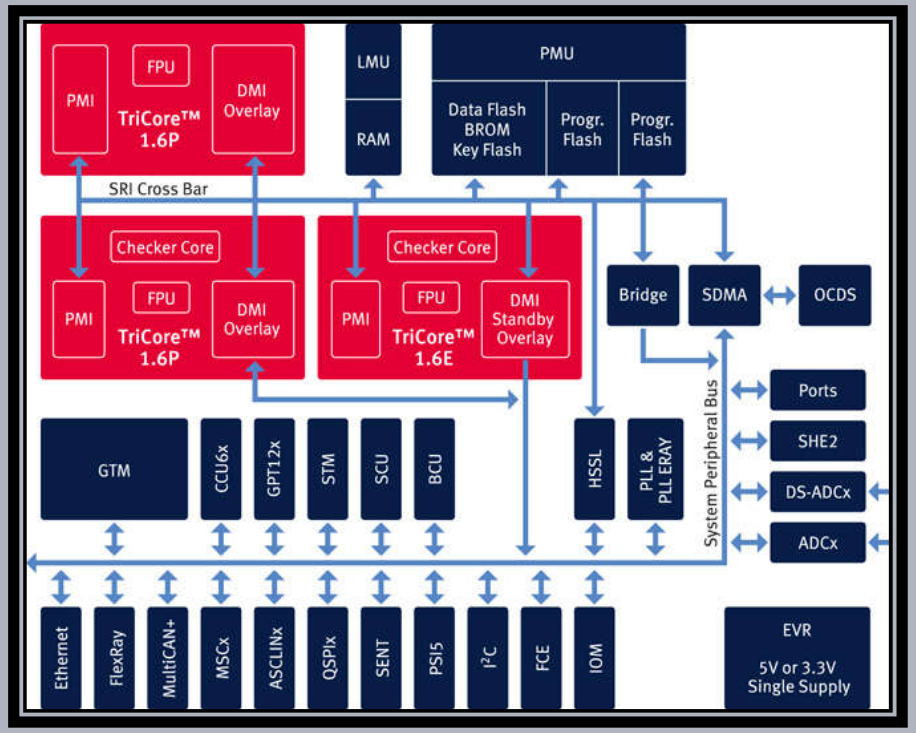
h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Functional Safety on Multicore Microcontrollers for Industrial Applications

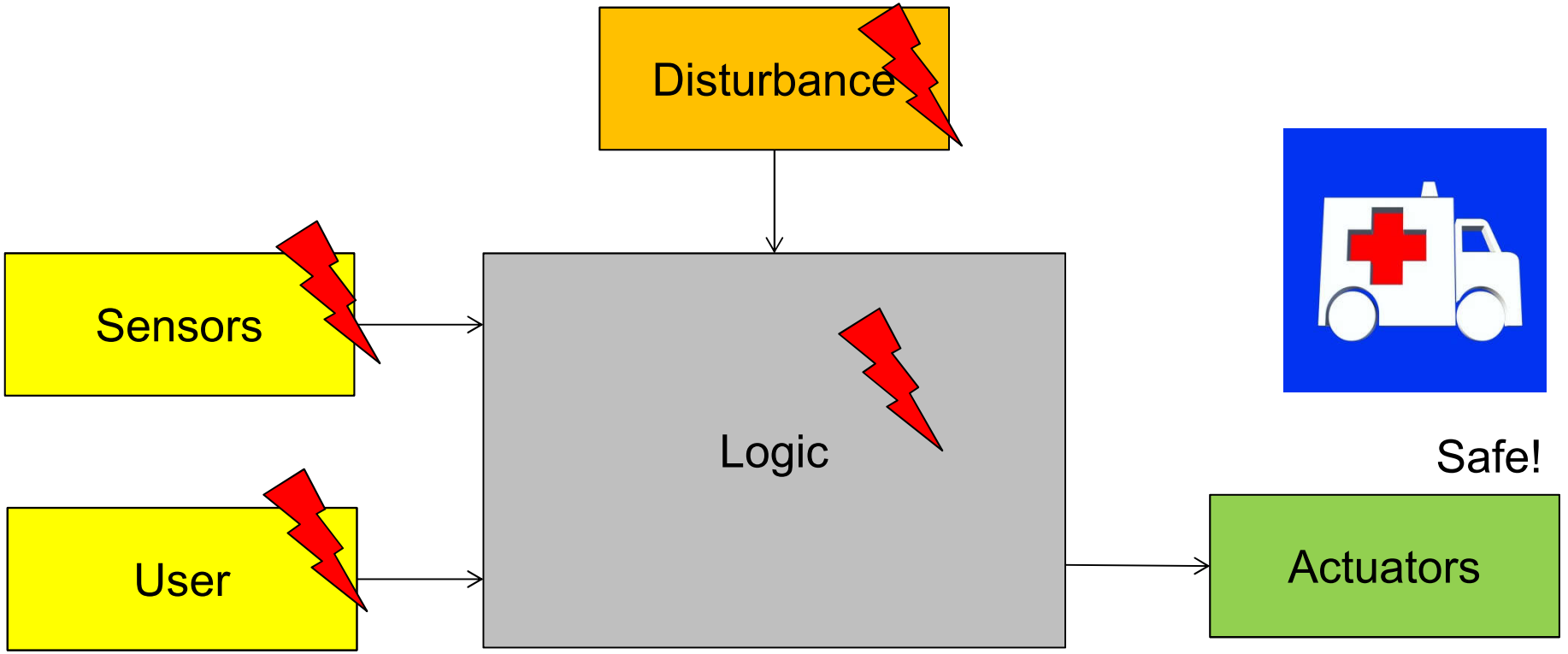
Thomas Barth (h-da)
Prof. Dr.-Ing. Peter Fromm (h-da)



