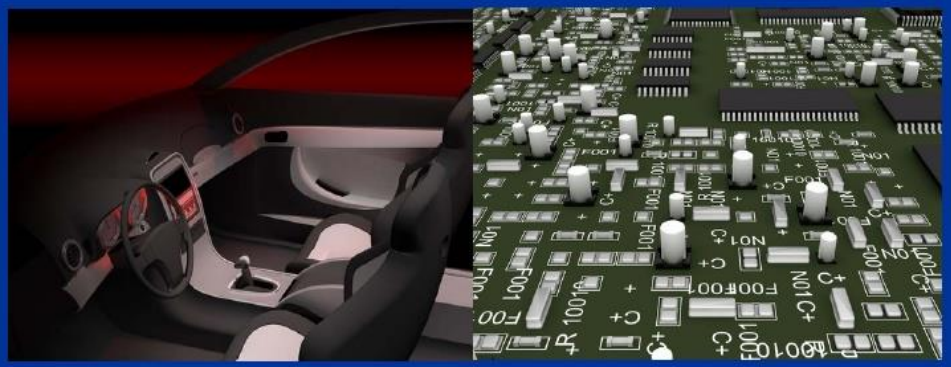
In-Vehicle Networking : a Survey and Look Forward

Nicolas Navet



INRIA

RTaW
RealTime-at-Work

Complexity Mastered

Workshop on Specialized Networks, ETFA09, Palma, Spain - 25/09/2009

Outline

1. Architecture of Automotive Embedded Systems
   - What they look like – example of BMW
   - Constraints in their design – case at Volvo
   - Need for optimizing resource usage (ECU, networks)
2. The Autosar Communication Stack
3. Automotive Networks
   - Time-Triggered versus Event-Triggered
   - Controller Area Network at high loads
   - FlexRay concepts and performances

Architecture of Automotive Electrical and Electronics (E/E) Systems

Electronics is the driving force of innovation

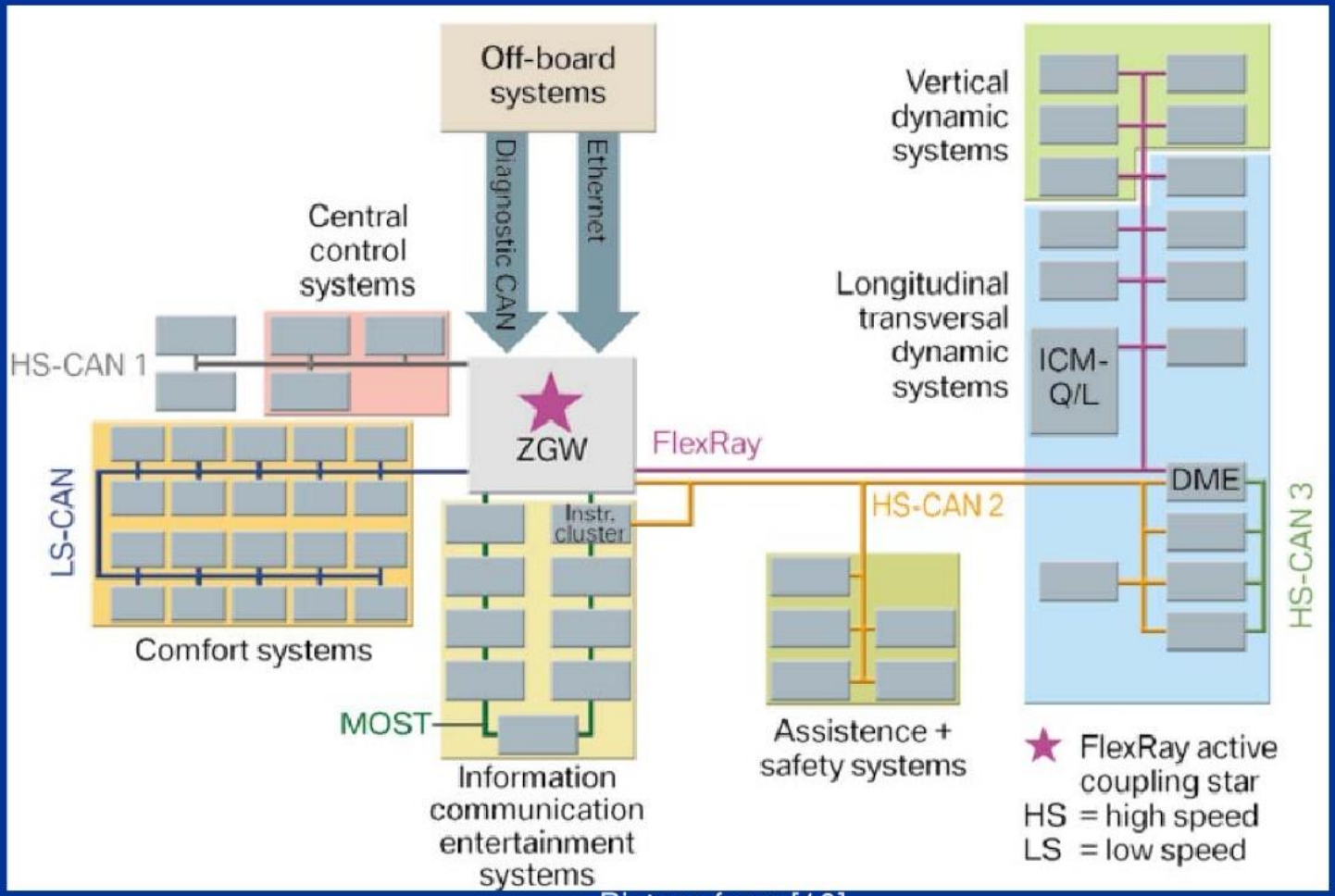
- 90% of new functions use software
- Electronics: 40% of total costs
- Huge complexity: 70 ECUs, 2500 signals, 6 networks, multi-layered run-time environment (AUTOSAR), multi-source software, multi-core CPUs, etc

Strong costs, safety, reliability, time-to-market, reusability, legal constraints !

INRIA

RTaW RealTime-at-Work

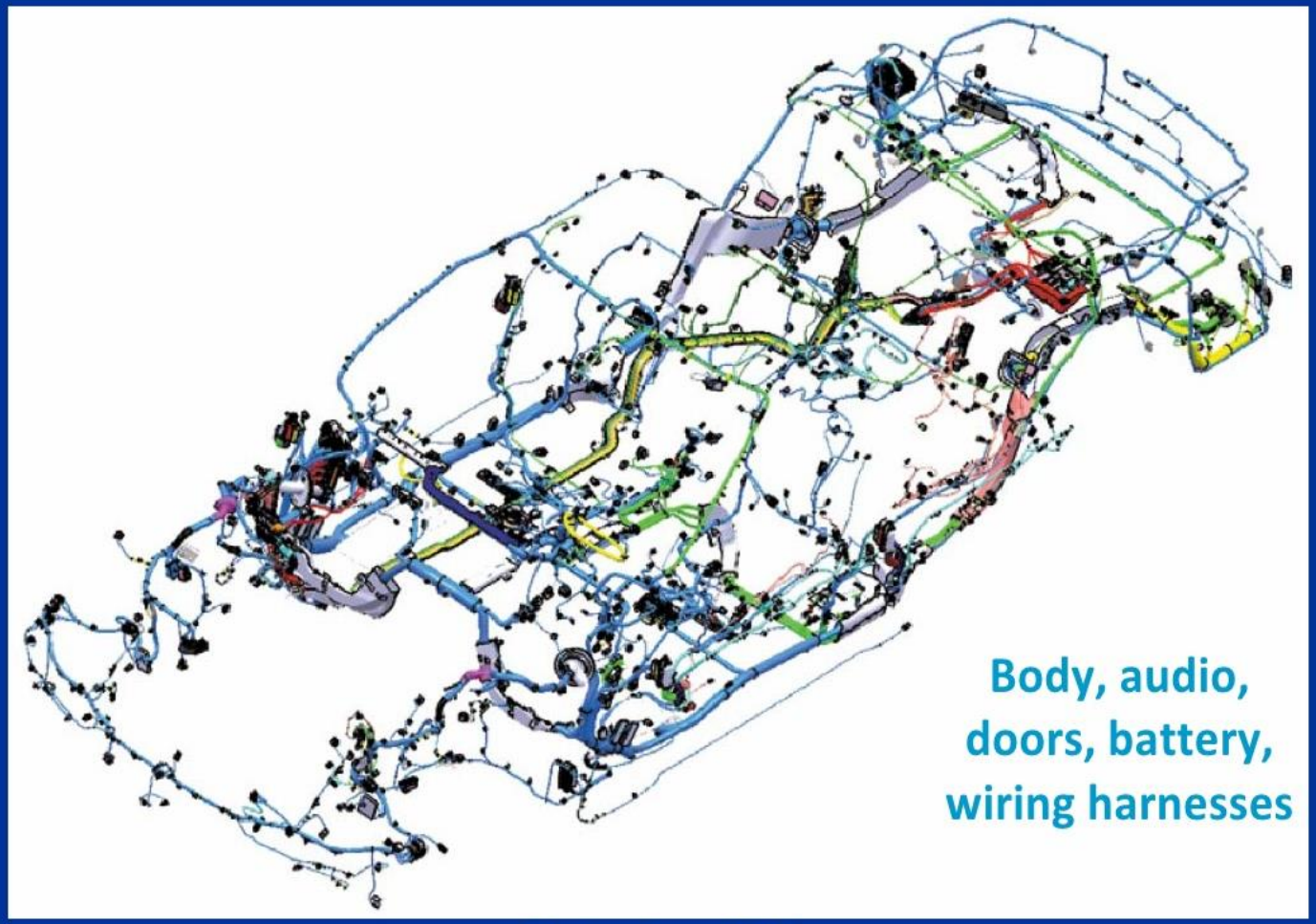
BMW 7 Series networking architecture [10]



