

# parMERASA

Theo Ungerer  
parMERASA Coordinator  
University of Augsburg, Germany  
[ungerer@informatik.uni-augsburg.de](mailto:ungerer@informatik.uni-augsburg.de)

**Sascha Uhrig**  
TU Dortmund, Germany  
[sascha.uhrig@tu-dortmund.de](mailto:sascha.uhrig@tu-dortmund.de)

Synopsis of FP7 Computer Systems and Transitioning to Horizon 2020  
Tel-Aviv, Israel, June 22, 2013

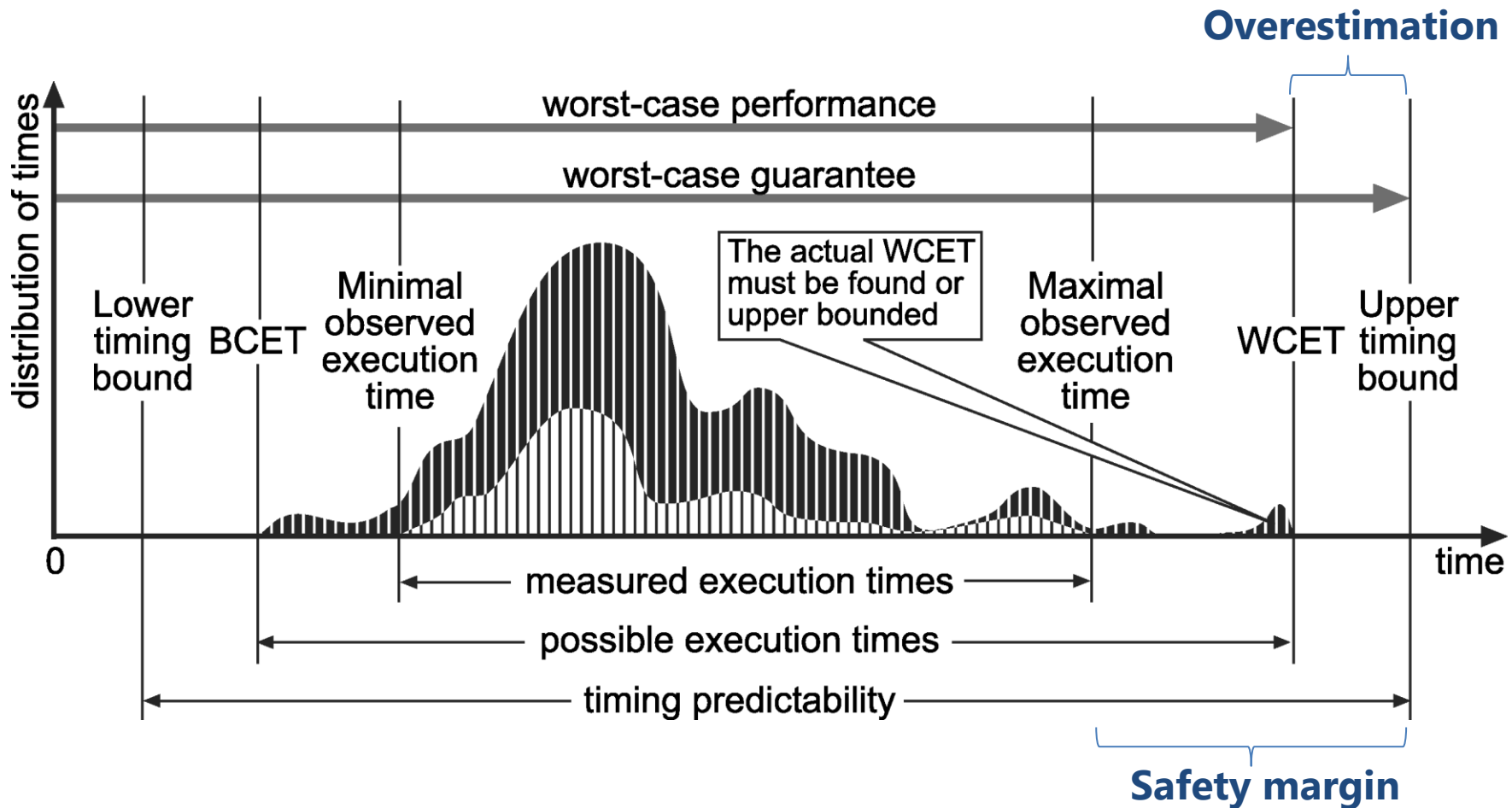
- The Hard Real-time Story
- Project Motivation
- parMERASA Overview
- parMERASA Structure and Achievements
- Conclusion

- Hard real-time:
  - Hard deadlines must never be missed
  - If missed it may cause harm to humans or equipment



- **Hard real-time** embedded systems require that a **deadline must never be missed**,
    - need a **proof of timing requirements by WCET (worst case execution time) analysis**, or
    - **at least**, demonstrate, depending on the criticality of system, that the **implementation meets its timing requirements**.
- ⇒ Most hard real-time systems are implicitly **safety related**

