IEEE 802 Plenary Session
March 11-16, 2012 - Waikoloa, Hawaii

General perspective on AVB 2 & Redundancy aspects relevant for backbone and control applications

Markus Jochim
General Motors

Structure of the presentation

- **PART I: Ethernet for backbone and control applications (Motivation)**
- **PART II: Redundancy related requirements**
- **PART III: General perspective on AVB Gen 2**

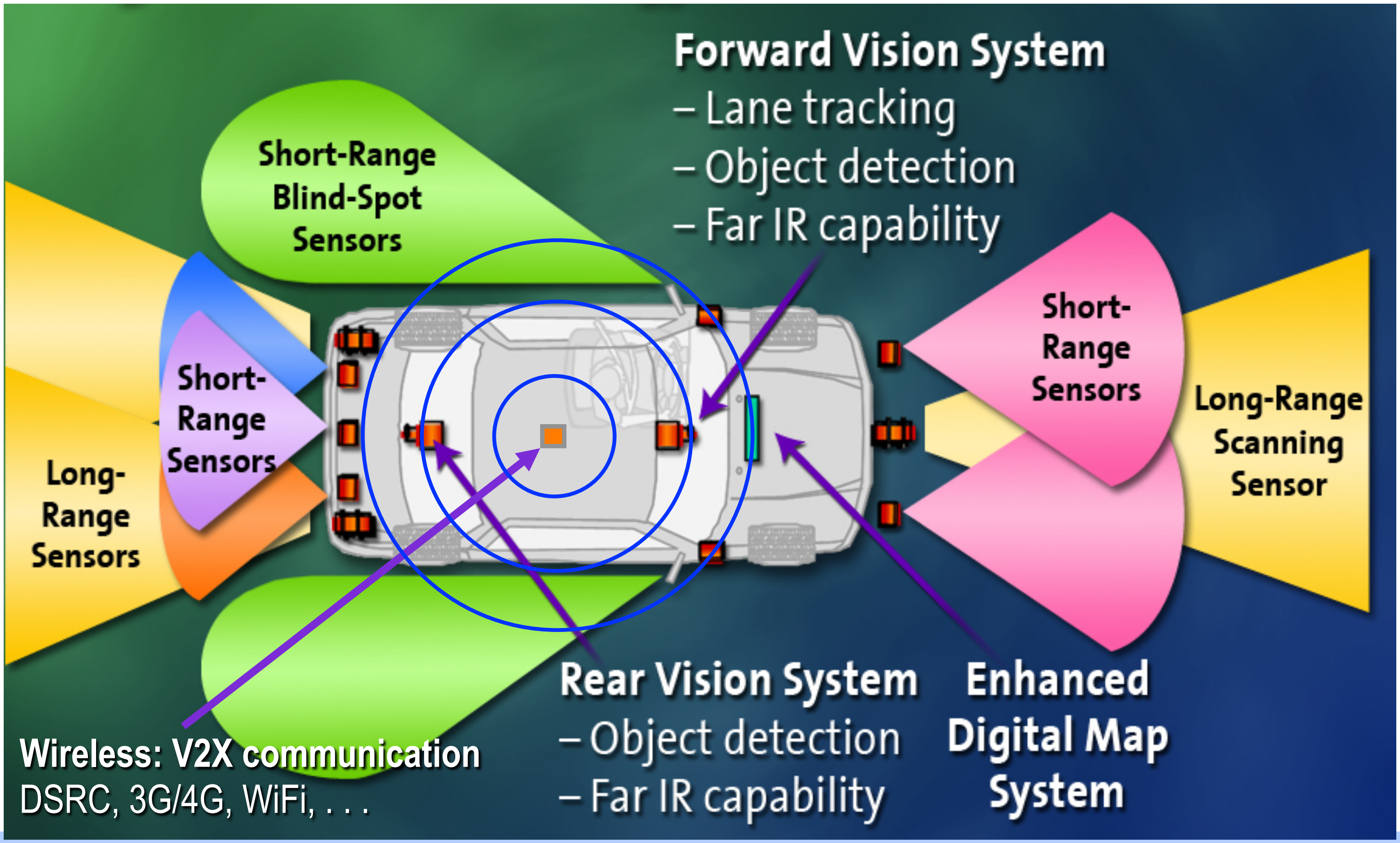
<u>**Note:**</u>

From the requirements perspective this presentation focuses on the redundancy aspects / the support for safety critical applications we would like to be addressed.

There are several other AVB Gen 2 concepts that are of relevance for future use cases that are not explicitly addressed in this presentation (e.g. "ultra low latency / time aware shaper / preemption etc.").

Part I:
Ethernet for backbone and control applications - Motivation -

Smart & Connected Vehicle



Ethernet for Control Applications

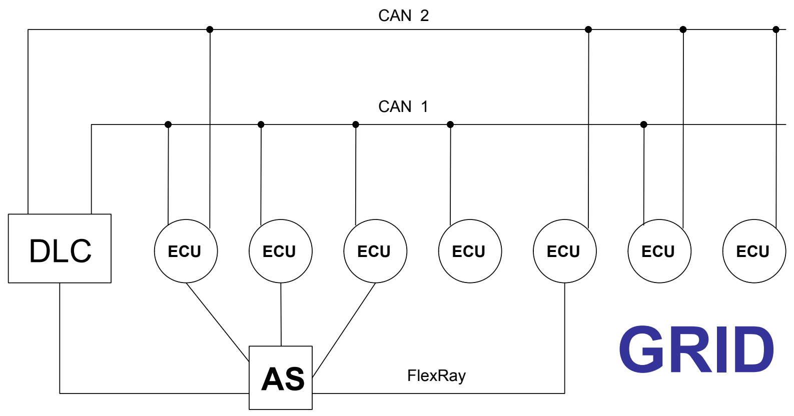
- The bandwidth requirements of control applications in Smart & Connected Vehicles of the future cannot be met by CAN.
- FlexRay is an option for an intermediate solution, but is limited by 2 x 10 Mbit/s and doesn't scale from bandwidth perspective.
- Ethernet has the potential to meet future bandwidth requirements.

