TUM CREATE

Schedule Integration for Time-Triggered Systems

Outline

Motivation
- Automotive software
- Automotive architectures
- Integration Challenge

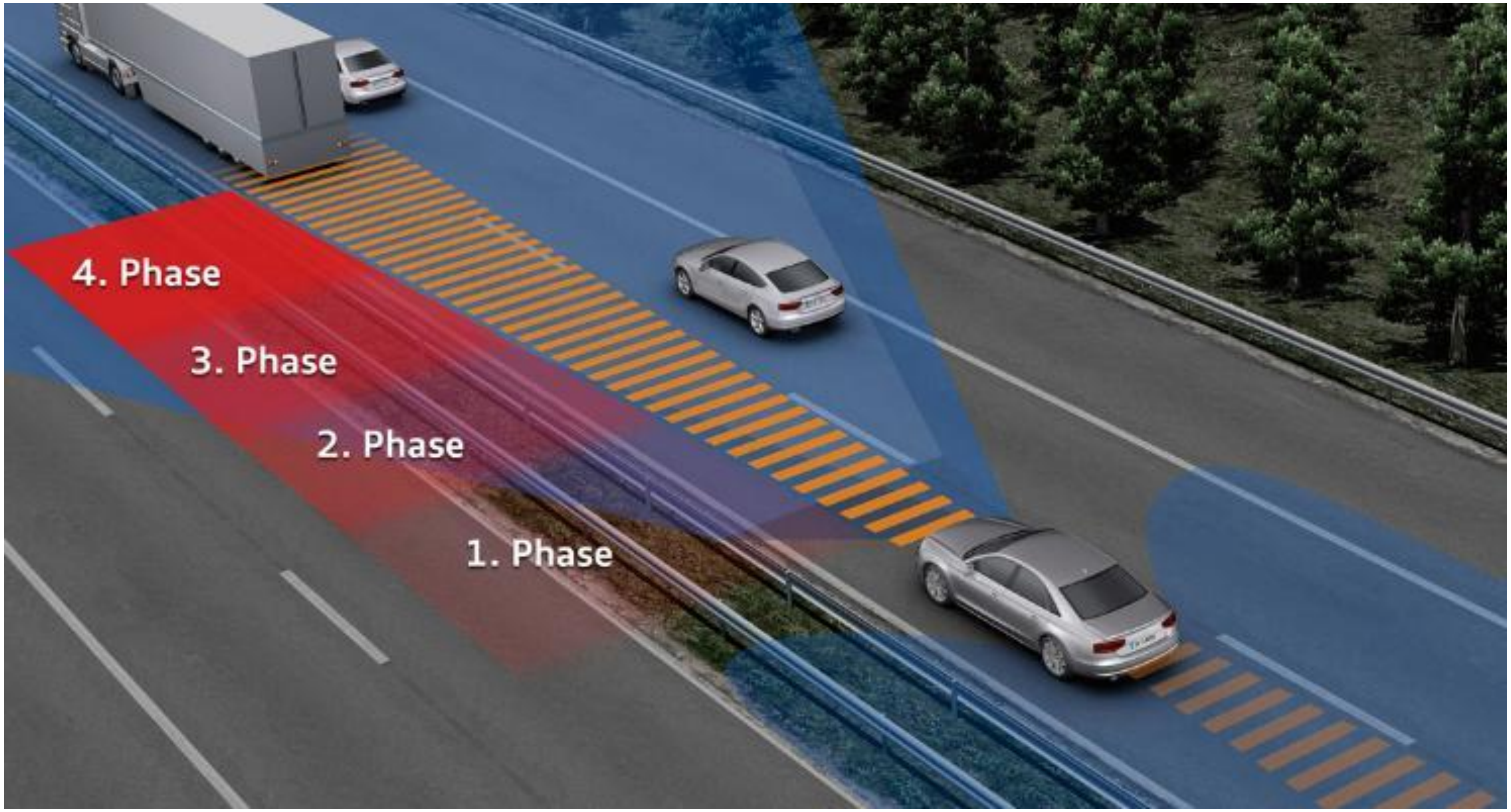
Time-triggered automotive systems
- Sychronization
- Schedule Integration
- FlexRay

ILP scheduling approach

Case study & Scalability

Conclusion

Example: Adaptive Cruise Control

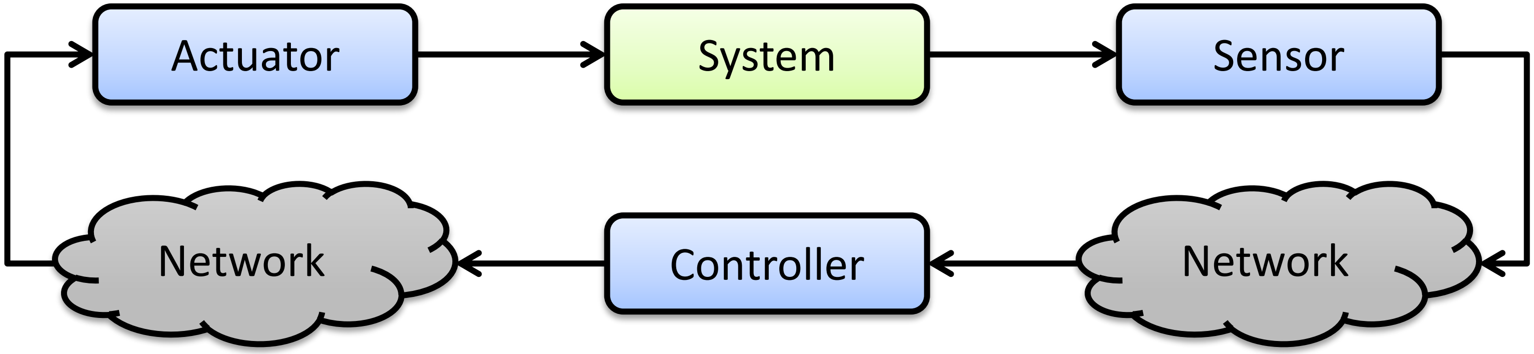


(AUDI AG, ACC)