Picture from [10]

- ZGW = central gateway
- 3 CAN buses
- 1 FlexRay Bus
- 1 MOST bus
- Several LIN Buses (not shown here)
- Ethernet is used for uploading code/parameters (End of Line)

BMW 7 Series architecture – wiring harness [10]



Picture from [10]

27Millions "variants"

Each wiring harness is tailored to the options

INRIA

RTaW RealTime-at-Work

There are many non-technical issues in the design of E/E architecture

The case at Volvo in [2] :

- Influence of E/E architecture wrt to business value? lacks long term strategy
- Lack of background in E/E at management level often mechanical background
- Lack of clear strategy between in-house and externalized developments
- Technical parameters are regarded as less important than cost for supplier / components selection
- Vehicle Family Management : How to share architecture and sub-systems between several brands/models with different constraints/objectives?
- Sub-optimal solutions for each component / function
- Legal / regulatory constraints

Architectural decisions often:

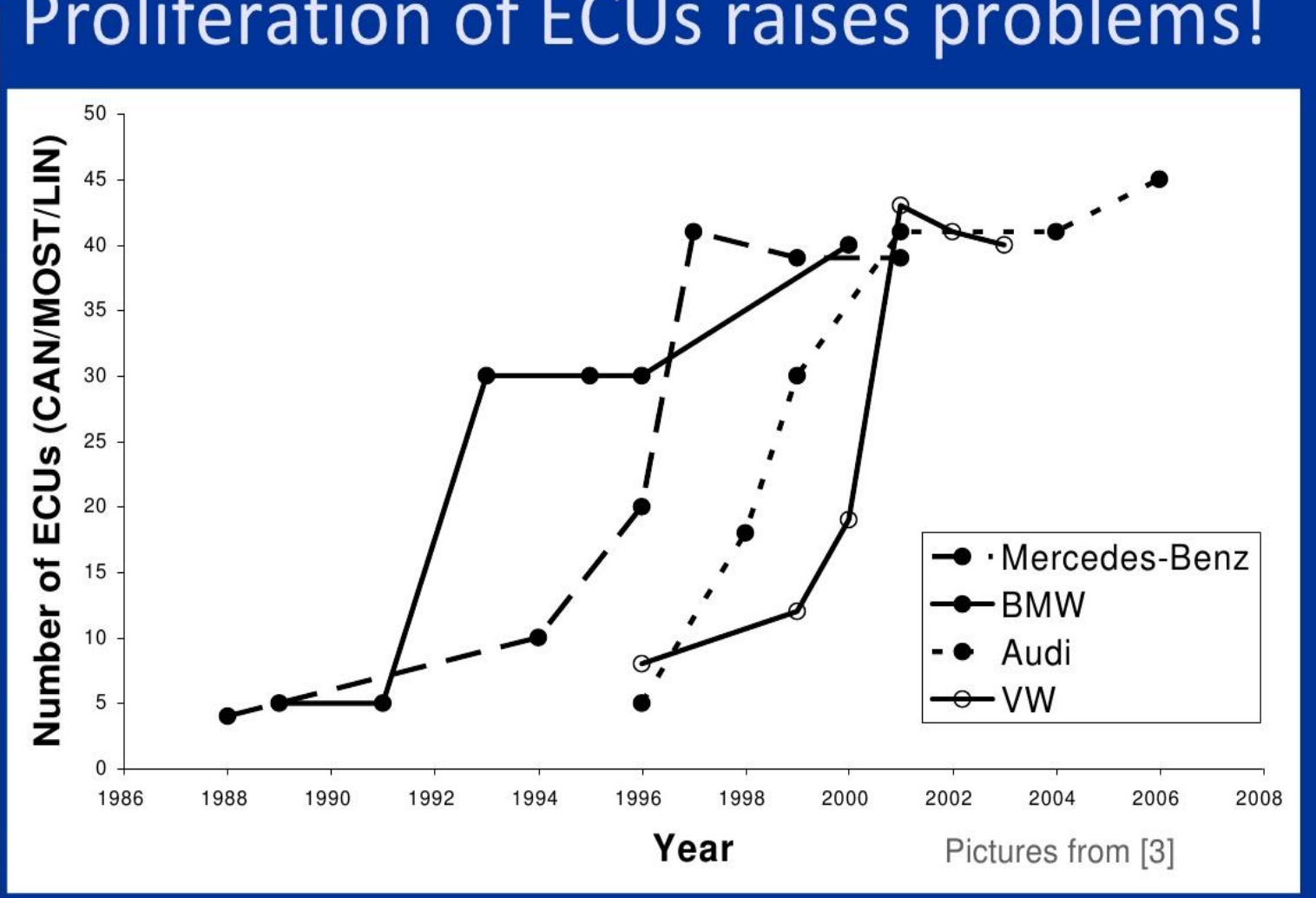
✓ lack well-accepted process

✓ are made on experience / gut feeling (poor tool support)

INRIA

RTaW RealTime-at-Work

Proliferation of ECUs raises problems!



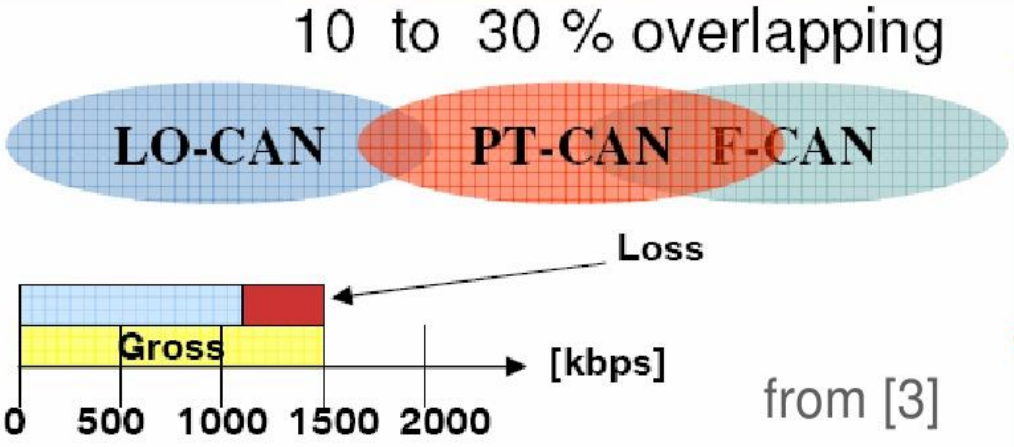
Pictures from [3]

Lexus LS430 has more than 100 ECUs [wardsauto]

Optimizing the use of networks is becoming an industrial requirement too

Good reasons for optimizing :

- Complexity of the architectures (protocols, wiring, ECUs, gateways, etc)
- Hardware cost, weight, room, fuel consumption, etc
- Need for incremental design
- Industrial risk and time to master new technologies (e.g. FlexRay)
- Performances (sometimes):
  - a 60% loaded CAN network may be more efficient that two 30% networks interconnected by a gateway
  - Some signals must be transmitted on several networks

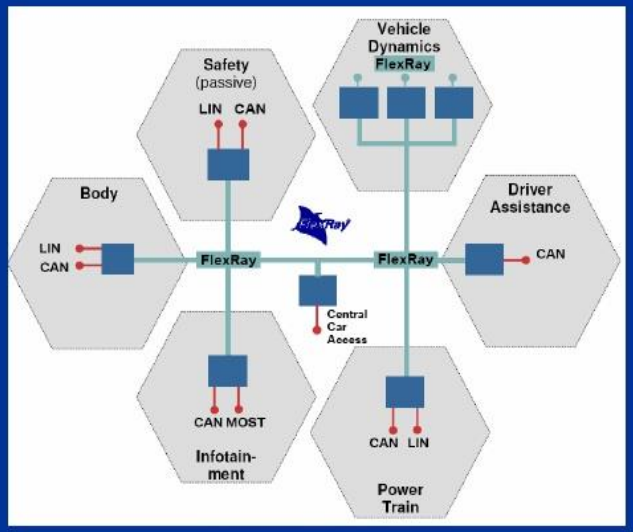


from [3]

Likely upcoming architectures

Fewer ECUs but more powerful

- Multi-core µ-controller
- Multi-source software
- Autosar OS strong protection mechanisms
- Virtualization ?
- ISO2626-2 dependability standard



Picture from [8]

FlexRay as backbone at BWM in a few years [8]

Backbone:

- High-speed CAN : 500Kbit/s
- FlexRay : 10 Mbit/s
- Ethernet ?

