POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA,
INFORMAZIONE E BIOINGEGNERIA

HEAP LAB

EMBEDDED SYSTEMS
Prof. William Fornaciari

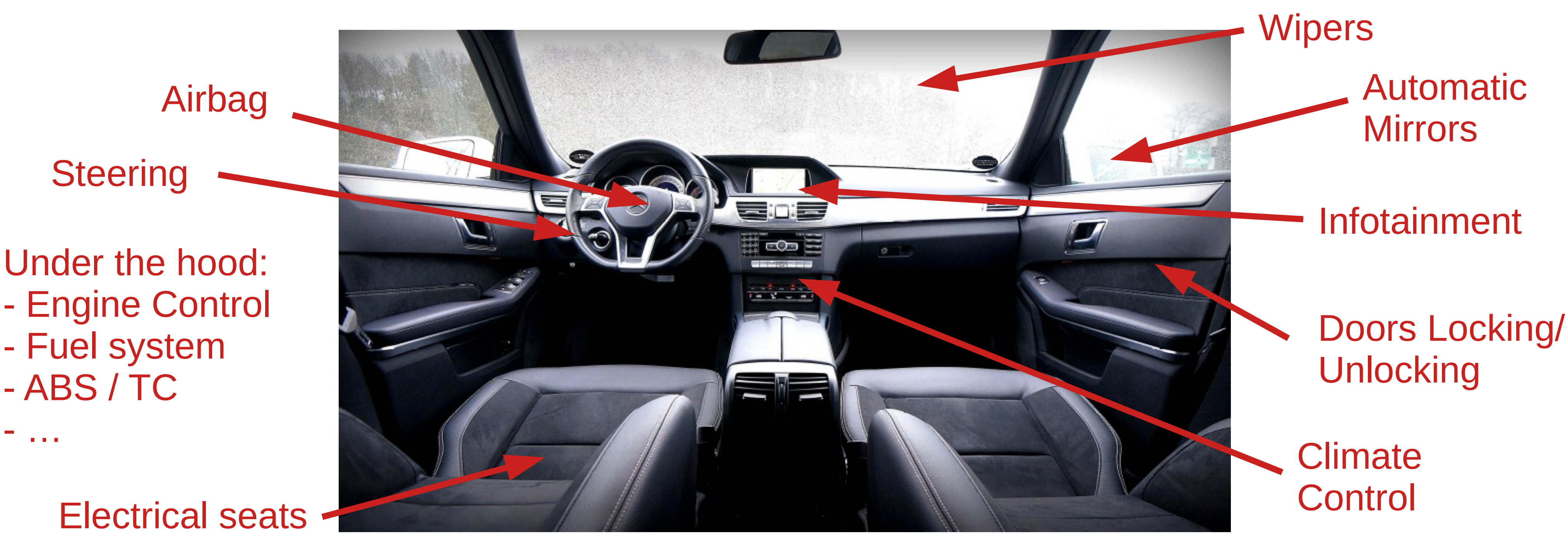
Mixed-Criticality Systems

William Fornaciari
<william.fornaciari@polimi.it>

Complex embedded systems

Modern cyber-physical systems are complex and contain several embedded computing nodes

- *Example*: a modern car contains more than 50 computing systems, from simple microcontrollers to multi-core Linux systems.



Complex embedded systems

Problems

- **Development and production costs**
  - Designing and building multiple unit requires multiple design processes and replicated software and hardware components
- **Resource under-utilization**
  - At any time instance, many systems are often in idle and their computational capabilities are wasted
- **Fault-tolerance**
  - Critical systems require fault-tolerant (FT) techniques and sometimes hardware replication and voting strategies.
  - Multiple critical systems require multiple FT implementations.

Complex embedded systems

Problems

- **System interoperability**
  - Which communication medium and protocol to use to enable multiple units to exchange information?
  - How to synchronize the distributed system?
  - The complexity of both increases exponentially with the increasing number of systems.
  - The problem is much more complex than intra-system communication and synchronization.

