1-6)



SP1 SoA - Embedded System Architectures

SP5
System Design
Platform,
Tools,
Models and
Interoperability

SP2 Executable Application Models and Design Tools for Mixed-Critical Multi-Core Embedded Systems

SP3 Dynamic Runtime Environments and Services

SP4 Multi-Core Hardware Architectures and Concepts

SP6 System Qualification and Certification

Project Structure
Living Labs (LL1-6)

LL1 Automotive applications

LL2 Avionics applications

LL3 Space applications

LL4 Industrial manufacturing and logistics

LL5 Internet of things

LL6 Cross Domain applications

Project Structure
Living Labs (LL1-6)

LL1 Automotive applications

LL2 Avionics applications

LL3 Space applications

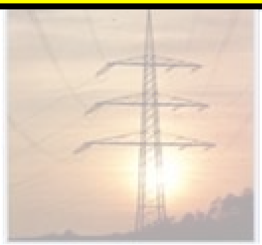
LL4 Industrial manufa

LL5 Internet of things

LL6 Cross Domain applications

Partners:

- Airbus Group,
- Fraunhofer Institute for Experimental Software Engineering (IESE),
- Instituto Superior de Engenharia do porto (ISEP),
- Technische Universitaet Braunschweig,
- Technische Universitaet Kaiserlautern.



EMC^2 Avionics Application (Airbus Group) Motivation

“Could bad code kill a person?”

2013 a big car manufacture accepted settlement to avoid punitive damages at an US court:

- Case: An inadvertent acceleration of one of the manufacturer's vehicles caused an accident in September 2007 that killed one woman and seriously injured another.
- The case was one of several hundred.
- Analysis of electronic throttle control system SW showed:
  - The source code is defective, and contains bugs -- including bugs that can cause unintended acceleration.
  - SW fail safes are defective and inadequate
    - A single bit flip is sufficient to cause an unintended acceleration.
    - The driver has no means to override this → drive loses control of engine speed.
    - The system does not provide any means to reliably detect the problem.
- Some of the identified root causes:
  - No proper SW development process → Complex source code with effectively infinite test space
  - No formal safety process → Some single point of failure were not identified.

Source: EE Times,”Single Bit Flip That Killed”, Junko Yoshida, 10/25/2013, www.eetimes.com
http://www.safetyresearch.net/Library/BarrSlides_FINAL_SCRUBBED.pdf

CS-25: **Regulation** for Certification of „Large Aeroplanes“
issued by European Aviation Safety Agency (EASA)

CS-25.1309:

(a) The aeroplane equipment and systems must be designed and installed so that:

(1) Those required for type certification or by operating rules, or whose improper functioning would reduce safety, ***perform as intended*** *under the aeroplane operating and environmental conditions.*

(2) Other equipment and systems are not a source of danger in themselves and ***do not adversely affect the proper functioning*** *of those covered by sub-paragraph (a)(1) of this paragraph.*

(b) The aeroplane systems and associated components, considered separately and in relation to other systems, must be designed so that -

(1) Any catastrophic failure condition

(i) is extremely improbable; and

(ii) ***does not result from a single failure****; and*

(2) Any hazardous failure condition is extremely remote; and

(3) Any major failure condition is remote.

Failure must not occur during the entire operational life of an entire system or fleet
(typically 10^9 flight hours)
→ Probability of less than 1 failure every 114.000 years

Source: CS- 25, “Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes”, Amendment 14, 19 December 2013

Federal Aviation Authority (FAA) Research

- Most complex hardware, including COTS microprocessors, goes through a process of demonstrating safety through the complete verification of the hardware design.
- This process is infeasible for some complex, nondeterministic COTS microprocessors.
- These microprocessors **should be assumed as potentially unsafe**, and system-level approaches for risk mitigation should be considered such as a **safety net.**

Employment of mitigations and protections at the appropriate level of aircraft and system design to help ensure continuous safe flight and landing.

- Safety device within the aircraft
- Protects against unexpected behavior, damage, injury, and instability
  - over the service life outside, or
  - at a level above the device itself.
- Multilevel approach

Source: Report DOT/FAA/AR-11/2, "HANDBOOK FOR THE SELECTION AND EVALUATION OF MICROPROCESSORS FOR AIRBORNE SYSTEMS", February 2011

Source: Research Report DOT/FAA/AR-11/5: Microprocessor Evaluations for Safety-Critical, Real-Time Applications: Authority for Expenditure No. 43 Phase 5 Report, May 2011

Multi-**C**ore **F**or **A**vionics (MCFA) Working Group

Launched on the initiative of semiconductor manufacturer Freescale.

Addressing the challenges related to the introduction of multi-core processors (MPCs) in the certified avionics domain.

MCFA had several meetings with certification authorities EASA and FAA.

- Discuss the challenges of certifying MPCs in avionics applications
- Exchange of viewpoints
- Feedback from industry to authority point of view on multi-core issues.

