Next Gen Automotive Ethernet Functions and the Implementation in an Ethernet MAC

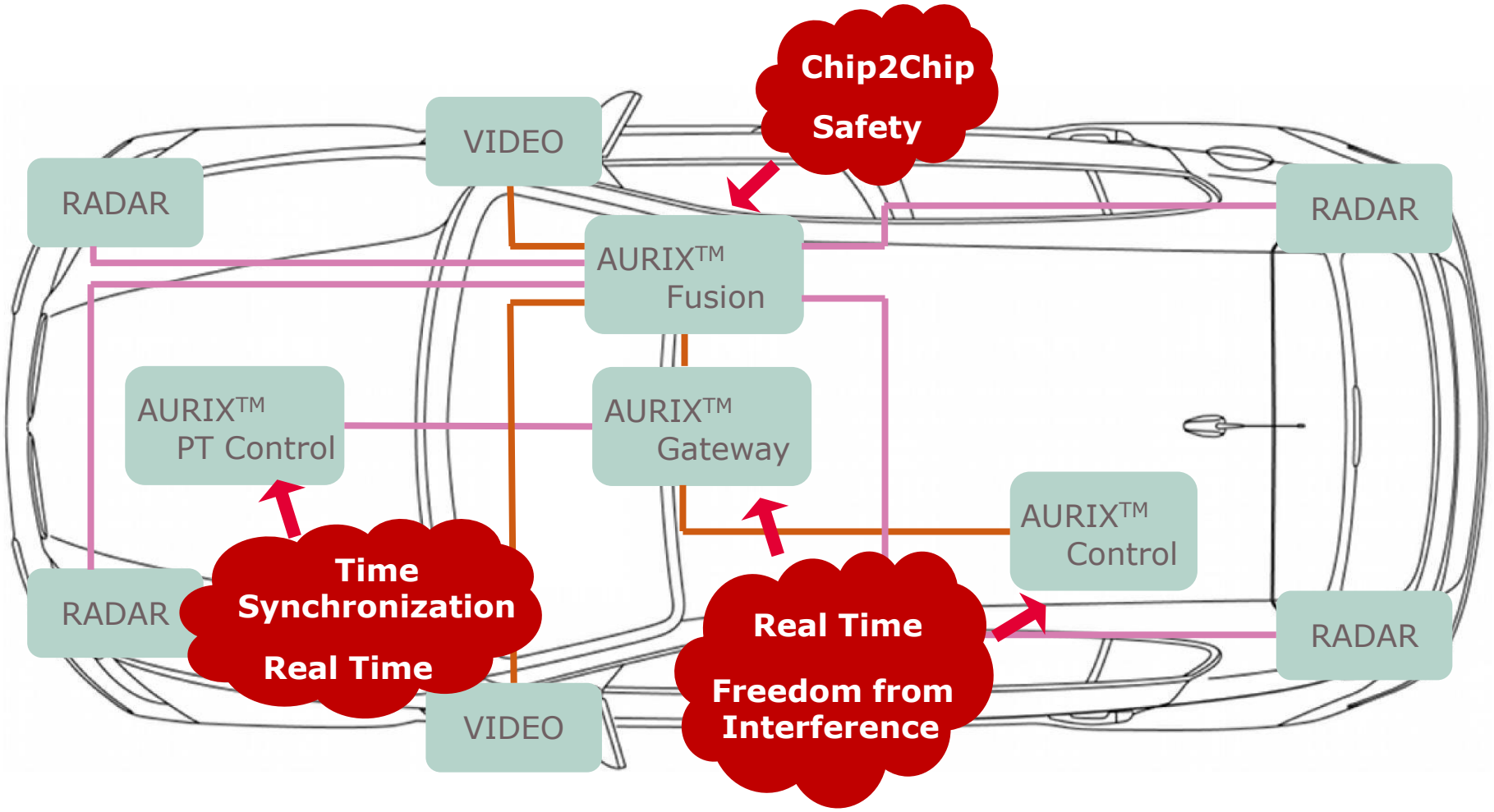
IEEE STANDARDS ASSOCIATION IEEE

2017 IEEE-SA Ethernet & IP @ Automotive Technology Day

Harald Zweck, Infineon Technologies

Infineon

Ethernet In-Vehicle Network Example



1Gbit

100Mbit

Note: The picture shows an example for a part of an In-Vehicle Network, and does not represent a comprehensive architecture.