Distributed Control System

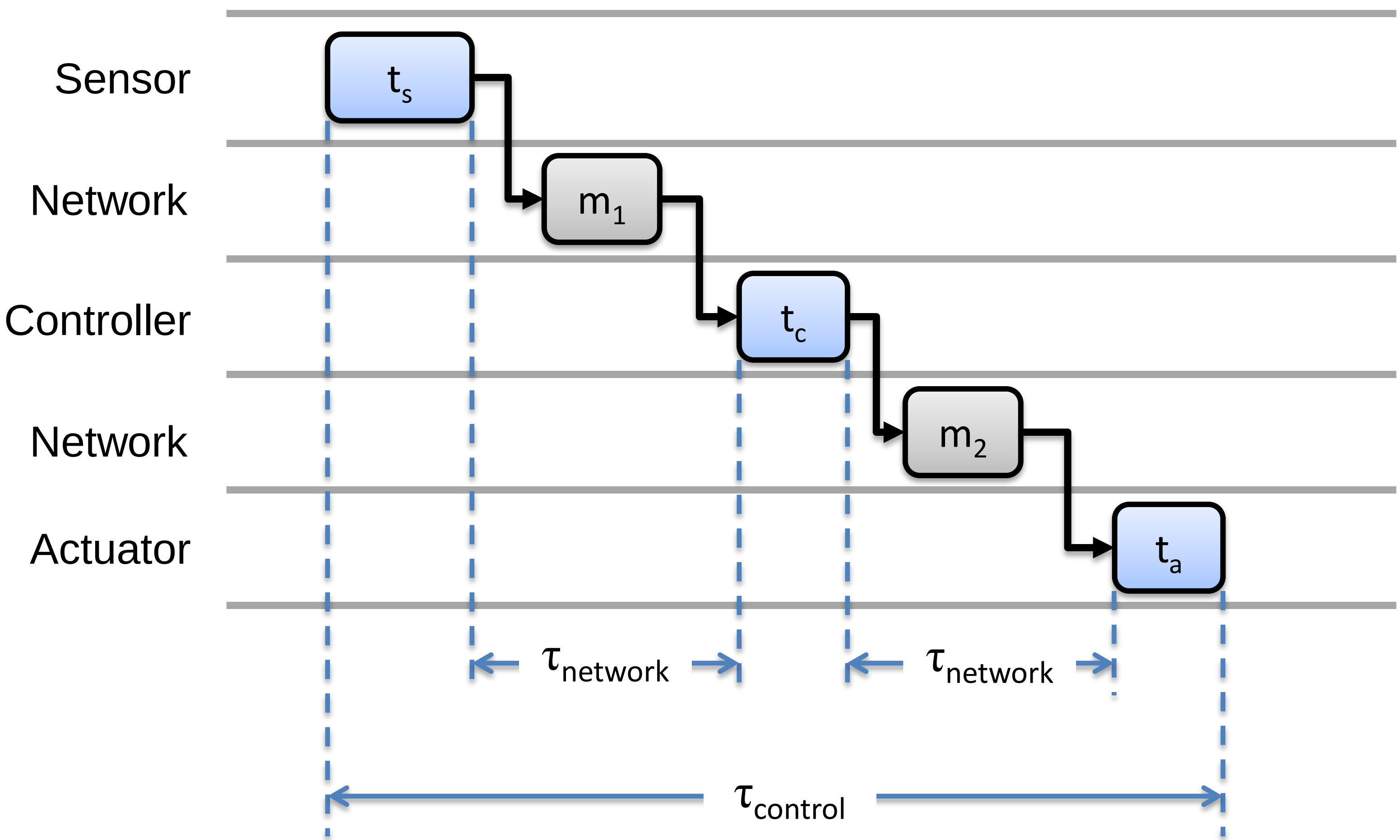
Speed Adjustment

Radar

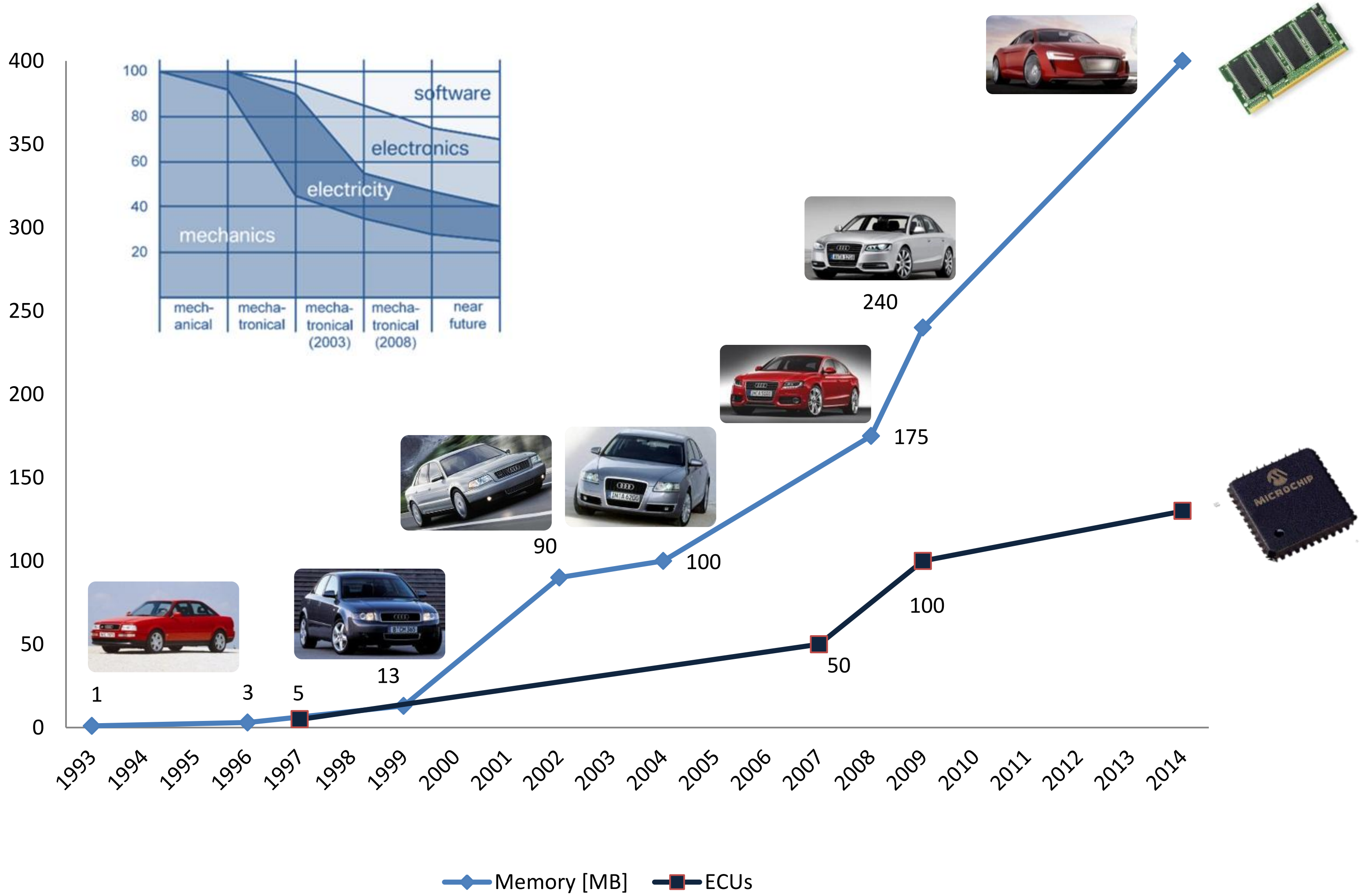


ACC ECU

Delay Requirements



Increasing Complexity in Automotive Electronics

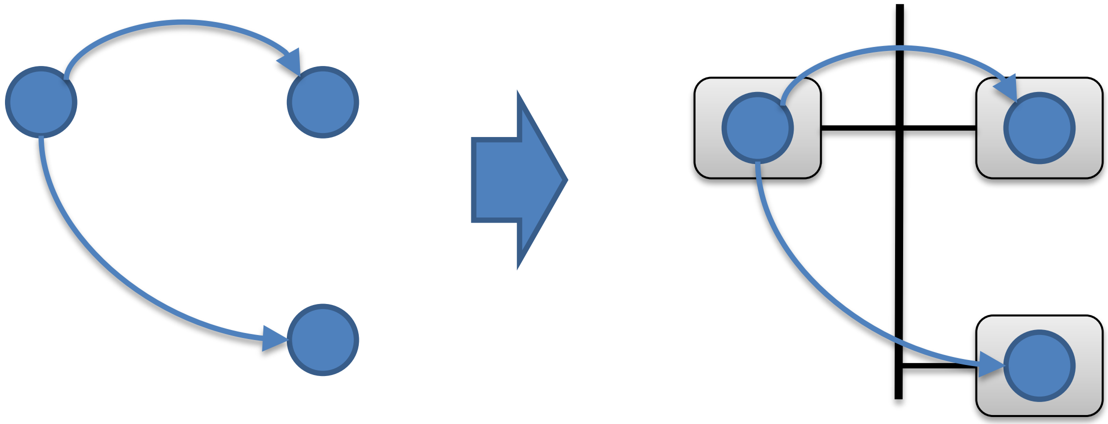


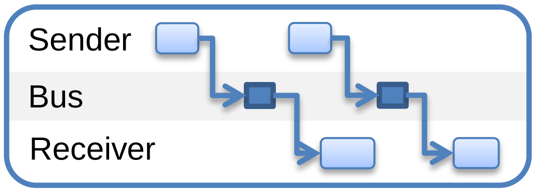
Sources:
Paul Milbredt, AUDI AG, EFTA 2010 - Switched FlexRay: Increasing the Effective Bandwidth and Safety of FlexRay Networks
BMW Group, FTF 2010 Orlando - Energy Saving Strategies in Future Automotive E/E Architectures