- **Static WCET analysis**
  - *Modeling the processor and memory system*
  - Modeling all potential paths of the program
  - Compute WCET bound
- **Measurement-based/hybrid WCET analysis**
  - *Measurement of basic block execution lengths*
  - Modeling all potential paths of the program
  - Compute WCET value, not necessarily an upper bound
  - **Extensive Testing and adding a safety margin**



**Figure: Basic notions concerning timing analysis of systems**

Source: Reinhard Wilhelm et al.: The Worst-Case Execution-Time Problem—Overview of Methods and Survey of Tools; ACM Transactions on Embedded Computing Systems, April 2008

- **Increasing demand** for functionality in current and future real-time embedded systems
    - Autonomic driving
    - 4D flight path prediction
    - Online modelling of dynamic vibrations and distortions
- ⇒ **Higher WCET performance required**



- **Multi-core processors** represent an excellent opportunity for providing such **higher levels of WCET performance** due to parallel execution.
  - However, **timing behaviour of COTS multi-cores** executing parallel applications is **not analysable / hard to analyse**
    - **COTS (common of-the shelf) processors** contain features that make a WCET analysis hard or even impossible, as e.g.
      - Complex branch prediction, out-of-order execution, two level cache hierarchy, simultaneous multithreading (SMT)
    - **COTS multi-core processors** bring in additional handicaps for hard real-time tasks
      - Shared busses, caches and memories, coherent first level cache
- ⇒ WCET analysis is **too pessimistic**/high **safety margins** required



- **Our approach** based on multi-core systems:
  - **predictable embedded multi-core** design
  - in concert with **WCET analysing technology**
- We have already worked on that topic in the preceding **MERASA** (2007-2010) project...

MERASA showed:  
Timing predictable multi-cores and  
WCET tools are **feasible!**

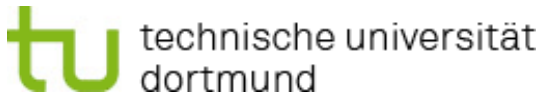
- **parMERASA** goes one step **beyond mixed criticality demands**:
  - **New hardware and software design paradigms** together with **new analysis techniques** are required.
  - Currently, **timing behaviour** of **parallel applications** is **not analysable** with current programming paradigms and timing analysis techniques.

**We target future complex control algorithms by parallelising hard real-time programs to run on predictable multi-/many-core processors.**

- Further increase **WCET performance** by
  - Many-core system
  - Analysable network-on-chip
  - Clustered architecture for supporting
    - Multiple parallel HRT applications
    - Mixed criticality applications
- **WCET technology, verification tools, parallelisation support**, and **system architecture** analysis
- exemplary **parallelisation of industrial applications**
- EC FP-7 project 2011-2014
- 3.3 Mio EC contribution
- Project webpage: <http://www.parmerasa.eu>



