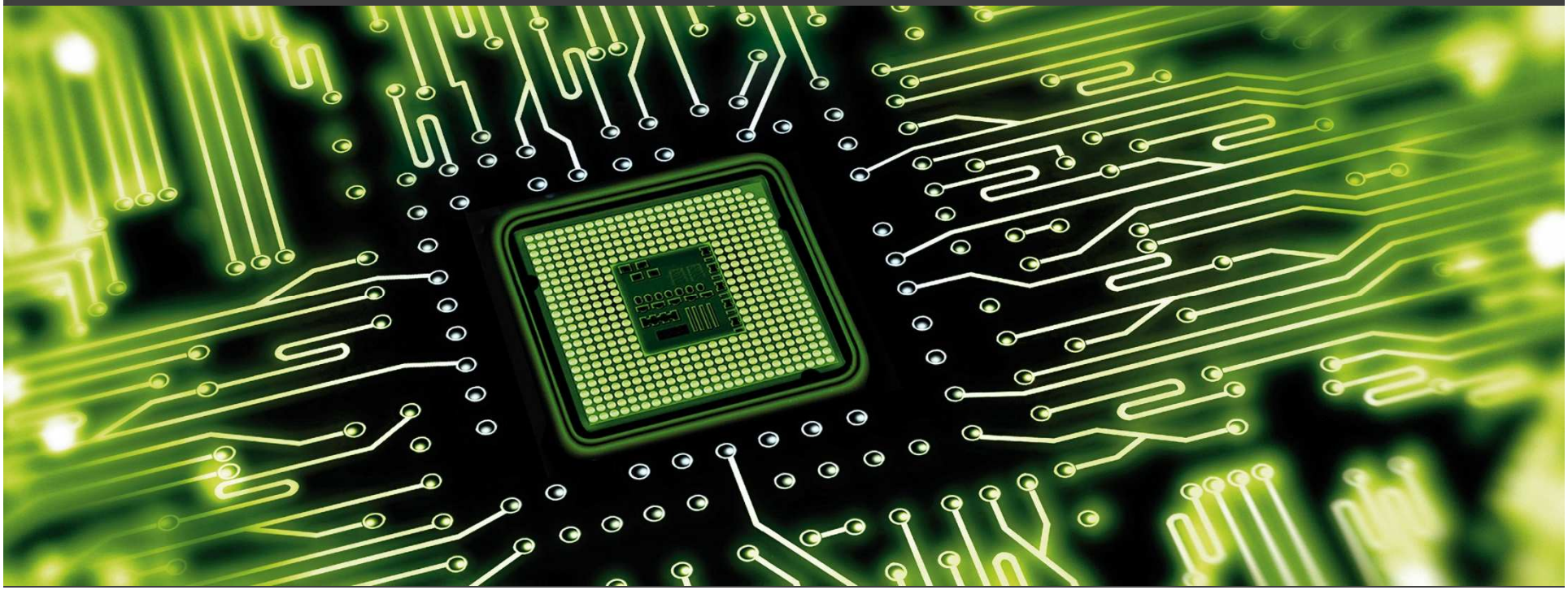
Multicore and Functional Safety; Solutions for new challenges