Integration Challenge

Application:

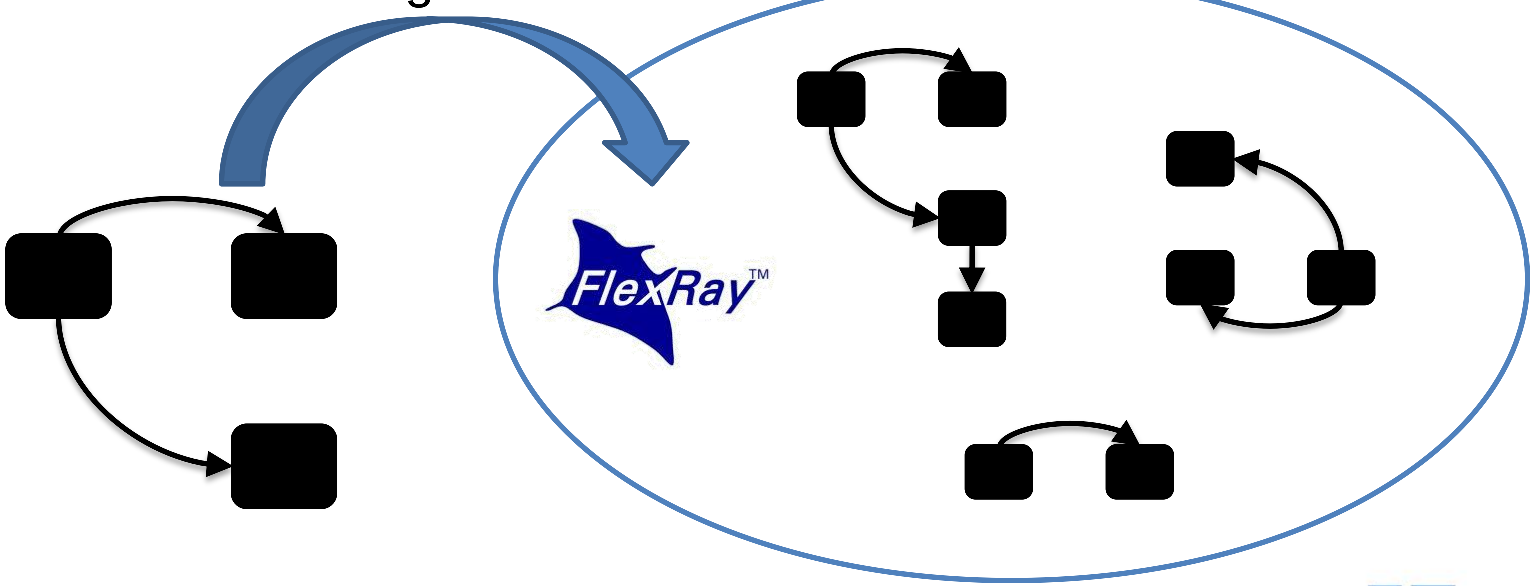
Implementation:





Mapping, Configuration and Schedule

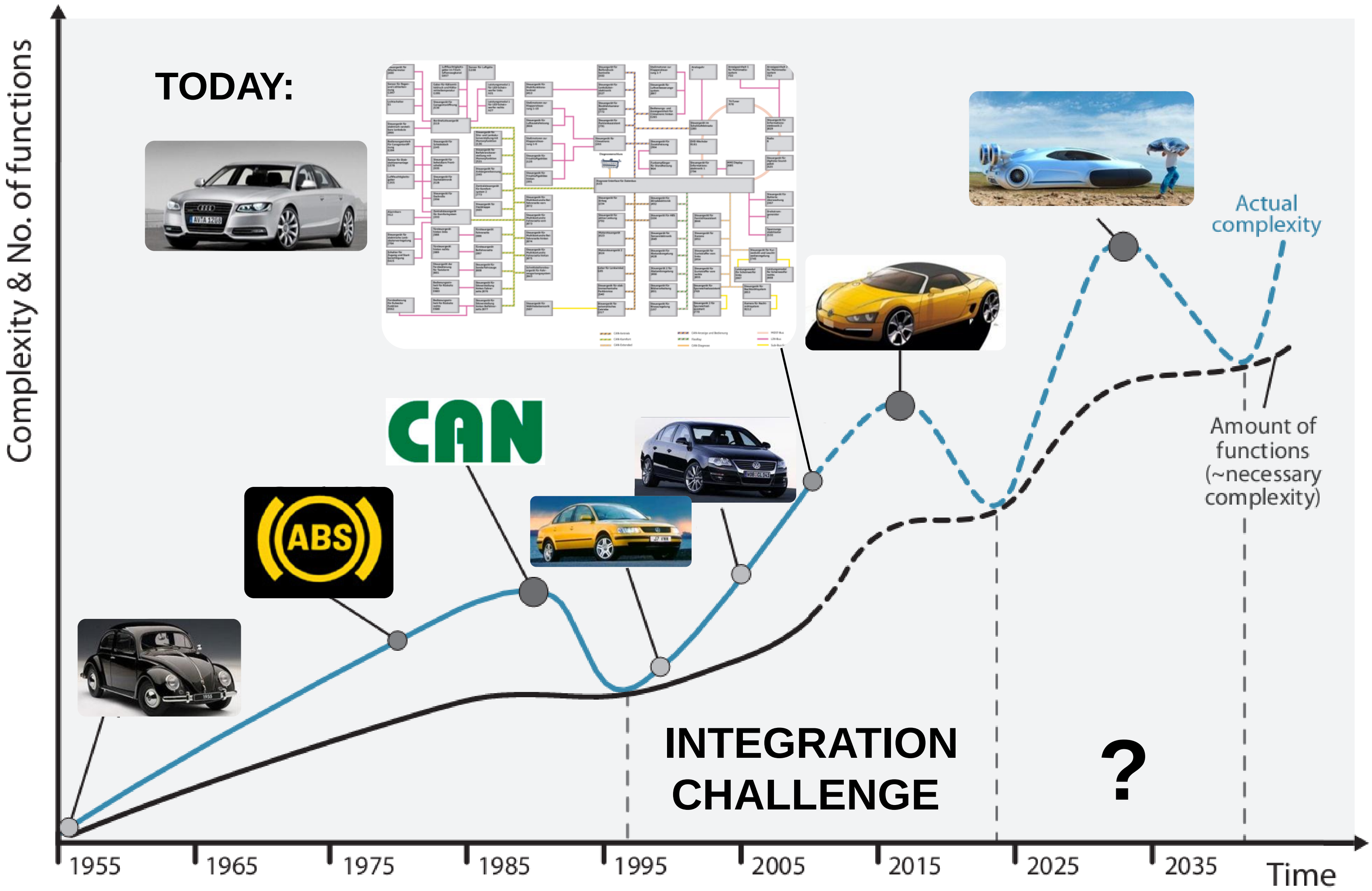
Integration:



Defined by:

- Message
- Timing
- Deadline

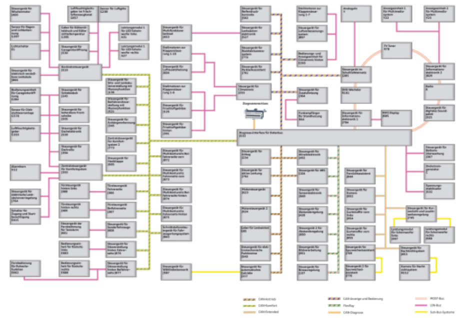
Rising Gap between Required and Actual System Complexity