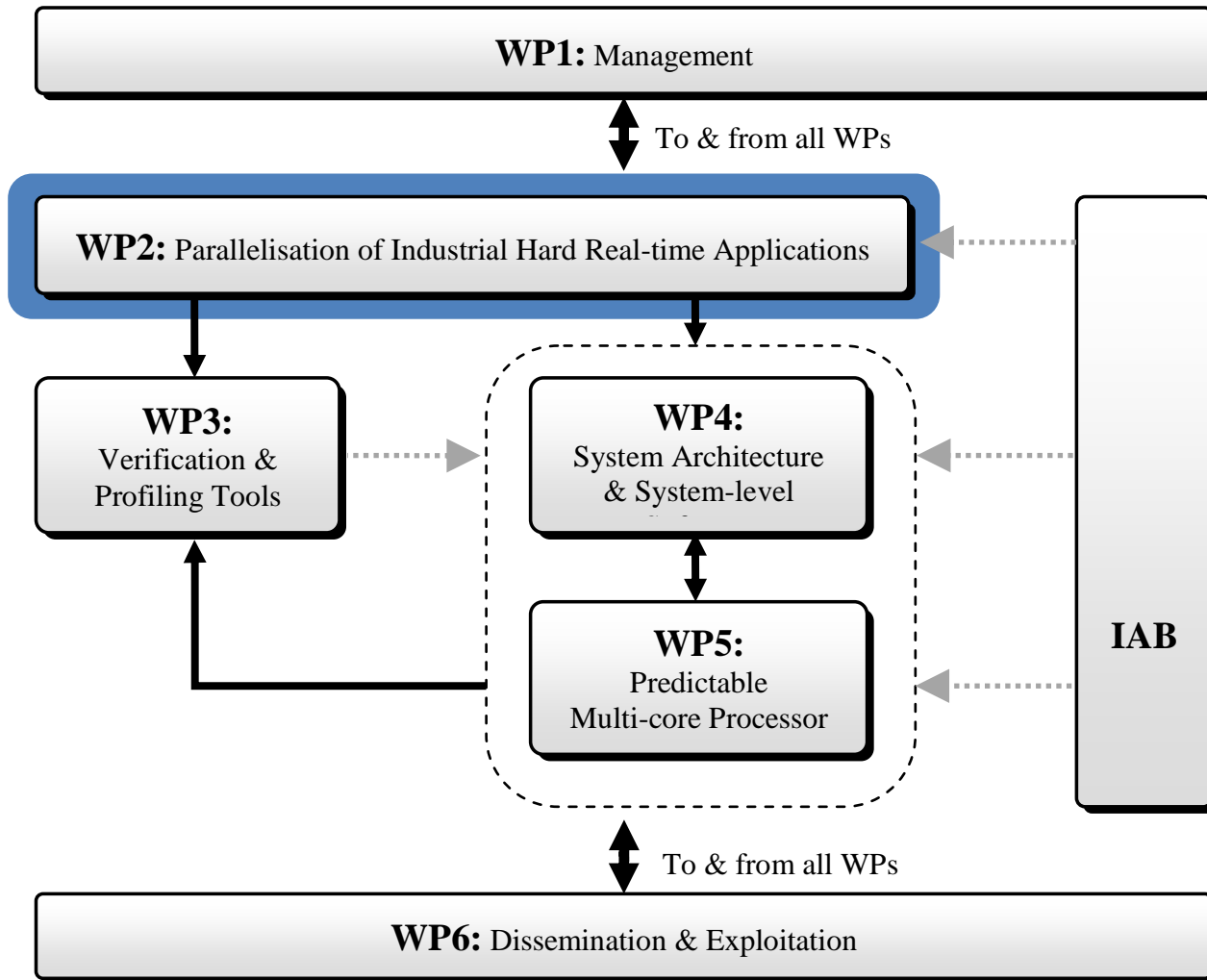
Universität  
Augsburg



- **University of Augsburg** (Project Coordinator)  
Germany
- **Barcelona Supercomputing Center**  
Spain
- **Université Paul Sabatier**  
Toulouse, France
- **Technical University of Dortmund**  
Germany
- **Rapita Systems Ltd.**  
York, UK
- **Honeywell international s.r.o.**  
Brno, Czech Republic
- **BAUER Maschinen GmbH**  
Schrobenhausen, Germany
- **DENSO AUTOMOTIVE Deutschland GmbH**  
Eching, Germany

- **Select and parallelise industrial hard real-time applications.**
- Find ways to **efficiently parallelise industrial applications** for embedded real-time systems.
- Provide **hard real-time support in system software, WCET analysis and verification tools for multi-cores.**
