Technische Universität Braunschweig

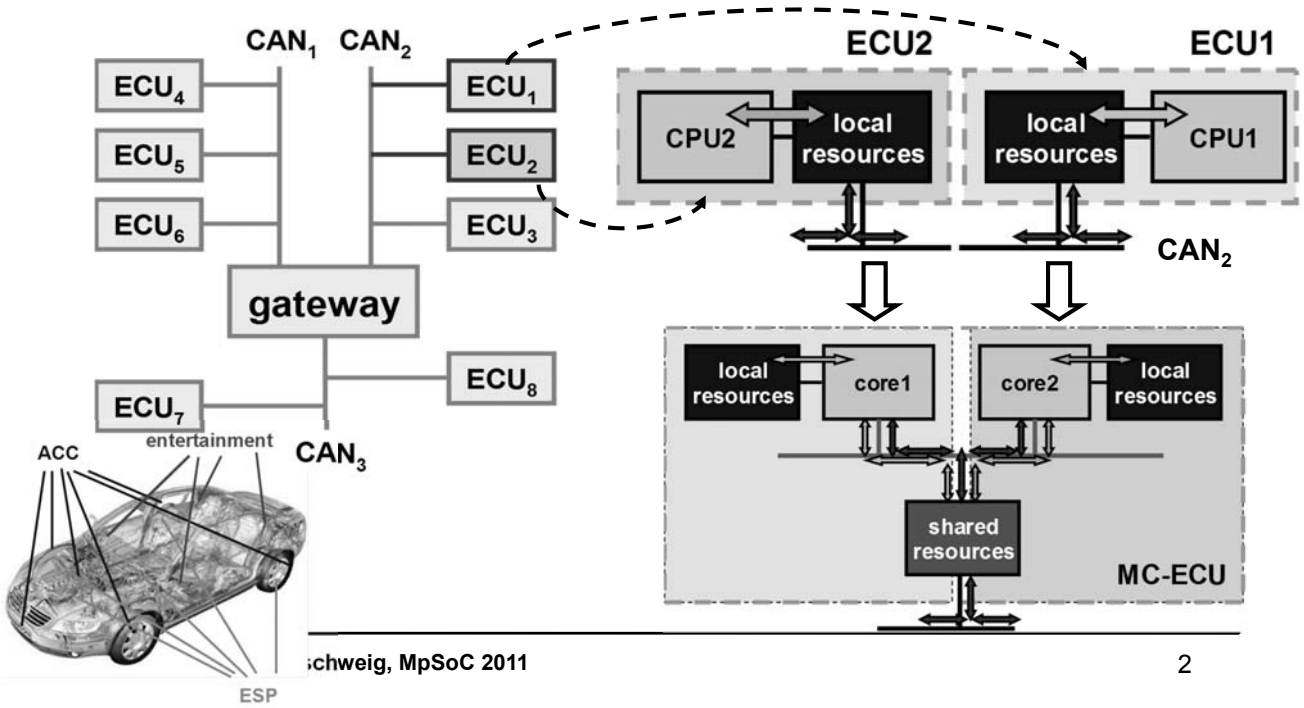
INSTITUT FÜR DATENTECHNIK UND KOMMUNIKATIONS-NETZE