INRIA

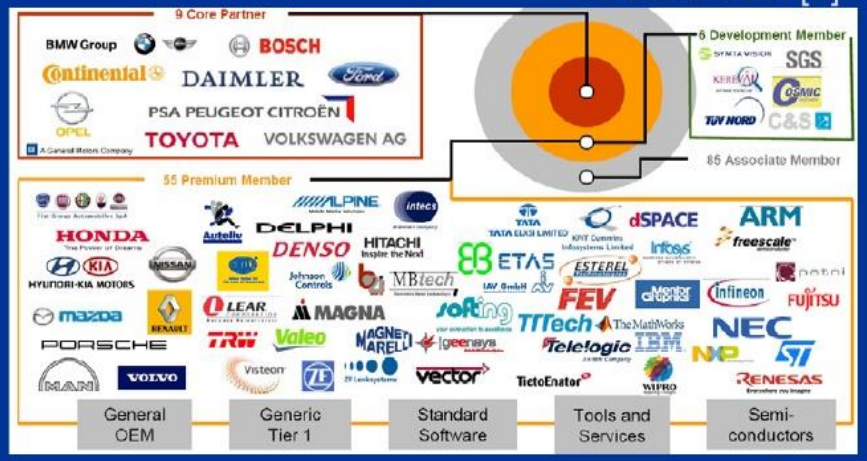
RTaW RealTime-at-Work

AUTOSAR Communication Stack

AUTOSAR at a glance - Automotive Open System Architecture

- Industry initiative that is becoming a de-facto standard
- Standardize: architecture (basic software modules inc. communication), methodology and exchange format, application interfaces
- "Cooperate on standards, compete on implementation"

Picture from [5]



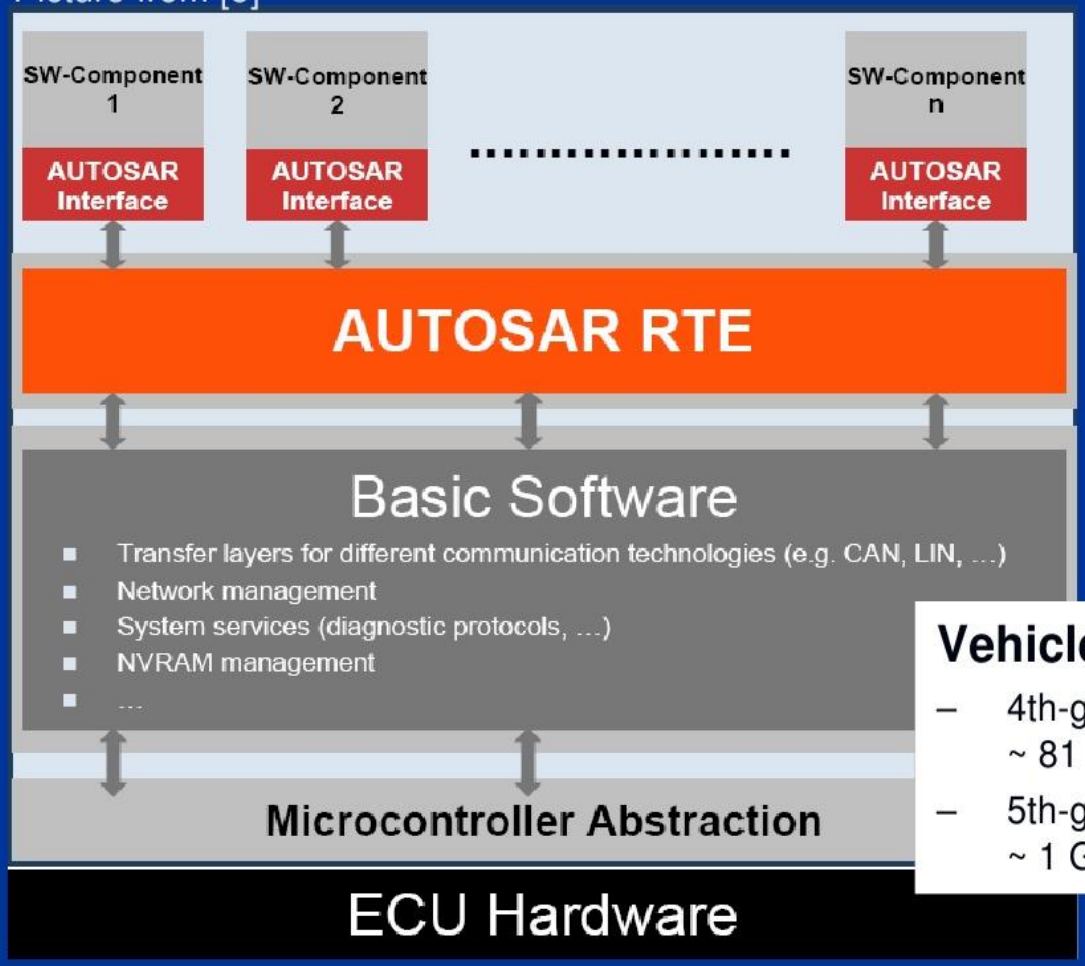
Benefits

- cost savings for legacy features
- quality through reuse and market competition
- focus on real innovation versus basic enablers
- ability to re-allocate a function
- helps to master complexity

Caveat: great complexity and still evolving specifications

AUTOSAR layered architecture: the global picture

Picture from [5]



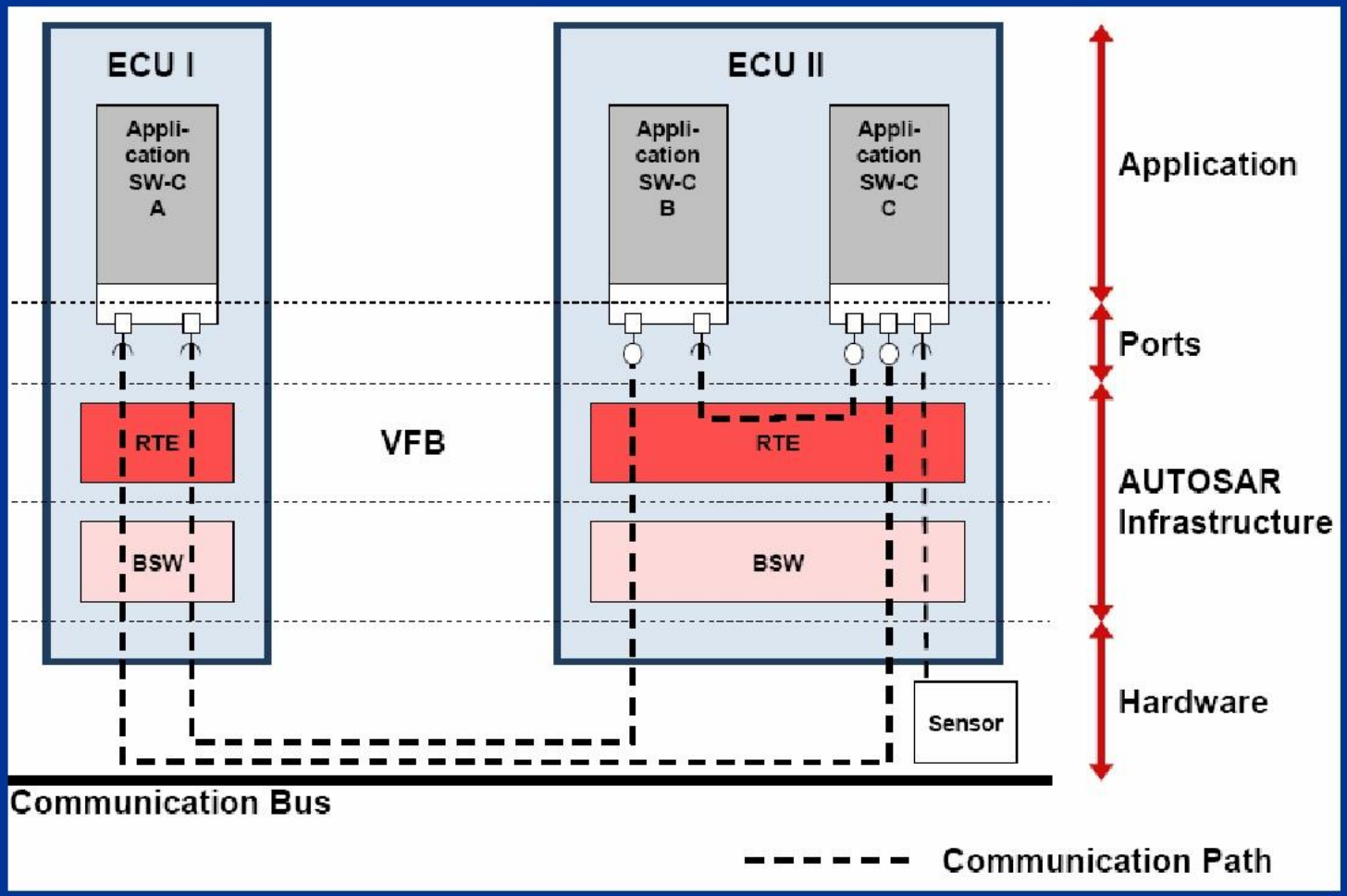
Supported networks are:

- CAN : Controller Area Network
- LIN : Local Interconnect Network
- MOST : Media Oriented Systems Transport
- Ethernet in the upcoming release for diag./upload

Vehicle Flashing Times [8]:

- 4th-generation BMW 7 series via CAN: ~ 81 MB in 10 h
- 5th-generation BMW 7 series via Ethernet: ~ 1 GB in 20 min

Intra- and inter-ECU Communication

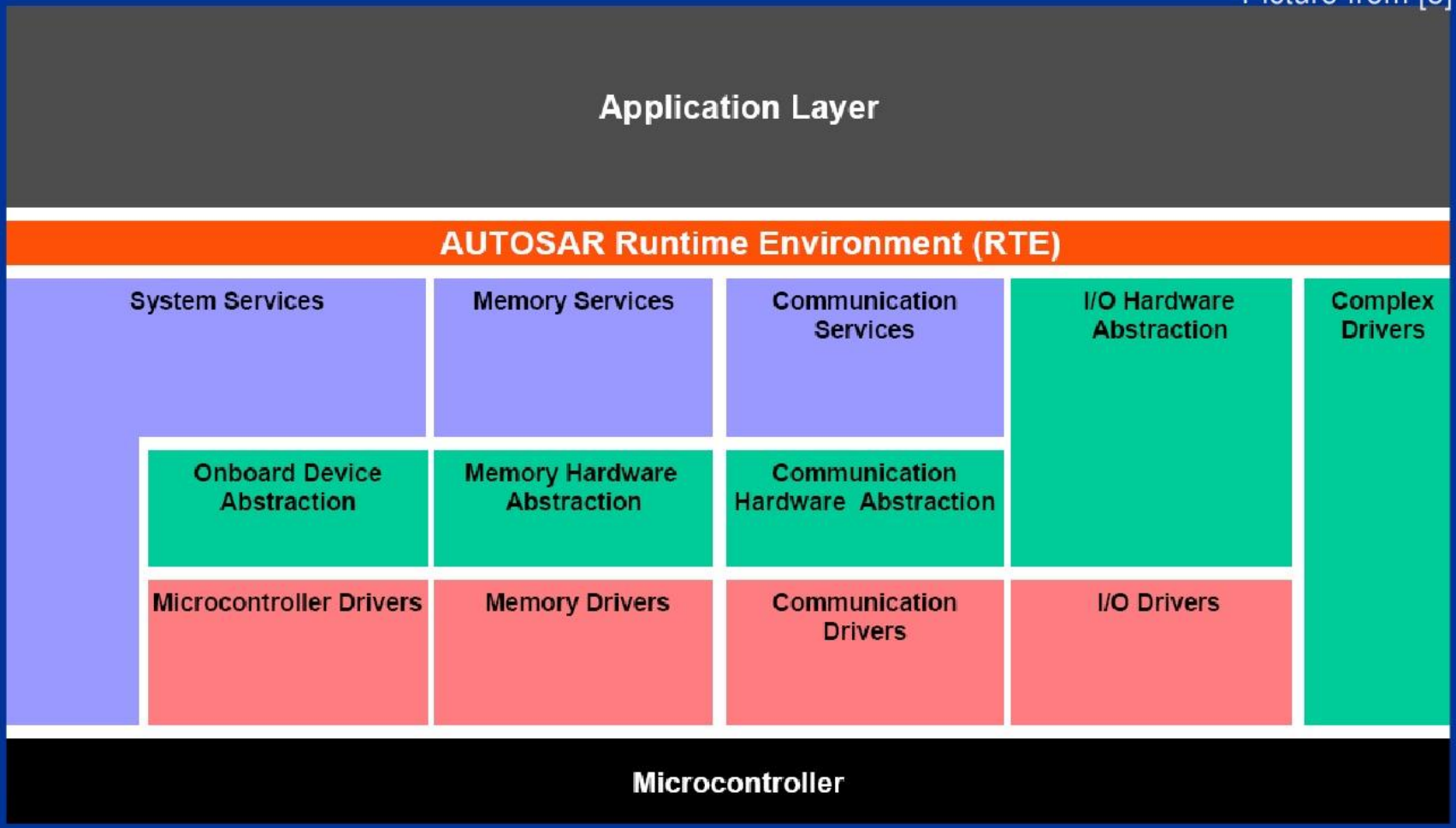


MW hides the distribution and the characteristics of the HW platform

Compliance: SW-C must only call entry points in the RTE

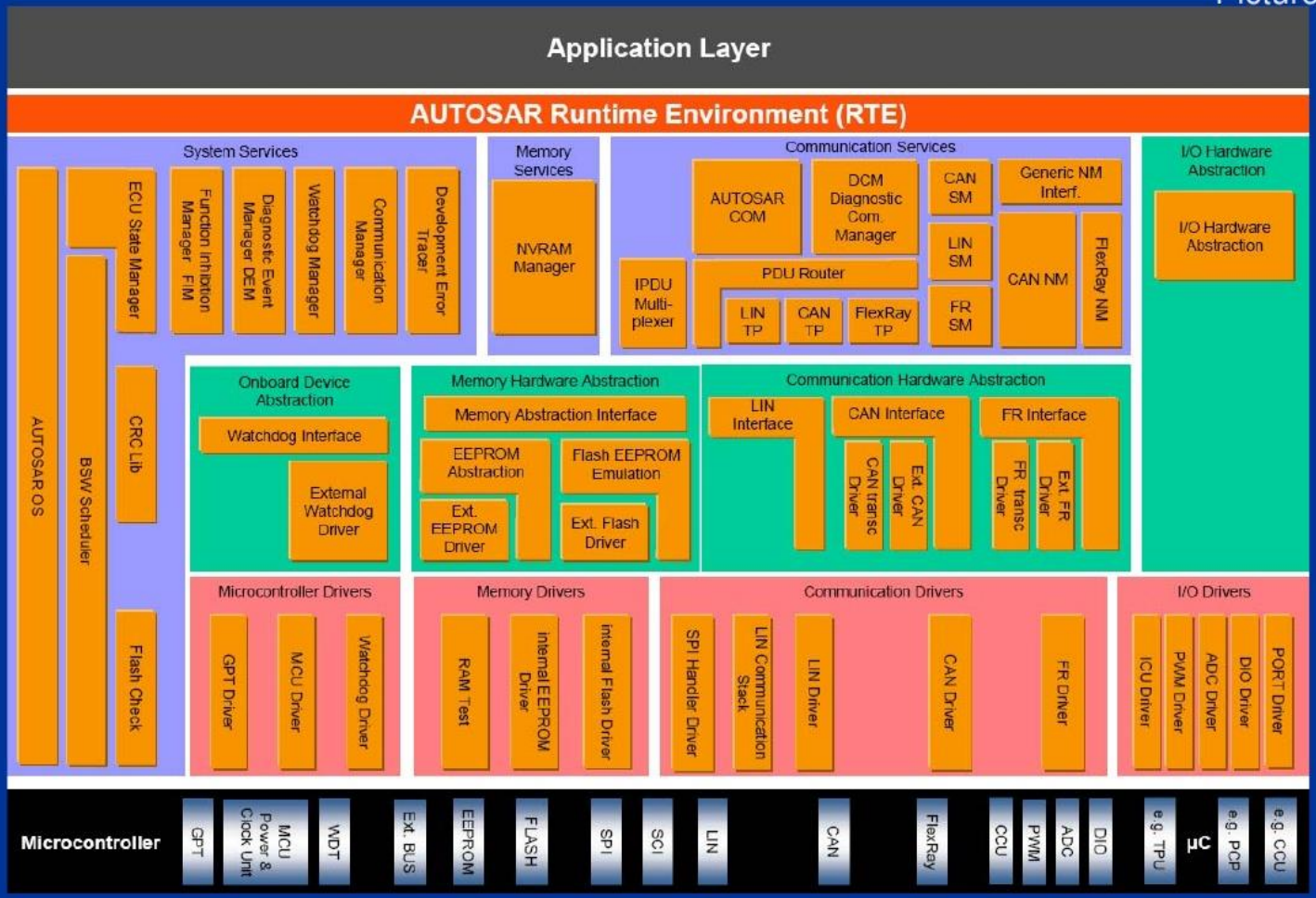
AUTOSAR layered architecture: some more details