Vehicle Development Cycle

Application Use Case

Solution Concept

Scalable, flexible and customizable in-vehicle network

Product Design

Product Verification

OEM

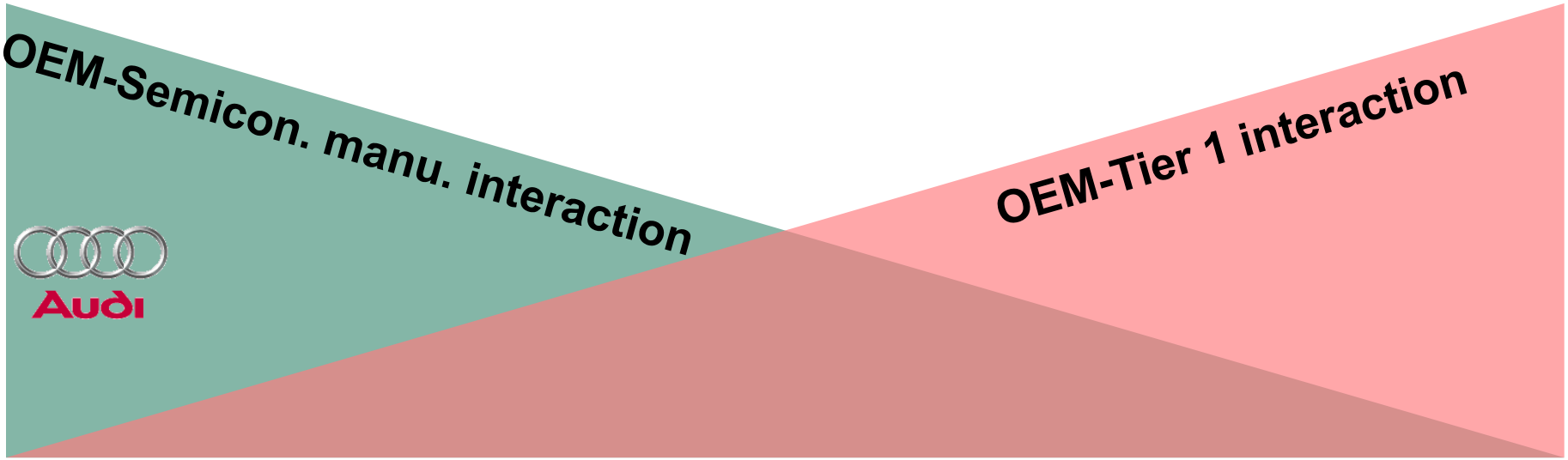
Pictures by courtesy of AUDI

Vehicle Development Cycle

X years to SOP

SOP

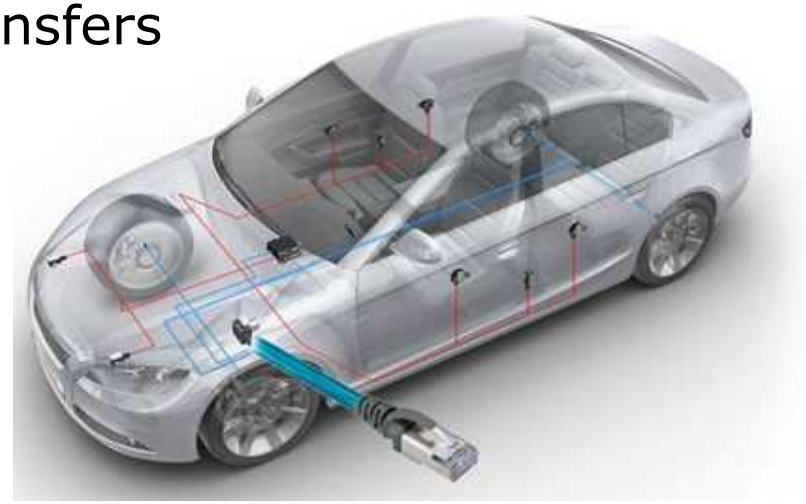
Concept Phase			Series Development		
Initial Discussion	Concept Discussion	Preparation & Planning	Coordination	Verification & Confirmation	Final Preparation



Picture by courtesy of AUDI

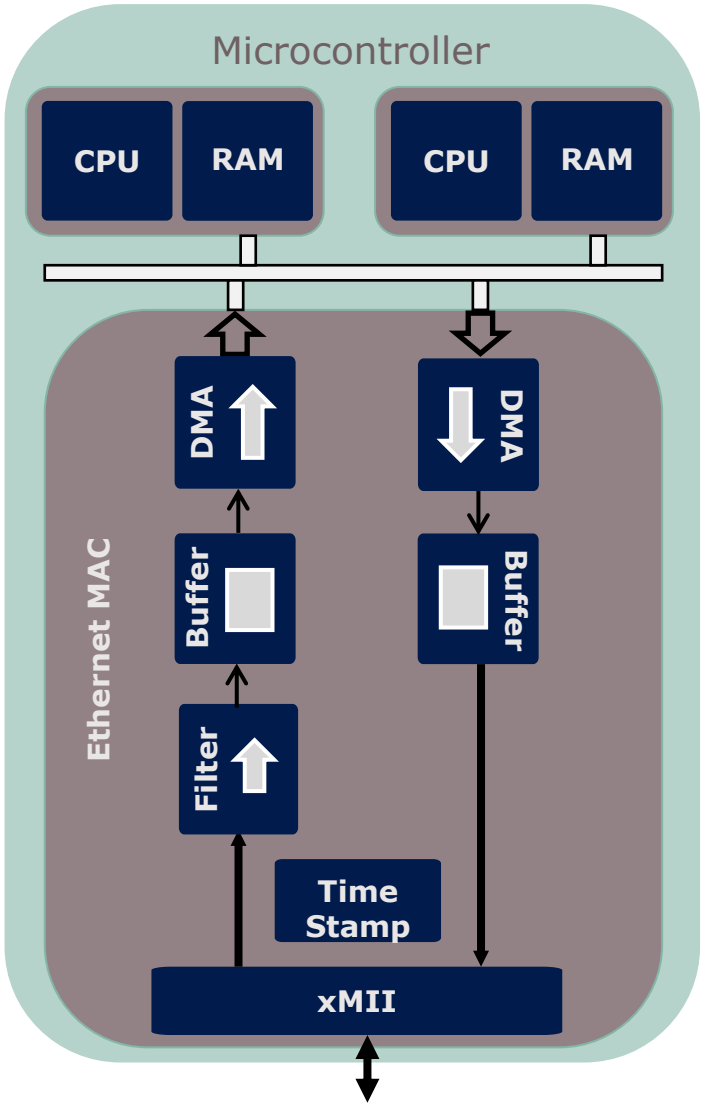
First Generation In-Vehicle Ethernet

- System requirement
  Simple Data Transfers
- Example Ethernet for diagnosis
  - Vehicle is in the repair shop
  - Vehicle does not move
  - Connection is based on standard Ethernet technology
  - No real time requirements for transfers