MpSoC for Safety Critical Applications – from Multicore to Manycore

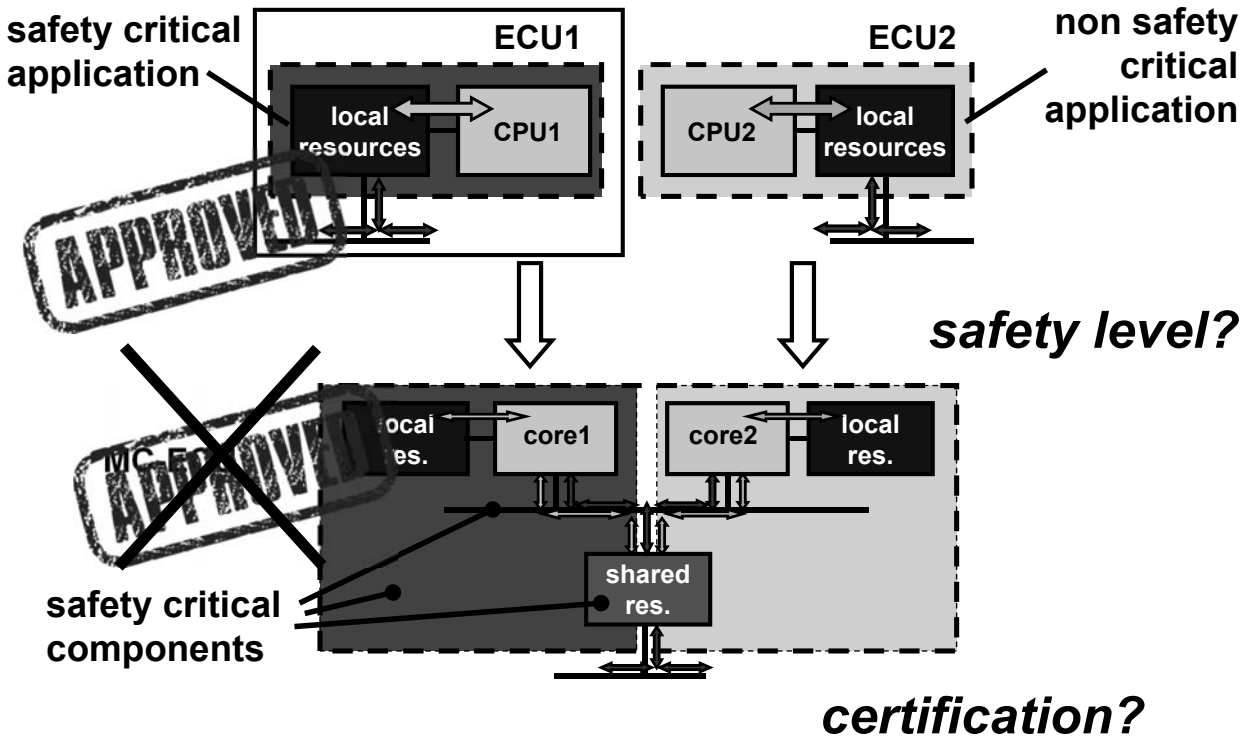
Rolf Ernst
Institut für Datentechnik und Kommunikationsnetze
MpSoC 2011, Beaune, France

Motivation

- **MpSoCs are an efficient platform for systems integration**
- **due to physical resource sharing, safety critical systems integration becomes more challenging**