Source: The Software Car: Information and Communication Technology (ICT) as an Engine for the Electromobility of the Future – ForTISS GmbH

Paradigm Shift

Event-triggered systems

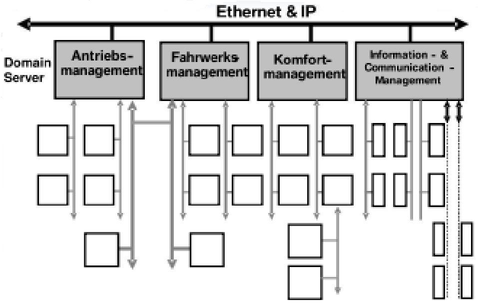


CAN

utilization
flexibility

(real-time properties need to be verified)

Time-triggered systems



FlexRay™

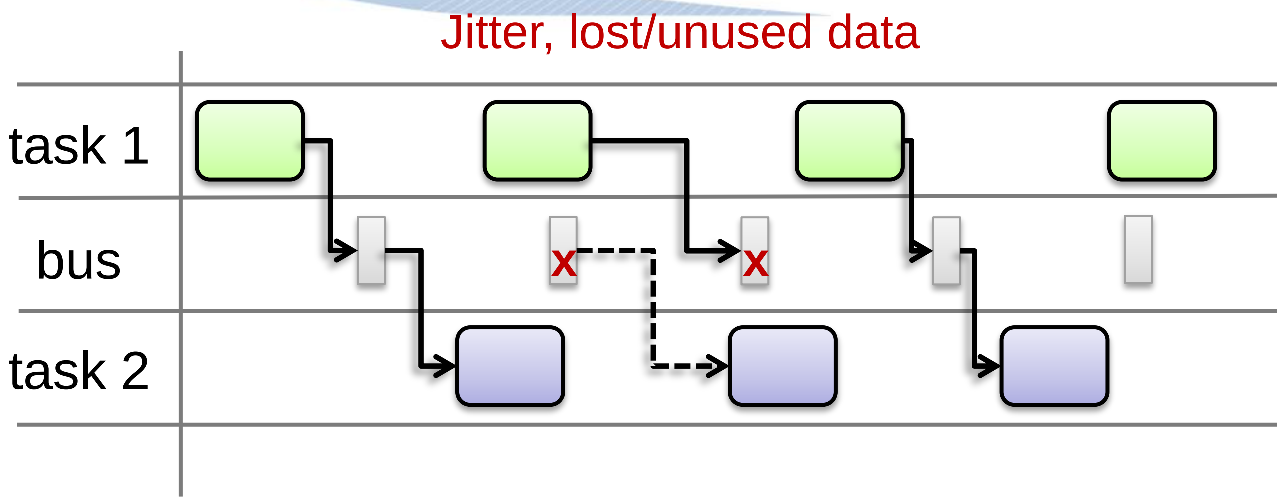
Ethernet (PTP)

predictability

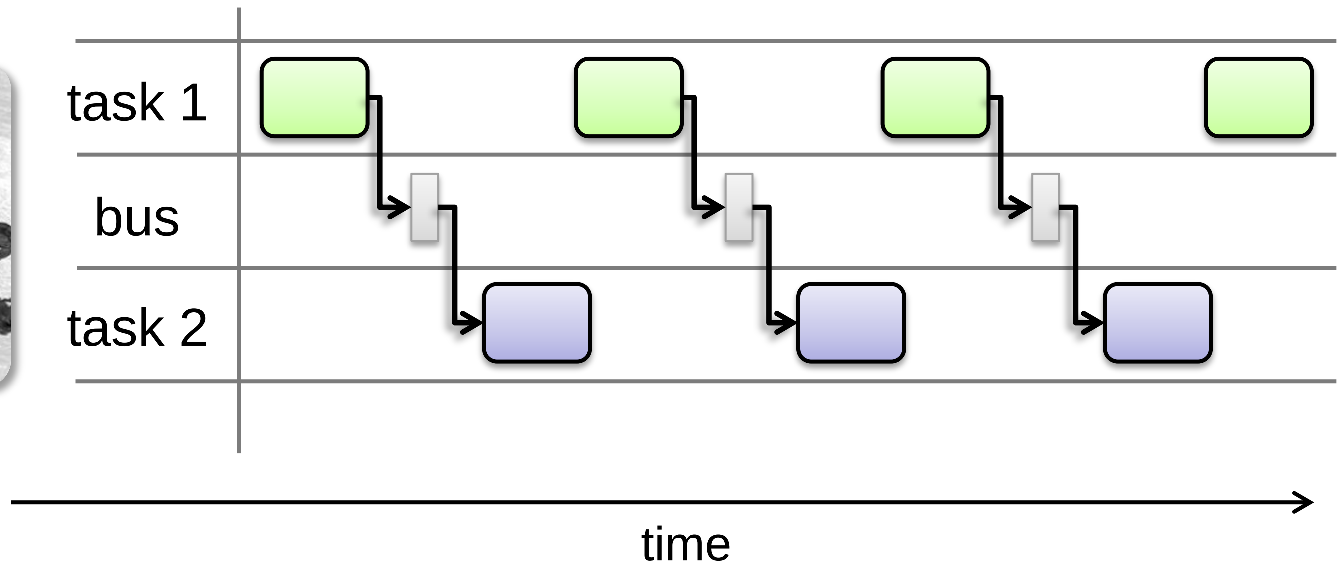
(real-time properties are guaranteed)