Picture from [5]



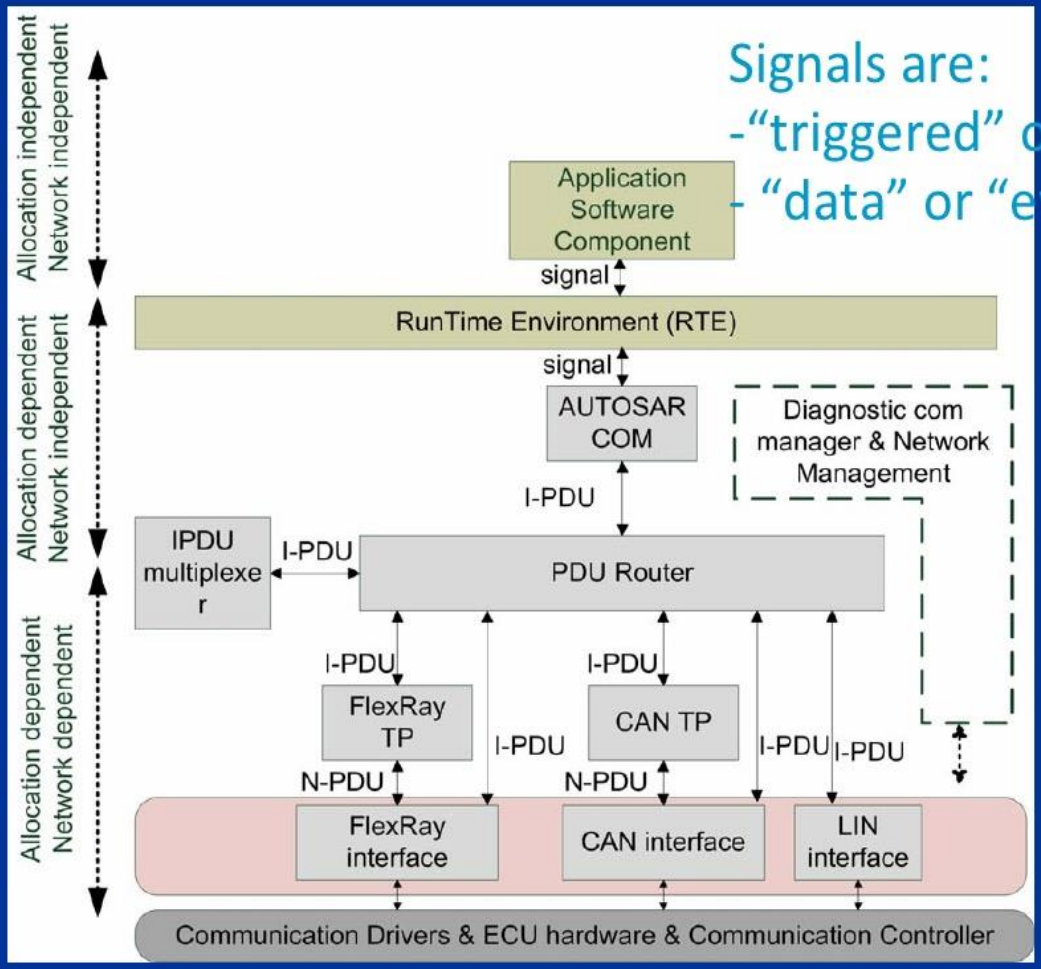
There are some 50 standardized basic software components (BSW) …

Picture from [5]

