Towards Computer-Aided, Iterative TSN-and Ethernet-based E/E Architecture Design

2020 IEEE Standards Association (IEEE-SA)
Ethernet & IP @ Automotive Technology Day
September 15-16, 2020 | Munich

Oliver CREIGHTON, BMW Group

Nicolas NAVET, Uni. Luxembourg / Cognifyer.ai
Patrick KELLER, Uni. Luxembourg

Jörn MIGGE, RealTime-at-Work (RTaW)

# 1. Challenges in the design of today's E/E architectures at BMW

1 Cultural shift from function/signal-oriented sub-architectures & solutions to a unified and <u>trusted</u> vehicle-wide layered Service-Oriented Architecture
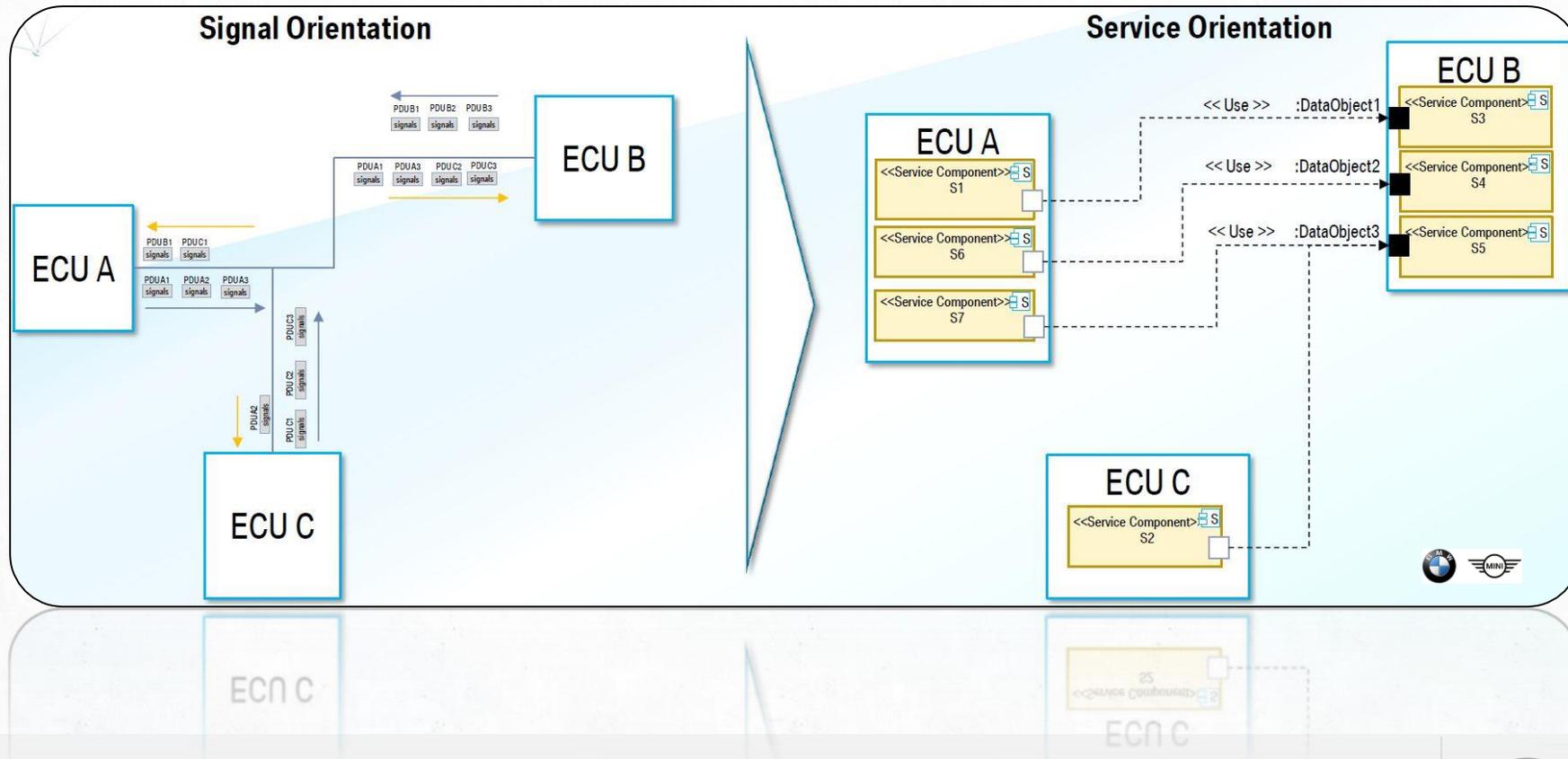
Consequence #1 : two key benefits