MpSoC for Critical Applications

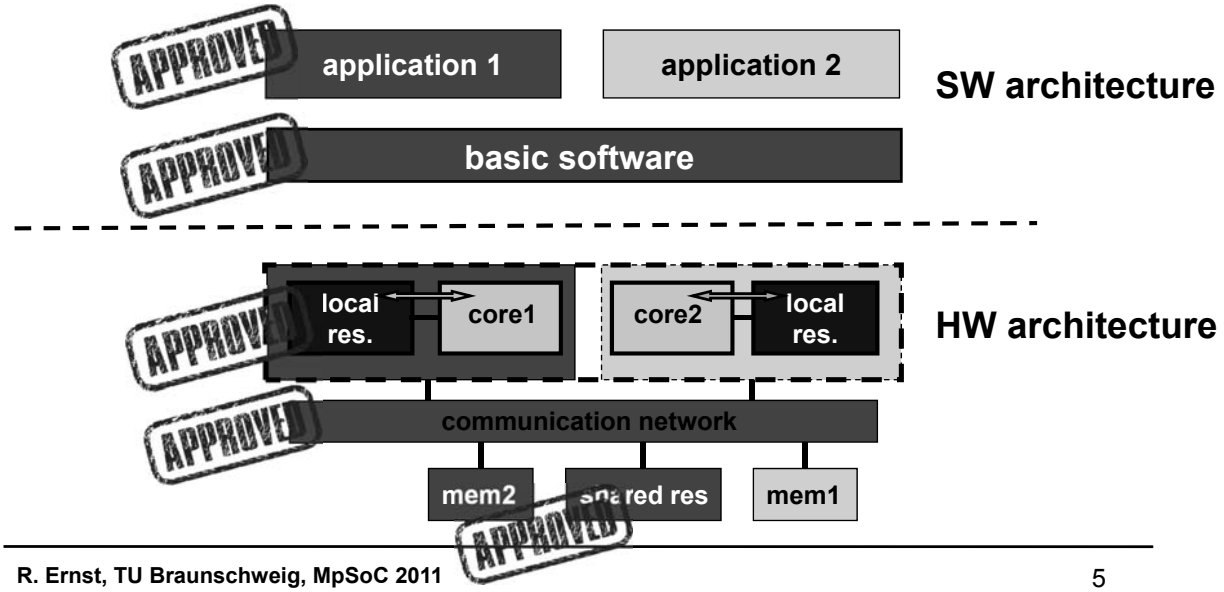


Mixed Critical MpSoC Certification/Qualification

- resulting multi-core ECU subject to highest safety standard involved
- high certification (or qualification, resp.) cost
  - must cover ALL applications and hardware
  - often qualified data of non-critical application not available
  - re-certification for any non-critical application update required
- alternative: isolation of different criticalities
  - requires certification/qualification of core components that control resources

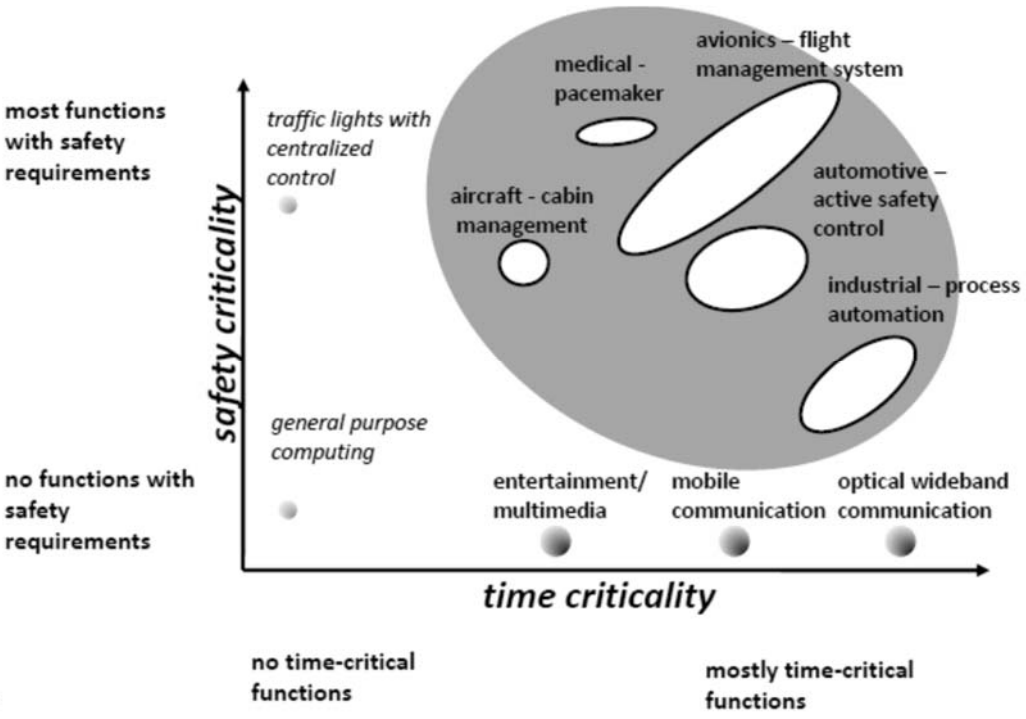
Isolation - Basics

- approve core components that control the resources used for any of the critical applications
  - basic software
  - communication
  - shared resources used for critical applications



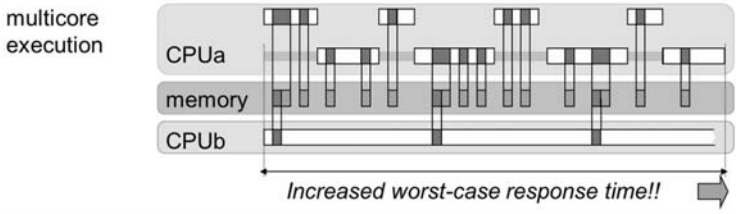
Safety and Time Criticality

- many safety critical systems are also time critical



MpSoC 2010 - Multicore

- virtualization/isolation used for functional isolation
- virtualization is not always sufficient for safety critical systems
  - competing accesses to shared resources requires temporal isolation/compensation