Contents

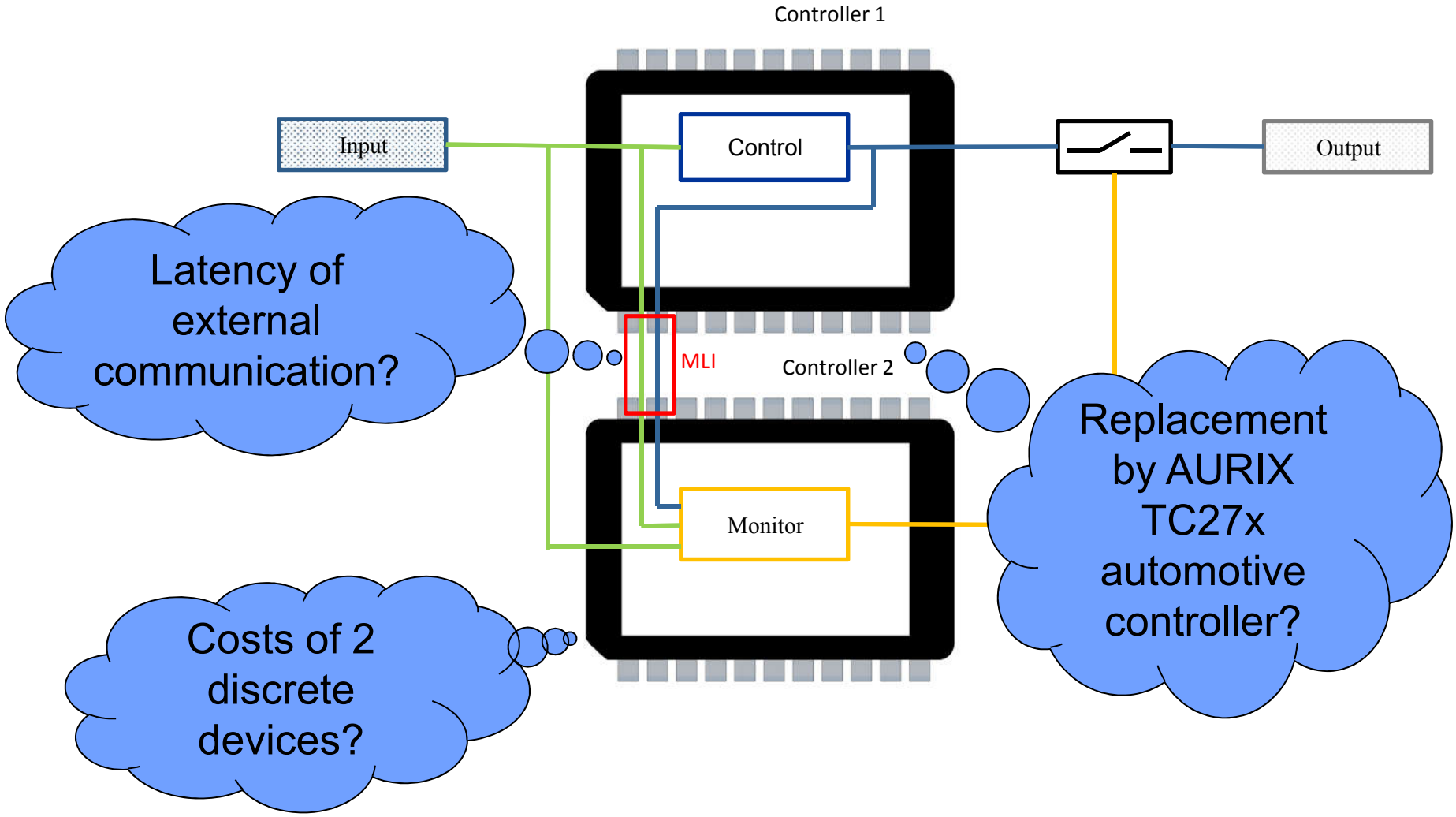
- Functional Safety
- Multicore Motivation
- ISO13849
- Implemented Software Architecture
- Hardware Features for Freedom from Interference
- Error Handling
- Conclusion and Outlook

Funktionale Safety – Overview



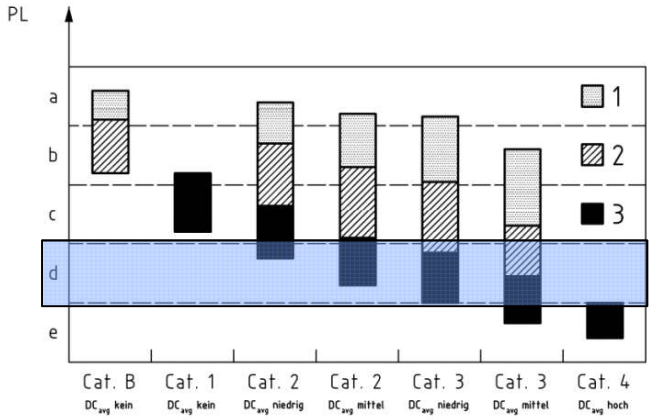
The objective of Functional Safety is freedom from unacceptable risk of physical injury or of damage to the health of people either directly or indirectly (through damage to property or to the environment).