1 Clear <u>separation of concerns</u> through layered SOA
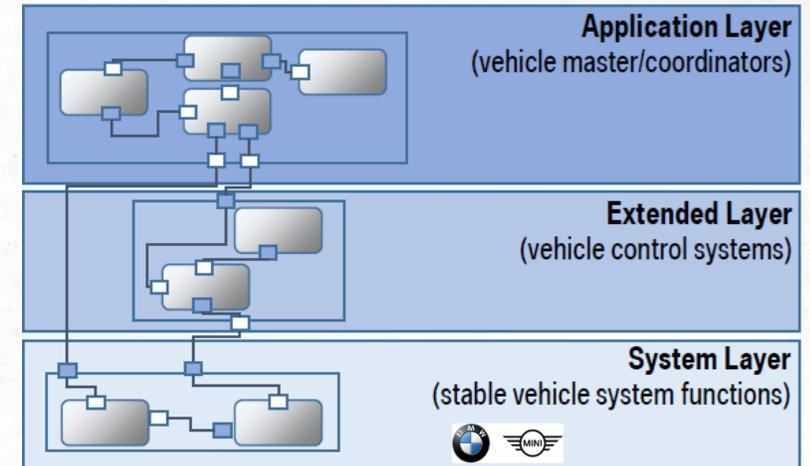
2 Well defined responsibilities between <u>infrastructure providers</u> and <u>consumers</u>

See "Service-oriented architectures as a mindset: Shaping the next EE architecture in a digital age" by Julian BROY (BMW Group) @ Automotive Networks (Hanser, 11/2019) for an in-depth discussion on SOA benefits, implementation & standardization issues.

Consequence #2 : More system knowledge must be encoded in the system itself, such as

1 Bounded latencies / deadlines

2 Bandwidth requirements and degradation options

3 Vehicle-wide runtime configuration (modes, start-up, shut-down), safety-required redundancy, authentication & authorization

Need for **self-aware automotive cyber-physical systems** "able, based on the understanding of their state and environment, to make self-explanatory decisions autonomously at runtime – despite limited resources, complex unforeseeable environmental dynamics, high expectations on their reliability, and substantial levels of risk associated with malfunctioning."

See *"Self-aware Cyber-Physical Systems"* by K. Bellman et al, ACM TECS, 2020/06.

Consequence #3 : Dynamic re-allocatability of resources means "**general purpose**" and "**highly integrated**" **hardware** that can serve multiple roles, possibly as a software-defined, virtualized infrastructure

Highway Pilot, L3 primary channel

BMW's Scalable Autonomous Vehicle Architecture uses for Level 3 & 4:
— Infineon's Aurix 3C and Renesas' 9C R-CAR SoCs
— Intel Denverton 8C and Intel Xeon 24C (level 4 only)