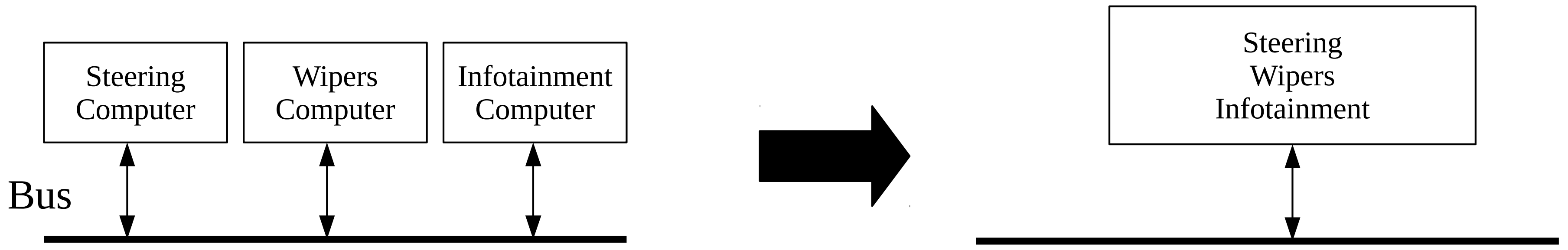
Systems' Consolidation

Possible solution

- Consolidate multiple systems onto a single powerful platform that performs all the required tasks.

- *Example*
  - We decide to consolidate the "Steering", "Wipers", and "Infotainment" computers into a single computational platform

    (This is not a realistic example, but it is for the sake of simplicity and for demonstration purposes in the next slides)



Systems' Consolidation

Advantages

- Lower design costs
- Better optimization of Power and Energy consumption
- Improved reliability of the overall system,
- Lower maintenance costs
- Reduced external bus congestion

Disadvantages

- Increased shared resource contention inside the computing node
- Large integration effort required
- Different applications may have different criticality levels.
  - How to manage them?

Mixed-Criticality Systems

Levels

- A mixed-criticality system is a system running tasks with different **criticality levels**.
  - *Example*: Automotive Safety Integrity Level (ASIL) – ISO 26262

Level	Impact of failure	Required failure rate
ASIL-D	Fatal / Survival uncertain	10^{-7} - 10^{-8} / h
ASIL-C	Fatal / Survival uncertain	10^{-6} - 10^{-7} / h *
ASIL-B	Severe injury - Controllable	10^{-6} - 10^{-7} / h
ASIL-A	Slight injury - Controllable	10^{-5} - 10^{-6} / h
QM (Quality Management)	No Effect	none