Dheeraj Sharma
June 2016

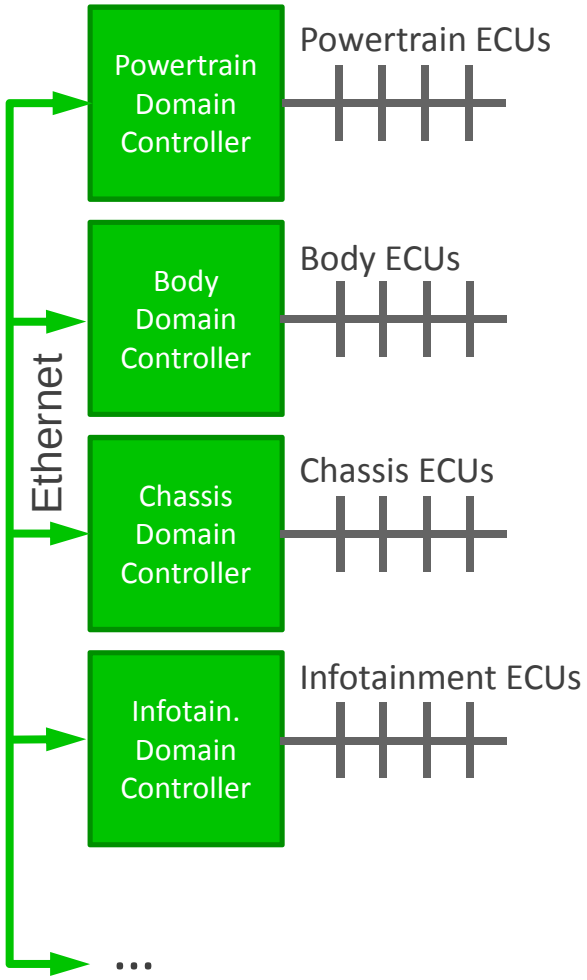


Elektrobit

Agenda

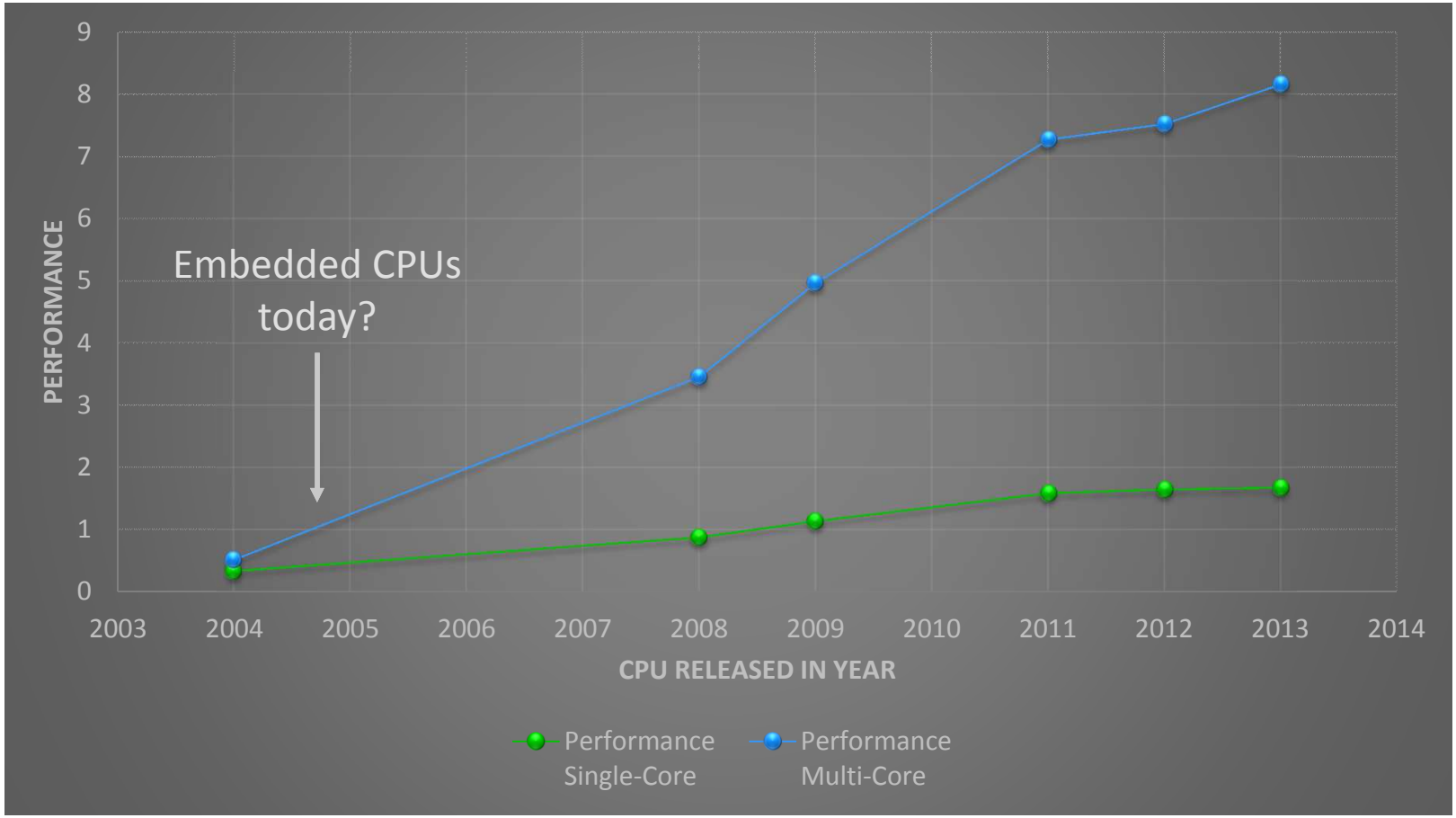
- **The Trend towards Multicore**
- What is Functional Safety ?
- Multicore in AUTOSAR
- Challenges presented by Multicore
- Summary

Next generation of car infrastructure



- Domain Controllers will be huge multi-MCU and Multicore systems
  - ECU independent function