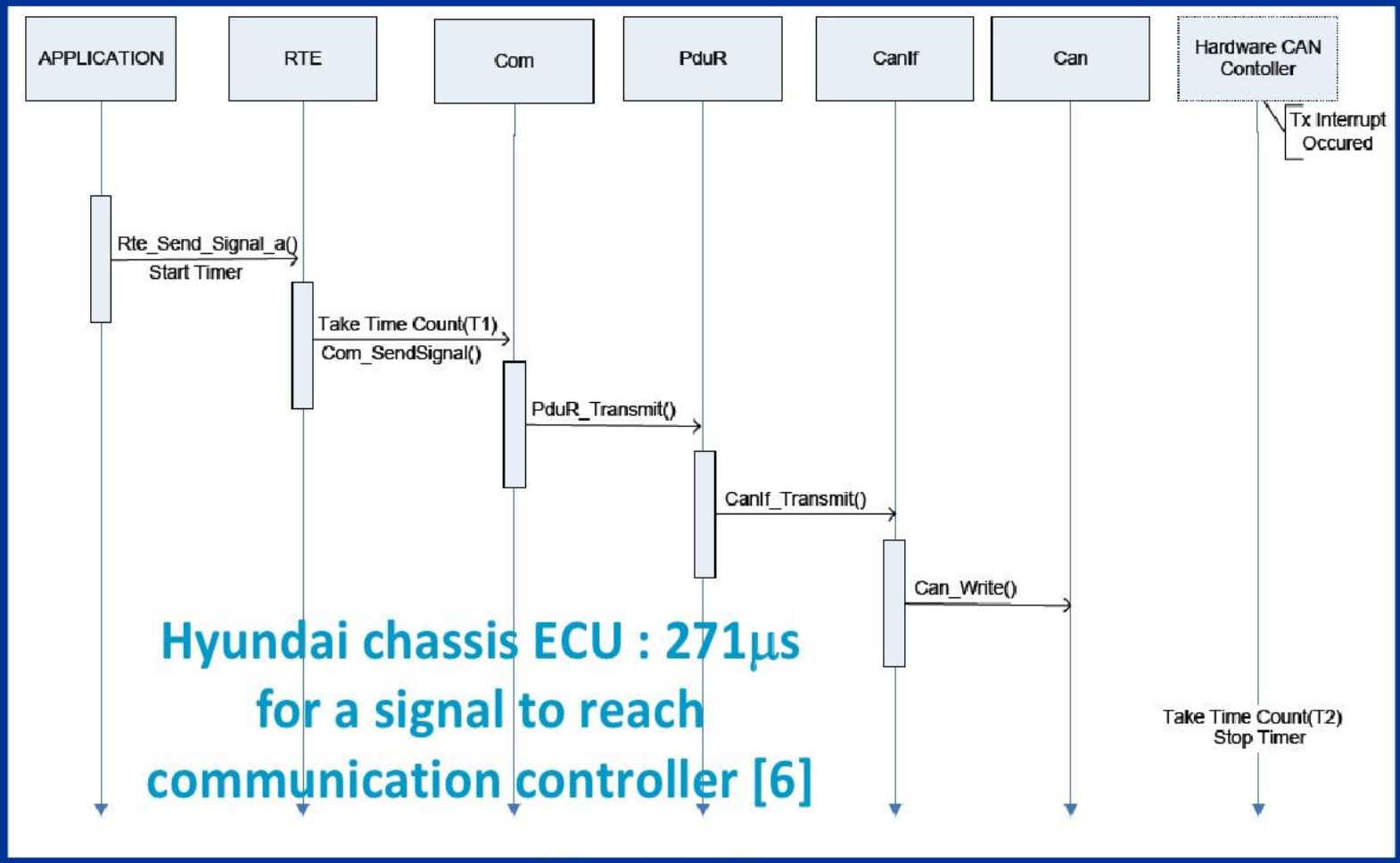


Zoom on the communication services

"Explicit" call to communication services or MW initiative: "implicit" mode

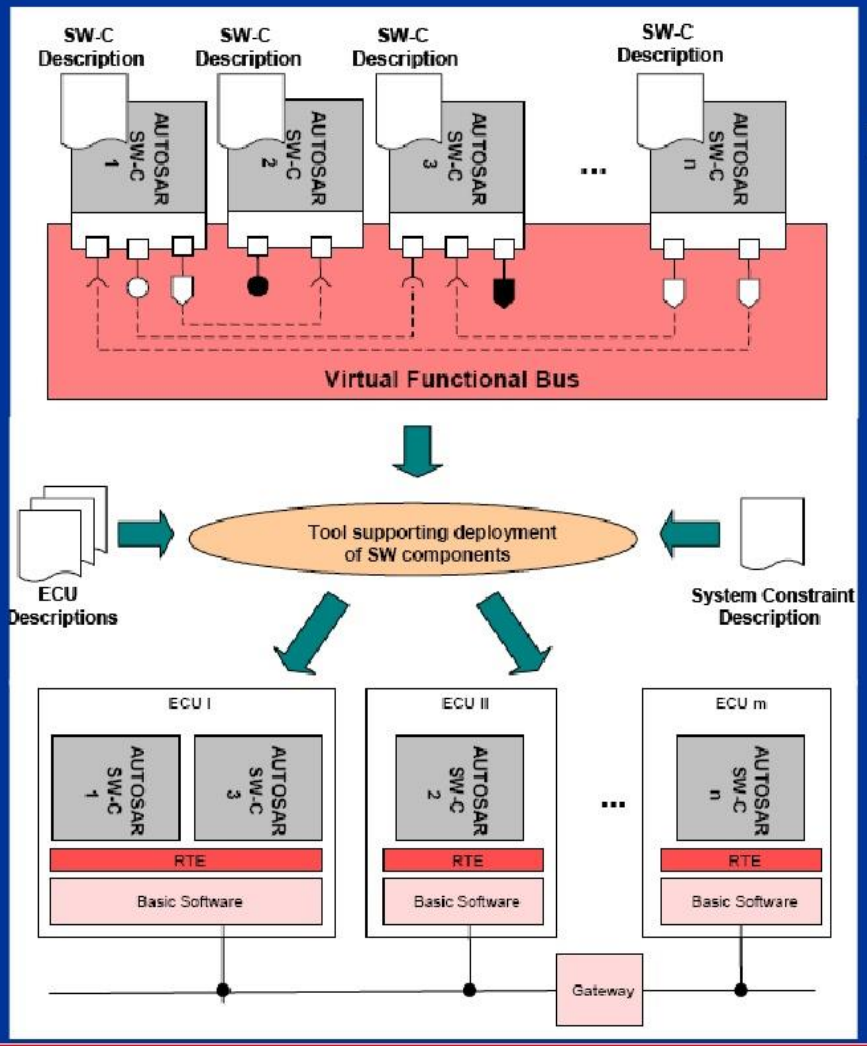


Sending a signal through the CAN communication stack [6]



Picture from [6]

Generation of the "operational" architecture



Picture from [5]

Automotive networks

Event-Triggered vs Time-Triggered Communication

Event-triggered communication

- Transmission on occurrence of events
- Collision resolution on the bus is needed
- Bandwidth efficient but performance degradation at high loads
- Incremental design and latencies computation non-obvious

Ex: CAN

Time-triggered communication

- frames are transmitted at pre-determined points in time
- Synchronization is needed
- Bandwidth not optimized but …
- Timing constraints are easy to check
- Missing messages are detected asap

Ex: static segment of FlexRay

INRIA

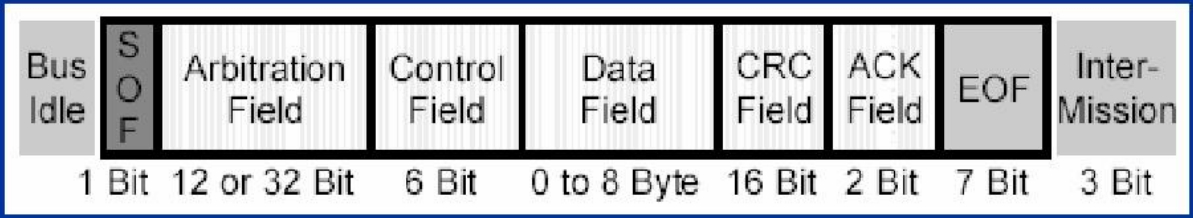
RTaW RealTime-at-Work

In practice “best of both world” approaches are needed and used

1. Offsets on CAN : impose some fixed de-synchronization between streams of messages on an ECU ⇨ less collision, better performances
2. FlexRay dynamic segment : reduce waste of bandwidth and increase flexibility
3. Upcoming FlexRay V3.0 : more flexibility with slot multiplexing also in the static segment

Controller Area Network: a Recap

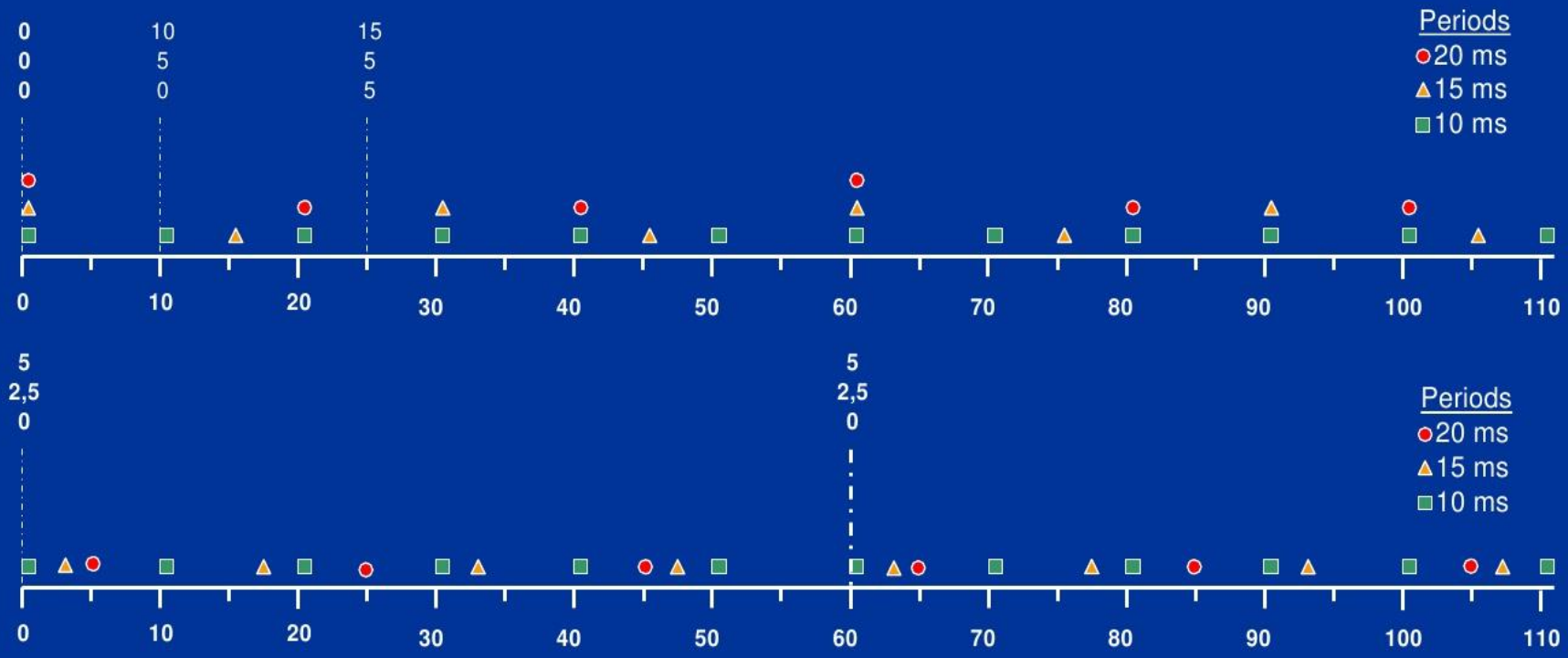
- Priority bus with non-destructive collision resolution
- Id of the frame is the priority
- At most 8 data bytes per frame



- Data rate up to 1Mbit/s (500kbit/s in practice)
- Normalized by ISO in 1994 – defacto standard in vehicles - more than 2 billions controllers produced

Scheduling CAN frames with offsets ?!

Principle: desynchronize transmissions to avoid load peaks

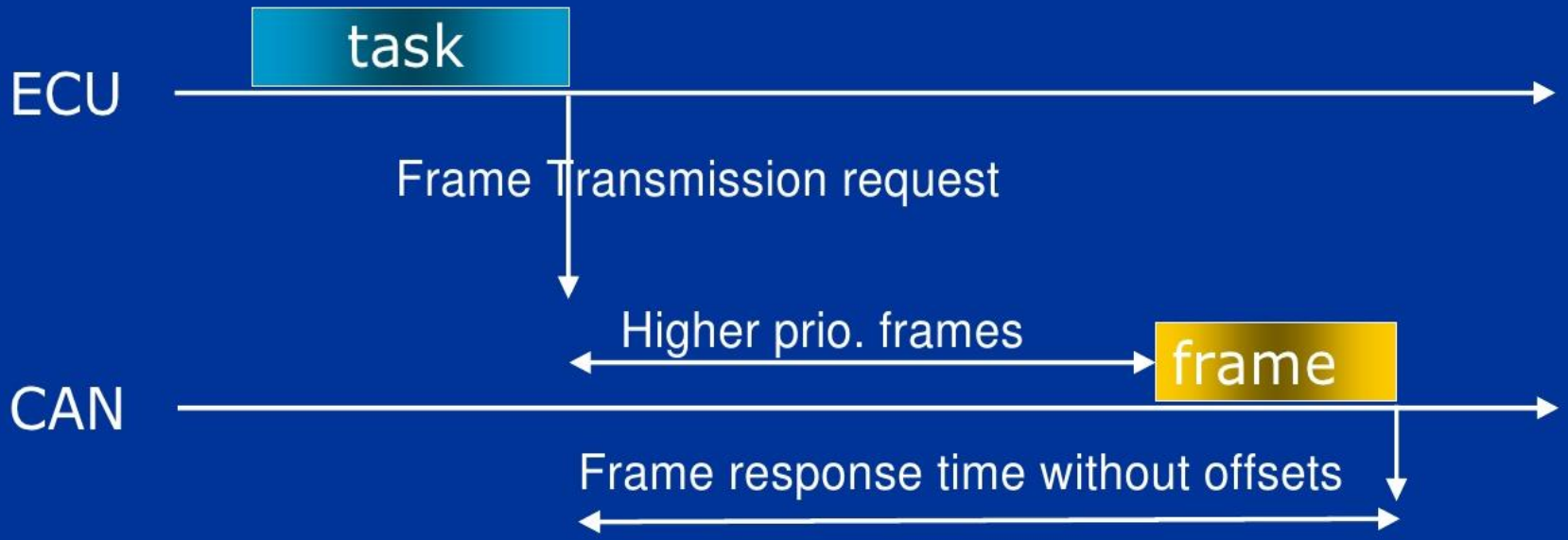


Algorithms to decide offsets are based on arithmetical properties of the periods and size of the frame [1]