* Additional fault-tolerance techniques mandatory compared to ASIL-B

Mixed-Criticality Systems

Back to the car example

ASIL-A

Wipers

Automatic Mirrors

Infotainment

Doors Locking/ Unlocking

Climate Control

QM

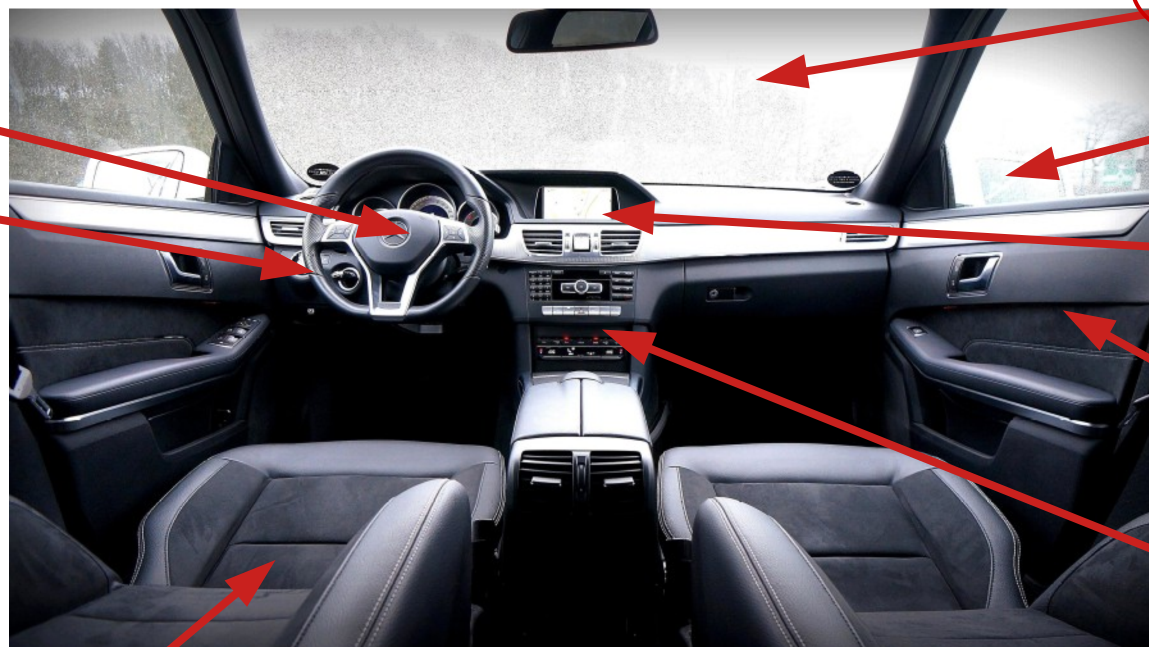
ASIL-D

Airbag

Steering

Under the hood:
- Engine Control
- Fuel system
- ABS / TC
- ...

ASIL-B or ASIL-C



Electrical seats

QM

Mixed-(Time) Criticality systems

Real-time requirements

- How to guarantee real-time requirements in mixed-criticality systems?
- Our example of mixed-criticality system contains:
  - *Steering* – Strictly hard real-time
  - *Wipers* – Possibly hard real-time
  - *Infotainment* – Soft real-time / Not real-time

- *Trivial solution*
  - Consider all the tasks with the maximum level of assurance, i.e. considering them hard real-time.

    Unfeasible: it may require too large computational power and/or it may be impossible to compute the WCET for Infotainment tasks

Task model

Mixed-criticaly task model

- To manage such tasks, mixed-criticality task model and scheduling algorithms has to be be used
- In the previous slides we have seen a task has many properties, including the WCET $\mathbf{C_i}$
- In mixed-criticality systems, in addition to the previous metrics, we have:
  - A **criticality level** χ_i:

    It can be expressed in many form, e.g. ASIL levels χ_i = {D, C, B, A, GM}
  - A **set of WCET** (not a single value) for each criticality level

Task model

Example

Task	X_i	WCET
Steering	D	$C_i(QM) = 2$, $C_i(A) = 5$, $C_i(B) = 7$, $C_i(C) = 8$, $C_i(D) = 10$
Wipers	A	$C_i(QM) = 5$, $C_i(A) = 7$
Infotainment	QM	$C_i(QM) = 100$