  - Connected to Actuator / Sensor ECUs
  - Reloadable functions
  - Connected via Automotive Ethernet
- Domain ECUs will be “small” Multicore or single core systems
  - Hard real time
  - I/O handling
  - Safety functions

Evolution of Desktop-CPU Performance



Source: Data from c‘t 7/2014, p. 127

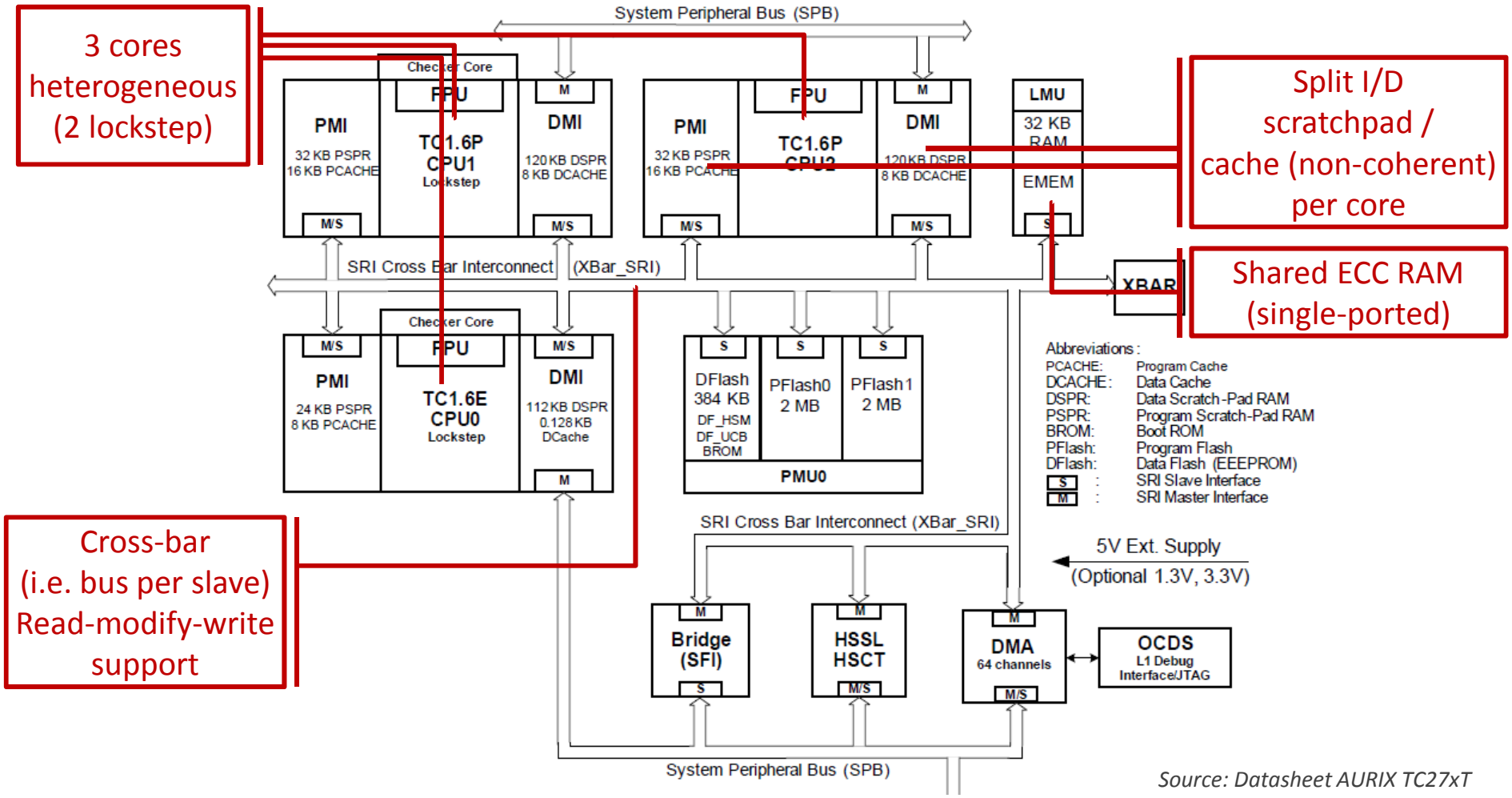
Typical HW architecture (Infineon AURIX TC27x)