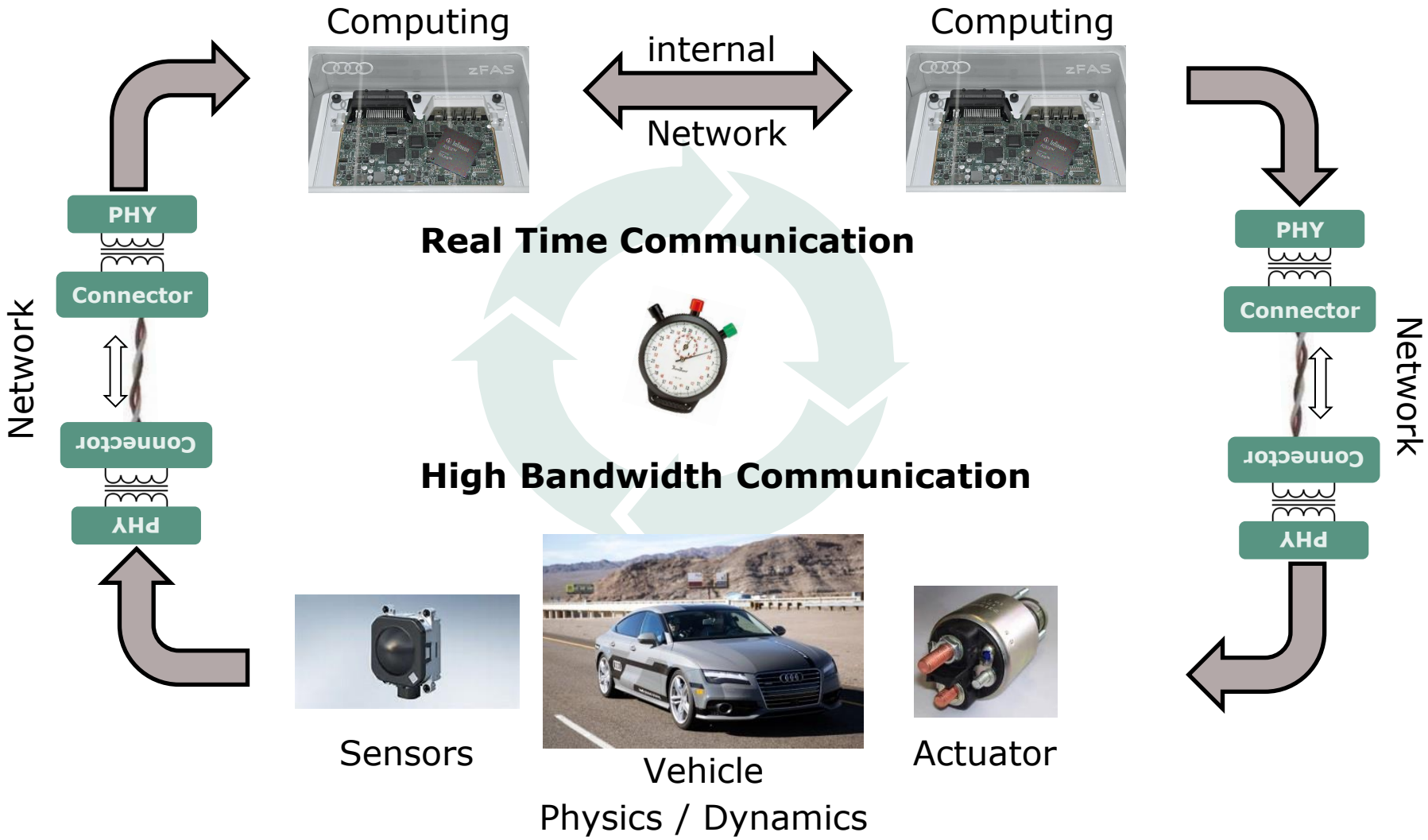
Use Case Diagnostic

- Implementation
  Buffered, DMA driven data transfer
  - Using IEEE 802.3
  - 802.1AS conforming Time Stamps
  - MAC address filter



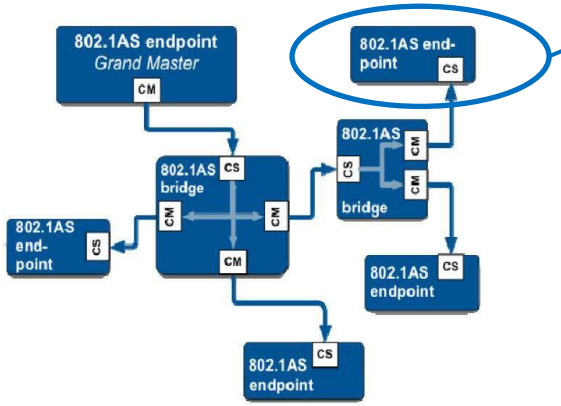
Second Generation In-Vehicle Ethernet

Vehicle Control Loop

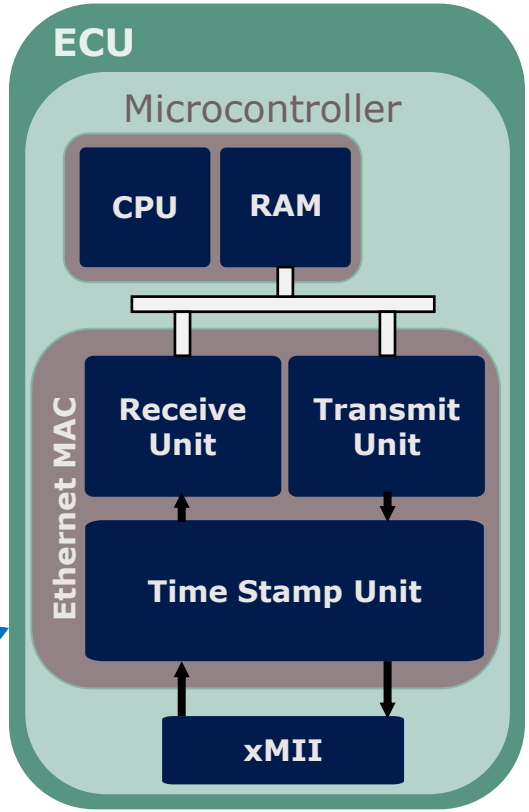


Use Case Time Synchronization

- System requirement:
  Synchronized time base for the IVN*
  - Time stamping compatible with IEEE 802.1AS
  - Feature is implemented in HW
  - Supports the implementation of a common time base for the IVN*
  - Required e.g. for AVB protocols