- Develop techniques for **time predictable multi-cores with 16 to 64 cores** which are commercially feasible.
- Contribute to **Standards** and **Open Source Software.**

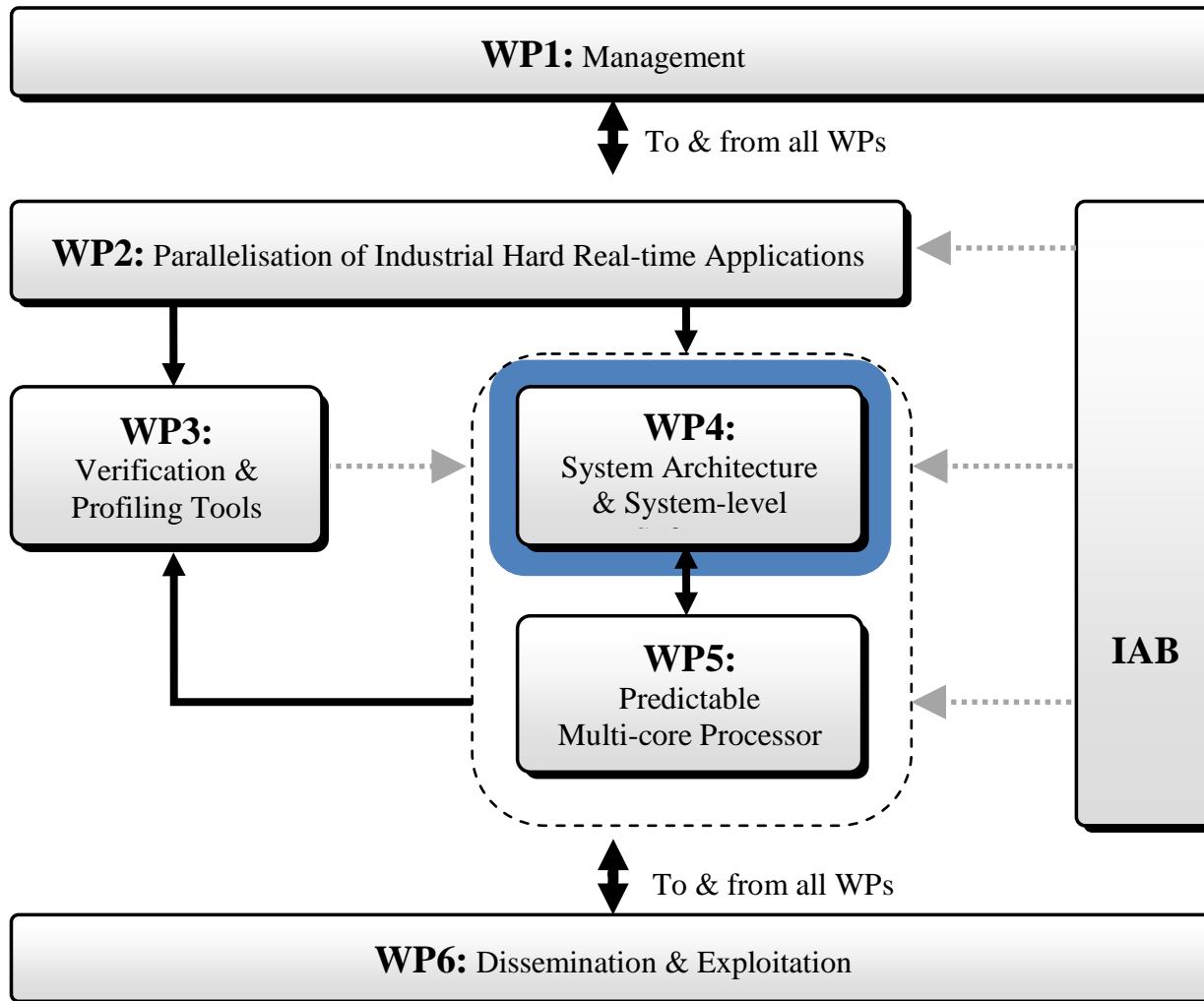
- **Phase 1 (Oct. 2011 – June 2012):  
Requirement Specification and Concept**  
Select applications, define requirements for system SW, tools, and multi-core design
- **Phase 2 (July 2012 – September 2013):  
Full Specification and Implementation of**  
System SW, tools, and multi-core design  
as well as parallelisation for maximum parallelism
- **Phase 3 (October 2013 – September 2014):  
Optimisation and Refinement**  
Agglomerate for optimal parallelism