Ethernet based infrastructure backbones
- Motivation -

Grid Topologies

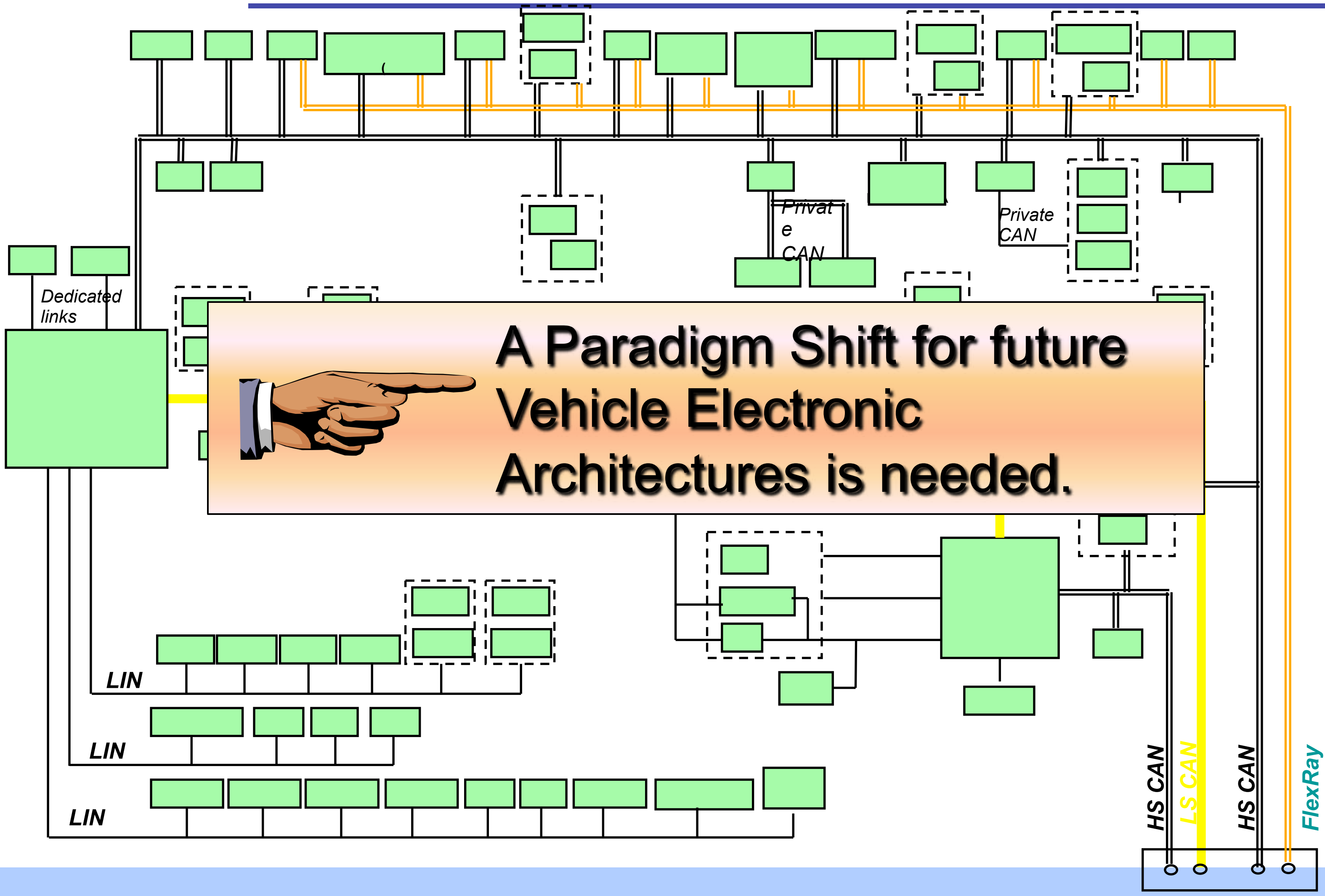
- Today's topologies are typically based on various technologies like CAN, FlexRay, MOST and Ethernet.
- No support for network protocols (OSI Layer 3) like IP.
(⇒ No direct communication across different serial data systems.)
- Instead: Some ECU's are connected to more than one communication system (e.g. CAN and FlexRay or different CAN domains).
(⇒ These ECU's are either just receiving information from different bus systems or have to carry out complex, customized gateway functions.)
- Adding more and more subsystems in an ad hoc manner increases the overall complexity tremendously.

⇒ Concerns: Analyzability, Complexity, Scalability, Maintainability



GRID

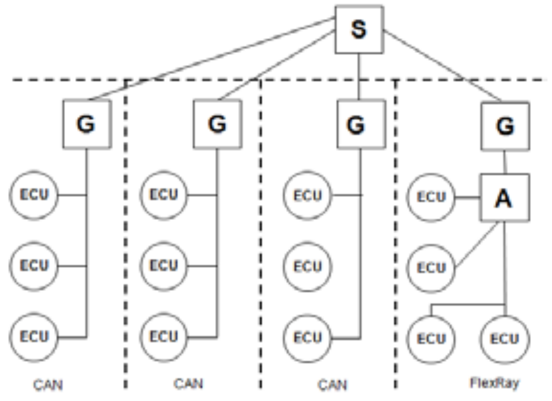
100 ECUs and 10 independent communication busses



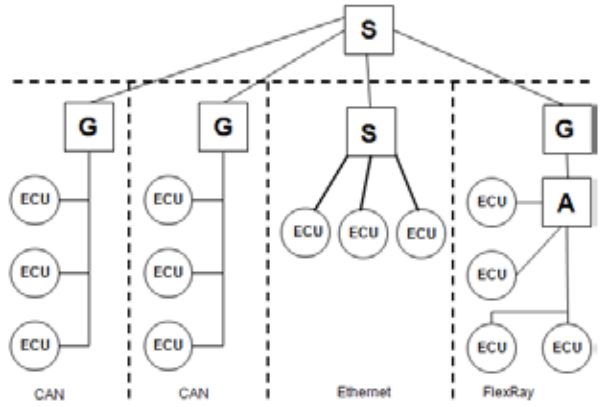
A Paradigm Shift for future Vehicle Electronic Architectures is needed.

Vision: Ethernet Backbone to Logically Flat Network

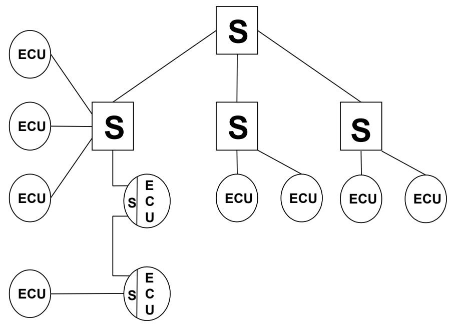
Step 1: Ethernet Backbone



Step 2: Backbone++



Step 3: Flat Network



- Step 1: A vision for the future is to move towards more regular Ethernet based topologies that avoid ad hoc connections. In a first step domain gateways (potentially integrated with domain ECUs) can connect the domains to the Ethernet backbone.

- Step 2: Gateways can be removed step by step as domains migrate to Ethernet.

- Step 3: The vision is a logically flat network where all communication and addressing is performed on lower ISO/OSI layers (Layer 2 and potentially Layer 3 (IP routing)).
  A "puristic" logically flat network is <u>not</u> a realistic topology for heterogeneous automotive networks.
  We see the idea of the logically flat network more in the sense of a vision that shows a direction.