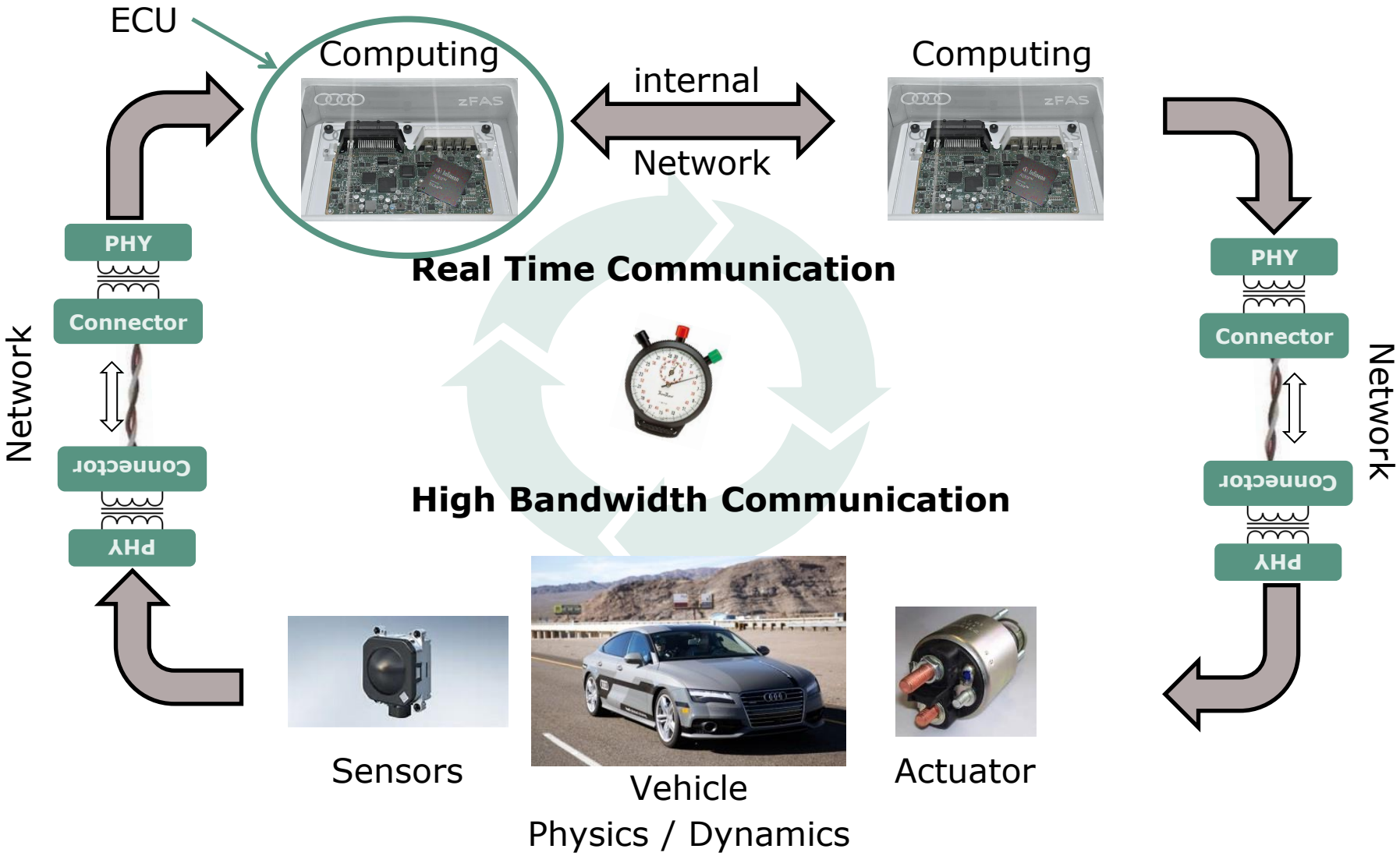


Picture by courtesy of IEEE 802.1



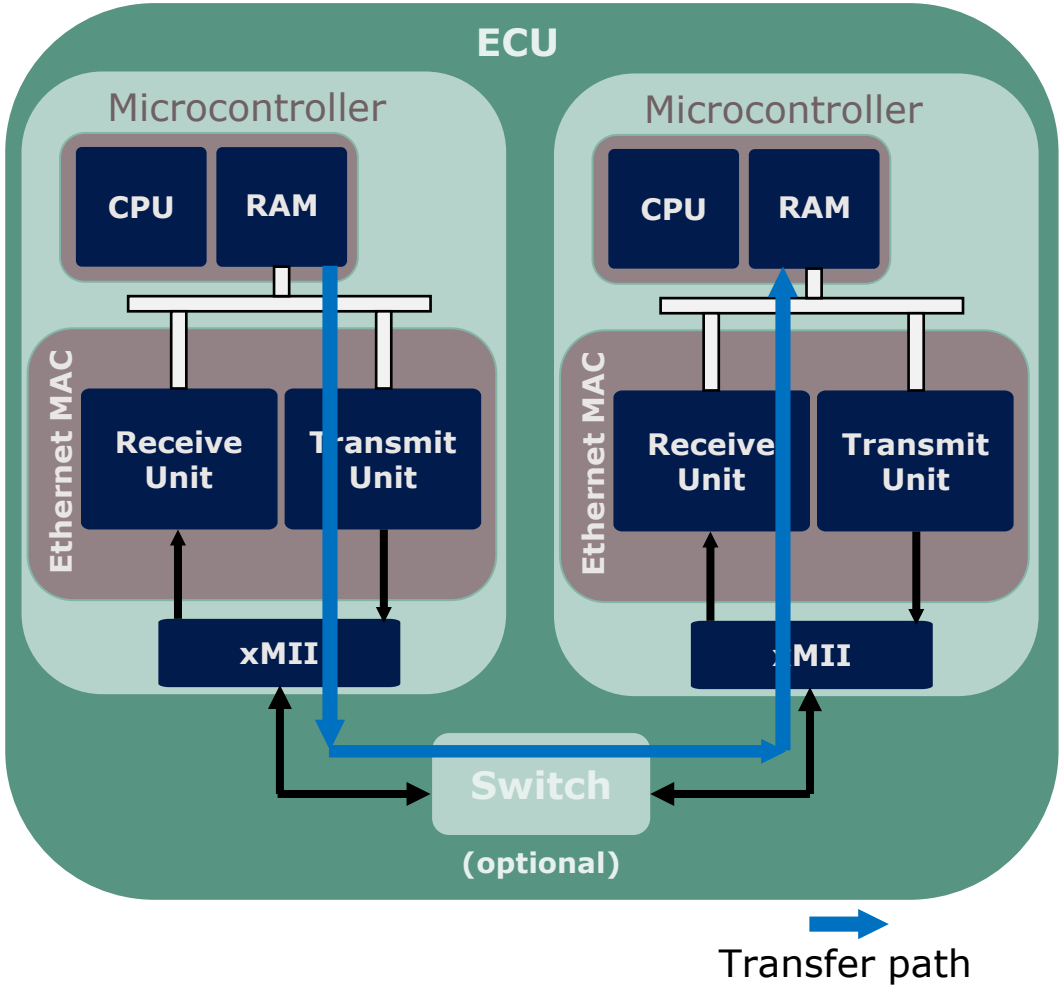
*IVN = In-Vehicle Network

Use Case In-ECU Communication



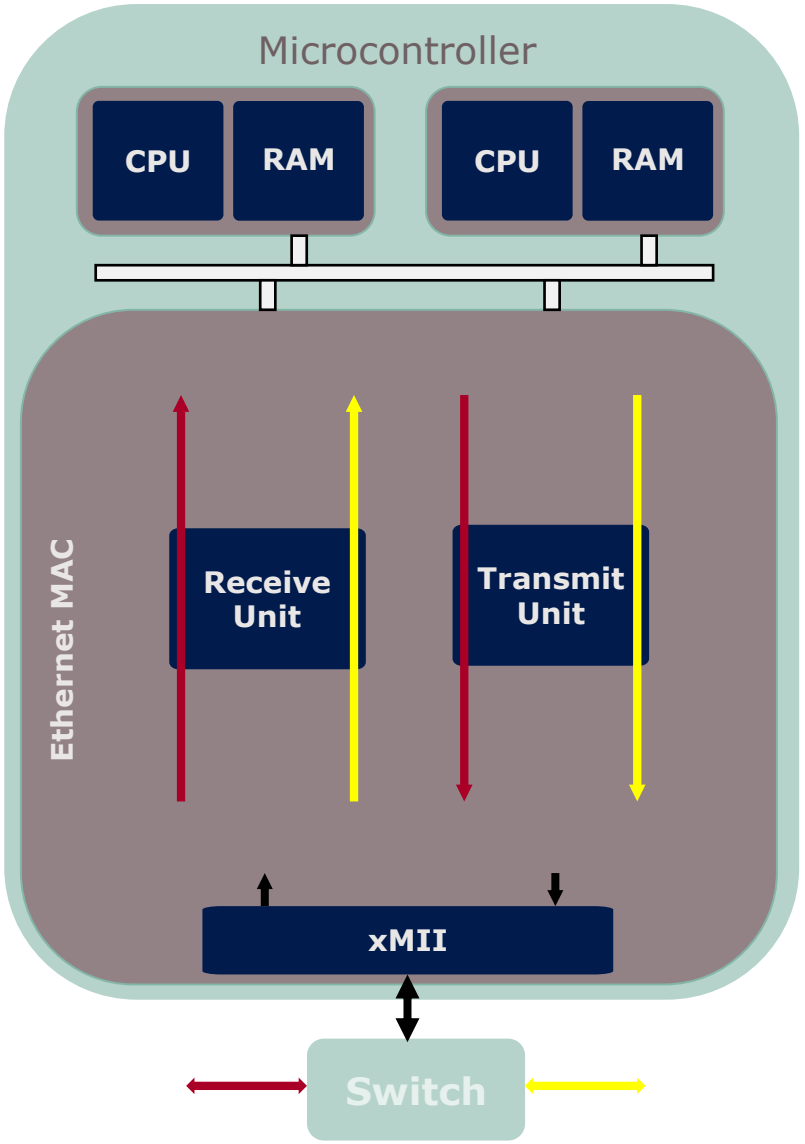
Use Case Memory-to-Memory Copy

- System requirement
  Synchronize content of variables
  - Data transfer between controllers (MCUs)
  - Data block copy process
  - The process is scheduled
  - The process execution time is constraint