## Use Cases for Parallelisation of Hard Real-time Applications

- **Avionics (Honeywell International s.r.o.)**
  - 3D Path Planning for airborne collision avoidance
  - Global Navigation Satellite System (GNSS)
  - Stereo Navigation for aircraft localization when in loss of GNSS
- **Automotive (DENSO AUTOMOTIVE Deutschland GmbH)**
  - Engine control for diesel fuel injection
- **Construction Machinery (BAUER Maschinen GmbH)**
  - Control algorithm for dynamic compaction machine



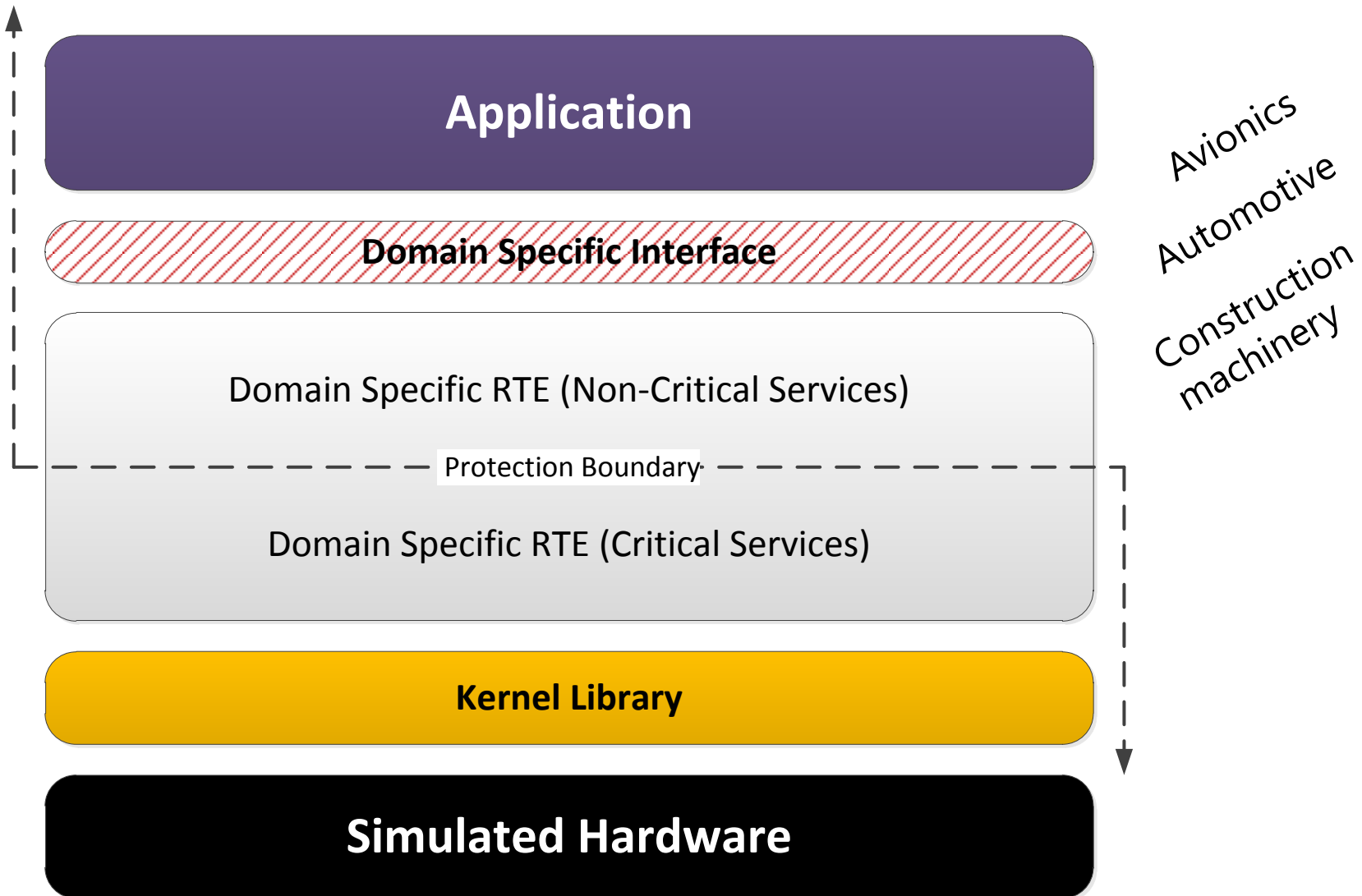


## **System Architecture and System-level Software**

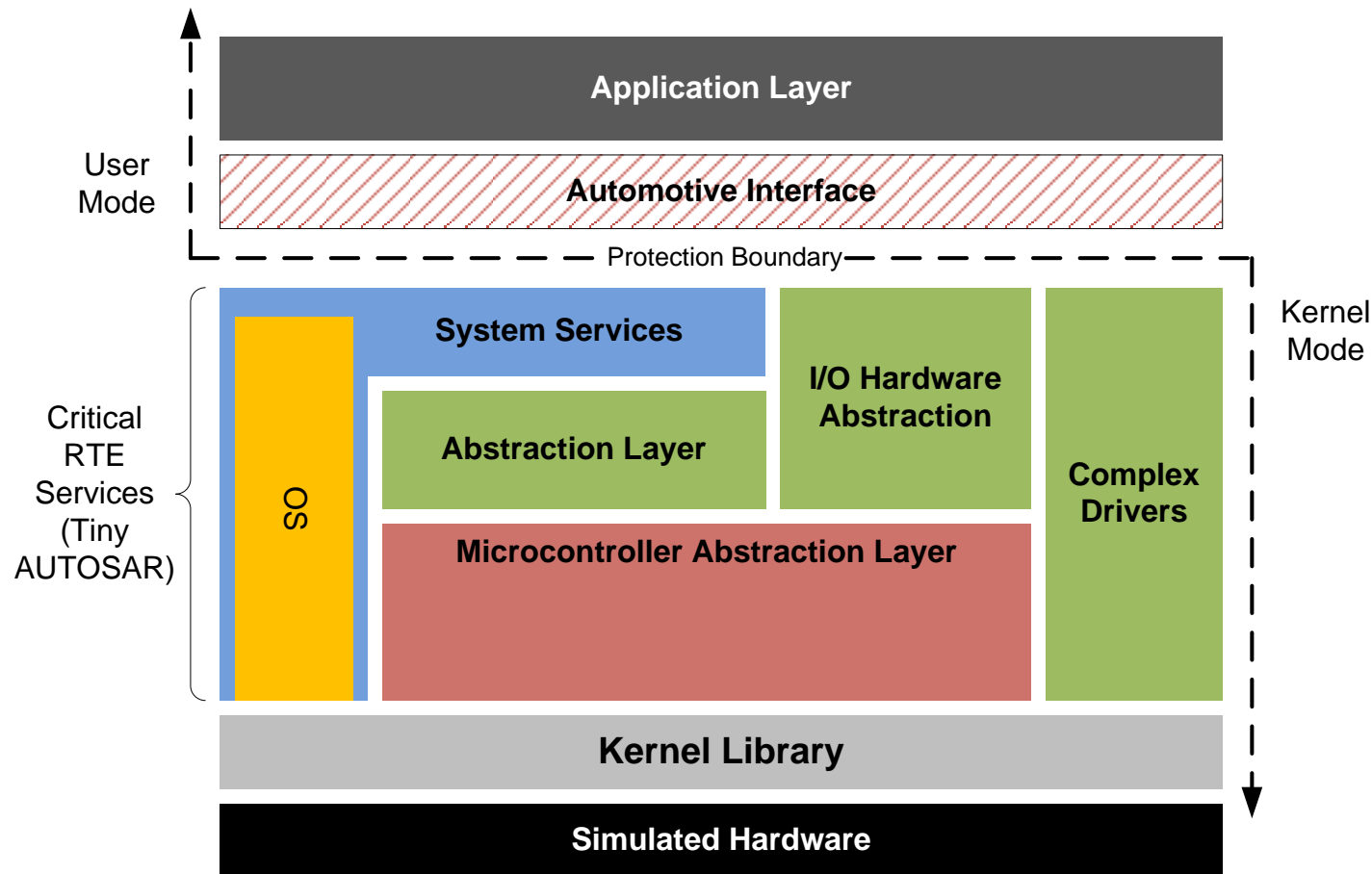
- System architecture with common kernel library for all three application domains
- TinyAUTOSAR, TinyIMA, Construction Machinery RTE