Introduction: Asynchronous vs. Synchronous

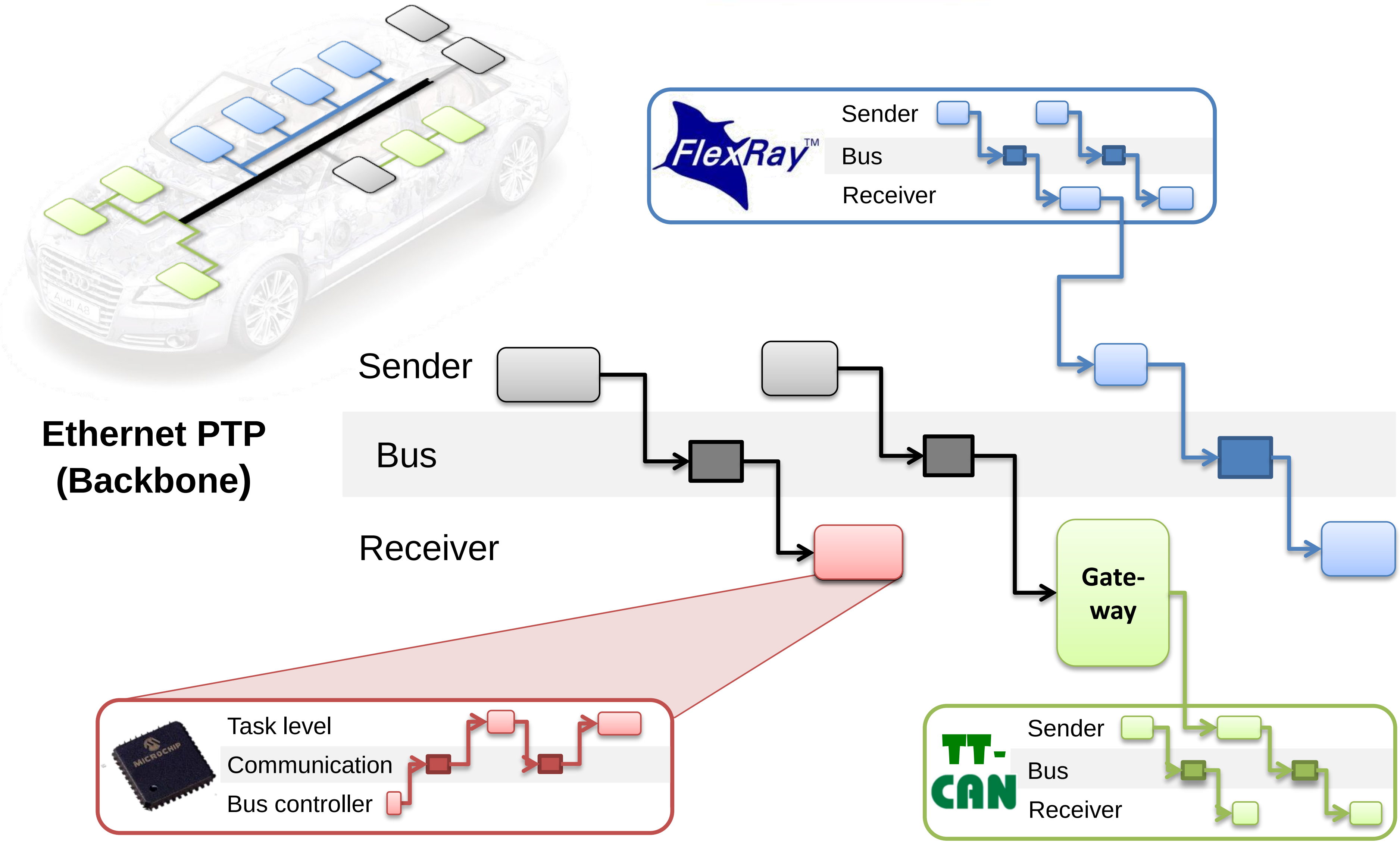
Asynchronous



Synchronous



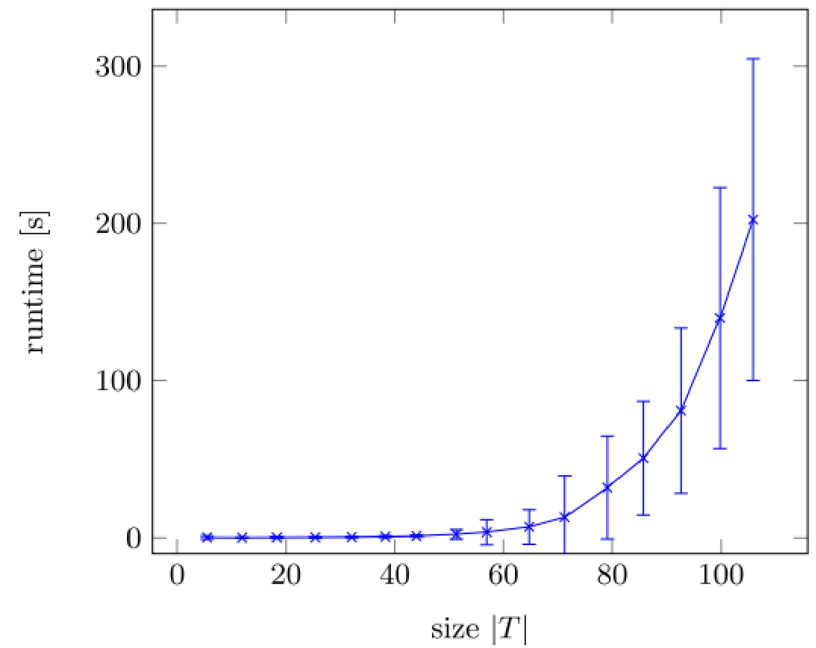
Vision: Holistic Scheduling for Time Triggered Systems



Related Work: Global Scheduling

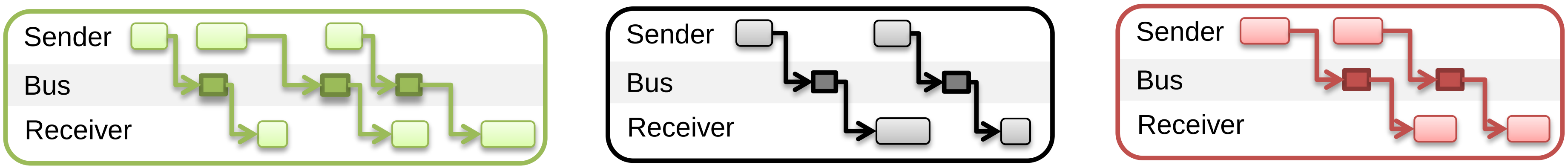
- **Schedule Optimization of Time-Triggered Systems Communicating Over the FlexRay Static Segment**
  H. Zeng, M. Di Natale, A. Ghosal, and A. Sangiovanni-Vincentelli.
  IEEE Transactions on Industrial Informatics, 7(1):1–17, 2011.

- **Modular Scheduling of Distributed Heterogeneous Time-Triggered Automotive Systems.**
  M. Lukasiewycz, R. Schneider, D. Goswami, and S. Chakraborty.
  In Proceedings of ASPDAC 2012, pages 665–670, 2012.