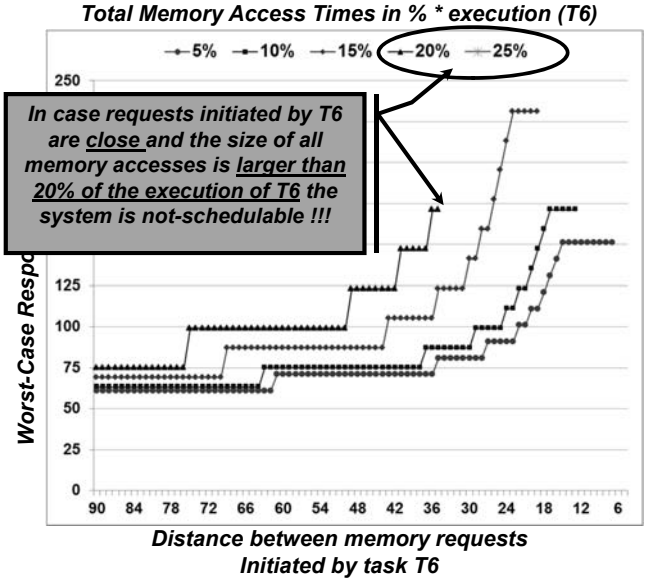


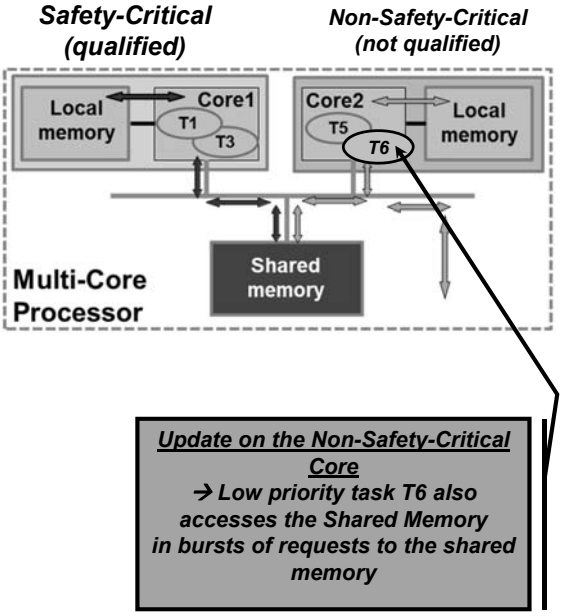
  - dependencies can be used for „performance attack“
    - uses memory arbitration
    - works even when timing is less critical

- approach: assign budgets and use formal analysis

“Performance Attack” via Update

- *misbehaving update T6 on Core 2 leads to failure of high priority critical task T1*



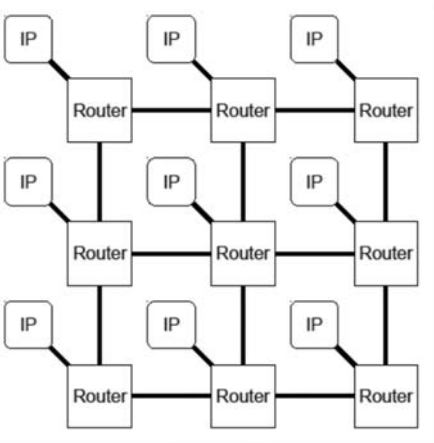


MpSoC 2011 - Manycore

- extend isolation to manycore systems
  - include NoC in function and performance isolation
- challenge:
  - guarantee performance for critical applications
  - minimize impact for non critical applications

Considering the Network-on-Chip

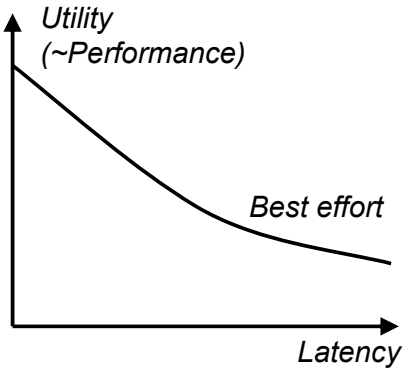
- routers forward individual packets
  - shared by everyone: Lots of interference
  - arbitration on contention
- use QoS techniques to guarantee service to individual traffic streams
- existing QoS mechanisms serve guarantees first! best-effort traffic is "second-class citizen"
  - static allocation of time slots (e.g. Aethereal)
  - dynamic scheduling of VCs + priorities (e.g. MANGO, QNoC)