3 cores heterogeneous (2 lockstep)

Split I/D scratchpad / cache (non-coherent) per core

Shared ECC RAM (single-ported)

Cross-bar (i.e. bus per slave) Read-modify-write support



Source: Datasheet AURIX TC27xT

Elektrobit

Agenda

- The Trend towards Multicore
- **What is Functional Safety ?**
- Multicore in AUTOSAR
- Challenges presented by Multicore
- Summary

From IEC 62304 Standard:

"There is no known method to guarantee 100 % safety for any kind of software.
*There are **three major principles** which promote safety for medical device software:*

1. *Risk management*
2. *Quality management*
3. *Software engineering"*

Risk Control as part of risk management

- The **goal** of risk control is to reduce the risk to an “acceptable level”.
- **Methods** of risk control (in order of preference!):

1. Eliminate the risk (by design)
   Easiest by omitting the functionality which creates the risk.

2. Implement protective measures
   E.g., fault detection, prevention of fault propagation (**freedom from interference**), …

3. Provide adequate information
   E.g., warnings, guidance in the safety manual, …

ISO 26262: Freedom from Interference

Memory

- Unintended writing to memory of another partition
- Register/Configuration corruption due to unintended writing to processor registers

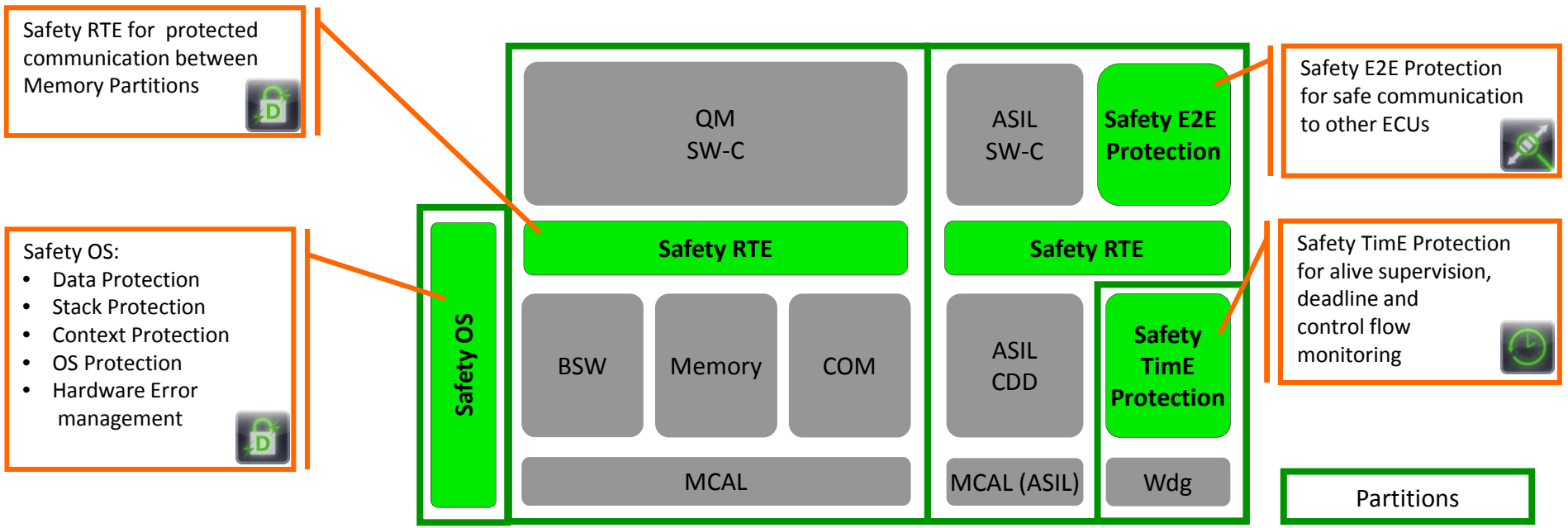
CPU Time

- Blocking of partitions
- Wrong allocation of processor execution time

Communication

- Loss of communication
- Insertions of messages

Memory Partitioning

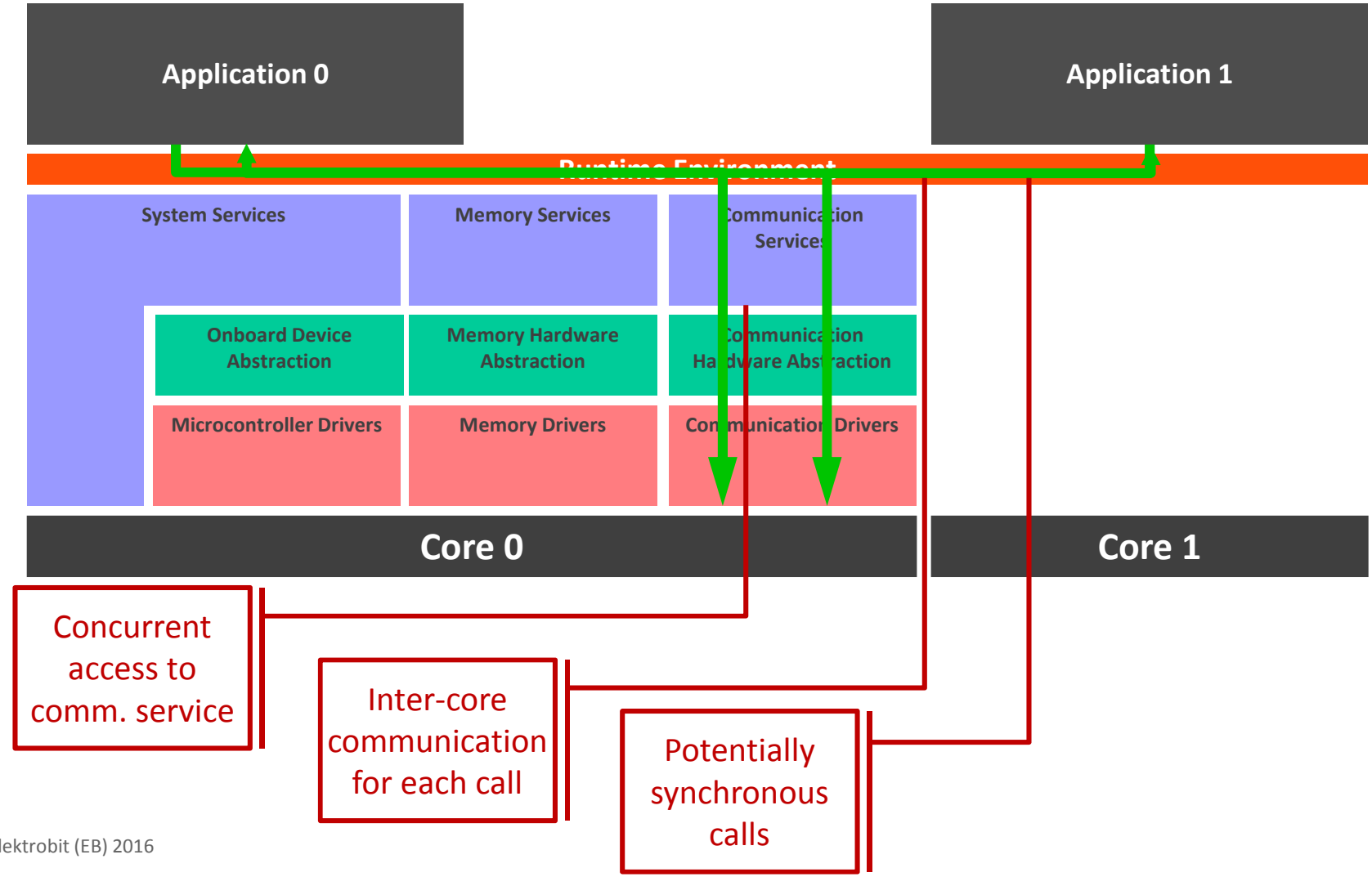


Elektrobit

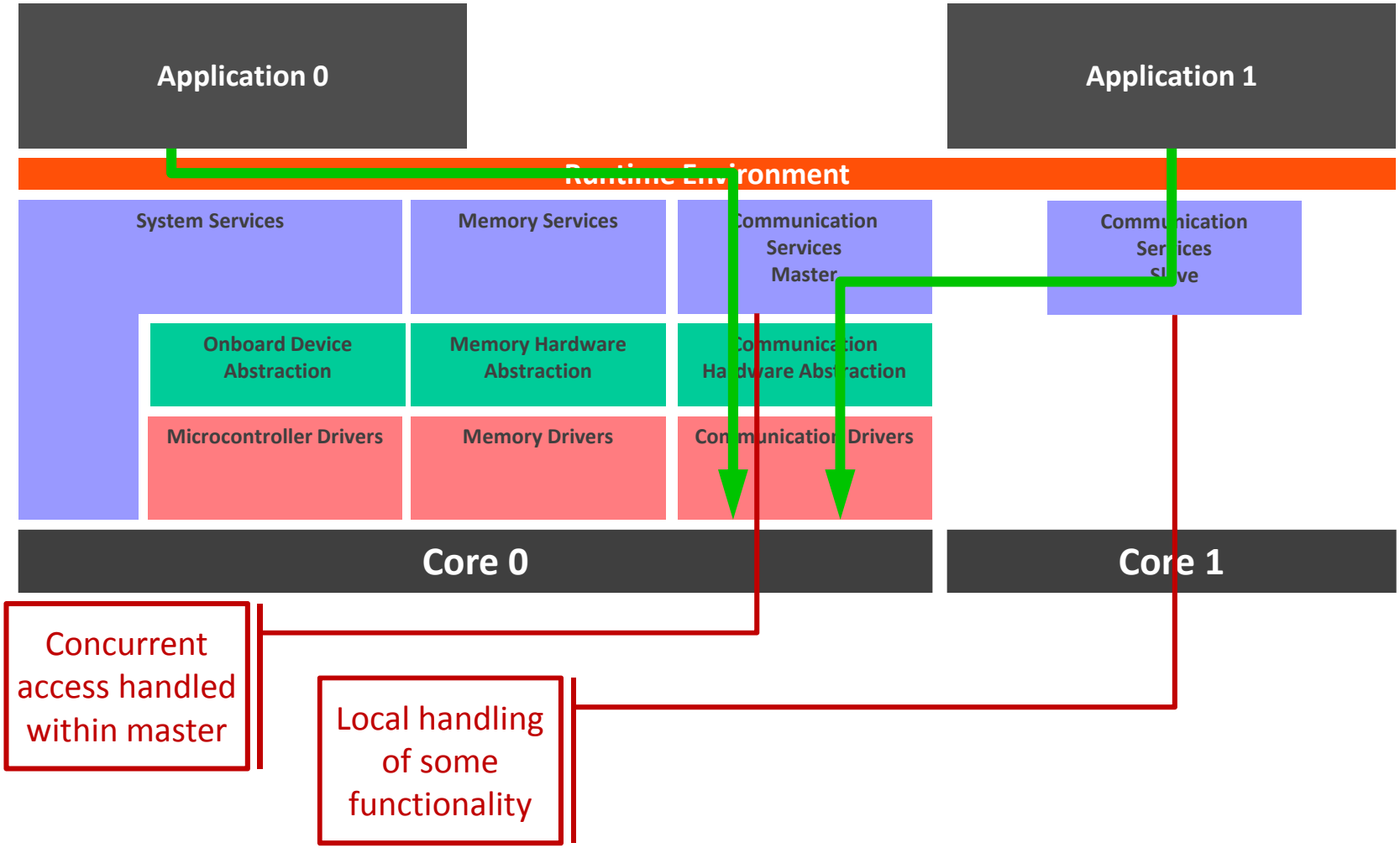
Agenda

- The Trend towards Multicore
- What is Functional Safety ?
- **Multicore in AUTOSAR**
- Challenges presented by Multicore
- Summary

Multicore in AUTOSAR 4.0



Basic Software Distribution in AUTOSAR >4.0



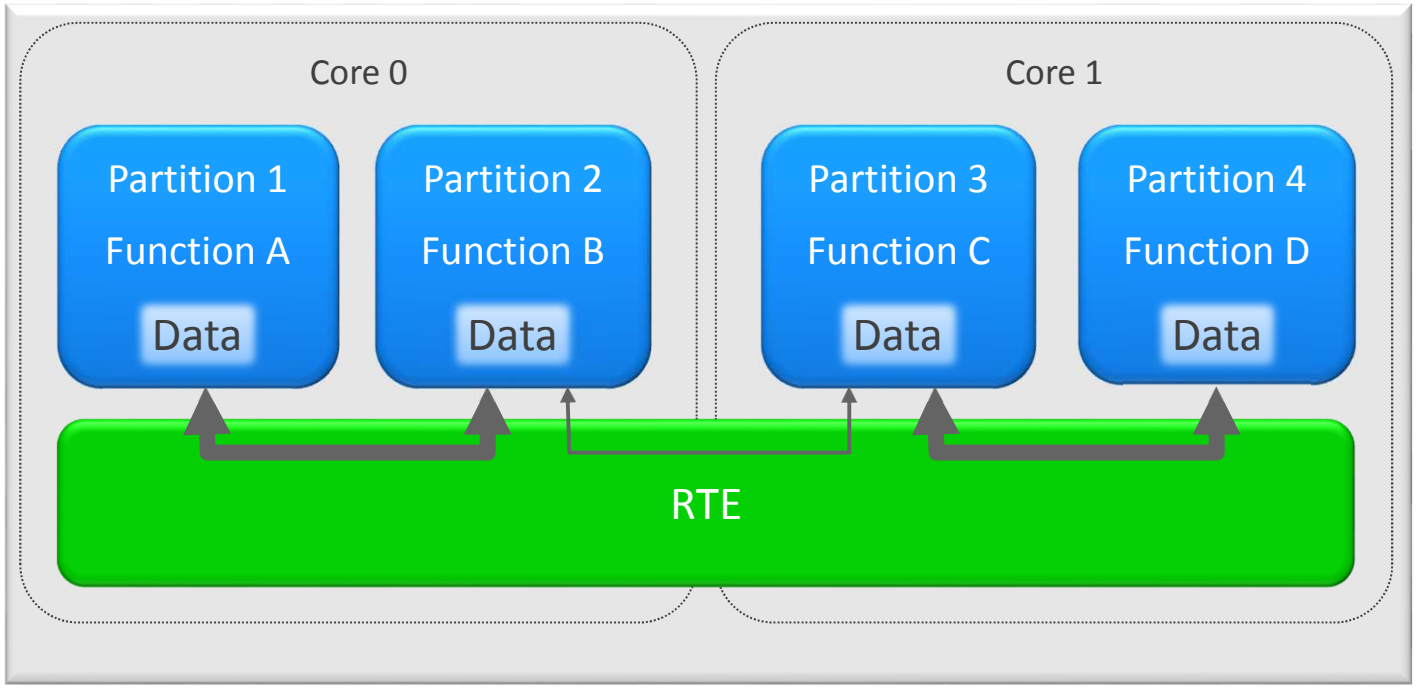
Elektrobit

Agenda

- The Trend towards Multicore
- What is Functional Safety ?
- Multicore in AUTOSAR
- **Challenges presented by Multicore**
- Summary

Cohesive Architecture

- **Partitioning and core allocation**
  - Functional Grouping
  - Related functions on same core
  - Freedom from Interference
- **Safety Analysis**
  - Only feasible when working on small units