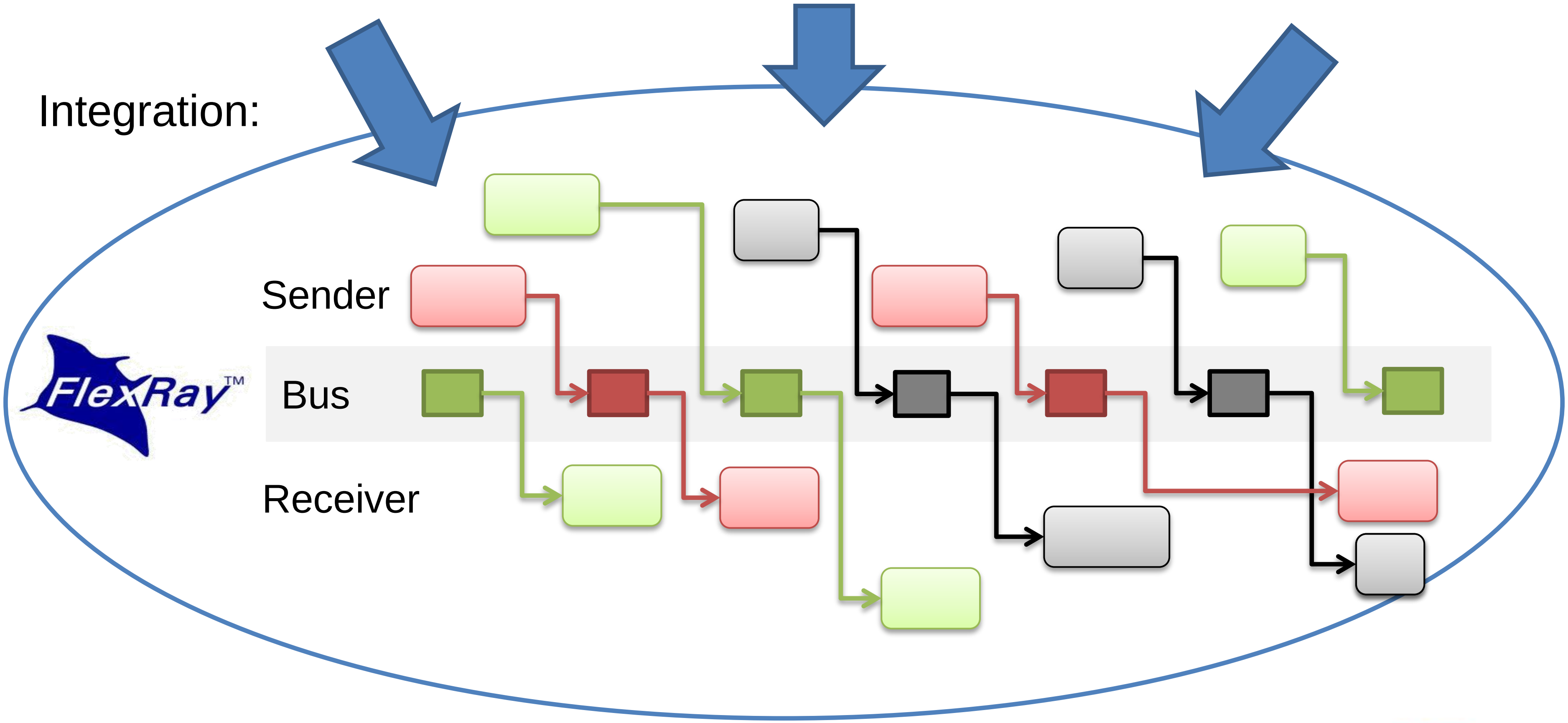


Schedule Integration approach

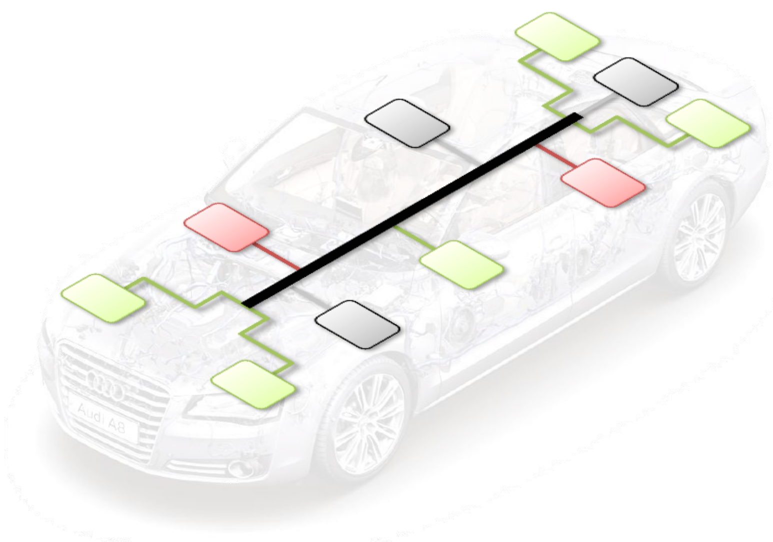
Cluster Schedules:



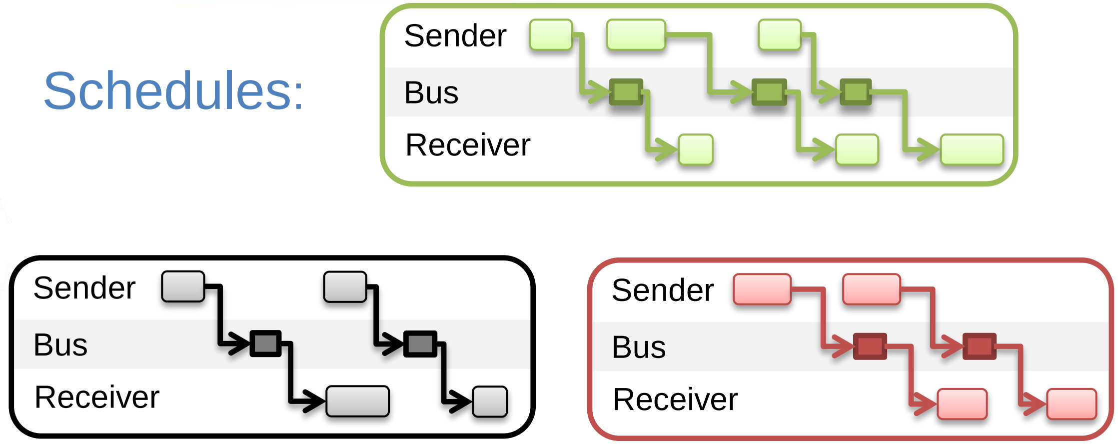
Integration:



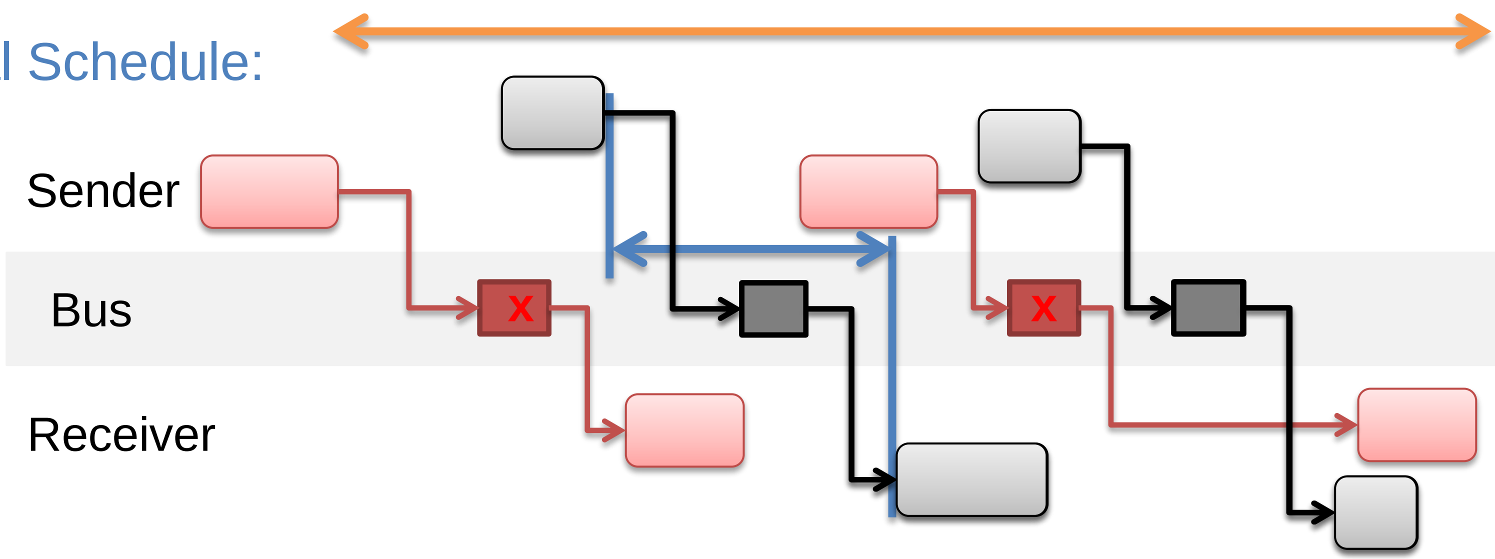
Degrees of Freedom



Schedules:



Global Schedule:



Introduction to FlexRay

FlexRay™

Foundation of FlexRay Consortium
(more than 100 members by 2005)