See *"Unveiled: BMW's Scalable AV Architecture" by* Junko Yoshida, EE|Times, 2020/04.
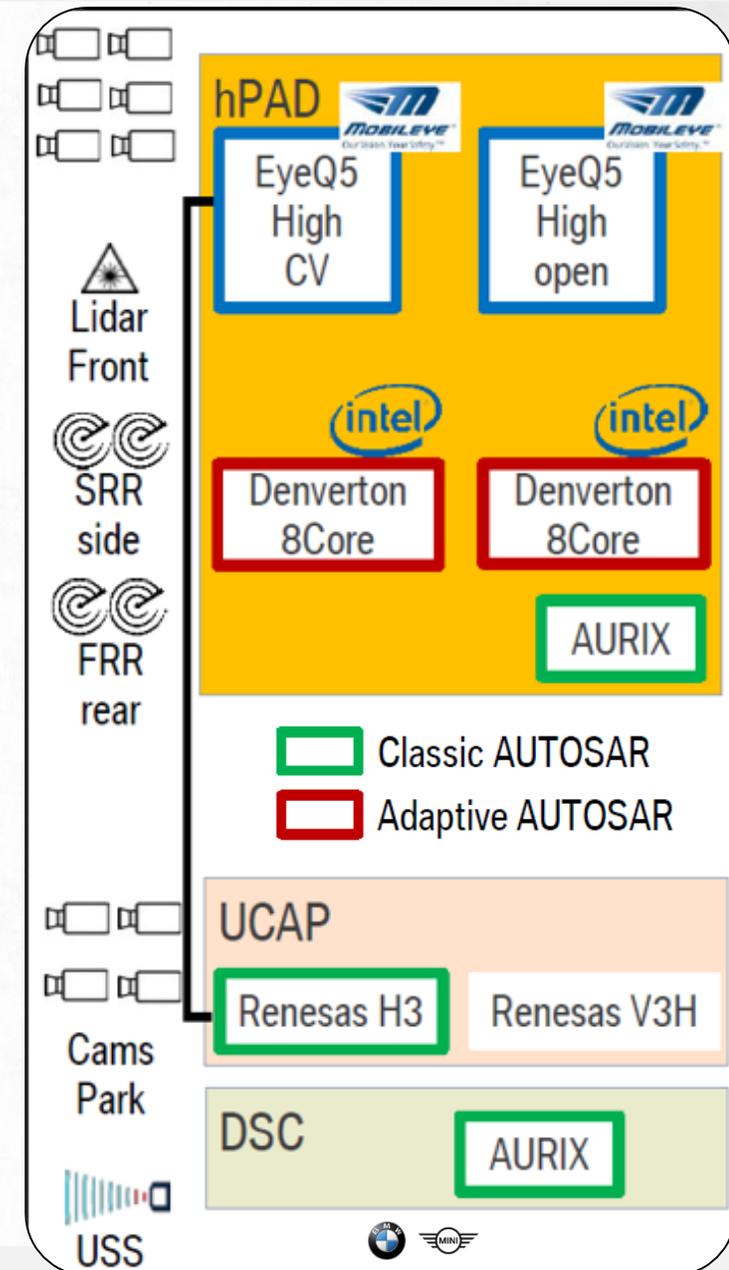
1 Modular privacy and trust: capabilities, roles, and rights must be centrally manageable, across individual vehicle boundaries

2 Modular safety case(s) needed: fault containment regions must be guaranteed by construction

SCALING OF BMW PLATFORM ARCHITECTURE.

L3   L4/5

L1   L2

L2 becomes a fallback for the L3
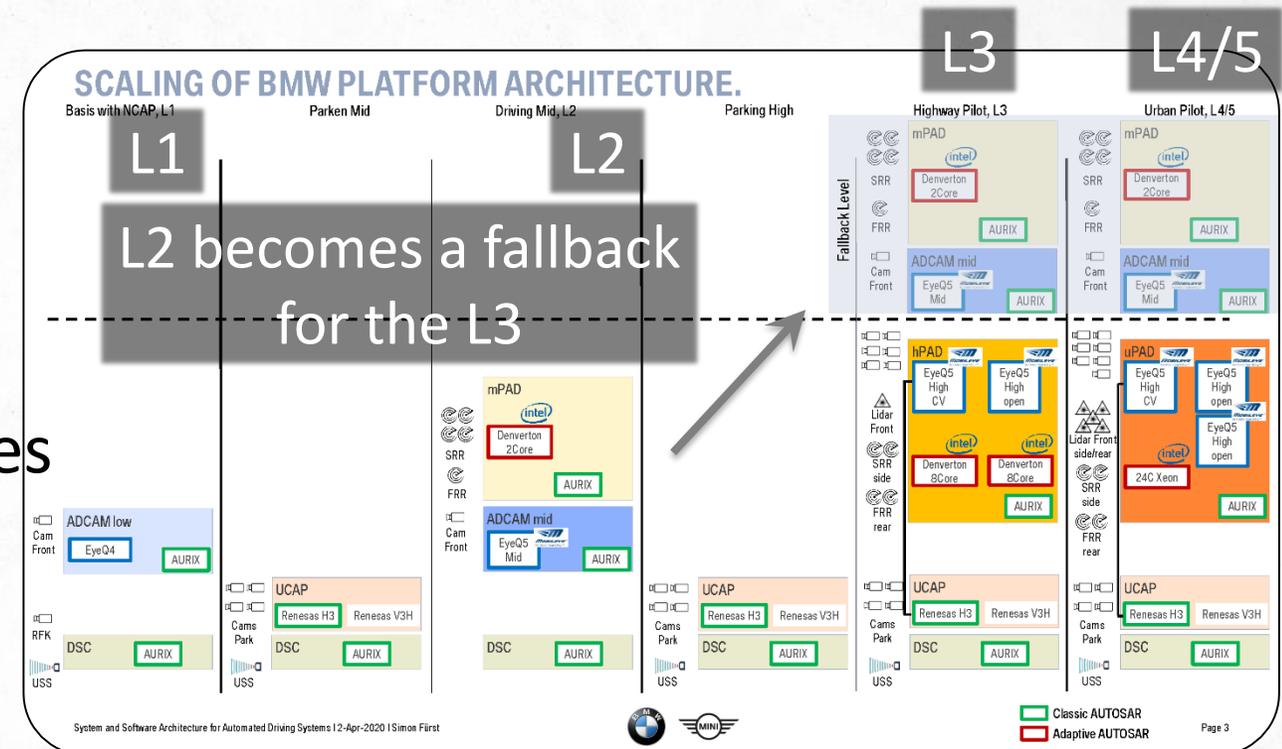
See *"System and Software Architecture for Automated Driving Systems"* *by* Simon Fürst (BMW Group), 2020/04.