  - MCU connection options
    - Directly coupled via xMII
    - Coupled via a switch



Use Case Memory-to-Memory Copy

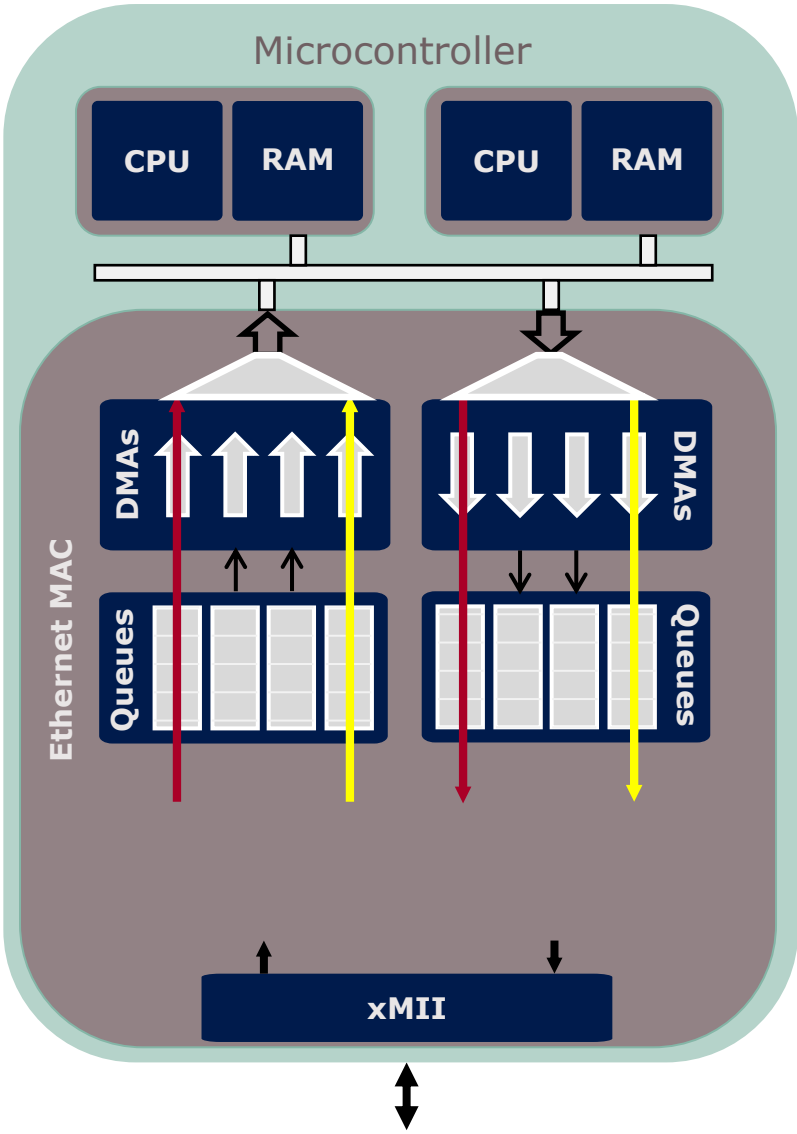
- MCU level requirement
  - Data flow separation inside the MCU – no dependencies
  - Any receive flow can be directed to any CPU core (in case of multicore MCU)
  - Any transmit flow can be sent from any CPU core
  - Precise scheduling for data transmission
  - Low software load