Current, redundant architecture



ISO 13849

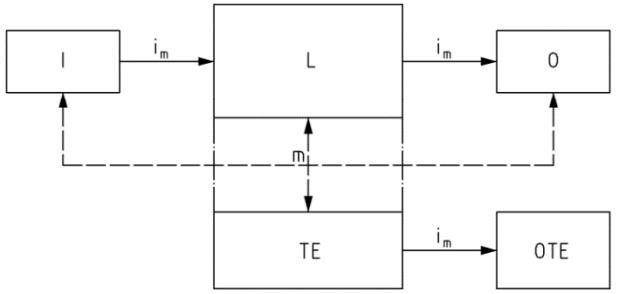
- Dependent on the possible consequences of failure, a certain probability has to be achieved (Performance Level/PL)
- Given application requires PL “d”



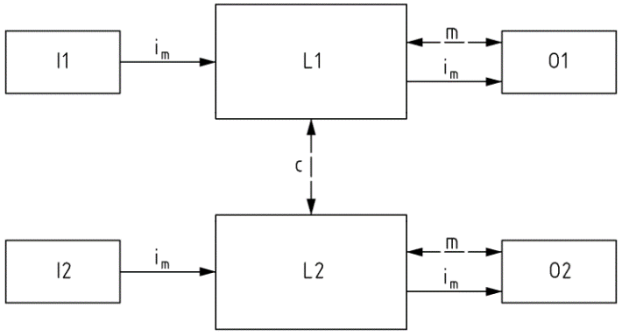
PL selection chart *

Legende

PL	Performance Level
1	$MTTF_d$ jedes Kanals = niedrig
2	$MTTF_d$ jedes Kanals = mittel
3	$MTTF_d$ jedes Kanals = hoch



Category 2 Architecture*



Category 3 Architecture*

* Source: ISO 13849

Freedom from Interference

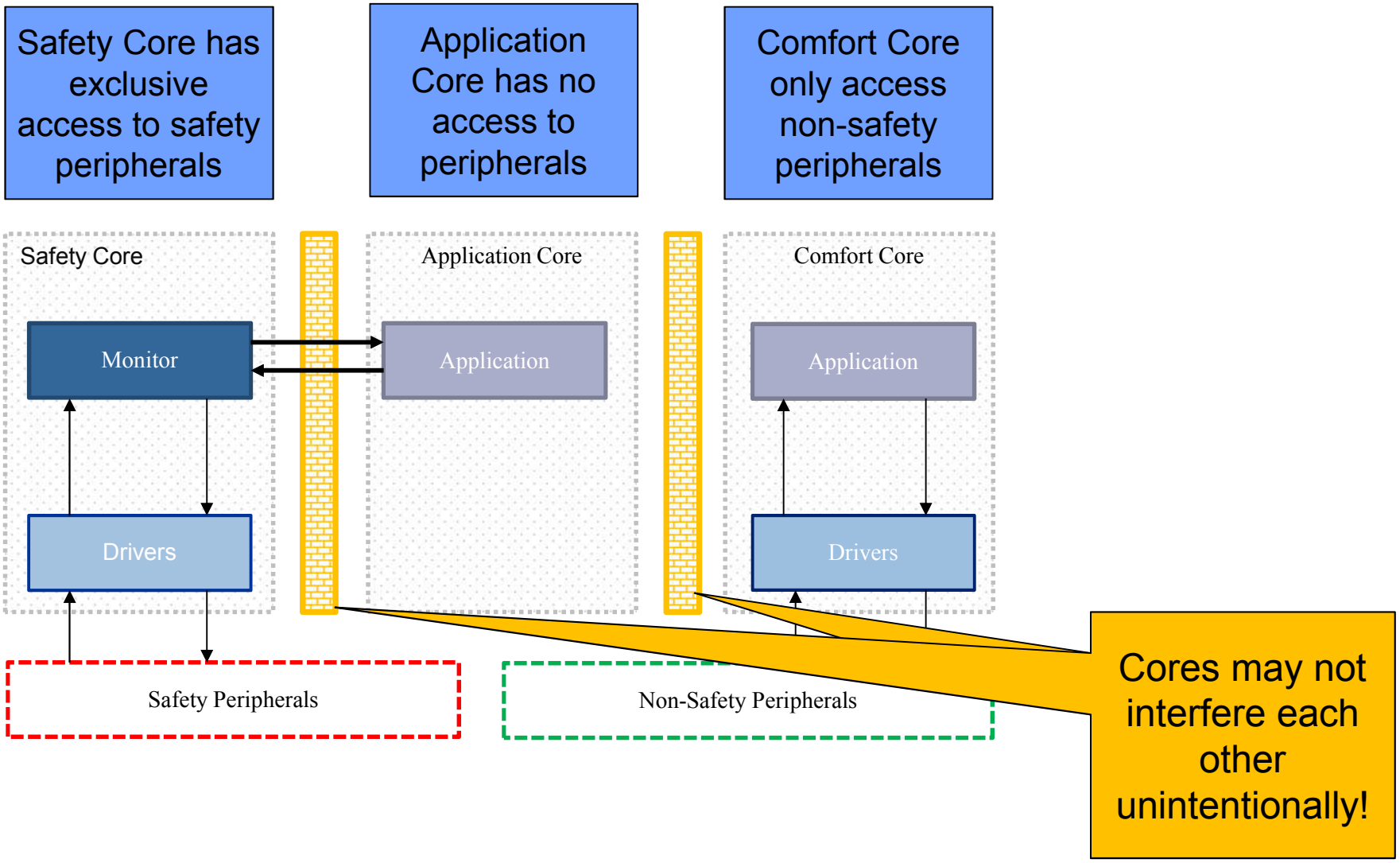
The utilization of multicore microcontrollers requires freedom from interference among the cores. Freedom from interference has to be achieved in:

- Data Domain
  One Core may not corrupt the data of other cores.

- Resource Domain
  Cores may not corrupt the configuration/inputs/outputs of shared resources/peripherals. Safety critical peripherals like the bus or PLL have to be protected against unintentional reconfiguration.

- Time-Domain
  Cores may not consume too much time blocking resources and the applications may not consume too much CPU time. When interacting with other cores, the timing behavior has to remain consistent.