③ High efforts & costs for integration & testing!

A Shift from "whole system tests" to continuous deployment & testing - Strong focus on automation needed

B Early-stage validation & verification on virtual platforms is key

C Test coverage must be measured in variability and validated execution paths, not in km driven

Large variety in methods and tools used in design
a way to intelligently combine their benefits is needed, not replacing them by something more complex

Execution Path

Time
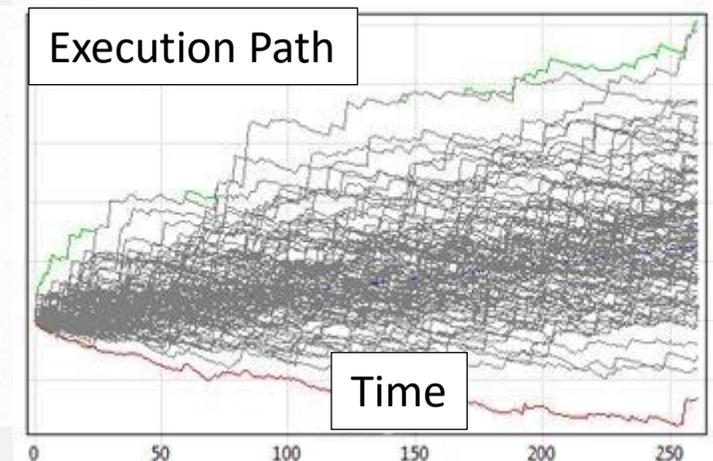
4 Design for SW and HW extensibility

✓ Architectural choices are made early in the design → software functions will be added during vehicle's development & once in customers' hands (eg, OS7 OTA)

2 scenarios of evolutions:

HW+SW update:
e.g, ADAS

Pure SW update

+
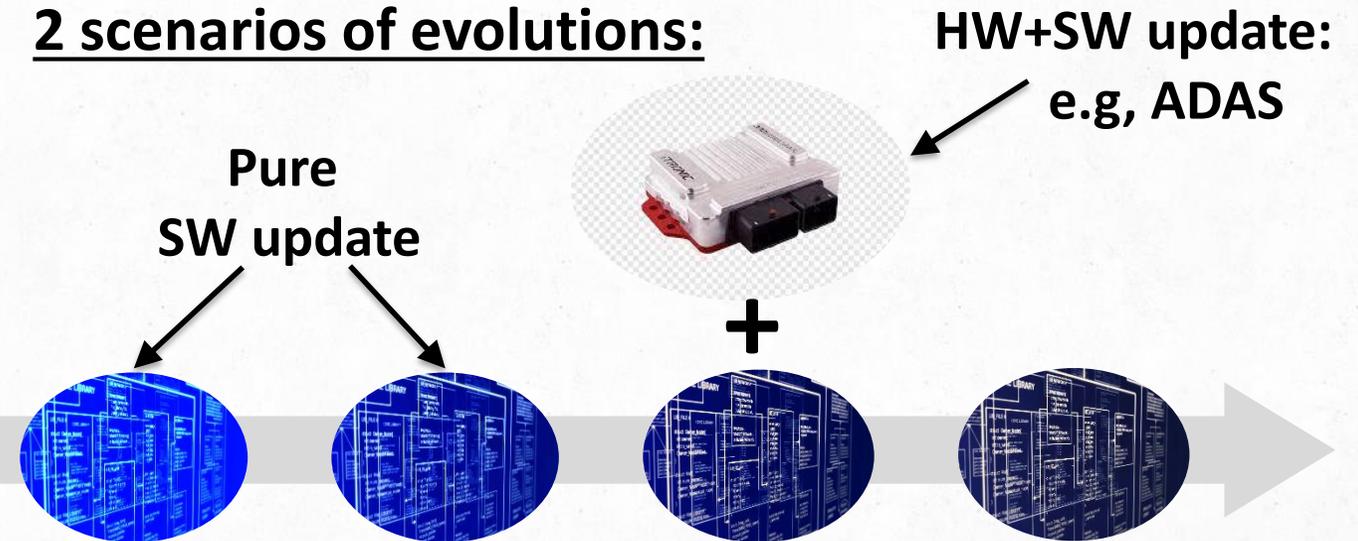
How to design "future-proof" E/E architectures? *i.e.,* make optimized design choices in terms of architecture, technologies (link speeds) & TSN protocol selection (e.g., Qbv? Qbu? CB? ...) ?