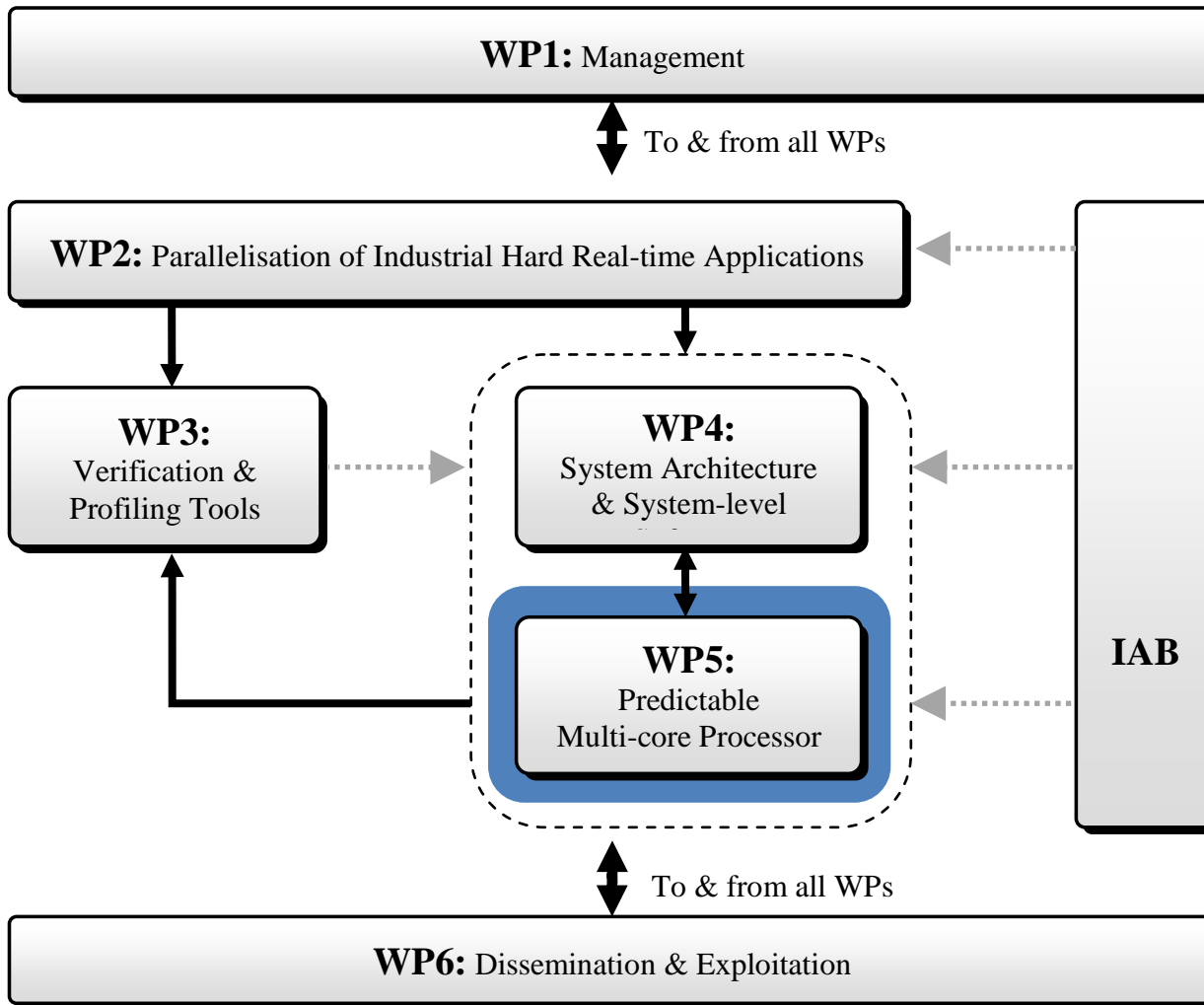
## **Approaches for Parallelisation of Hard Real-time Applications**

- Parallelisation approach and parallel pattern catalogue
- Parallelisation started towards suitable parallel design patterns in all application domains