Certification Authority Concerns on MPCs

- MCPs are highly complex COTS SOCs, many of which appear to be primarily designed for speed, not specifically for safety, integrity or deterministic behaviour.
- Most MCPs include features to speed up and control data transfers that are not on single core processors.
  - E.g. shared cache, shared memory, hypervisors, also ‘coherency fabrics/modules’ that control access to memory and peripherals of the MCP.
- Some complex features were not developed to DO-178B or DO-254 so can’t be thoroughly tested and could therefore include unintended functionality.
- These features and contention for processor resources from software of two cores can lead to cache problems, data jitter, large worst-case execution time (WCET) increases (e.g. 181%), denials of access to peripherals *).

Authority guidance at present is limited to MCP installations with only two cores activated and with software applications from one system only.

*) *Moscibroda, T. and Multu, O., “Memory Performance Attacks: Denial of Memory Service in Multi-Core Systems,” Proceedings of the 16th USENIX Security Symposium, 2007, pp. 257-274.*

Objectives for Certification of Safety Nets

- Verify that the means of detecting and handling errors, including their MCP safety net, are capable of detecting and handling the errors they are intended to detect.
- If the MPC contains (at least partially) a fail-operative safety-critical functionality, **a means has to be implemented to provide that functionality even after errors are detected in the MCP or its hosted software.**

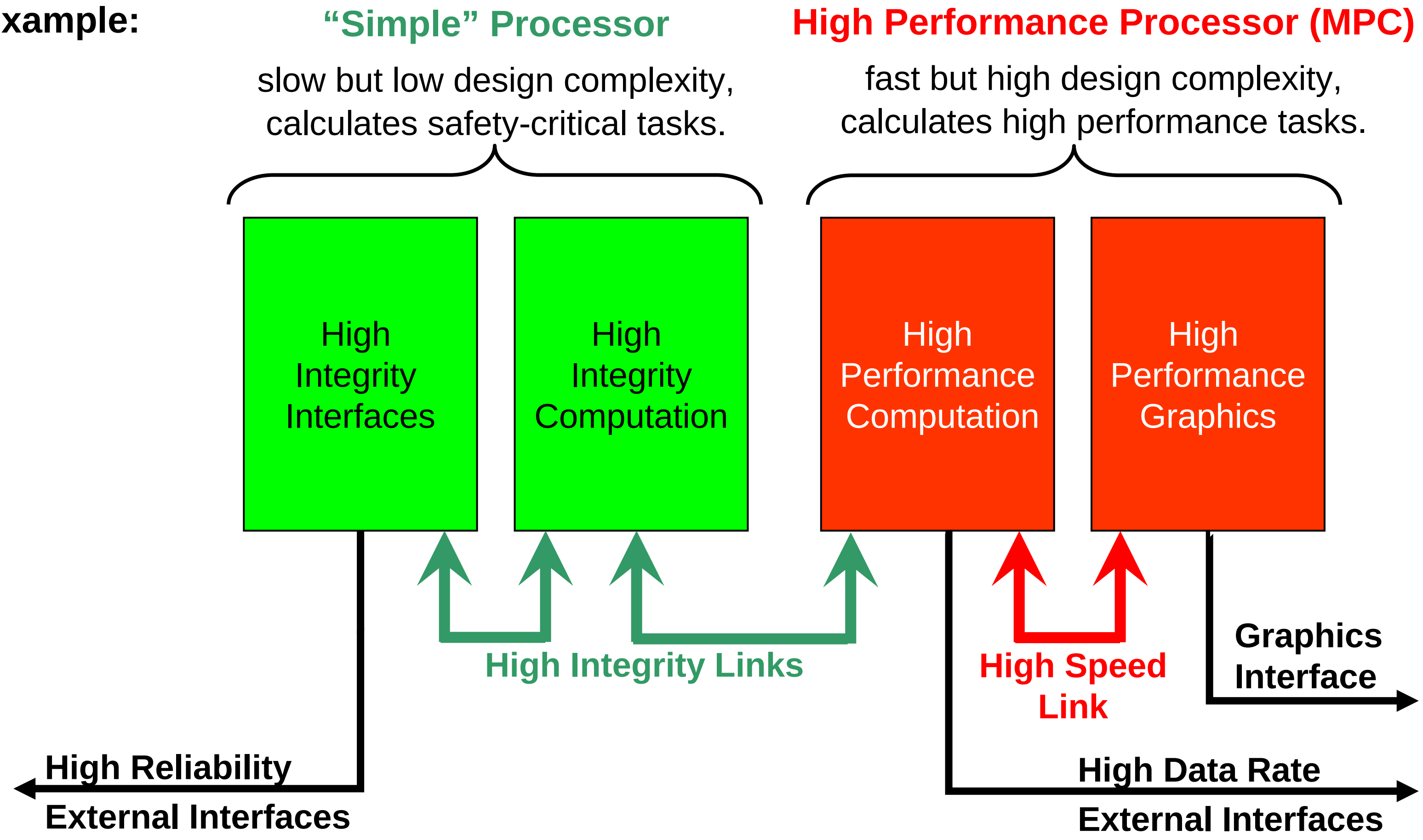
In other words: We need at least 2 **independent** sub functions

Implications for Airbus Group Avionics Application

Goal: To demonstrate a solution for the Objective “**a means has to be implemented to provide that functionality even after errors are detected in the MCP or its hosted software**”.