Possible solutions offered by algorithmic tools

1 Transition to service orientation → **Big data and AI algorithms for correlating many of the various existing design specifications**

2 - Modular privacy and trust
- Modular safety case(s) needed →

- Transitive trust algorithms for a centralized security model
- Mathematical models of fault probabilities within fault containment regions and their resulting "module error rates"

3 High efforts for integration & testing ↓

- **Design complexity metrics and test coverage calculators**
- **Simulation of "full-stack" system behavior with varying degrees of precision, potentially plugging in real components for "software-in-the-loop" or "hardware-in-the-loop" testcases, in order to build trust in the overall OA.**

Highest challenge

Focus on challenge ④
Use-cases for algorithmic tools: COTS & R&D
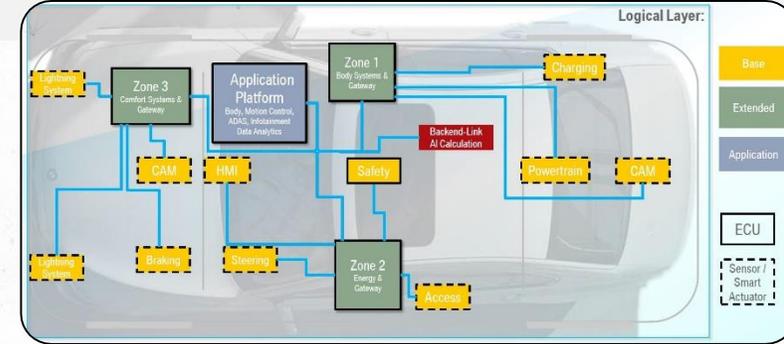
A Quantify network extensibility wrt TSN technological options
Topology Stress Test ®
IEEE SA Ethernet TechDays 2019

B Identify bottlenecks in E/E architecture and remove them
Topology Optimizer ® - AEC2020

C Assess and optimize communication reliability
AEC2020 + IEEE SA Ethernet TechDays 2020 (NXP, UL, Cognifyer)

D Cost-optimize by reducing link speeds & # of ECUs
Topology Optimizer ® - AEC2020

Selecting cost-optimized TSN scheduling solutions

E **E/E architecture synthesis**   **Our focus next**

Total capacity
Bottlenecks
Reliability
Cost-optimize
Synthesis

Candidate solution

Solution Refinement

Solution Creation

Enabling technologies for E/E Architecture Design Automation

create configure evaluate

- **Model-Based System Engineering**: comprehensive system description over entire dev. process **1**
- **Configuration algorithms** that automate **all** parameters setting & optimize resource usage **2**
- **Fast performance evaluation tools:** both simulation & worst-case evaluation **3**
- **"Virtual Design Assistants"** **explor. the design space:** cost/capacity/.. optimisation, architecture synthesis **4**
- **AI for scalability** : predicting solution feasibility and technology-independent configuration algorithms **5**

"Centaur Era": teaming design engineers with machine by "marrying human experience and creativity with computer's brute force ability"