Implemented Software Architecture



Available features for Freedom from Interference on the AURIX

Data Domain
- Memory protection unit (MPU)
- Bus protection unit (BUS MPU)

Time Domain
- Core separation
- Internal watchdog
- Safety watchdog

Resource Domain
- Register access protection (RAP)
- Memory protection unit (MPU)
- Privilege level
- ENDINIT and S(afety)ENDINIT signals

Diagnostics / Exception Handling
- Safety Management Unit (SMU)
- TriCore error detection

Álvaro Salvá CC BY-NC-SA 2.0

CPU-MPUs with RTOS

The primary feature for freedom from interference are the CPU-MPUs. Every Core is equipped with an on CPU-MPU which is monitoring outgoing communication and is able to distinguish between read/write/execute access.

CPU-MPU configuration is managed by the RTOS PxROS from HighTec. Key advantages:

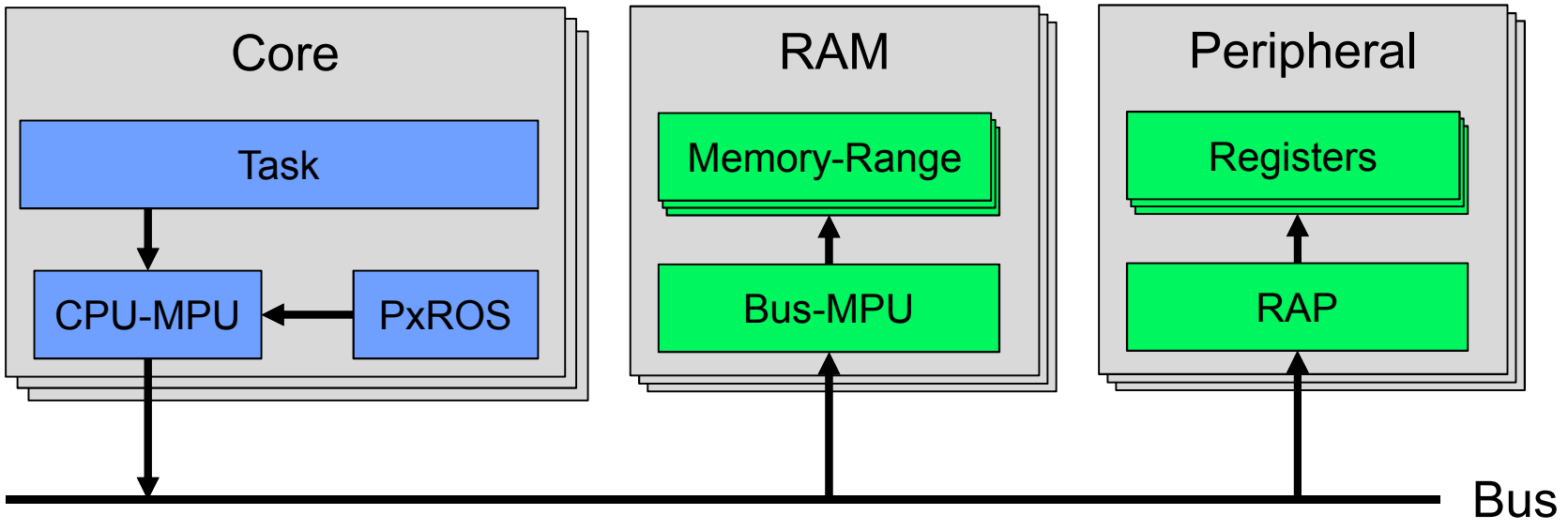
- Tasks can dynamically be assigned to different cores.
- Each Task has an own set of access rights for memory and peripherals.
- Tasks can exchange data in a non-blocking way, even between cores. I.e. the assignment of tasks can easily be reconfigured without the change of application-code.
- There is no limitation to the number of possible MPU-entries

PXROS-HR
embedded safety

Bus-MPU / Register Access Protection (RAP)

While the CPU-MPUs only monitor outgoing traffic, the Bus-MPUs and RAPs monitor ingoing traffic and allow to assign memory ranges and peripherals to certain cores. Only write-access can be limited.

Each RAM of the AURIX is equipped with an own Bus-MPU.
The granularity for peripherals depends on the module, for GPIOs each Port can be assigned.



Data- and Resource-Domain protection

The CPU-MPUs along with an Safe RTOS like PxROS have been proven to be an easy to use feature.

To complement the CPU-MPUs and to harden the system against wrong configurations, mechanisms like the Bus-MPUs and RAPs can be used. While the CPU-MPU configuration is handled by the OS in a dynamic way, the Bus-MPU and RAP configuration is more static.