Challenges

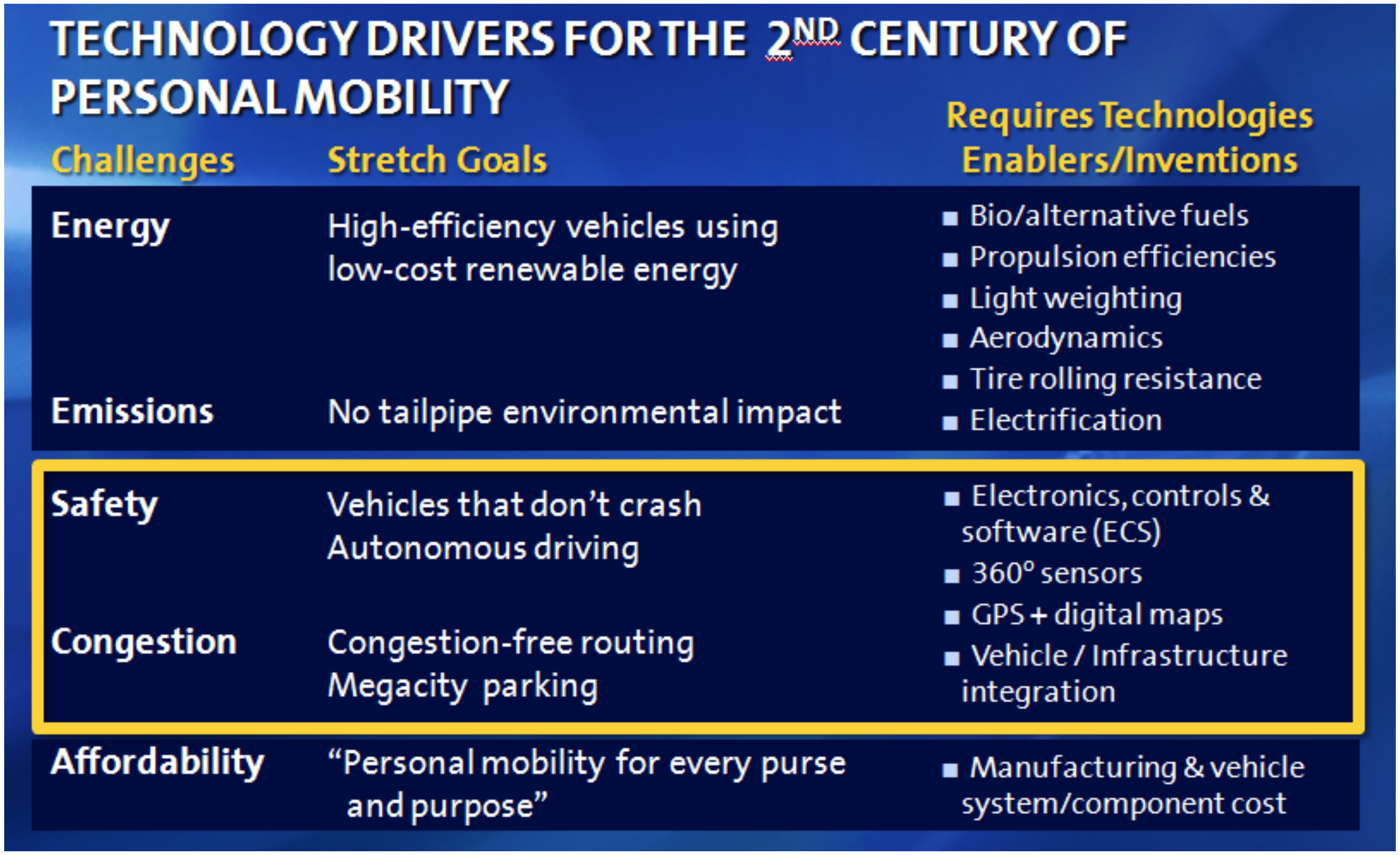
What challenges are we facing with "Ethernet" (= without AVB or TTEth.) for Backbone and Control ?

- **Guaranteed Delivery / Bandwidth?**
  Even if QoS features are used, frames can still be dropped in switches.
- **Latency Guarantees?**
  In general no bounded maximal latency.
- **Time Synchronization?**
  No low level support for time sync (like in FlexRay).
- **Redundancy?**
  Only limited inherent support for redundancy and fault tolerance (e.g. link aggregation, STP).

Currently assessing Std. Ethernet (without AVB), AVB Ethernet and TTEthernet for Backbone and Control.