2. Illustration on a prototype TSN-based zonal SOA architecture – evolution scenario considered: addition of new services by software update
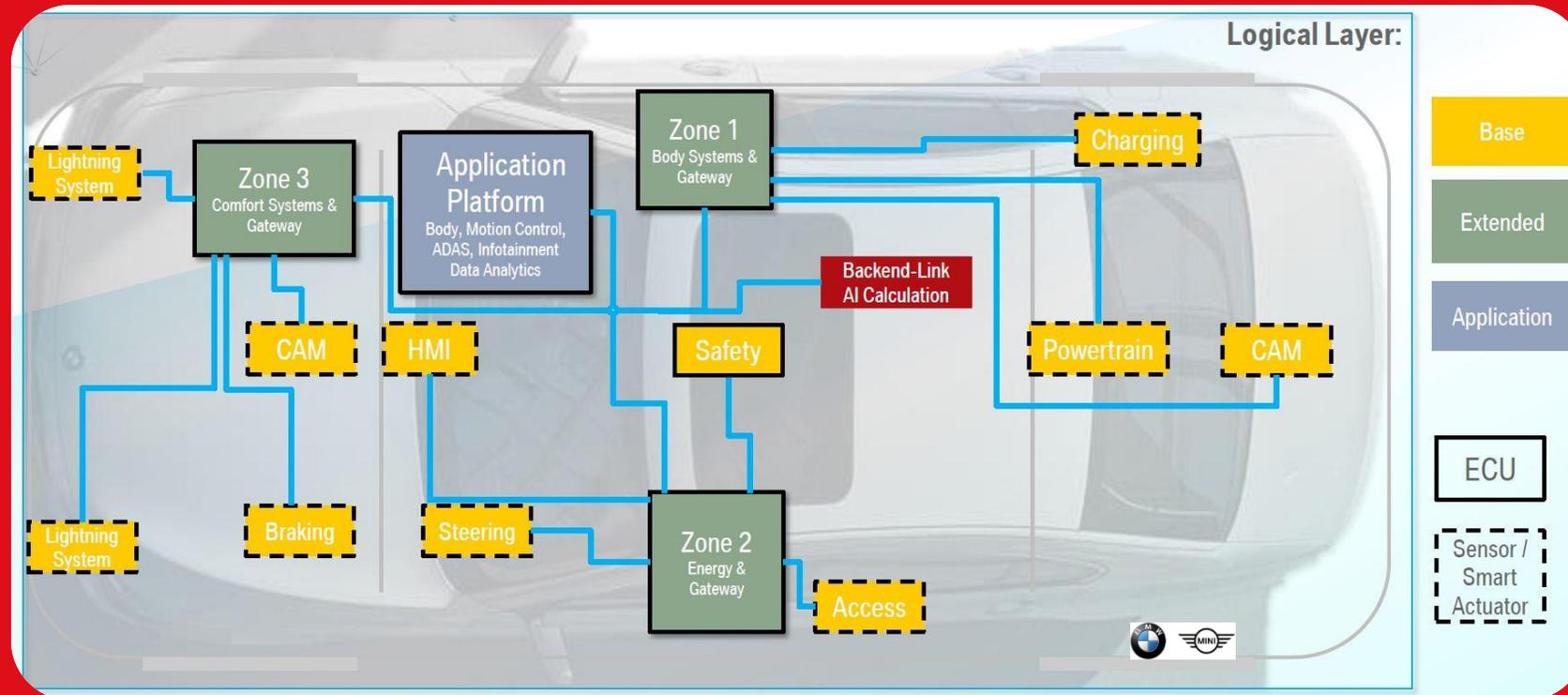
See *"Service-oriented architectures as a mindset: Shaping the next EE architecture in a digital age"* by Julian BROY (BMW Group), Automotive Networks, Hanser, 11/2019.

Model of the core TSN Network

Redundant Central Computer ("application platform"): body, motion, data analytics, ADAS