How: Investigate Safety Net on a Hybrid Avionic Integrated Architecture.

Chosen Example:



“Simple” Processor
slow but low design complexity, calculates safety-critical tasks.
High Performance Processor (MPC)
fast but high design complexity, calculates high performance tasks.
High Integrity Interfaces
High Integrity Computation
High Performance Computation
High Performance Graphics
High Integrity Links
High Speed Link
Graphics Interface
High Reliability External Interfaces
High Data Rate External Interfaces

Airbus Group Avionics Application

Display-oriented application

- Converts multiple sensor data sources.
- Generates surveillance map to assist pilot.

Hybrid Platform

"Simple" Processor:

- High criticality but low performance.
- Provides essential information to the pilot.

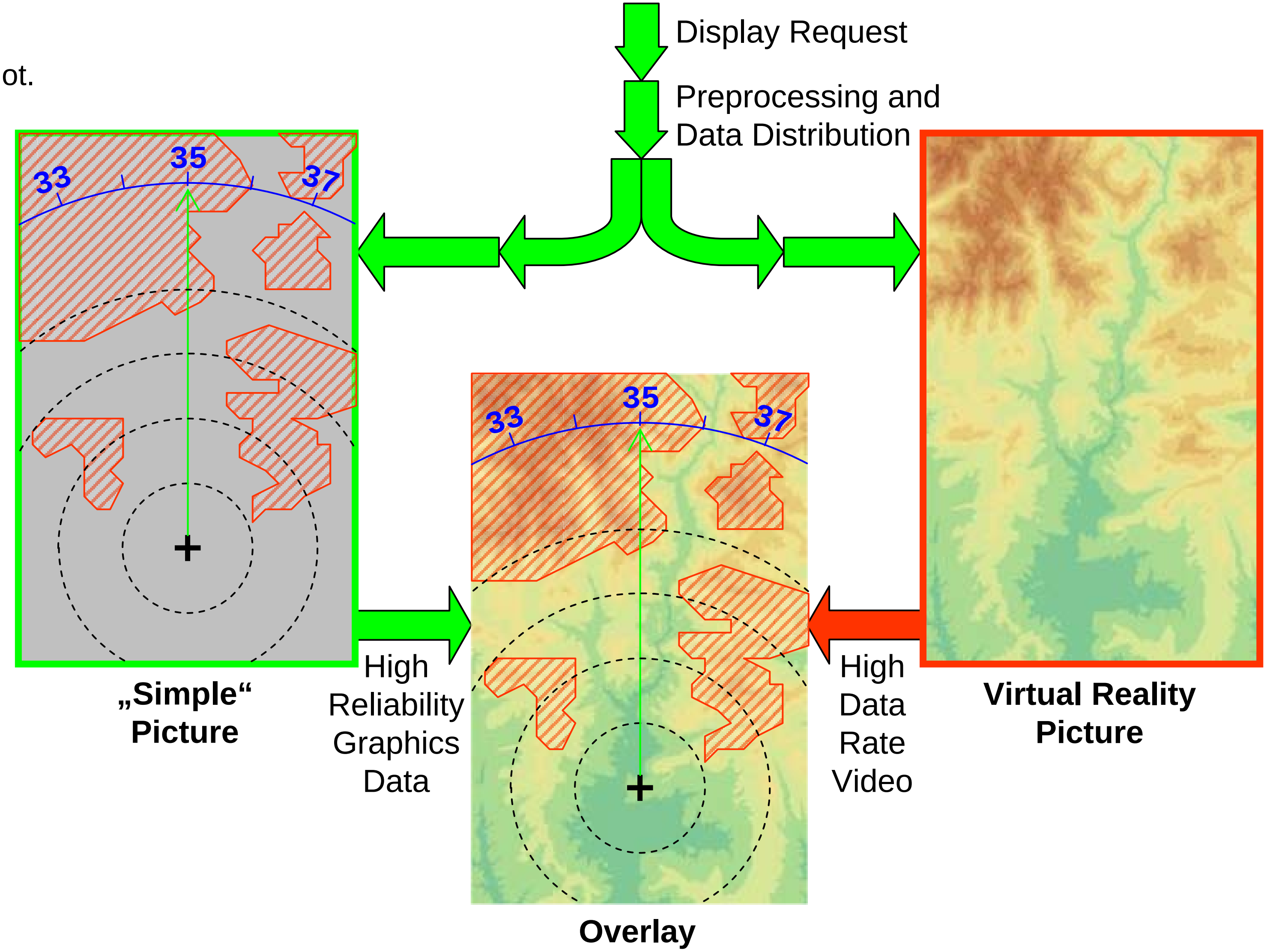
High Performance Processor:

- Low criticality but high performance.
- Multi-core / Graphics Processor.
- Computes complex and enhancing visual elements.

Pilot is able to decide on correctness based on the high-criticality output.

Voting implemented by

- Human comparison (pilot) or
- Dedicated hardware.



Thank you for your attention!