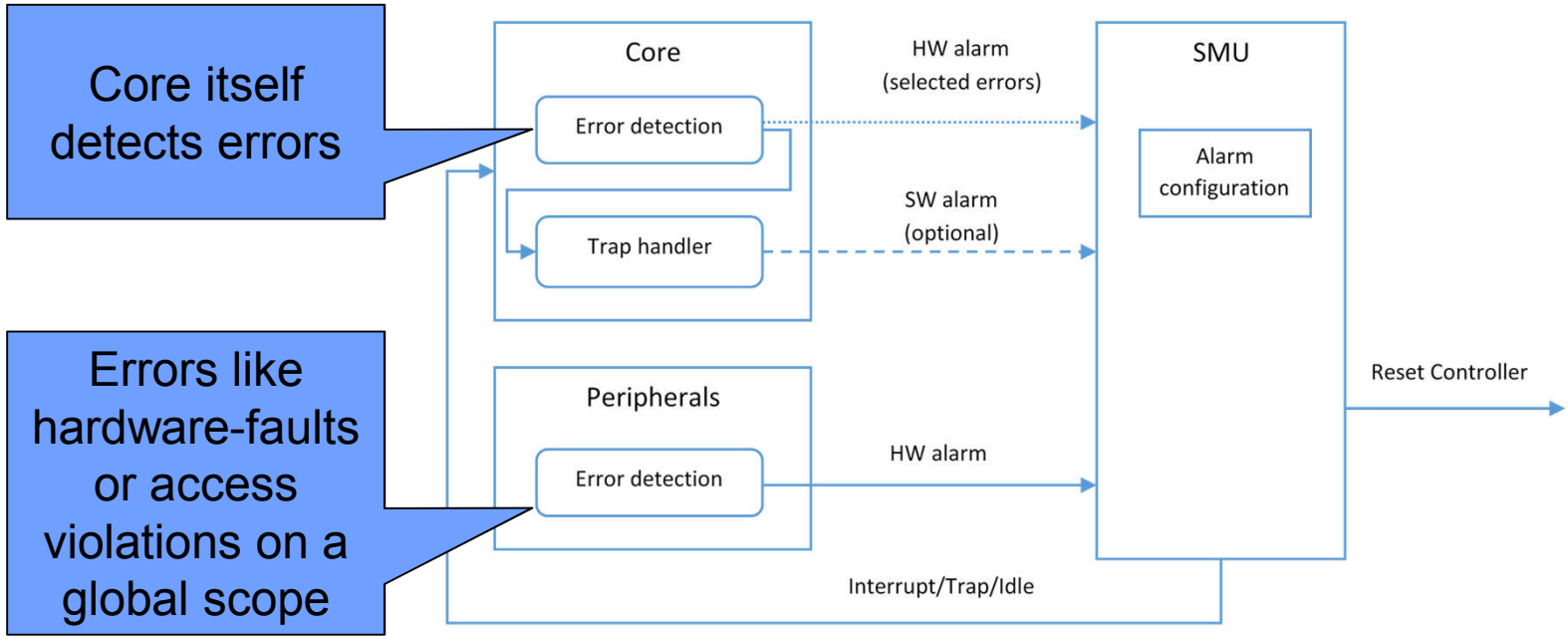
The granularity of the RAPs sometimes is not fine enough, e.g. separation of ADC channels.

Keeping the configuration of all features consistent can be challenging.

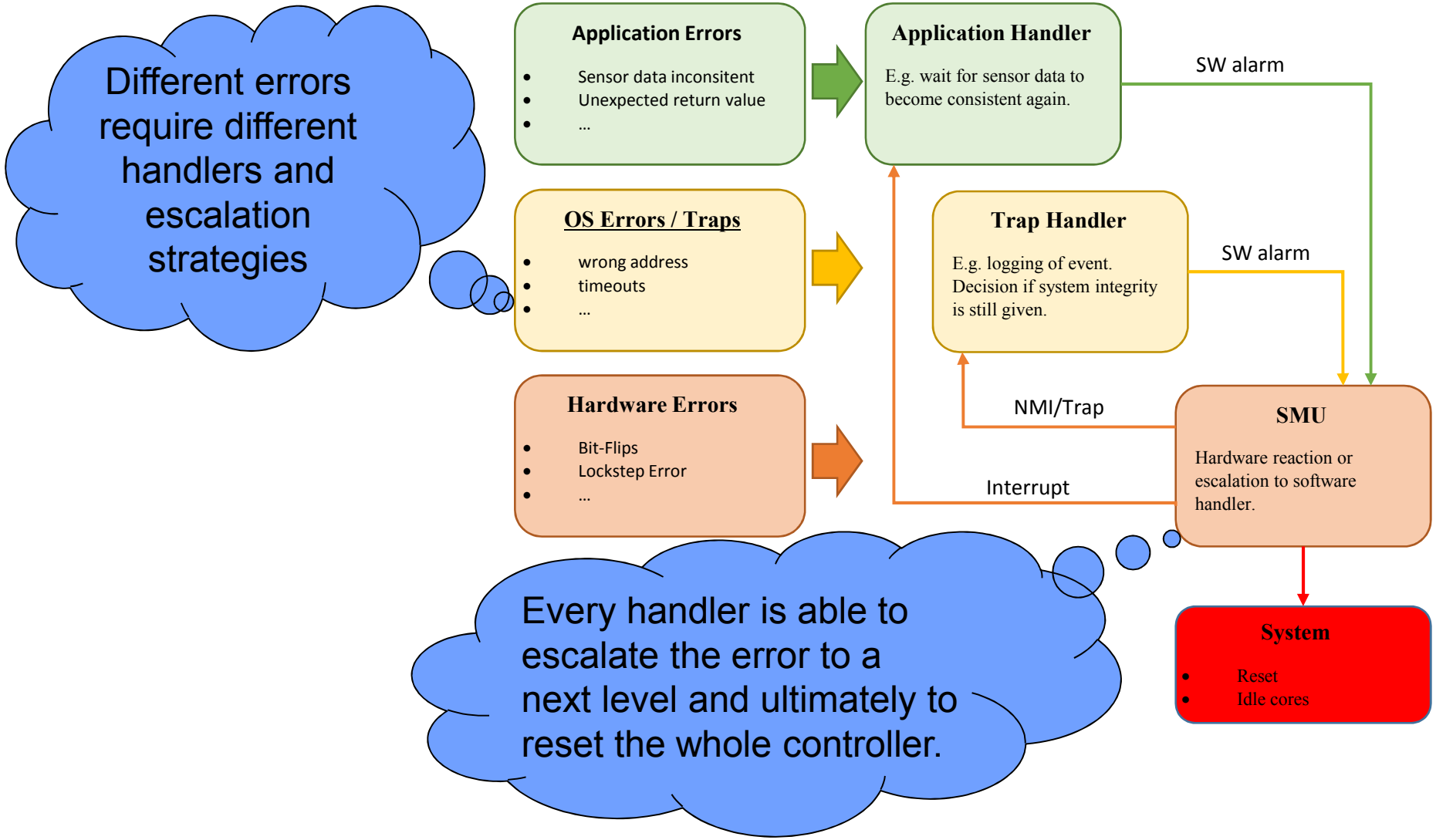
RAP can be used to secure safety critical peripherals like the PLL against reconfiguration.