3 Zone Controllers

17 ECUs incl. HMI, powertrain, charging, lightning systems, camera, AI backend calculator, access, etc
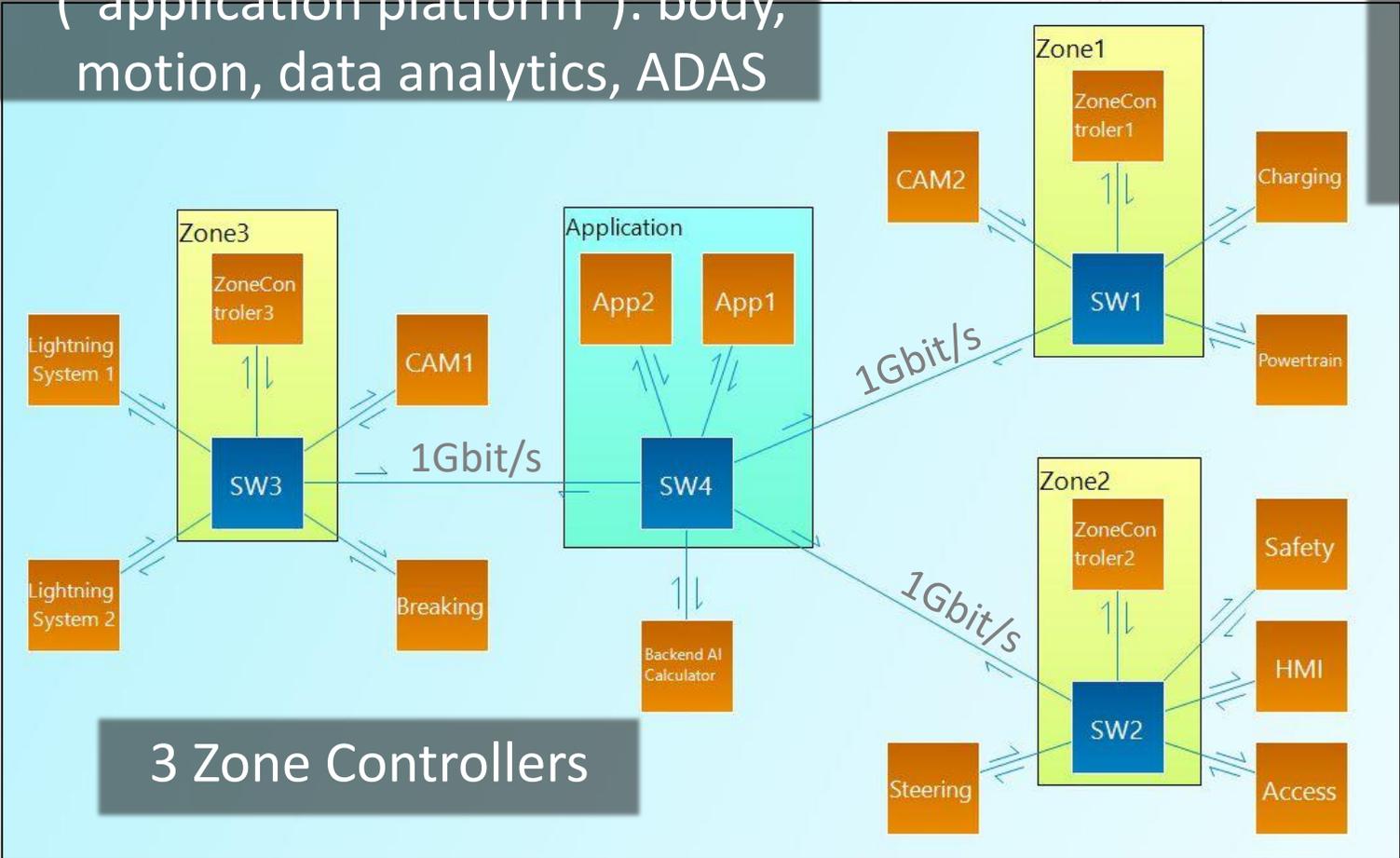
# Nodes	17
# Switches	4
Link speed	1Gbit/s: inter-switch links 100Mbit/s: all other links
# TFTP streams	6 → 320Kbit/s overall
Standard automotive traffic	Command & Control (≈30% of the streams), Audio (5%), Video incl. ADAS (5%), Misc. Services (60%)

[RTaW-Pegase screenshot]

Breaking down the design problem into smaller problems answered using algorithmic tools

| Overload Analysis	Total Network Capacity for each TSN solutions	Cost / Extensibilty Analysis	Cost & Capacity Optimization	Architecture Synthesis based on a Core Topology
• **Determine upper bound on architecture extensibility** • Independent of TSN protocols • Fast, coarse-grained analysis	• **Assess the relative ability of TSN scheduling solutions to support additional traffic** • Allows estimate architecture lifetime • Precise, compute-intensive analysis	• **Consider the "cost" of the different TSN scheduling solutions** • Cost can be a function of dev. time price, risk, …	• **Remove performance bottlenecks** trough local improvements • Reduce link speeds • Reduce # of ECU by relocating functions	• **Extend core topology by adding HW components** (individual components or "patterns") • In a first step, benchmark manually-created candidate architectures

Probability that the network is overloaded when new services are deployed

Overloaded network = the load of one link or more is higher than 100% → no TSN policy can meet the timing constraints

% of overloaded network configurations

- 10% of overloaded networks when adding 90 services, overload % then increases steeply

- This suggests that, whatever the TSN policy - under our traffic assumptions - this architecture is suited to support at most 60-80 additional services

[RTaW-Pegase screenshot]