Note:
Red arrows represent MCU to MCU flow
Yellow arrows represent other flows

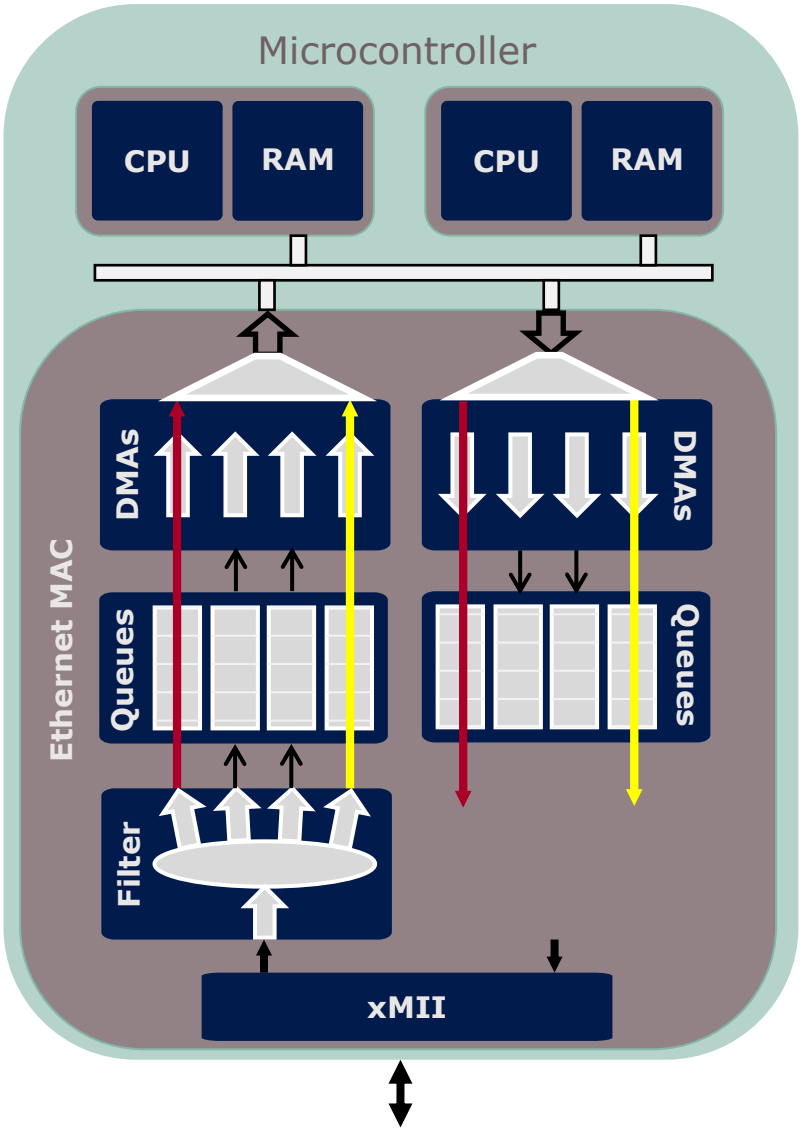
Use Case Memory-to-Memory Copy

- Implementation
  Separation of data flows
  - IEEE 802.1Q VLAN separation
  - Introduction of several channels as “queues”
  - Each queue holds data independently
  - Each queue is emptied / filled by its own DMA



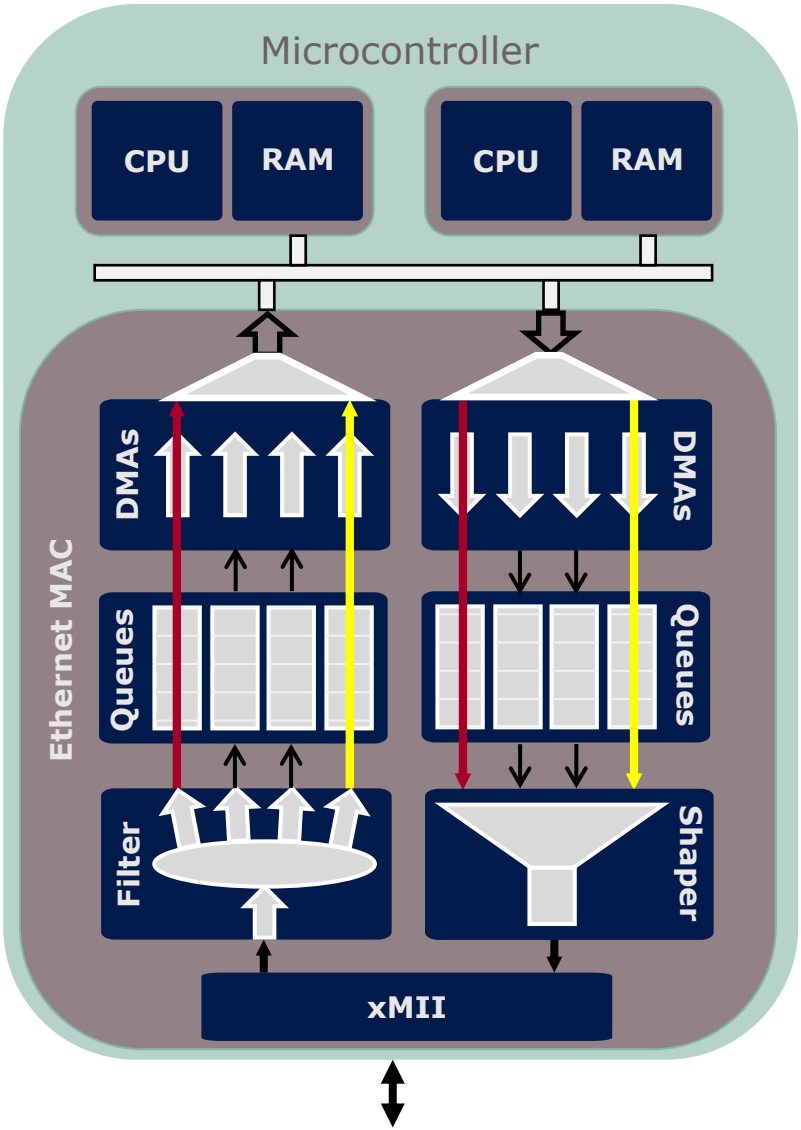
Use Case Memory-to-Memory Copy

- Implementation
  Separation of data flows (inbound)
  - IEEE 802.3 frame header filter
  - Separates inbound traffic based on destination address
  - Forwarding to according queue
  - In turn selects destination CPU



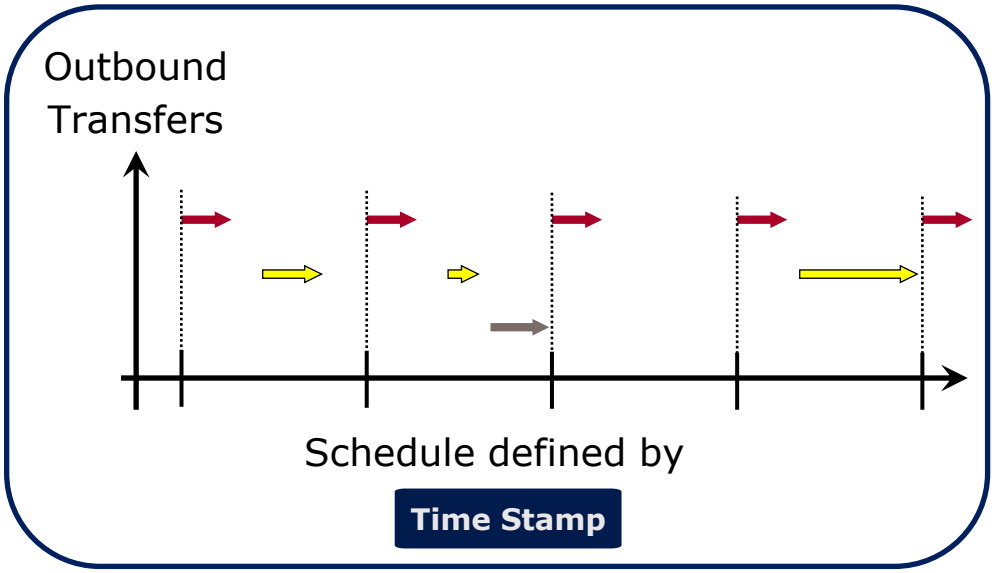
Use Case Memory-to-Memory Copy

- Implementation
  Separation of data flows (outbound)
  -> re-merging of the flows
  - IEEE 802.1Q shaper
  - Separate outbound traffic is merged before the xMII interface
  - Merging is based on IEEE 802.1Q shaper rules