Error Handling

The required PL d with a Cat. 2 Architecture requires at least a medium DC, which means that at least 90% of all possible dangerous errors have to be detected. The AURIX provides a big set of various features that are able to detect errors. Errors are reported to the SMU (Safety Management Unit) which allows a configurable reaction.



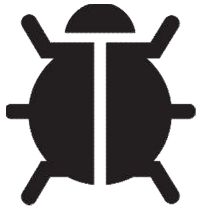
Error Handling



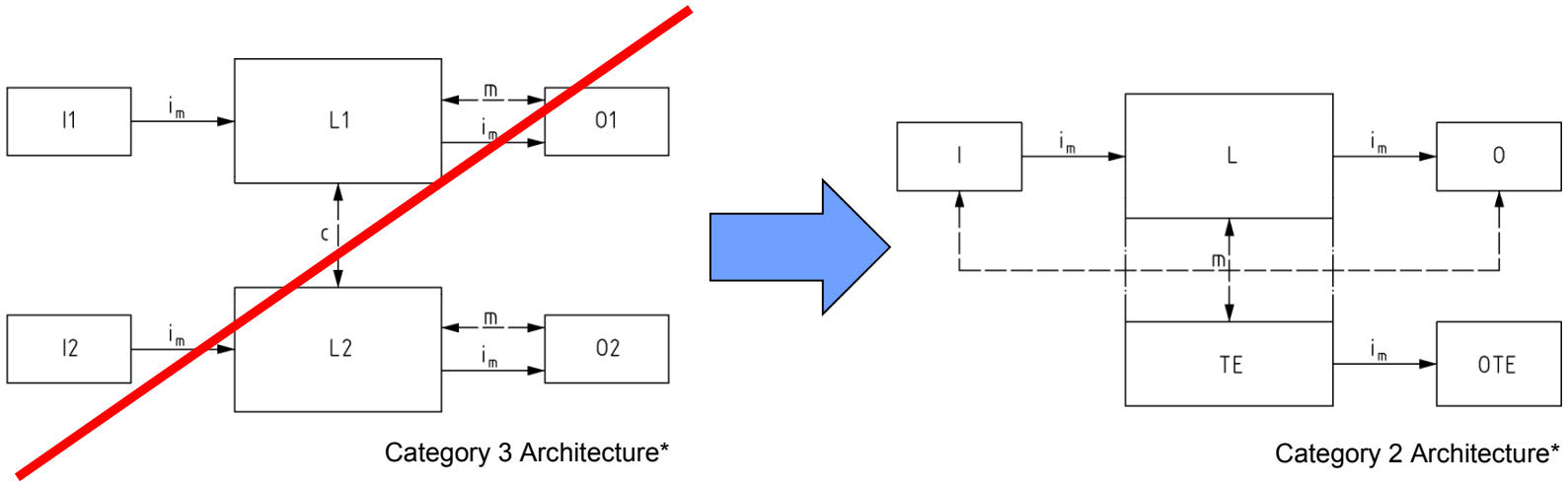
Error Handling

The possibilities for diagnostics and exception handling on the AURIX are extensive. There are ~120 available hardware error detection mechanisms plus error detection of the TriCore CPUs and software/user errors.

The design of an error escalation strategy can be challenging due to the amount of possible error sources and the individual consequence for the overall system. The configuration therefore requires very good knowledge about standards, software, hardware and the controlled machinery.



Conclusion



Category 3 Architecture*

Category 2 Architecture*

Common Cause Failures (CCF)

- Specifications of ISO 13849 regarding CCF in 2-channel systems are probably not achievable due to the close physical coupling of the cores. Therefore Investigation was focusing on a Category 2 architecture with at least medium (>90%) Diagnostic Coverage (DC)

* Source: ISO 13849

Conclusion

- Multicore architectures seem to be an interesting option for safety-related applications.
- Freedom from Interference in the data- and resource-domain seems to be primarily achievable with the CPU-MPUs and a hardware optimized RTOS like PxROS from HighTec.
- CPU-MPUs can be complemented with Bus-MPUs and RAP.
- Utilization of the SMU and strategies for error-handling are crucial to meet the requirements of the ISO 13849 regarding Diagnostic Coverage.
- Multicore parametrization/programming/debugging is challenging!

Outlook

- The presented safety-mechanisms complement each other only if the configuration is kept consistent. Tools have to be developed to ease parametrization.

- While the current architecture is fail-safe, advancement towards fail-operational is challenging in terms of freedom from interference.

Contact

M.Sc. Thomas Barth
Prof. Dr.-Ing. Peter Fromm

Hochschule Darmstadt - University of Applied Sciences
FB Elektrotechnik und Informationstechnik (EIT)
Birkenweg 8
64295 Darmstadt