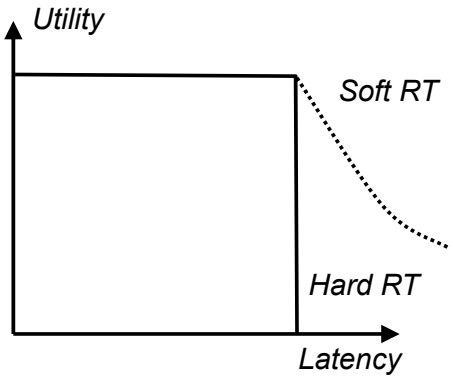
- not optimal for mixed-critical applications
  - best-effort is an important traffic class, but often *latency-sensitive*
    - BE traffic suffers from high latency
    - RT traffic has no benefit from reduced latency (deadline driven)

Mixed Criticality: Best-effort and Real-Time Streaming

- best-effort applications
  - most existing applications, major role in user experience
  - unpredictable and bursty resource usage
  - latency-sensitive

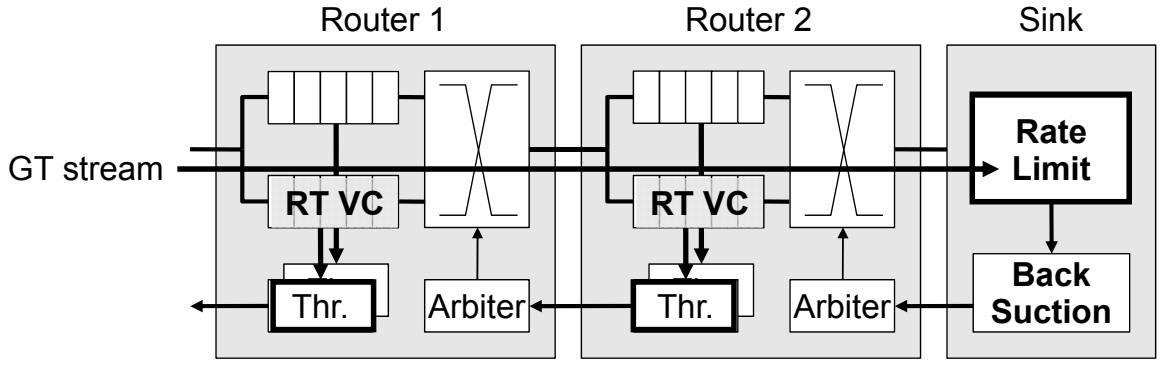


- real-time streaming applications
  - require resource and timing guarantees
    - resource sharing must be under control for efficient co-execution
  - regular access patterns → latency-tolerant



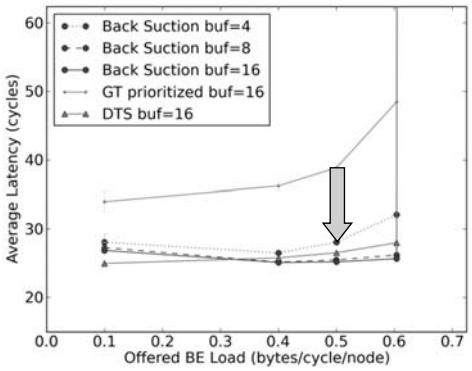
Solution: "Back Suction"

- prioritize RT traffic based on downstream buffer occupancy
- let sink control the priorization of RT traffic by creating a "suction" that pulls data towards the sink
- suction propagates backwards towards source
  - Threshold module at every virtual channel buffer monitors occupancy
  - limit rate (to guaranteed rate) at which sink may assert back suction



Result: Guarantees and Improved BE Latency

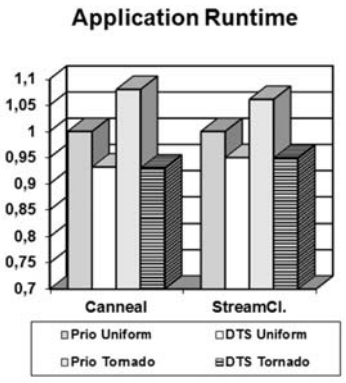
- **mechanism provides throughput guarantees to individual real-time streams**
  - **proven by formal analysis**
- **BE latency is improved significantly**
  - **application runtime improves accordingly**