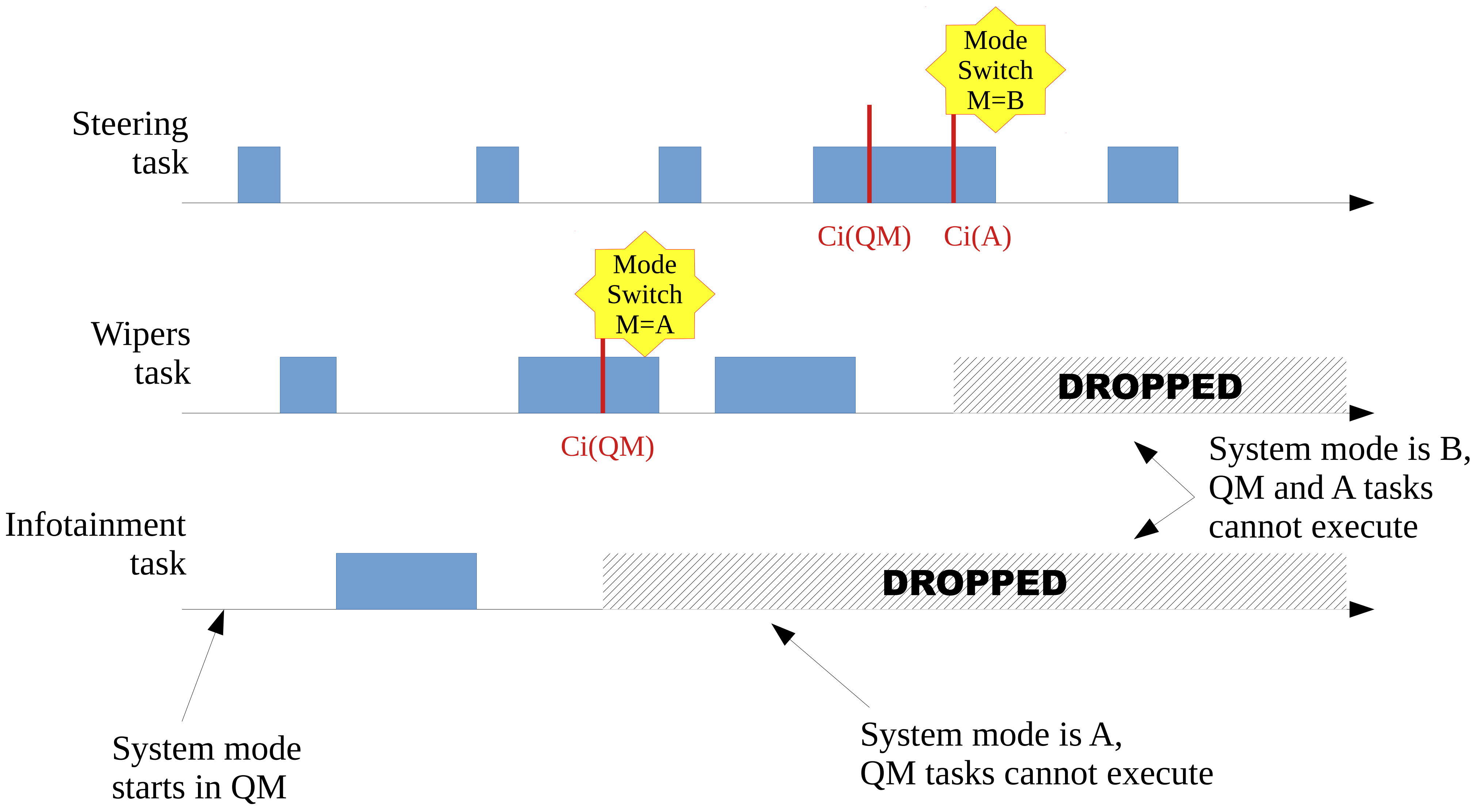
- Each task has as many WCETs as the value of its criticality level
  - Tasks with QM have one WCET value,
  - Tasks with A have two WCET values, and so on
- Only $C_i(D)$ is assumed to be a non-underestimated WCET, while other WCETs are possibly unsafe, thus only D tasks are guaranteed to be scheduled

Traditional mode-switch concept

Behaviour

- When a task overrides one of its WCET, a **system mode switch** occurs
- The system mode M can assume the same values of the criticality levels
- If the system is in the criticality mode m, i.e. M=m, all tasks with criticality level lower than M are not scheduled and **dropped**
- This allows tasks with higher criticality levels to have more time to execute and complete within their deadline

Traditional mode-switch concept



Traditional mode-switch concept

Strategies

- Different strategies exist to decide when to switch back to normal mode QM
  - ➔ Example: as soon as the system is in idle

- **Important**: ASIL-D tasks are always scheduled and cannot be dropped: no task can overrun its $C_i(D)$

- The WCET values $C_i(x)$ shoud be designed such that...
  - ➔ The ASIL-D tasks are executed in any condition
  - ➔ Mode switches occur very rarely, to minimize the disruption to other tasks

Scheduling mixed-criticality systems

Scheduling algorithms

- Many scheduling techniques have been developed in the last years
- The problem is non-trivial and no optimal algorithms exist
- The most famous one is a modified version of EDF called the **EDF-VD** (Virtual Deadlines)

- Further reading
  - https://www-users.cs.york.ac.uk/burns/review.pdf

Questions?

Contacts

- William Fornaciari
  DEIB Politecnico di Milano
  Building 20, 1st flloor
  william.fornaciari@polimi.it

- **HEAPLab**
  DEIB Politecnico di Milano
  Building 21, Floor 1
  Phone: 9613 / 9623
  http://heaplab.deib.polimi.it/

- **Slides credits**:
  - Federico Reghenzani <federico.reghenzani@polimi.it>