INRIA

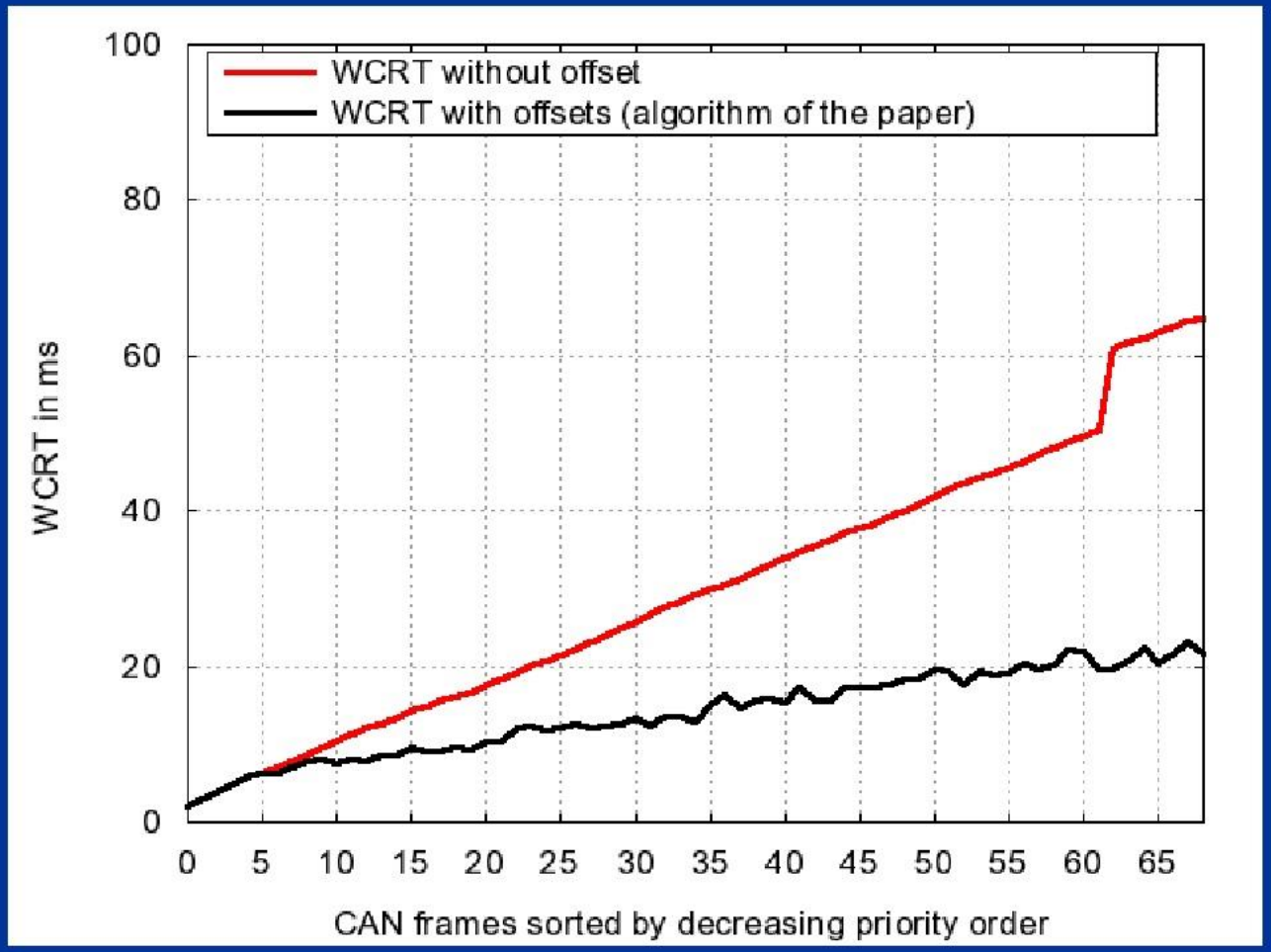
RTaW RealTime-at-Work

But task scheduling has to be adapted otherwise data freshness is not much improved …