Part II:
Redundancy related requirements

Motivation

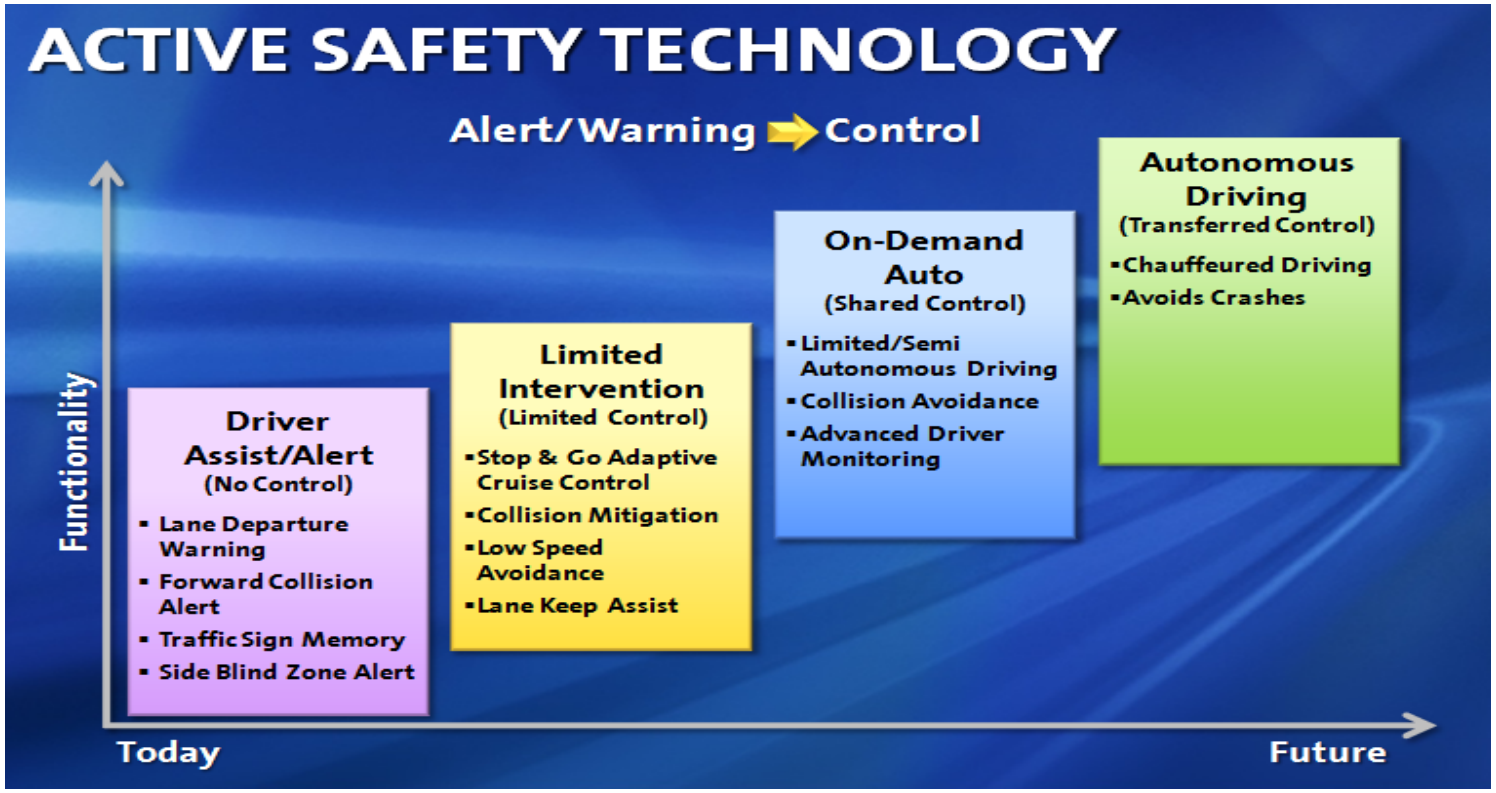


TECHNOLOGY DRIVERS FOR THE 2ND CENTURY OF PERSONAL MOBILITY

Challenges	Stretch Goals	Requires Technologies Enablers/Inventions
Energy	High-efficiency vehicles using low-cost renewable energy	■ Bio/alternative fuels ■ Propulsion efficiencies ■ Light weighting ■ Aerodynamics ■ Tire rolling resistance ■ Electrification
Emissions	No tailpipe environmental impact	
Safety	Vehicles that don't crash Autonomous driving	■ Electronics, controls & software (ECS) ■ 360° sensors ■ GPS + digital maps ■ Vehicle / Infrastructure integration
Congestion	Congestion-free routing Megacity parking	
Affordability	"Personal mobility for every purse and purpose"	■ Manufacturing & vehicle system/component cost

Motivation

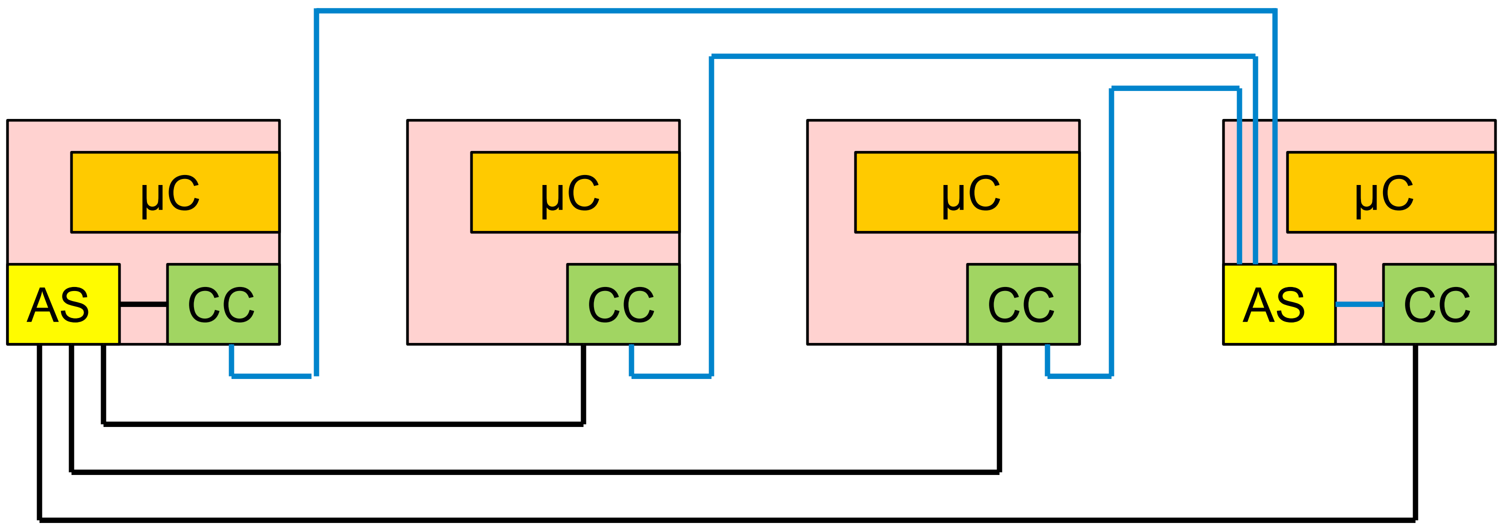
Increased demand for safety critical / fail operational applications!



➢ Communication system should support the design of such systems !

Example: FlexRay

- Example: Dual channel FlexRay system with integrated Active Stars (= Hubs)



CC = FlexRay Communication Controller

AS = Active Star

Blue = Channel A
Black = Channel B

- In safety critical systems, structural redundancy is typically not used to increase bandwidth, but to send redundant information over redundant paths.
(This discussion does NOT focus on structural redundancy for increased bandwidth!)

Example: FlexRay

- Support for safety / fault tolerance was designed into FlexRay:
  - Dual channel support
  - Redundancy management
  - Fault tolerant clock sync algorithm
  - Fault tolerant startup algorithm
  - Fault isolation capabilities in transceivers and active stars including limited protection against babbling idiots.

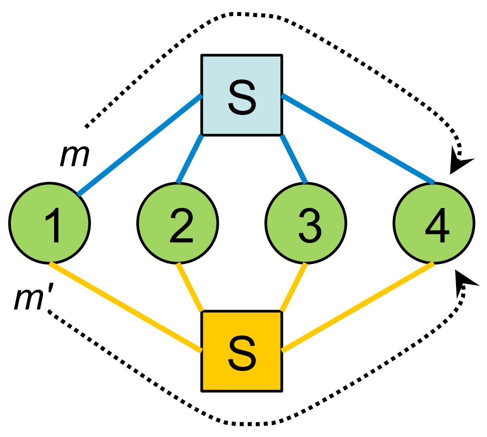
- Dual channel FlexRay systems are planned to be used for safety critical automotive applications in the foreseeable future!

Support for Safety & Fault Tolerance

- What type of S&FT mechanisms should be part of Gen 2?
- Following the guiding principles:
  - P1: "As much as possible for as little as possible"
  - P2: "Make expensive features optional"
- Redundancy / fault tolerance related mechanisms discussed on the following slides:
  - Static structural redundancy
  - Dynamic structural redundancy
  - Redundancy management
  - Error propagation / Fault isolation / Fault tolerance

Static structural redundancy (1/5)

Example 1: Massiv structural redundancy ("2 channels")



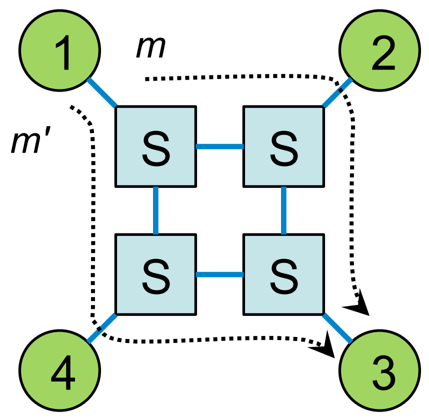
4 nodes, 2 LANs, m=m'

- Various implementation options. For example:
  - "2 fully independent LANs":
    Two Eth. controllers per node are sending out redundant messages on two independent LANs. Currently a protocol like PRP would be required.
  - "1 LAN":
    3-port switches integrated into nodes receive a message via MII and will send the message out on two outgoing ports. Approach requires protocol modifications / is not compatible with RSTP.
- No network reconfiguration required if a link/switch fails.
  ⇒ Link or switch failure will not affect real time guarantees.