- Example: Implementation of AUTOSAR stack

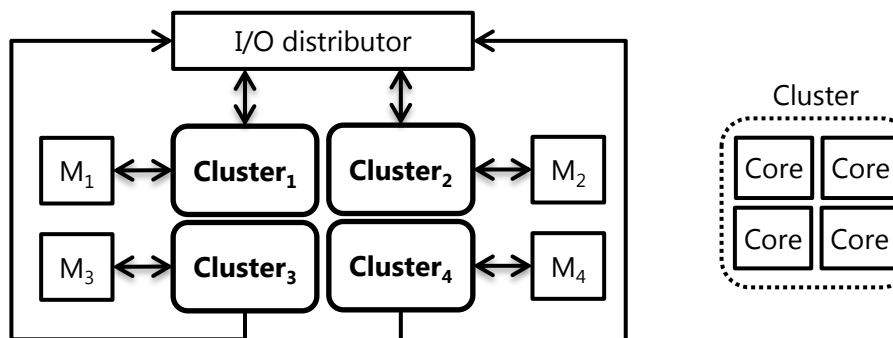




## Predictable Multi-core Processor

### ▪ parMERASA *Generic Many-core Architecture*

- Clustered many-core architecture based on simple cores and predictable interconnect
- New predictable NoC structures; new coherency cache defined

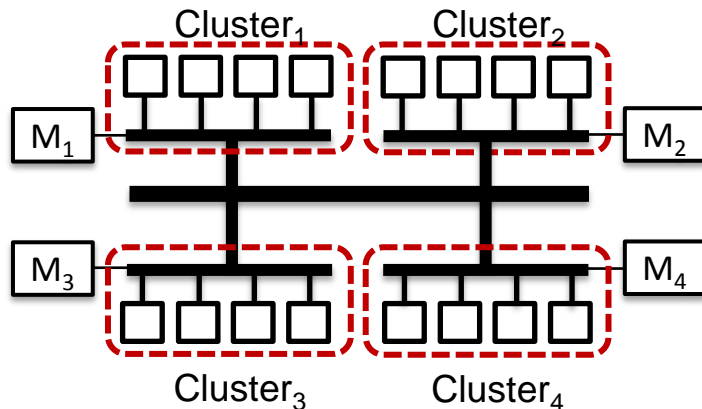


## parMERASA *Generic Many-core Architecture*

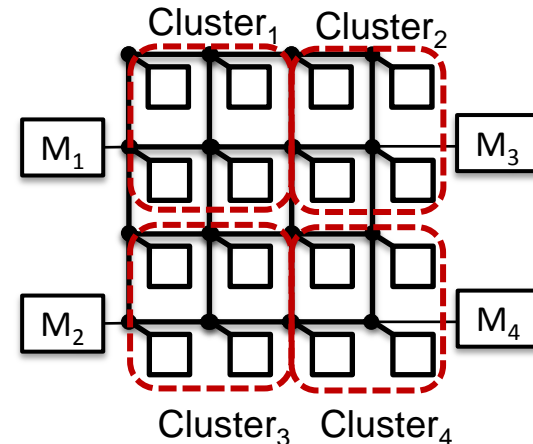
- Multiple topologies will be investigated
- Basic ideas:
  - Temporal isolation of clusters
  - One application per cluster
  - Interference-free coherency of memory accesses

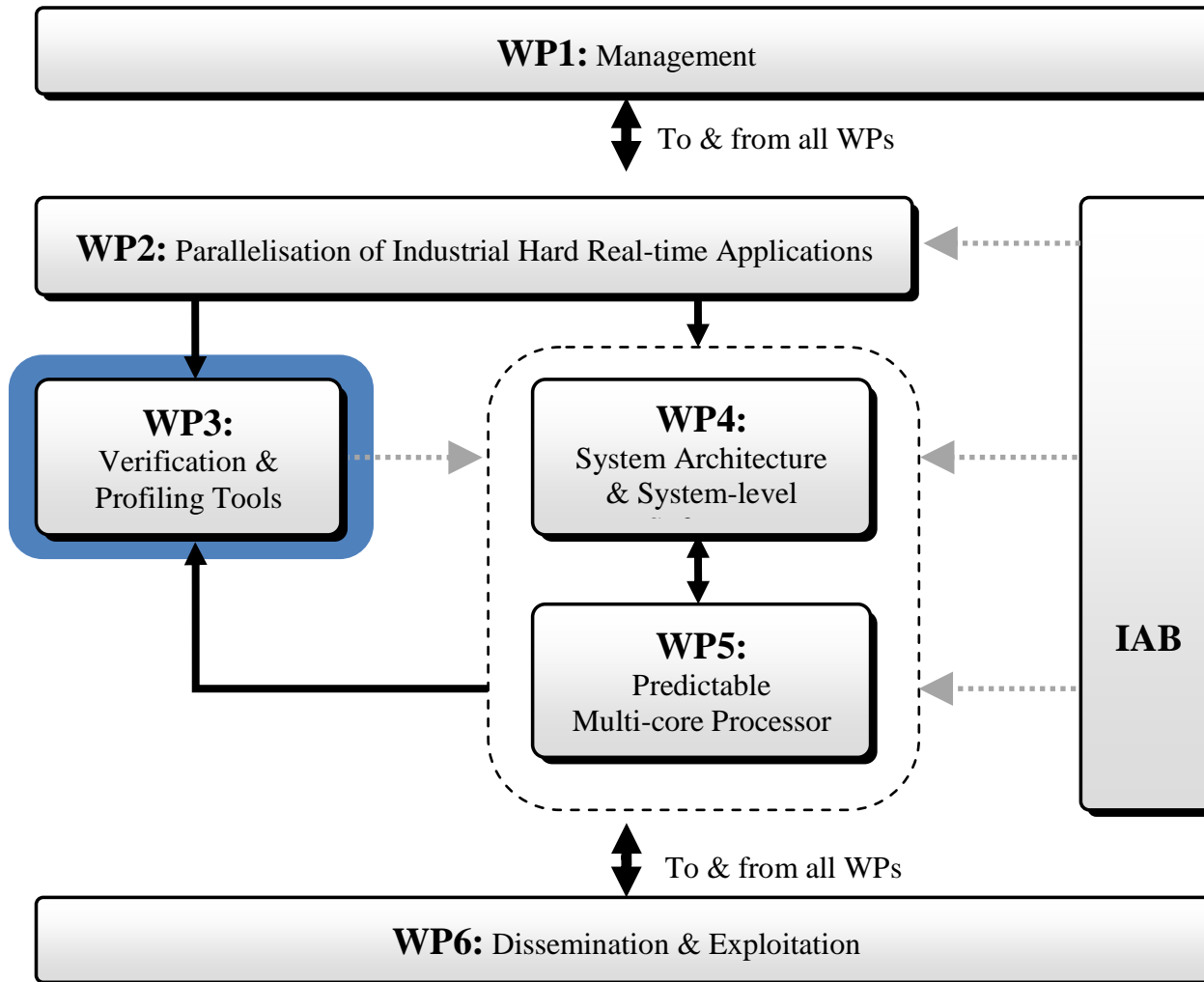
⇒ **Isolation as required by ISO26262 and ARINC653**