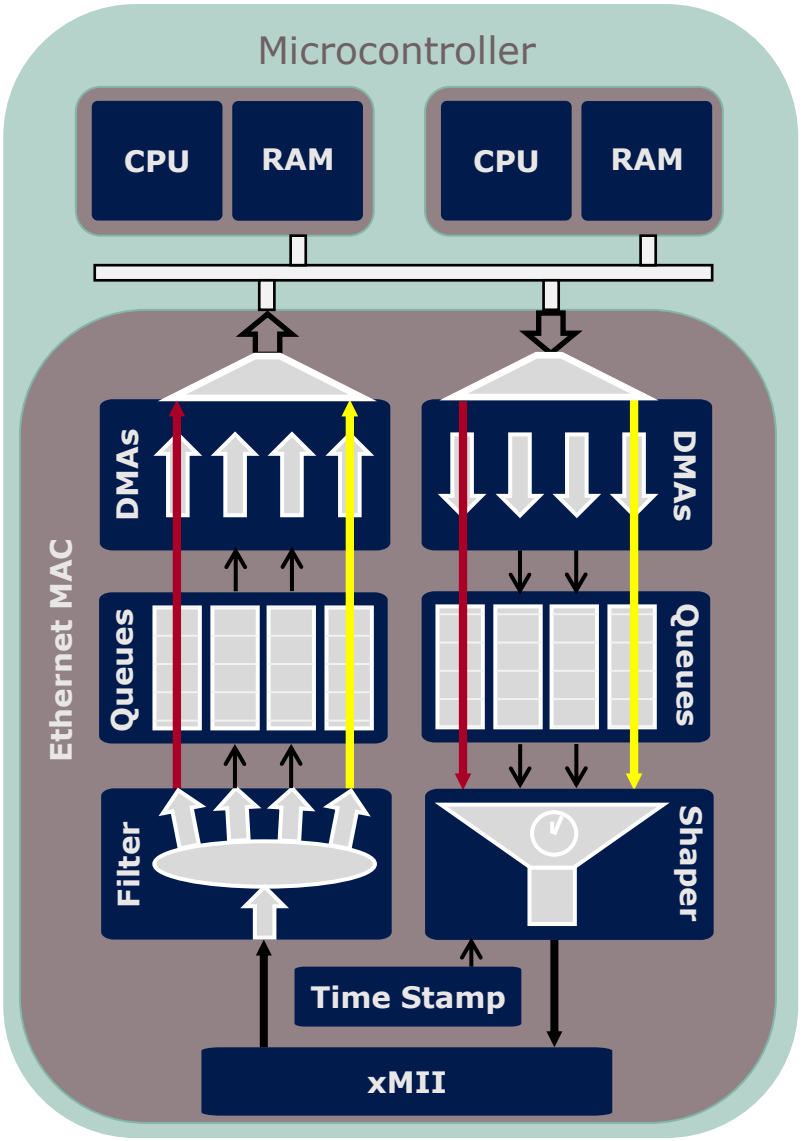


Use Case Memory-to-Memory Copy

- Implementation
  Outbound data flow is scheduled
  - Using IEEE 802.1AS
  - 802.1AS drives a scheduler
  - Queue output is driven by the scheduler (in “shaper” block)

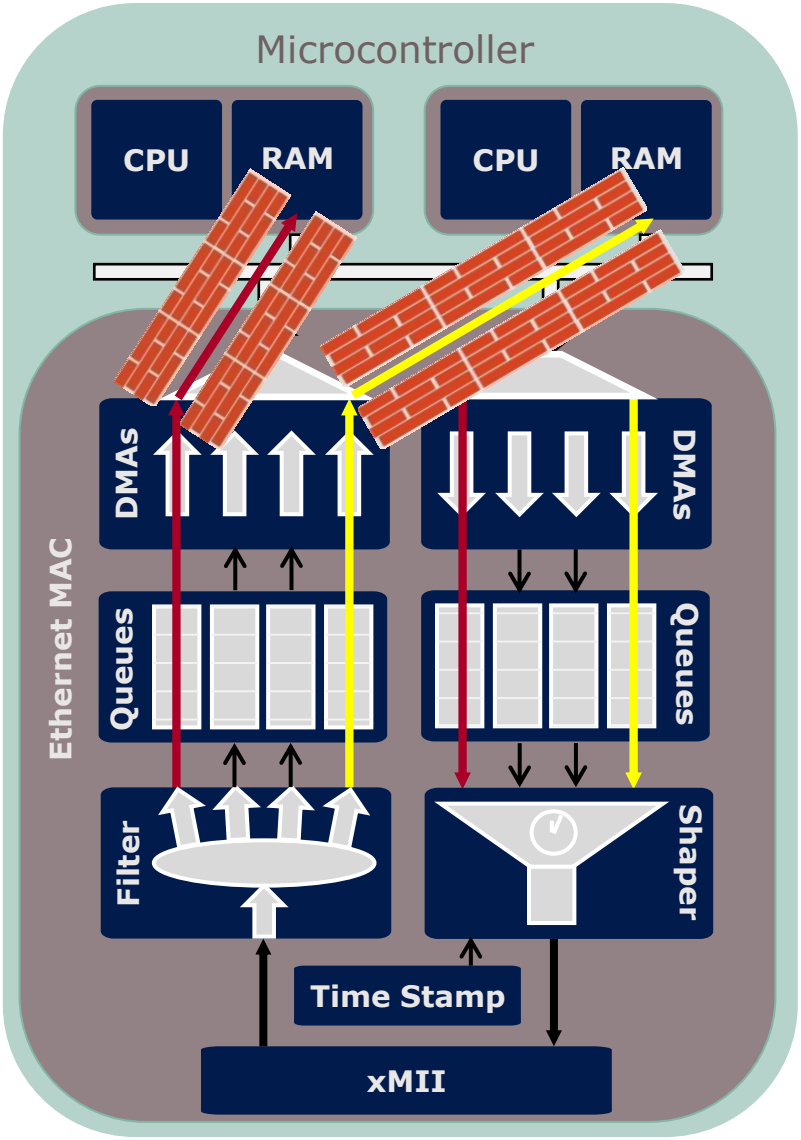


Red arrows represent MCU to MCU flow
Yellow and grey arrows represent other flows