- Receiving nodes need to perform redundancy management (see slides 34ff).
- Conceptually simple approach
  (Simplifies the complex task to analyze / validate system behavior in presence of faults.)
- High costs due to high degree of redundancy

Static structural redundancy (2/5)

Example 2: Moderate structural redundancy (Ring topology)



- 4 nodes, 1 LAN, m = m'
- Three port Switches (potentially integrated into nodes) are sending redundant messages in two directions.
  (Similar to HSR)
- No network reconfiguration required if one link or switch fails
  ⇒ Link or switch failure will not affect real time guarantees.
- Receiving nodes need to perform redundancy management
  (see following slides).
- Moderate costs due to low degree of redundancy
  - Failure of e.g. a switch may however disconnect a single node from the network while the rest of the network remains operational.
- Note:
  While critical messages are sent in both directions, it is sufficient for non-critical messages to be sent in one direction only (bandwidth consideration).

Static structural redundancy (3/5)

- If AVB Gen 2 does not support static structural redundancy, the need for static structural redundancy could be addressed at the system level.
- Example:
  Connect each node to two independent LANs and manage synchronization and redundancy management at the application layer.
  Adds complexity and puts a burden on the SW and the microcontroller. ☹

Our perspective:

- Static structural redundancy is relevant for the implementation of some automotive safety critical applications that require challenging real time deadlines in presence of faults.
- Support for the implementation of safety critical automotive systems by means of an IEEE standardized low level mechanisms (layer 2) that support static structural redundancy is desirable.

Static structural redundancy (4/5)

Questions: Which mechanism should be used to enable static structural redundancy?

- "Seamless Redundancy" (= static structural redundancy)
  - Listed in AVB Gen 2 assumptions presentation (www.ieee802.org/1/files/public/docs2011/avb-pannel-gen2-assumptions-0711-v5.pdf).
  - Is it already decided which mechanism will be used to achieve this?

- Some options:
  - Use VLANs not under the control of either STP or SPB to create multiple paths between particular nodes in a network
    www.ieee802.org/1/files/public/docs2012/new-avb-nfinn-spb-tsn-0112-v01.pdf (Slide 18)
  - Configuring two VLANs that will utilize / block different paths based on VLAN specific spanning trees?
  - Redundancy mechanisms available in TTEthernet?
  - PRP for independent LANs?
  - HSR for ring topologies?

Other alternatives? What are the Pros and Cons?

Static structural redundancy (5/5)

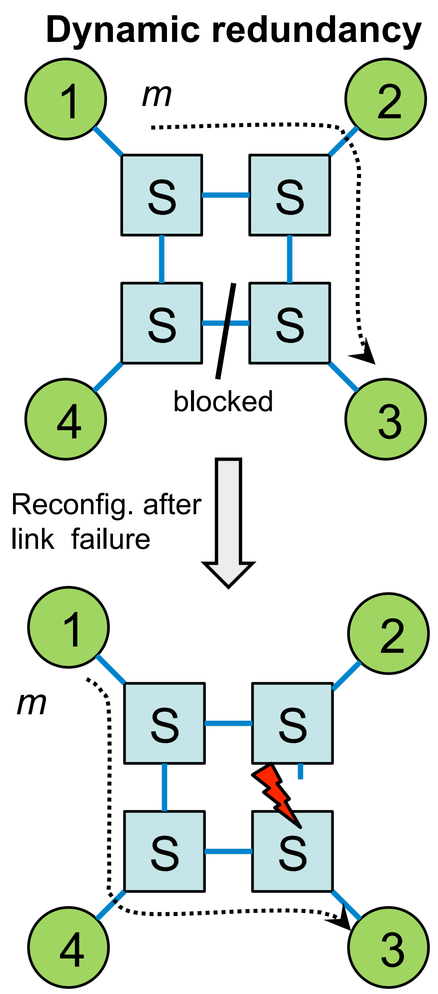
Further questions / discussions:

- Redundancy support seems most important in the context of the new ultra low latency class. (Time aware shaper, Preemption). Any "compatibility" issues?

- Costs:
  - Principle P2: "Make expensive features optional"
  - Static structural redundancy can be costly

    ⇒ There should be no cost impact on implementations that do not require structural redundancy.
  - Is this a simple goal to achieve?

Dynamic structural redundancy (1/4)



Example 1: Ring topology

- Example: 4 nodes connected to 3-port-switches
  (Switches can be integrated with nodes)
- RSTP will reconfigure system in case of link / switch failure.
- Receiving node does not need to perform redundancy management.
- Reconfiguration takes time.
  ⇒ Impact on worst case real time guarantees.
- Moderate costs due to low degree of redundancy
  - Failure of e.g. a switch may however permanently disconnect a single node from the network while the rest of the network will be operational after reconfiguration.

Dynamic structural redundancy (2/4)

- Comparing:
  i. "Ring Topology - Dynamic structural redundancy" with
  ii. "Ring Topology - Static structural redundancy"
- Question: What is a use case for option i) if option ii) is less complex and has zero reconfiguration time?
- Answer:
  In applications that can afford the required reconfiguration time option i) is of interest since the network is not burdened with redundant messages.

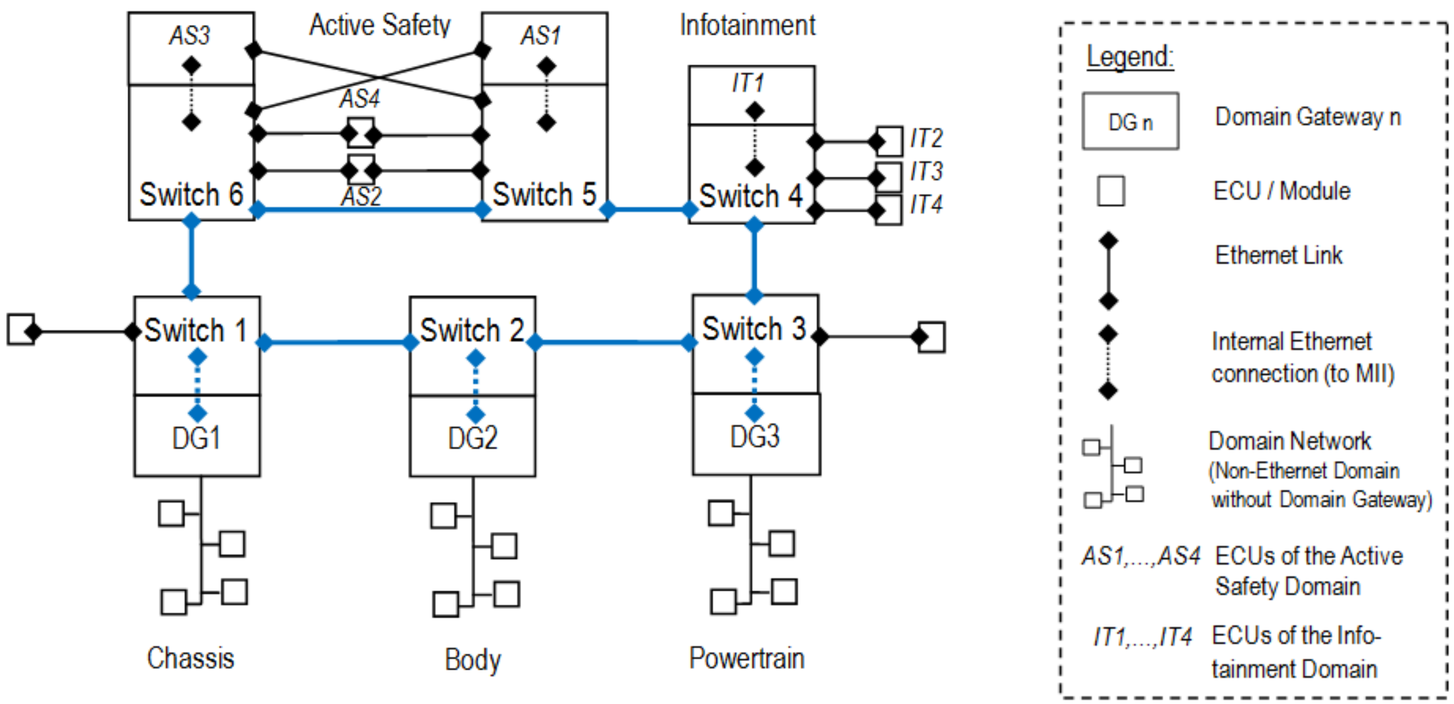
Our perspective:

Dynamic structural redundancy is of interest for automotive use cases that have moderate latency requirements as well as for use cases with tight latency requirements that can tolerate the presence of a fault to result in deadline violations for a limited time t. Knowledge of the max. value of t must be available at system design time.

Dynamic structural redundancy (3/4)

Example:
The moderate degree of redundancy ("ring" instead of "daisy chain") can increase the robustness of an Ethernet backbone (blue) that connects various automotive domain networks.

Utilizing dynamic redundancy avoids burdening the backbone with redundant messages.

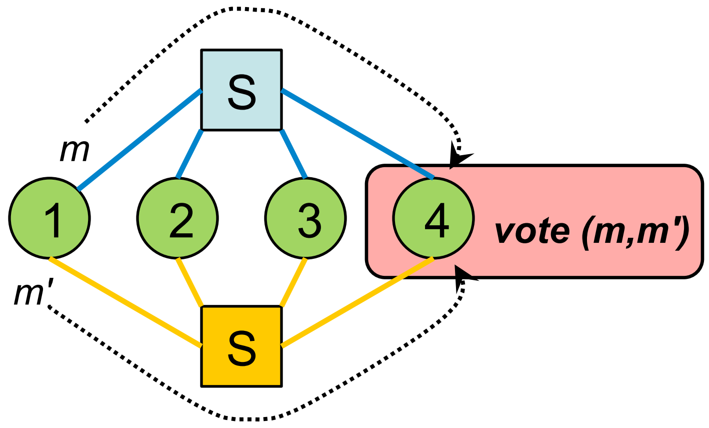

Active Safety
AS3
AS1
AS4
AS2
Switch 6
Switch 5
Infotainment
IT1
IT2
IT3
IT4
Switch 4
Switch 1
Switch 2
Switch 3
DG1
DG2
DG3
Chassis
Body
Powertrain
Legend:
DG n
Domain Gateway n
ECU / Module
Ethernet Link
Internal Ethernet connection (to MII)
Domain Network (Non-Ethernet Domain without Domain Gateway)
AS1,...,AS4 ECUs of the Active Safety Domain
IT1,...,IT4 ECUs of the Infotainment Domain

Dynamic structural redundancy (4/4)

Questions: Reconfiguration time?

- Assumption:
  Small engineered networks with a small number of nodes and a small number of redundant links.
- Question:
  - Can lookup table based decisions on the activation of previously blocked links be used as an alternative to RSTP to reduce the network reconfiguration time?
  - What is the performance improvement when compared with RSTP?
    (A lower bound for table lookup based decision is the time required for physical link failure detection. Lower ms or 100's of µs range assumed.)

Redundancy management (1/4)



- Receiving nodes need to perform redundancy management
- Purpose of redundancy management:
  Assess / compare both messages m, m'.
- Fault free case: m = m'.
  Conclusion: Message m can be delivered to the application.

- Protocol needs to be able to associate m with m'
  (E.g. based on seq. number or time / sequence of reception)

- Two alternative concepts for implementing Red. management:
  - Red. management at data link layer
  - Red. management at higher layers

  (See following slides.)

Redundancy management (2/4)

- **Concept 1 ("Red. management performed at data link layer"):**
  - Some standard redundancy management voting strategies could become part of the data link layer.
  - Examples for voting strategies that can be implemented:
    - Pick first:
      Whichever message (m or m') arrives first will be delivered.
      Only "standard checks" (E.g. CRC, frame format) are performed.
      Potential issue: Latent fault on one channel may remain undetected.
    - Compare:
      Deliver a message only if m and m' are both received and m=m' holds.
  - For a concrete application the data link layer could then be configured to activate one of the preconfigured strategies.
  - Pros:
    - Preconfigured strategies will cover the majority of all cases.
    - Simple concept: Only 1 message is delivered to the higher layers.
    - The solution is transparent to application & middleware !

Redundancy management (3/4)

- **Concept 2 ("Red. management at higher layers"):**
  - The data link layer could be enabled to deliver both messages m and m' to the higher layers.
  - Enables the implementation of application specific redundancy management functions at a higher protocol layer.
  - Allows monitoring of all redundant messages by the application / middleware.

Our perspective:
Supporting both concepts would provide maximum flexibility.

Redundancy management (4/4)

Questions / Discussion:

- Assumption:
  - Both concepts could be implemented without too much overhead?

  Is that a correct assumption?

- Assumptions on costs (Principles P1 & P2):
  - Redundancy management in endstations will have no noticeable cost impact.

  Correct?

Error propagation / Fault isolation / Fault tolerance (1/7)

- **Purpose of the following six slides:**
  - Give a rough idea of some of the concepts and typical concerns that are relevant when designing safety critical automotive systems.
  - Raise the sensitivity for the need to address topics that are related to fault isolation / fault tolerance.