### Hardware Cluster Definition



### Virtual Cluster Definition







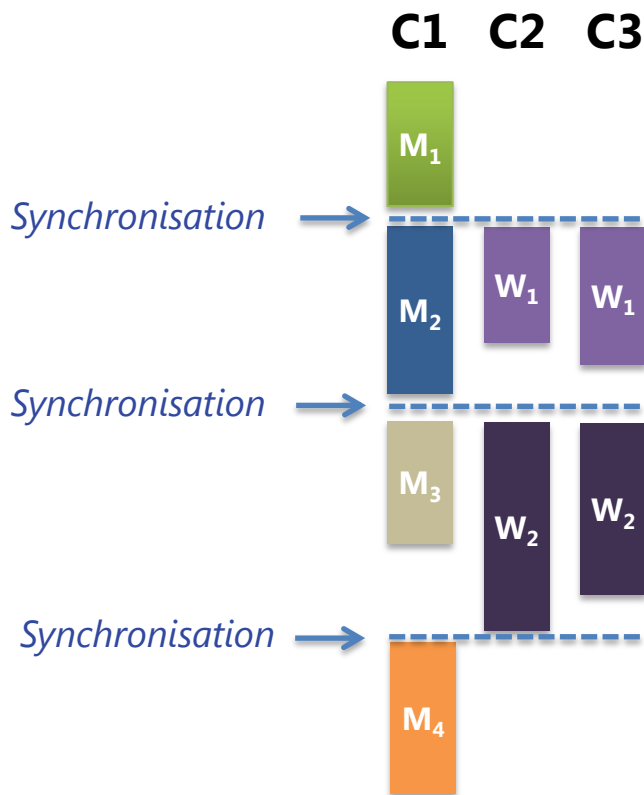
## Verification and Profiling Tools

- **OTAWA tool**

- specification of annotation format for source code annotations,
- analysis of synchronisation primitives

- **Five verification and parallelisation support tools**

- WCET analysis tool RapiTime enhanced for parallel programs;
- Tool to assist with the parallelisation of existing sequential software;
- Visualisation and profiling tool for parallel programs;
- On-target code coverage tool for parallel programs;
- Memory, cache and stack analysis tool for parallel programs.



Worst-case guarantee

```
global_wcet =
  wcet(M1)
+ max (wcet(M2), wcet(W1))
+ max (wcet(M3), wcet(W2))
+ wcet(M4)
```

- Separation of execution time and waiting time
- Synchronisation functions must be timing predictable
- Waiting time for synchronisations must be bounded

- Current multi-core architectures and parallel application software is **not time predictable**
- parMERASA targets **parallelising hard real-time programs** to run on predictable multi-/many-core processors, including **WCET analysis tools**.
  - Baseline timing analysable many-core architecture defined
  - Cross-domain lower layer RTE implemented
    - Upper layers under development
  - Analysis tools under development
    - Static WCET analysis
    - Analysis of parallelism and interdependencies

⇒ **parMERASA will provide new techniques for future parallel hard real-time applications**

Make **timing predictable** techniques  
commercially feasible to **increase safety**  
in **avionics, automation** and  
**automotive** domains!