Architectural considerations

Challenges

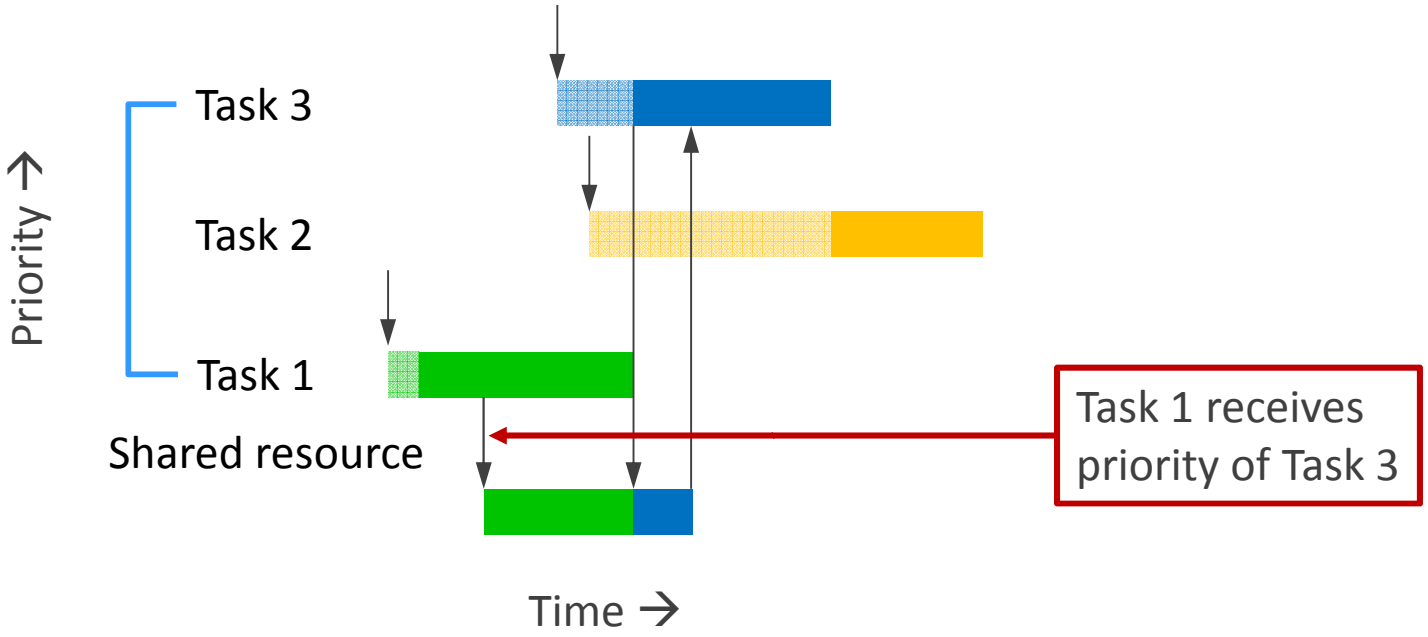
- Partitions are mapped to cores
  - Groups of tasks etc. are distributed
- Cross-core communication takes time
  - Beware: Transparent communication in AUTOSAR
- Locks are expensive
  - Locking only possible via OS
  - Spinlocks can block entire CPU
- True parallel execution
  - Race conditions
  - Task priorities work only on one core

Countermeasures

- Keep partitions cohesive
- Locate related and communication-intensive applications on same core or partition
- Where possible, use lock-free algorithms
- Reduce cross-core communication to a minimum

Single-core scheduling with shared resources

- AUTOSAR supports preemptive and non-preemptive tasks
- **Shared resources to protect access to global data**
- AUTOSAR uses Priority Ceiling Protocol (PCP)
- Upon occupying a resources, the task receives the priority of the highest task that can access the resource



Shared resources on Multicore

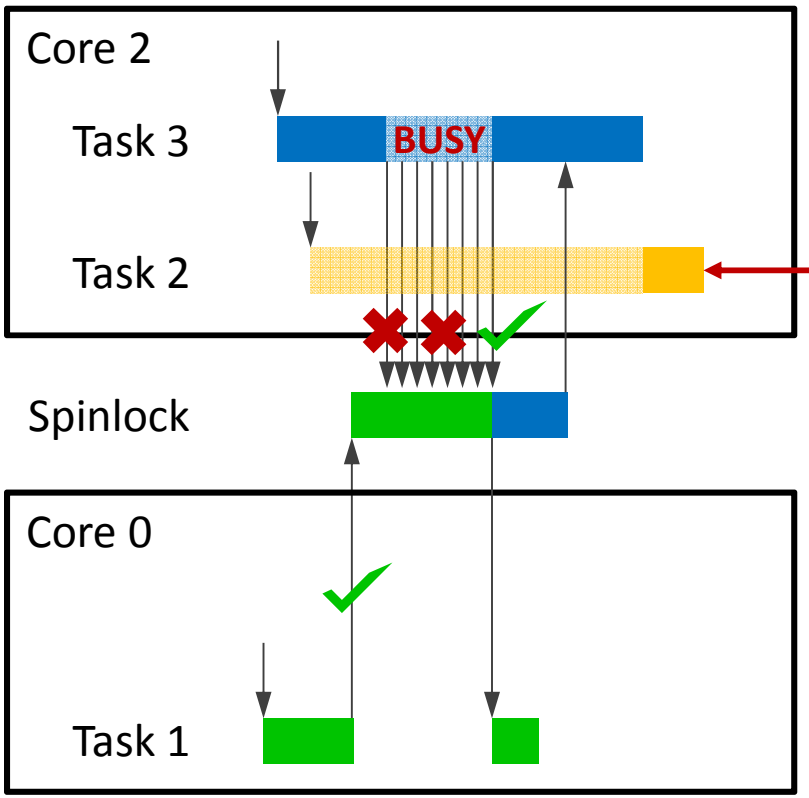
- **Priority Ceiling Protocol** (PCP) does **not work** for Multicore
  - Separate schedulers
  - No common notion of priority