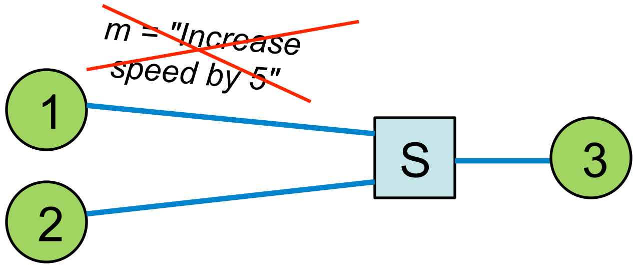
Error propagation / Fault isolation / Fault tolerance (2/7)

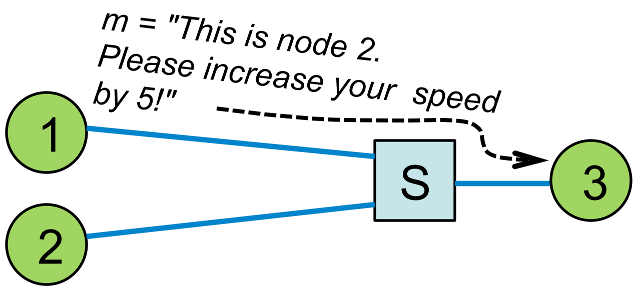
Fault containment regions:

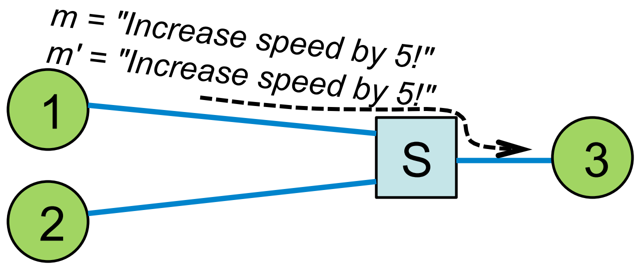
- A network consists of components: node 1, node 2, switch a, switch b, link 1a, etc. (Other levels of granularity are of course possible. E.g.: µController of node 1, memory of node 1, PCB etc.)
- Example of a goal for a safety critical application:
  - A single fault that affects a component must not lead to an event that has been defined to be unacceptable.
  - What constitutes a reasonable goal and what constitutes an unacceptable event typically depends on the properties of the application.
- A common approach when designing safety critical systems:
  Partitioning of a system into Fault Containment Regions (FCR)
- A FCR is a set of components. The failure of a component that is element of FCR X may propagate and cause other components within X to fail.
- This means:
  Components within the same FCR are not guaranteed to fail independently.
- But: A fault that affects a FCR x must not result in failure of component within FCR y!

Error propagation / Fault isolation / Fault tolerance (3/7)

- **Failure modes:**
  - Components can fail in different modes. Examples:
    - Fail silent: *"An Ethernet controller affected by a fault stops sending frames."*
    - Masquerading faults: *"A node 1 affected by a fault copies the MAC address of node 2 into the source address field of a frame"*
    - Duplication of messages (unintended): *"A faulty microcontroller sends a message twice."*







Error propagation / Fault isolation / Fault tolerance (4/7)

- The question which components can fail in which mode typically depends on:
  - the technical properties of the component.
  - the fault the component is exposed to (e.g. EMI, undervoltage, resistive short, stuck at fault in a register, bit flip of a memory location, instruction set of µController affected by a fault, etc.).

- Error propagation:

  The failure of a component affects other components in the system, which in turn become faulty components.
  - OK: If these components are located in the same FCR
  - Not OK: If they are located in different FCRs !!!

- Fail silent behavior is often desirable since it typically prevents error propagation.

- But: More complex failure modes may require fault isolation mechanisms or fault tolerant protocols to prevent error from propagating to other FCRs !

Error propagation / Fault isolation / Fault tolerance (5/7)

- **Example 1: "Babbling Idiot"**
  - A fault causes a node to (permanently or erratically) transmit "additional" frames.
  - Absent fault isolation, the fault propagates from the node to the link, the switch and other nodes.
  - The babbling idiot may then consume bandwidth that other nodes require (Violation of QoS guarantees).
  - Fault isolation by traffic policing in switches (= bus guardian function):
    - RC traffic:
      A bus guardian function integrated into the switch can filter out frames that violate the TSpec associated with a given stream (RC traffic).
    - Time triggered traffic ("Time aware shaper"):
      A similar mechanism needs to be defined for time triggered traffic.
    - Best effort:
      No need / no chance to perform traffic policing for best effort traffic.

Error propagation / Fault isolation / Fault tolerance (6/7)

- **Example 2: "Clock synchronization"**
  - A fault can cause a node in a synchronized system to develop a wrong notion of time.
  - Countermeasures may be required, to prevent the faulty node from distributing faulty timing information that will mess up other nodes notion of time. (Depends on the details of the time sync algorithm.)
  - Examples:
    - FlexRay uses the fault tolerant midpoint algorithm which limits the effect that faulty nodes can have on the time synchronization and guarantees a known precision of the time synchronization even in presence of a tolerable number of faulty sync nodes.
    - TTEthernet also addresses the problem by means of a fault tolerant clock sync. algorithm.

Error propagation / Fault isolation / Fault tolerance (7/7)

- Conclusions related to "Fault isolation / Fault tolerance" :
  - As stated before, a longer term strategic decision to base future automotive EE architectures on Ethernet & IP will require a technology that forms a basis for a multitude of automotive use cases.
  - In order to be able to address safety critical control use cases Ethernet needs to be able to compete with systems like FlexRay, that have been designed for these use cases.
  - If problems like the babbling idiot or the need for a fault tolerant clock sync are not addressed within the IEEE standards, it may (depending on the properties of the application) become difficult to implement safety critical systems on the basis of Ethernet without applying system level redundancy.

Our perspective: Fault isolation and fault tolerance related topics should be carefully addressed within the standard !!!

"Babbling idiot" and "fault tolerant clock sync" are examples of such topics.

Part III:
General perspective on AVB Gen 2

General perspective on AVB Gen2

- It is our perception that some AVB / AVB2 discussions within the automotive industry tend to focus on current requirements.
- Examples:
  - Do we really need QoS guarantees for control applications? We lived without them until now!
  - Do we currently need ultra low latency for Ethernet based systems?
  - Do we need to be able to send critical and non-critical data on the same link? We have never done that before!
  - Do we really need the bandwidth that Ethernet offers?
  - FlexRay's support for redundancy and fault tolerance has not really been used in series up until now. Should AVB2 really support redundancy and fault tolerance?
- While these are not only legitimate but also important questions, AVB2 in our opinion needs to go beyond today's known requirements (see following slides)!

General perspective on AVB Gen2

- Often the concern behind the questions raised is that new Gen 2 mechanisms could drive up the costs.
- Mindset: "Don't increase the price of today's hardware components by adding costs for a feature that is not required for early Ethernet use cases."
- Understandable, since our industry, due to high volumes, is very cost sensitive.
- There is a lot of "wild guessing" going on in our industry, on what the cost impact of the different mechanisms that are planned for AVB Gen 2 (e.g. Preemption) might be.
- Conclusion:
  The costs associated with AVB 2 need to be understood and communicated.
  Some features may only require "a couple of more gates" with no noticable cost effect, but that is not always obvious. Also indirect costs must be considered. Example: FlexRay unlike CAN requires to plan all static frames duing design time. When FlexRay was first applied this had a huge impact on processes and tools that needed to be introduced.