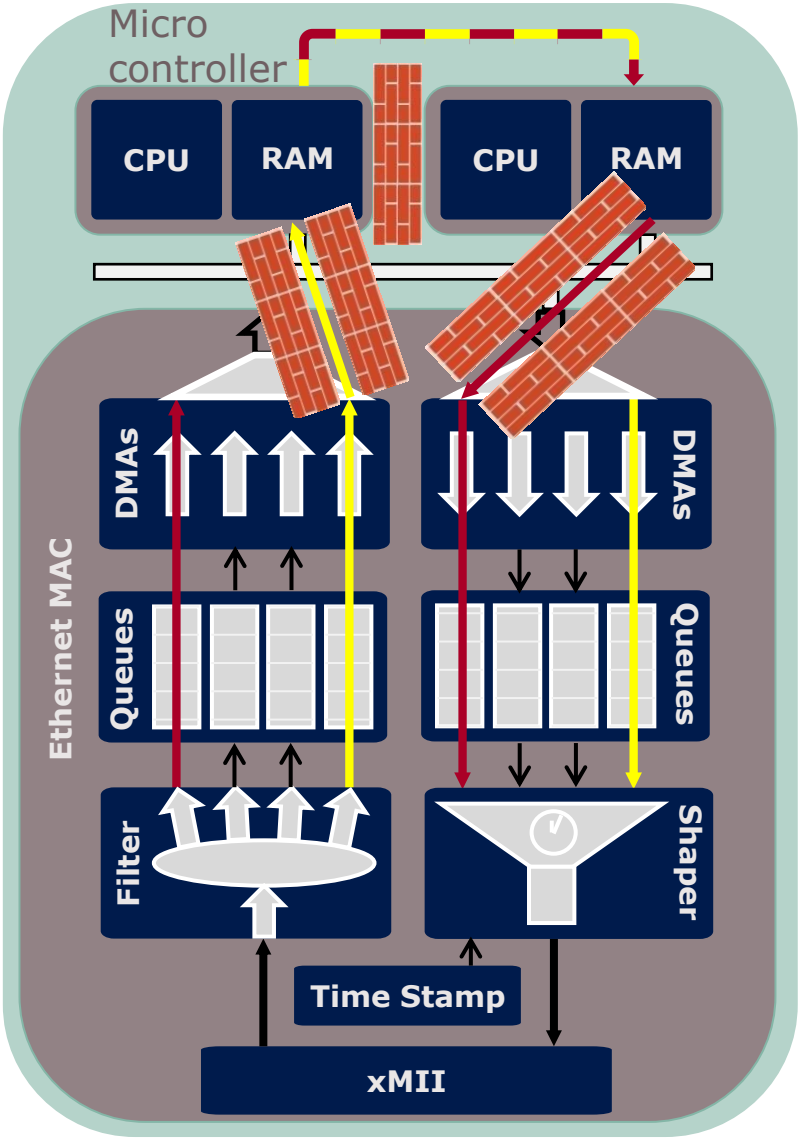
Use Case Memory-to-Memory Copy

- Implementation
  Data flow separation in the MCU
  - Needs freedom from interference
  - Transfer paths of DMAs are safe guarded
  - Potential DMA write targets can be protected



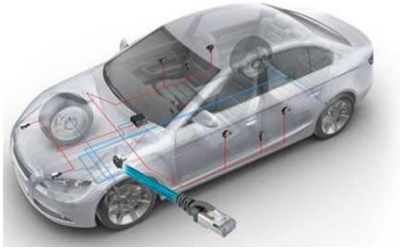
Use Case Memory-to-Memory Copy

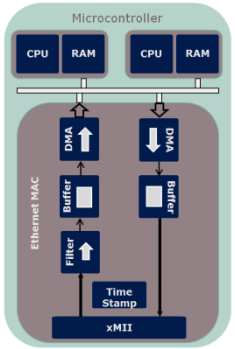
- Implementation
  Data flow separation in the MCU
  - Freedom from interference enables firewalling the flows
  - Processes and their data flows can be separated to different CPUs

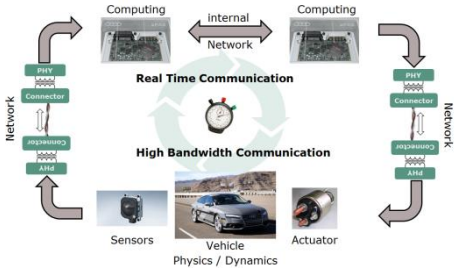


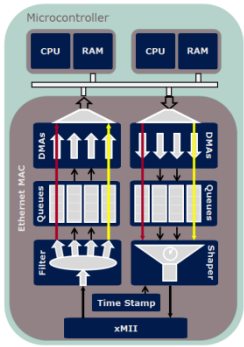
Summary

- The next gen Ethernet implementations will expand the support for Automotive application use cases
- There will be HW features for use cases requiring deterministic data transfer
- IEEE 802 offers the standards to make the step successful

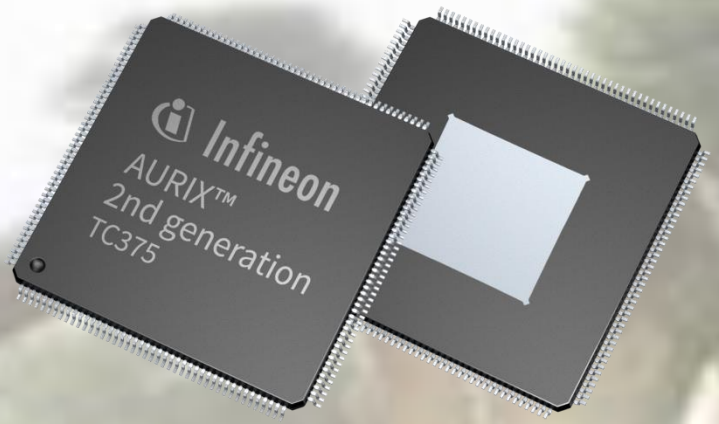








Thank You !



CAN-FD

FlexRay

IEEE Ethernet

1 Gbit/s
100Mbit/s

ASC LIN

SENT

PSI5