Tasks and messages scheduling should be designed jointly…

Offsets Algorithm applied on a typical body network



65 ms

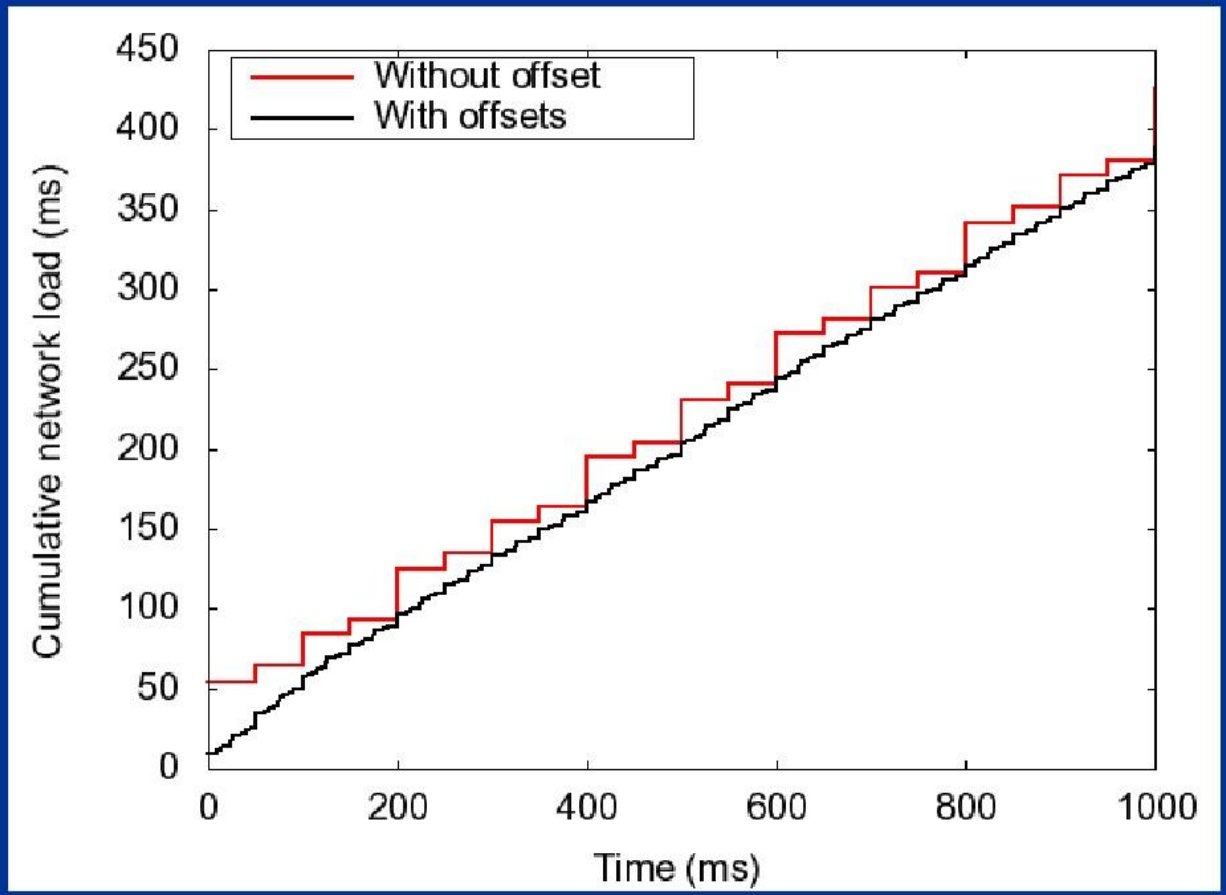
21 ms

INRIA

RTaW RealTime-at-Work

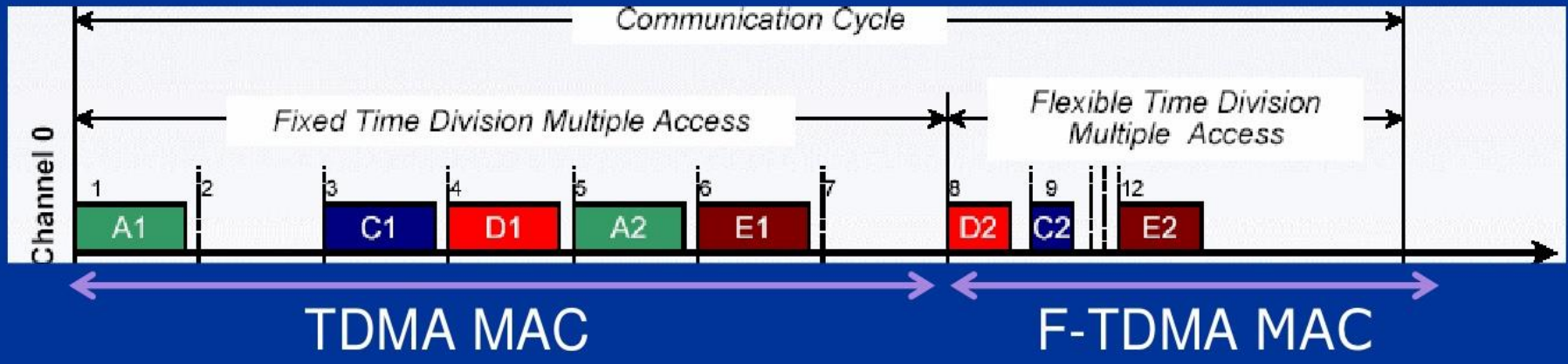
Efficiency of offsets some insight

Work = time to transmit the CAN frames sent by the stations



- Almost a straight line, suggests that the algorithm is near-optimal

FlexRay protocol basics



- Typically ST segment: 3 ms and DYN: 2ms
- Frames: up to 254 bytes, size is fixed in the static segment (BMW:16bytes)
- Data rate: between 500kbit/s and 10Mbit/s
- 64 ≠ communication schedules max. (but a slot always belongs to the same station)

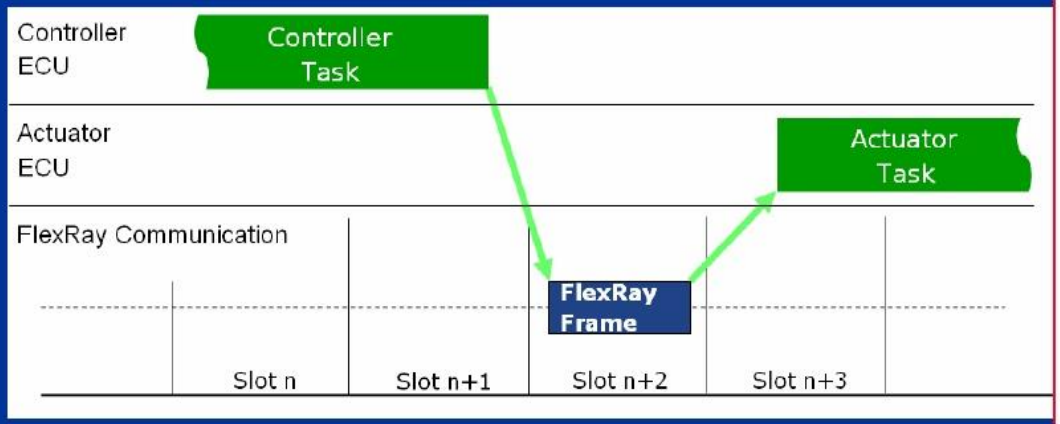
FlexRay bus design and configuration

Requirements on FlexRay

- Performance requirements: response times, jitters,
- Incrementality requirements: additional functions or ECUs
- Dependability requirements: fail-silence, babbling idiot, …
- Platform requirements: platform wide frames (e.g., NM), carry-over of ECUs, etc

Complex Problem

- Mixed of TT and ET scheduling
- Tightly linked with task scheduling
- Large number of parameters (>70)
- AUTOSAR constraints (OS, COM, etc)
- …



Crucial question : applicative software synchronous or not wrt FlexRay ?

– all applicative modules are synchronized with FlexRay global time ?

– all applicative modules are running asynchronously ?

– combination of synchronized and asynchronous modules (likely) ?

✓ Optimal solutions probably out of reach but there are good heuristics, e.g. [11]

FlexRay VS (multi-)CAN [11]

Useful load (signals)	FlexRay 2.5Mbit/s		FlexRay 10Mbit/s		1x CAN 500Kbit/s	
Load 1x ($\approx$ 60kbit/s)		free slots		free slots	network load	31%
	ST	23	ST	100	R without offsets	15.3
	DYN	9	DYN	43	R with offsets	7.8
Load 2x ($\approx$ 120kbit/s)		free slots		free slots	network load	57%
	ST	21	ST	98	R without offsets	49.6
	DYN	9	DYN	43	R with offsets	14.9
Load 3x ($\approx$ 180kbit/s)		free slots		free slots	network load	85%
	ST	19	ST	96	R without offsets	148.5
	DYN	7	DYN	41	R with offsets	79.7
Load 4x ($\approx$ 240kbit/s)		free slots		free slots	non-schedulable 2x CAN 500 OK	
	ST	19	ST	96		
	DYN	7	DYN	40		
Load 5x ($\approx$ 300kbit/s)		free slots		free slots	non-schedulable 2x CAN 500 depending on the overlap	
	ST	15	ST	92		
	DYN	6	DYN	40		
Load 10x ($\approx$ 600kbit/s)		free slots		free slots	non-schedulable with two CAN buses	
	ST	**3**	ST	**84**		
	DYN	**0**	DYN	**36**		

In our experiments, between 2 and 2.5 MBit/s of data can be transmitted on FlexRay 10Mbit/s

INRIA

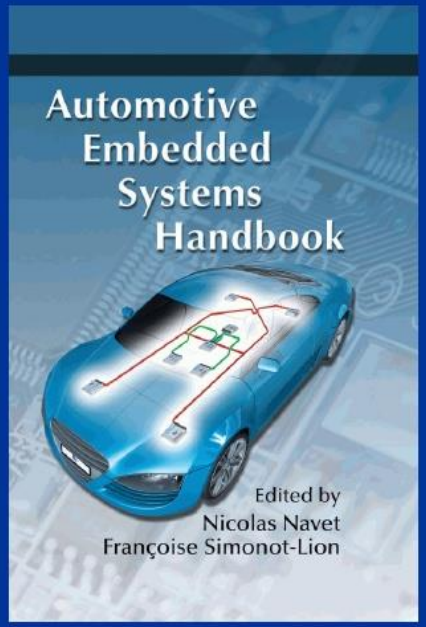
RTaW RealTime-at-Work

Conclusion

- Automotive MAC protocols are well mastered technologies that respond to the current needs
- Com. systems architectures will change
- AUTOSAR will probably require one or two car generations to replace all what exists
- Dependability will create new needs:
  - Increasing safety-related functions (X-by-Wire)
  - Certification in the context of ISO26262

References

References



Automotive Embedded Systems - General

[1] N. Navet, F. Simonot-Lion, editors, The Automotive Embedded Systems Handbook, Industrial Information Technology series, CRC Press / Taylor and Francis, ISBN 978-0849380266, December 2008.

[2] P. Wallin, Axelsson, A Case Study of Issues Related to Automotive E/E System Architecture Development, IEEE International Conference and Workshop on the Engineering of Computer Based Systems, 2008.

[3] T. Nolte, Hierarchical Scheduling of Complex Embedded Real-Time Systems, Summer School on Real-Time Systems (ETR'09), Paris, 2009.

AUTOSAR

[4] AUTOSAR layered software architecture, part of release 3.1, V2.2.2.

[5] AUTOSAR – an open standardized software architecture for the automotive industry, Simon Fürst, 1st Autosar Open Conference, 2008.

[6] Performance of AUTOSAR Basic Software modules in a chassis ECU, HYUNDAI MOTOR Company HYUNDAI & KPIT Cummins, 1st AUTOSAR Open Conference, 2008.

[7] J. Buczkowski, Keynote address to the AUTOSAR conference, Ford, 1st AUTOSAR Open Conference, 2008.

[8] T. Thomsen, G. Drenkhan, Ethernet for AUTOSAR, EB Automotive Gmbh, 2008.

FlexRay

[9] A. Schedl, "Goals and Architecture of FlexRay at BMW", slides presented at the Vector FlexRay Symposium, March 2007.

[10] H. Kellerman, G. Nemeth, J. Kostelezky, K. Barbehön, F. El-Dwaik, L. Hochmuth, "BMW 7 Series architecture", ATZextra, November 2008.

[11] M. Grenier, L. Havet, N. Navet, "Configuring the communication on FlexRay: the case of the static segment", Proceedings of ERTS'2008.

INRIA

RTaW
RealTime-at-Work

Questions / feedback ?

Please get in touch at:
nicolas.navet@realtimeatwork.com

http://www.realtimeatwork.com