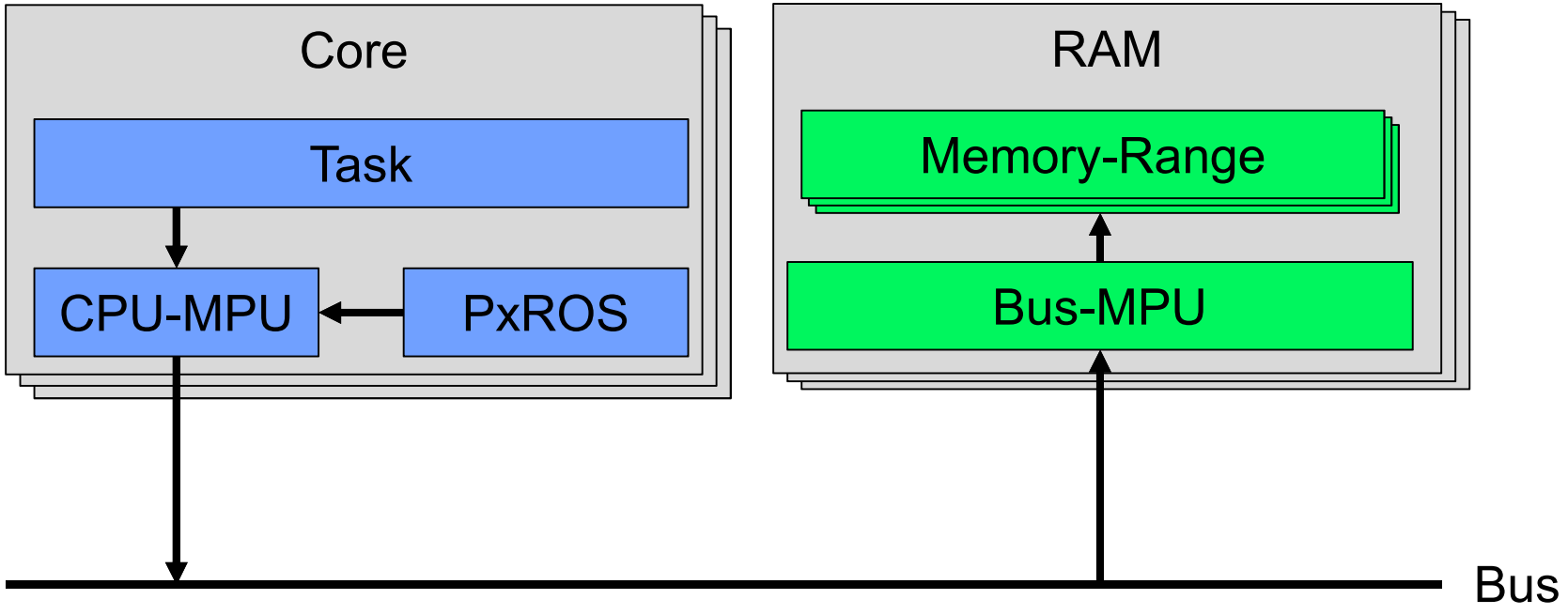
Email: thomas.barth@h-da.de; peter.fromm@h-da.de
Web: www.eit.h-da.de

BACKUP

Bus-MPUs

While the CPU-MPUs only monitor outgoing traffic, the Bus-MPUs monitor ingoing traffic and allow to assign memory ranges to certain cores. Only write-access can be limited.

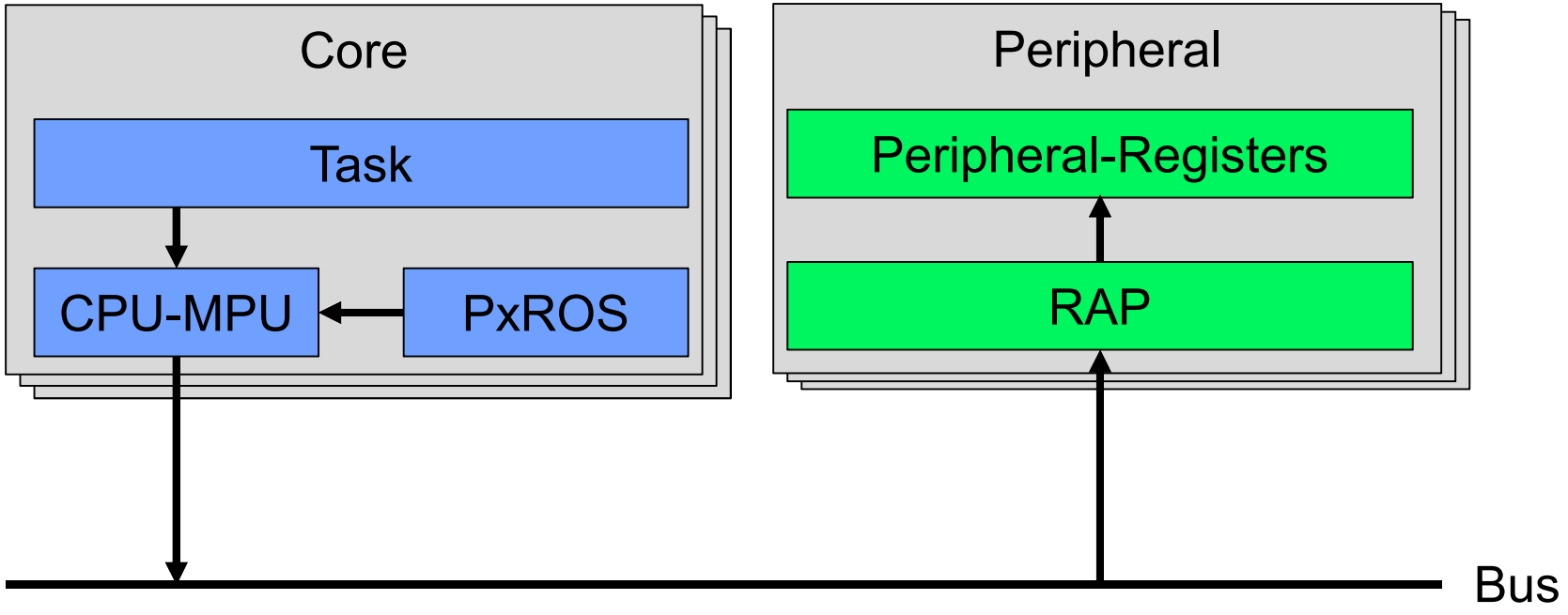
Each RAM of the AURIX is equipped with an own Bus-MPU.