2000

2005

Release of version 2.1, the current industry standard

FlexRay™ 2.1

Presentation: "FlexRay – Past Present Future Perfect" - Christopher Temple

Used in active damping system of BMW X5

2006

2008

Fully introduced with BMW 7 series

BMW 7-series

FlexRay™ 3.0

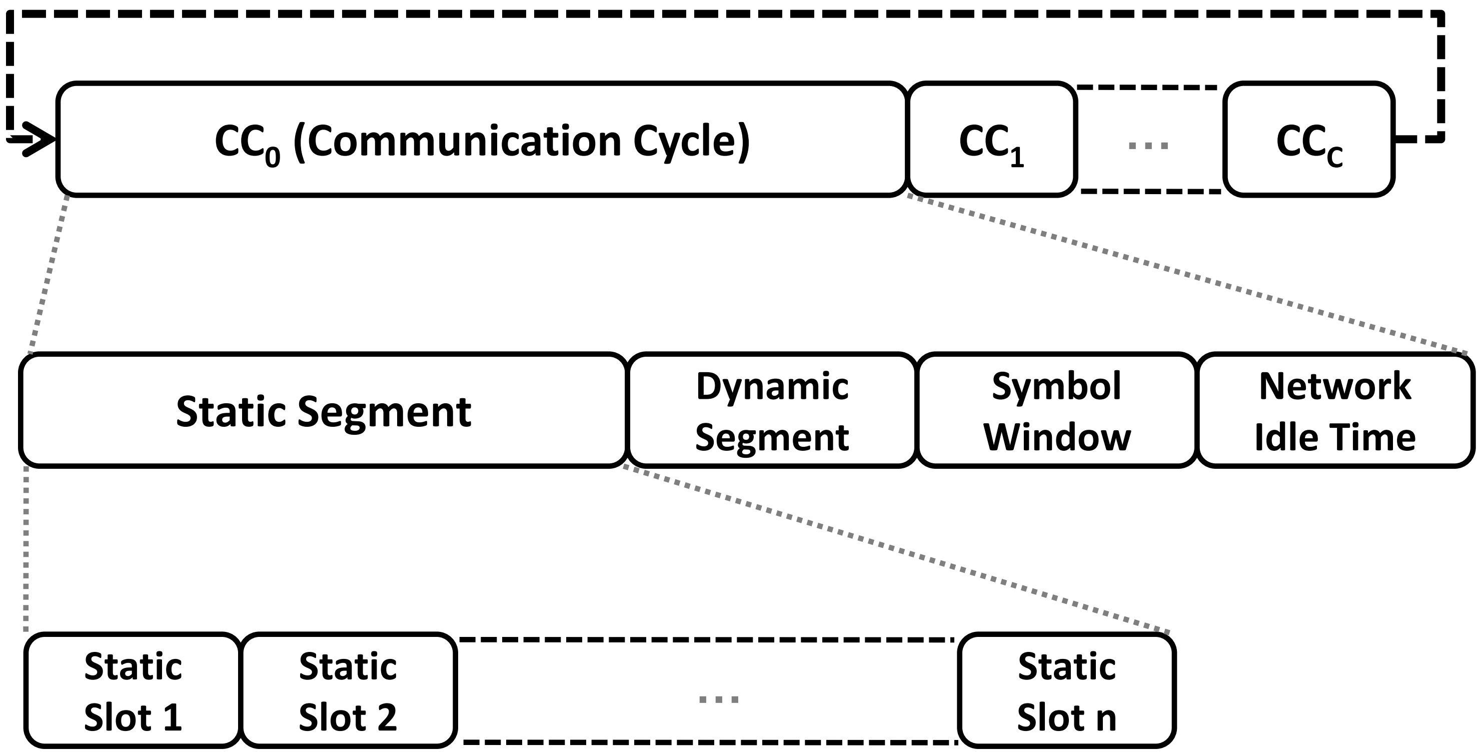
Release of version 3.0, Disbandment of consortium

2010

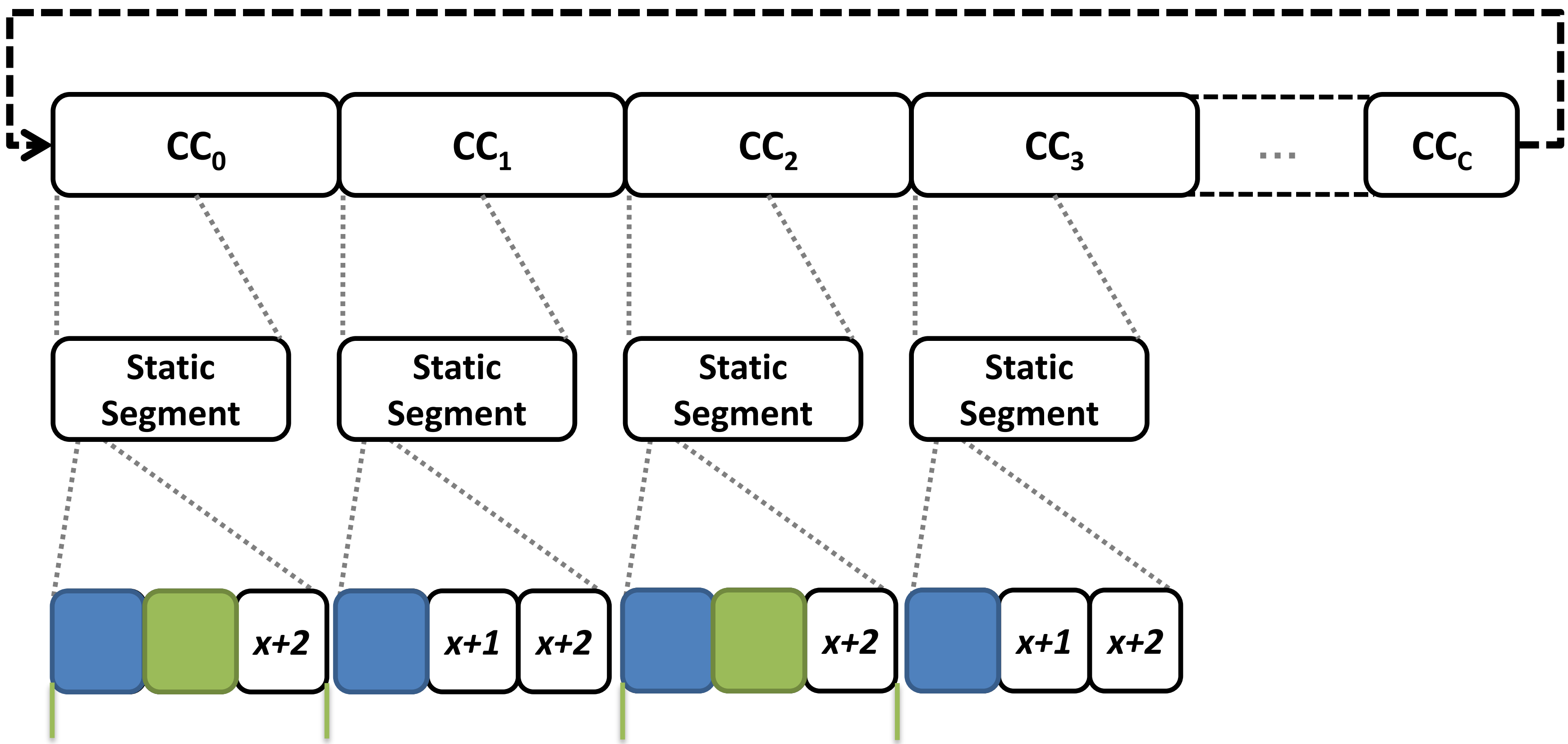
future

ISO standard in preparation

FlexRay Protocol (Static Segment)



Schedule Integration for FlexRay



ILP-based Schedule Integration Approach

$\forall D \in \mathcal{D}$:

$$\sum_{i \in o(D)} \mathbf{o_{D,i}} = 1 \quad (9)$$

$\forall D \in \mathcal{D}, f \in D$:

$$\sum_{i \in v(f)} \mathbf{o_{f,i}} = 1 \quad (10)$$

$\forall D \in \mathcal{D}, f \in D, i \in v(f), j \in o(D)$:
$\exists (s(f) + j + i)\% n_{all} \geq n_{static}$:

$$\mathbf{o_{f,i}} + \mathbf{o_{D,j}} \leq 1 \quad (11)$$

Only occupy static FlexRay segment

$\forall D \in \mathcal{D}, f \in D, i \in v(f), j \in o(D)$,
$(s(f) + j + i)\% n_{all} < n_{static} : c \in c(f)$,
$\tilde{c} = (((s(f) + j + i) + c \cdot n_{all})\%(n_{all} \cdot n_{cycles}))$:
$\exists \tilde{c} < n_{all} \cdot n_{rep}$:

$$\mathbf{o_{f,i}} + \mathbf{o_{D,j}} - \mathbf{x_{D,f,\tilde{c}}} \leq 1 \quad (12)$$

$c \in \{0, .., n_{rep} \cdot n_{all} - 1\}, c\% n_{all} < n_{static}$:

$$\sum_{D \in \mathcal{D}} \sum_{f \in D} \mathbf{x_{D,f,c}} \leq 1 \quad (13)$$

Each static slot is only occupied once

Search Space Reduction:

Filter Offsets:

$\forall i \in o(D), f \in D$:
$\exists j \in v(f), i + j + s(f)\% n_{all} < n_{static}$:

$$i \in \tilde{o}_f(D) \quad (14)$$

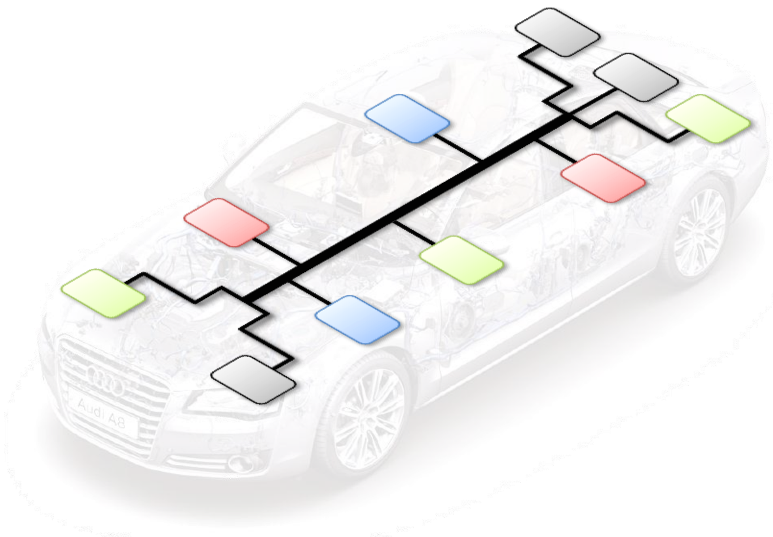
$$\tilde{o}(D) = \bigcap_{\forall f \in D} \tilde{o}_f(D) \quad (15)$$

Cycle Breaking:

$\forall D \in \mathcal{D}, f, \tilde{f} \in D, f \neq \tilde{f}, c(f) = c(\tilde{f}), s(f) < s(\tilde{f})$,
$I = \{s(f) + i | i \in v(f)\} \cap \{s(\tilde{f}) + j | j \in v(\tilde{f})\}$,
$i, j \in I, i > j$:

$$\mathbf{o_{f,i-s(f)}} + \mathbf{o_{\tilde{f},j-s(\tilde{f})}} \leq 1 \quad (16)$$

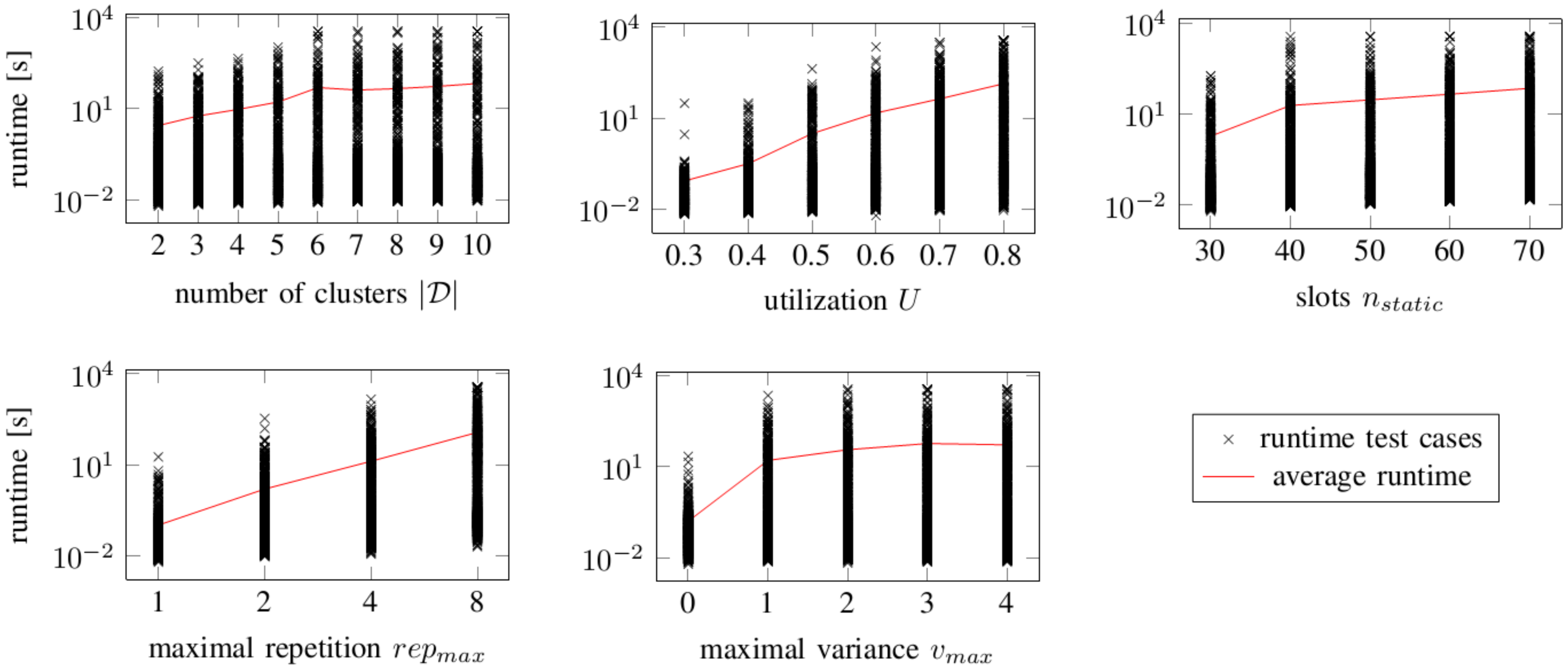
Case Study

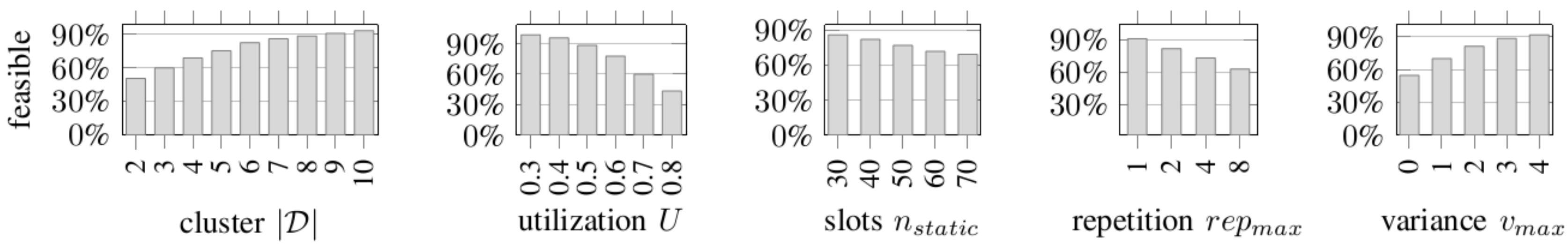


cluster	ECUs	frames	repetition	variance
body	2	9	2,4	3-217
chassis	3	12	2,4	0-85
information	2	11	4,8	9-278
electric	3	10	1,2	0-55
total	**10**	**42**	**1,2,4,8**	**0-278**

- Utilization of FlexRay bus: 29%
- Feasible schedule for **FlexRay 2.1** in 163ms (Xeon 3.2GHz Quadcore,12GB RAM)
  - ILP formulation of 5207 variables and 146926 constraints
- Feasible schedule for **FlexRay 3.0** in 208ms
  - ILP formulation of 14895 variables and 1070730 constraints

Scalability Analysis with 5400 Synthetic Test Cases





Conclusion

- Automotive functions benefit from a synchronous schedule
- Concurrent scheduling of ECUs and buses necessary
- Concurrent scheduling allows reduction of integration efforts
- ILP based scheduling