[NOCs 10]

~ 30% latency improvement over standard prioritization scheme

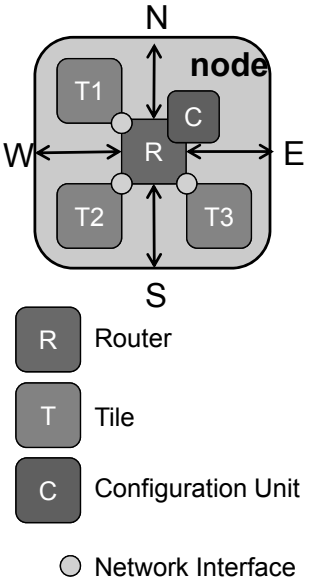
improves application performance by >10%



[ASP-DAC 10]

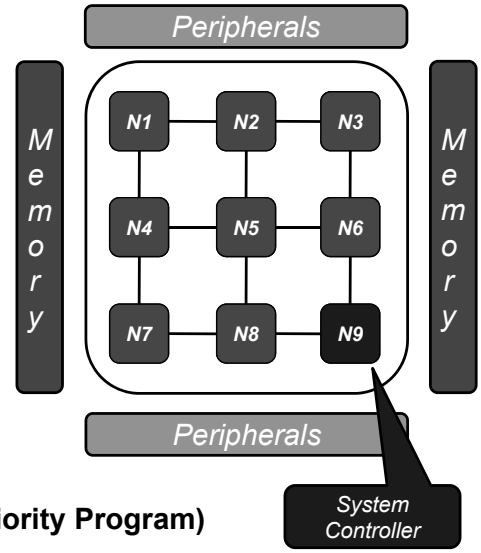
Functional Isolation and System Virtualization

- **compose manycore system from tiles**
  - **existing IP components, e.g. LEON3-Sparc processor**
  - **individually qualified/certified or pre-certified**
- **network interface isolates tiles**
  - **provide controlled access to the rest of the system**
  - **transparent mapping of memory and peripherals**
    - **translate tile-local address space to system-wide addresses**
- **configuration unit**
  - **controls network interface**
  - **configured via central, trusted resource manager**
  - **monitors behavior of tiles to detect malfunctions**



Manycore Research Platform

- European project RECOMP
  - reduce certification costs for MpSoC
  - TUBS: manycore research platform for mixed-critical applications
- FPGA based prototype
  - based on SPARC LEON3 processor tiles
  - NoC with back suction
  - safety-functions in Network-Interface
    - system virtualization
    - power- and performance-monitoring



- German project ASTEROID (DFG Research Priority Program)
  - introduce platform redundancy for increased reliability requirements
  - redundancy in space and time (retransmission, checkpointing, rollback)
  - targeted to application and OS safety requirements
  - collaboration w. TU Dresden (L4 microkernel)

Conclusion

- MpSoCs are used for safety critical applications
- mixed criticality is a challenge and a serious certification/qualification and maintenance cost driver
- research has started to look into manycore systems for mixed critical applications – many new challenges
- several European and national projects targeting MpSoC for mixed criticality - first results presented

Thank you!

Literature

- **RECOMP**
  - **www.recomp-project.eu**
  - **TUBS part**
    www.ida.ing.tu-bs.de/en/research/projects/recomp

- **for the challenge of MpSoC performance dependencies see**
  - **Mircea Negrean, Simon Schliecker, Rolf Ernst. "Response-Time Analysis of Arbitrarily Activated Tasks in Multiprocessor Systems with Shared Resources." In *Proc. of Design, Automation, and Test in Europe (DATE)*, Nice, France, April 2009.**

- **for the manycore research platform**
  - **Jonas Diemer and Rolf Ernst, "Back Suction: Service Guarantees for Latency-Sensitive On-Chip Networks," in Proceedings of the 4th ACM/IEEE International Symposium on Networks-on-Chip (NOCS'10), May 2010**
  - **Jonas Diemer, Rolf Ernst, und Michael Kauschke, "Efficient Throughput-Guarantees for Latency-Sensitive Networks-On-Chip," in Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC 2010), January 2010**