- AUTOSAR specifies **spinlocks for Multicore synchronization**
  - Busy waiting on memory location via atomic instruction *(e.g. TAS, CAS, LL/SC)*

- **Multicore Priority Ceiling Protocol** (MPCP) assigns priority higher than highest core-local priority to each inter-core resource
  - Can be implemented in AUTOSAR (spinlock and interrupt lock)
  - Used to prevent deadlock
    (e.g. when task holding a spinlock is preempted by task that also tries to obtain the spinlock)

Elektrobit

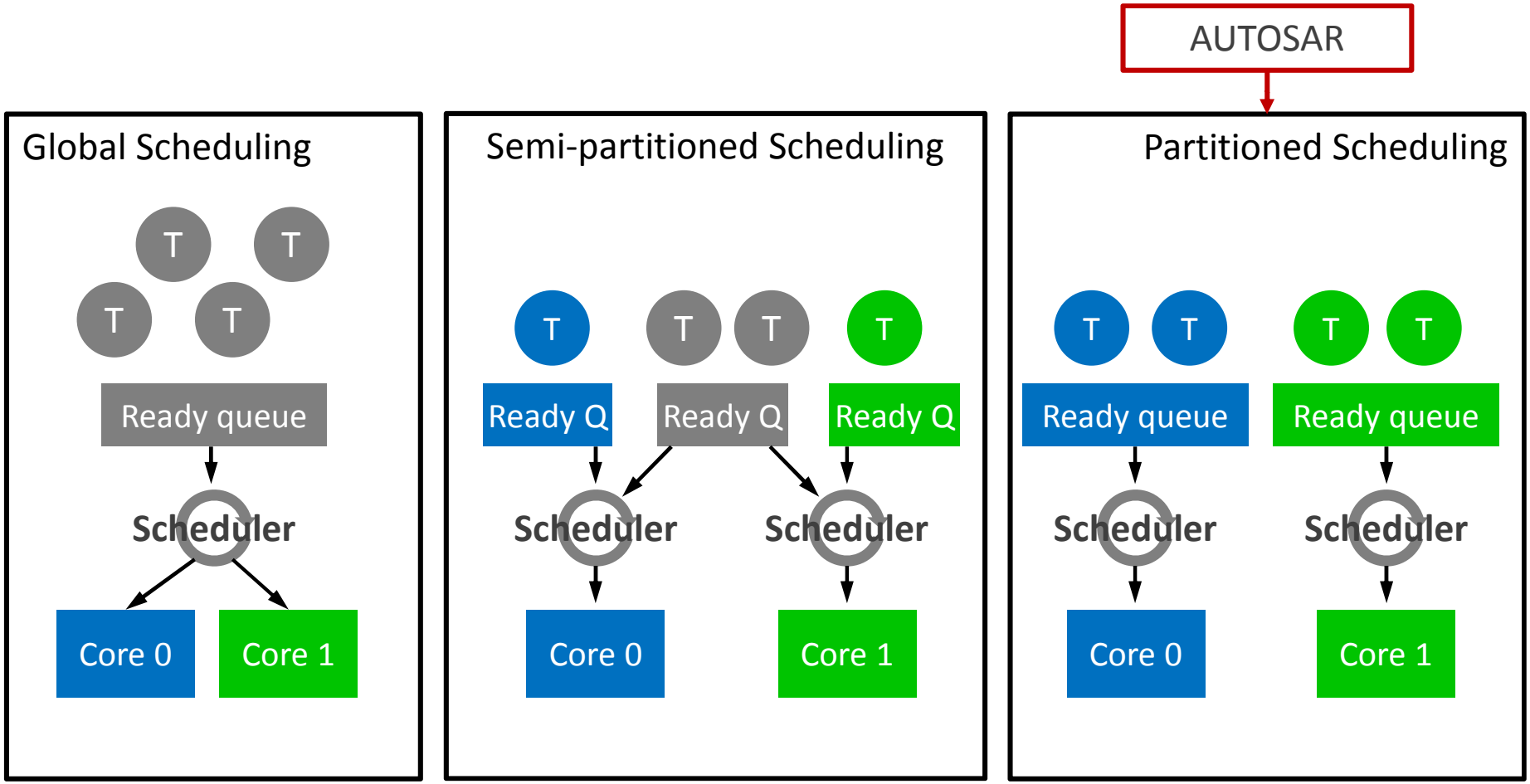
Multicore scheduling with shared resources



- Spinlocks **waste processing time**
- Spinlocks create **high traffic on** the processors **interconnect**
- Inter-core locking creates **scheduling dependencies**