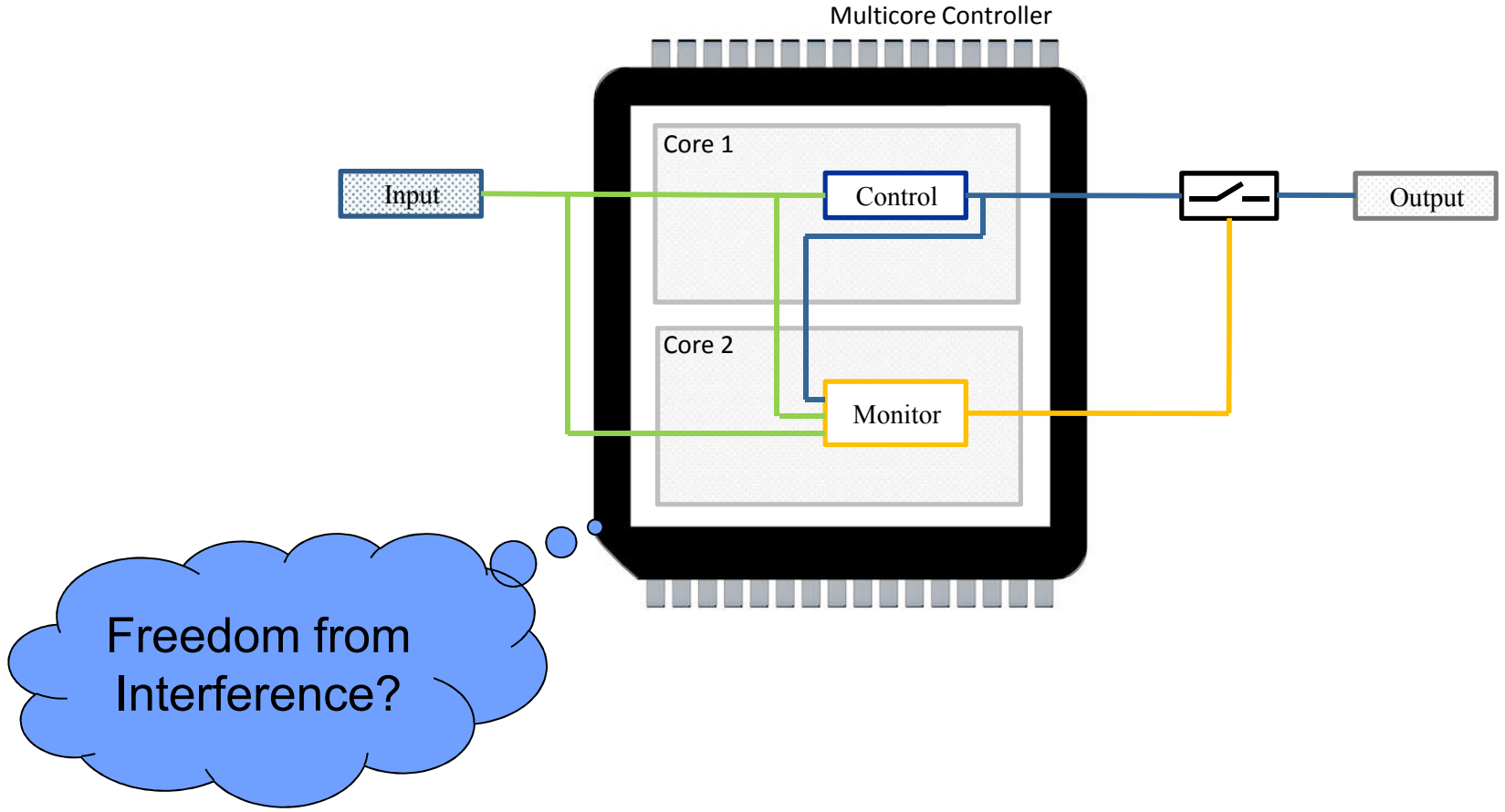
Register Access Protection (RAP)

Similar to the Bus-MPU, the Register Access Protection can be used to assign peripherals to certain cores. Only write-access can be limited.

The granularity depends on the peripheral, for GPIOs each Port can be assigned.

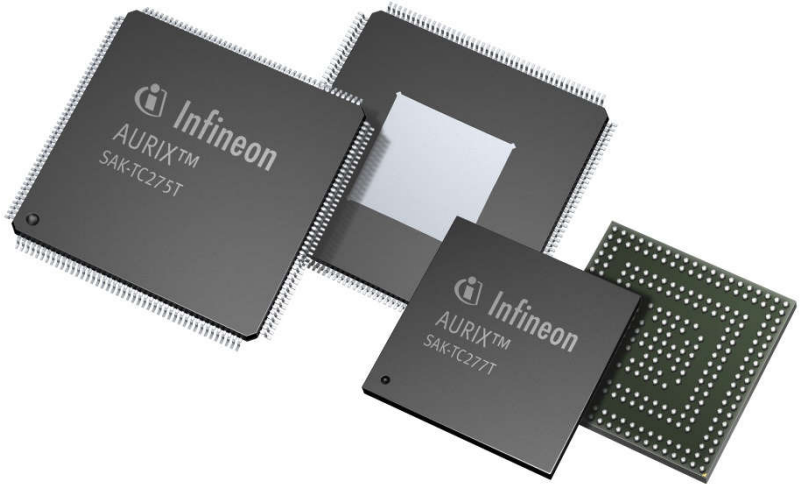


Future architecture?



Multicore Motivation

Main reasons for multicore

- Less latency compared with external communication
- Costs (2 devices vs. 1)
- Tamper-resistance

The AURIX TC27x multicore microcontroller was chosen for evaluation as it is an automotive grade microcontroller that is certified according to ISO 26262 and provides a wide set of safety mechanisms.