General perspective on AVB Gen2

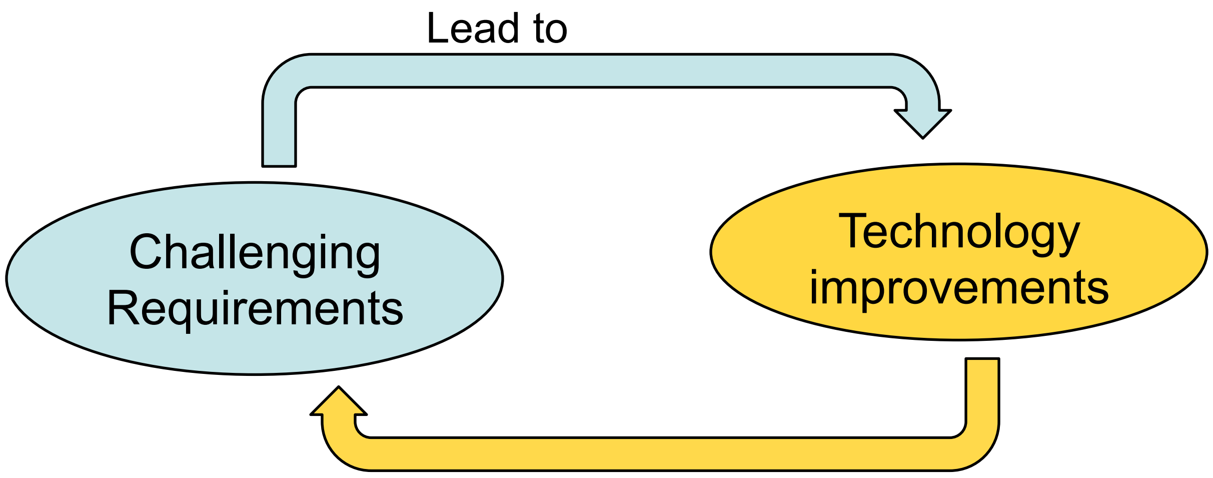
- **Short term** use cases:
  For a single domain specific application a simple argument like "cost reduction due to cheaper cables" might be sufficient to motivate the use of Ethernet.

  BUT:

- **Longer term strategic decisions**:
  Strategic decisions to base future automotive EE architectures on Ethernet & IP will require a technology that:
  - forms a basis for a multitude of very different use cases
    (E.g.: Infotainment, Active Safety, Control, Safety critical Control, Backbone)
  - scales well and has sufficient "headroom" to accommodate future requirements.
  - is very competitive from a cost perspective!

General perspective on AVB Gen2

- We see AVB2 as a technology that will enable new applications and at the same time generate new requirements.



Enable new applications that require even more bandwidth, lower latency, more fault tolerance, . . .

General perspective on AVB Gen2

Analogy: "PC market"

Cite from 1981:
"Nobody will ever need more 640kByte RAM for a PC"

1985: 1MByte RAM is not uncommon.

AVB2 has a long lead time:

ideas ----> concepts & PARs ----> IEEE Gen 2 Specs available ----> early AVB 2 components available --->
2012 *2016* *2017?*

2018 ??? to 2028 ???
----> AVB 2 components as a commodity -> AVB 2 in series applications of some OEMs

⇒ It is in general very difficult to do a reasonable job of anticipating trends & requirements that will be relevant that far into the future!

General perspective on AVB Gen2

Conclusions:

⇒ Do not exclusively focus on today's requirements!

⇒ Build a platform / a technology that enables a multitude of different use cases (Infotainment, Active Safety, Control Applications, Fault tolerant applications, . . .)

⇒ Understand and communicate cost impact of planned Gen 2 mechanisms!

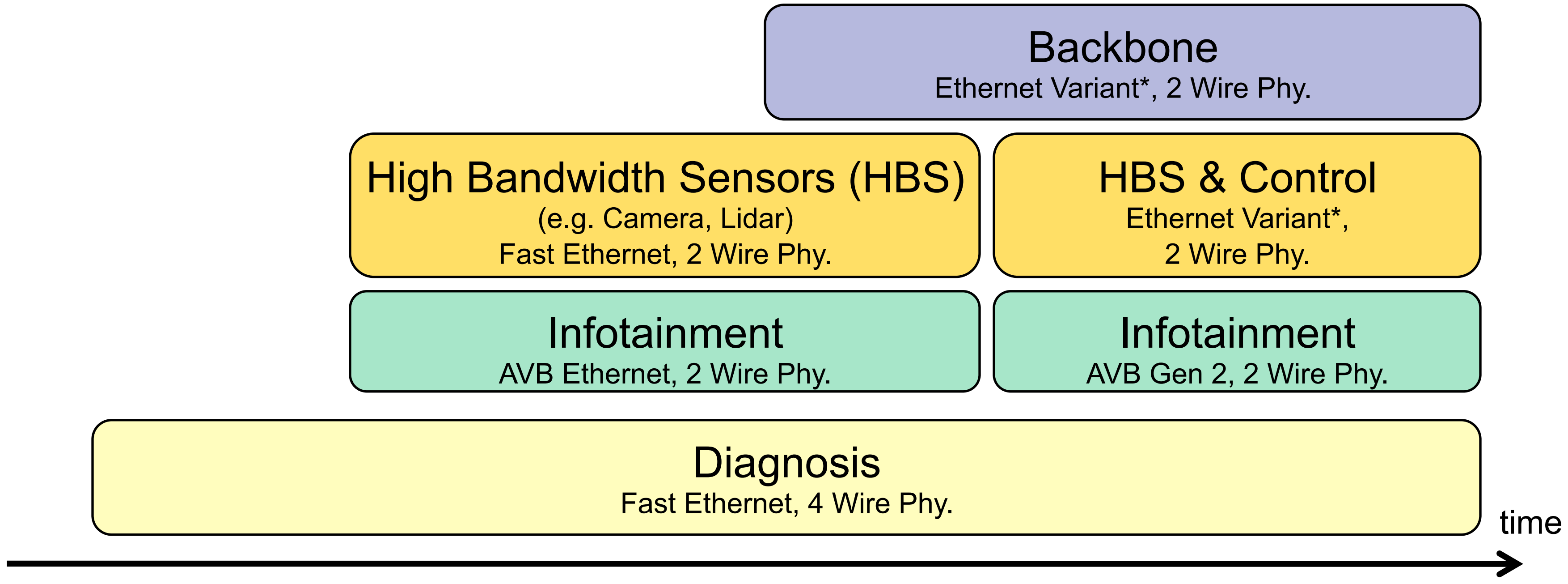
⇒ For features without noticable cost impact
Push the limits and enable what is possible rather than just what is already required or anticipated for the next couple of years!

⇒ New AVB2 features should either come at very low cost or should be optional, so that they can be scaled away when not needed.
Example: Structural redundancy.

Guiding principles: P1: "As much as possible for as little as possible."
P2: "Make expensive features optional"

Backup Slides

Automotive Ethernet Applications


Backbone
Ethernet Variant*, 2 Wire Phy.
High Bandwidth Sensors (HBS)
(e.g. Camera, Lidar)
Fast Ethernet, 2 Wire Phy.
HBS & Control
Ethernet Variant*,
2 Wire Phy.
Infotainment
AVB Ethernet, 2 Wire Phy.
Infotainment
AVB Gen 2, 2 Wire Phy.
Diagnosis
Fast Ethernet, 4 Wire Phy.
time

* Enhanced Ethernet or TTEthernet or AVB2