Task 2 could have completed while Task 3 was waiting for the spinlock

Multicore scheduling in AUTOSAR



Elektrobit

Multicore scheduling in AUTOSAR

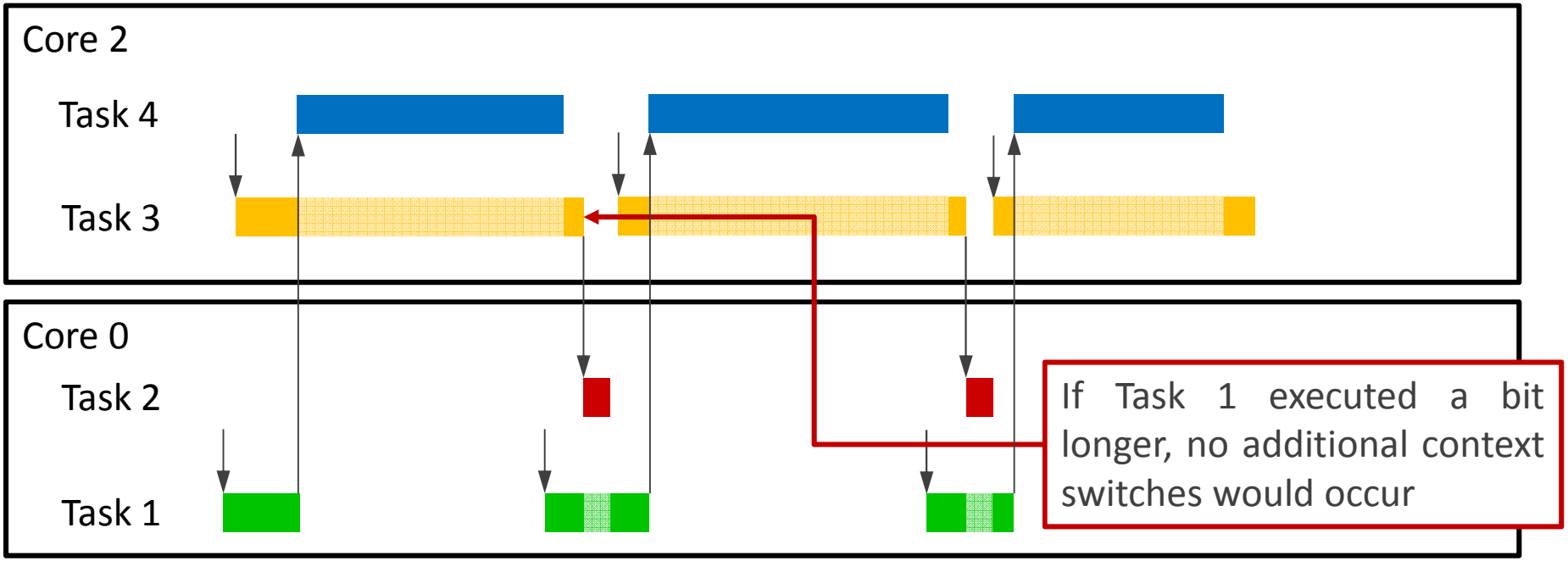
- Logical separate **OS instance per core**
  - OS resources are effective only per core
  - Priorities defined per core
  - (Memory protection configured per core)
- **Static mapping** of tasks to cores (i.e. partitioned scheduling)
  - No fancy features like dynamic load-balancing

- **Cross-core interrupts** possible
- **Cross-core events** possible
- **Inter-core locking** via spinlocks; but spinlocks can be expensive.

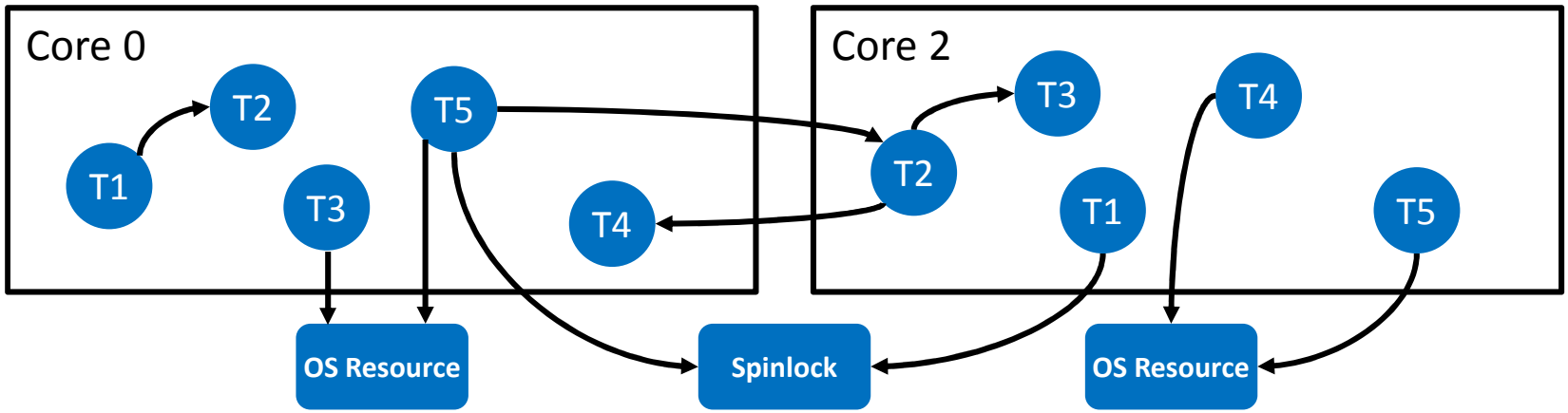
Elektrobit

Partitioned Multicore scheduling

- Cross-core task activations/interrupts/callbacks/sync. C/S calls create **scheduling dependencies** *(remember independent schedulers/timebases)*
- Cross-core scheduling dependencies can **increase context switches**
- **Analysis is complex**



Full-blown scheduling complexity



- What are the scheduling dependencies here?

„In my thesis I have shown how incredibly complex the analysis is and ***quite frankly you simply should not build systems like this****.“*

Dr. Mircea Negrean

(Title of thesis: „Performance Analysis for Multi-Mode Multicore Systems with Shared Resources – Principles and Application to AUTOSAR“)

Elektrobit

Agenda

- The Trend towards Multicore
- What is Functional Safety ?
- Multicore in AUTOSAR
- Challenges presented by Multicore
- **Summary**

Elektrobit

Corollaries

- **Reducing inter-core locks**
  - Increases potential parallelism
  - Decreases complexity
- **Reducing core-scope** of global data
  - Reduces need for inter-core locks
  - Increases potential use of caches
- **Separating global data**
  - Reduces need for inter-core locks
  - Helps establishing memory protection
- **Reducing consistency**
  - Increases potential parallelism

Questions to ask

- **Where is data accessed?**
- **What kind of lock is required?**
- **Is atomic access possible?**
- **Where is data accessed?**
- **Are separate per-core instances possible?**
- **Do members of a struct/elements of an array have different access requirements?**
- **Is the order of calls relevant?**
- **May data be inconsistent in intermediate states?**

Contact us!

Elektrobit

www.elektrobit.com
dheeraj.sharma@elektrobit.com