# of additional services from 25 to 175

Network extensibility for ≠ TSN QoS options

Using CBS + a top priority express class, 55 new services can be added (75% assurance level) – similar results with CBS + TAS at top priority level
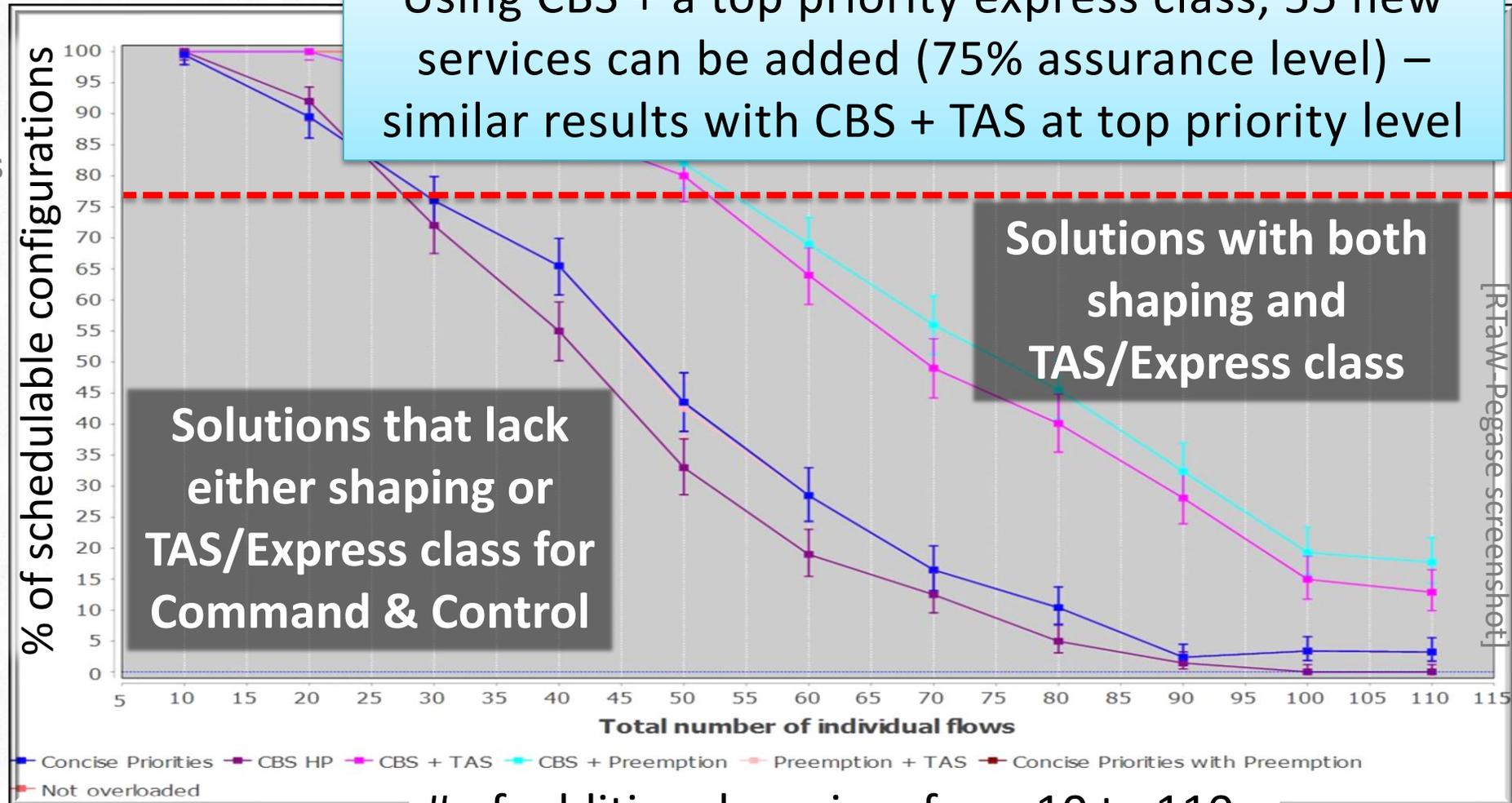
User-defined stream priorities

☐ User priorities ☐ CBS
☑ CBS/HP
☐ PreShaping

☐ Preemption ☐ TAS
☐ Preemption+PreShaping ☑ TAS+CBS
☑ Preemption+CBS ☐ TAS+PreShaping
☑ TAS+Preemption

Stream priorities optimized

☑ Concise priorities

☑ Concise priorities+Preemption

Solutions with both shaping and TAS/Express class

Solutions that lack either shaping or TAS/Express class for Command & Control

% of schedulable configurations

Total number of individual flows

[RTaW-Pegase screenshot]

Concise Priorities · CBS HP · CBS + TAS · CBS + Preemption · Preemption + TAS · Concise Priorities with Preemption · Not overloaded

# of additional services from 10 to 110

Cost /
Extensibilty
Analysis

Adding cost into the equation

✓ Cost can be any quantity, expressed in relative or absolute values, possibly calculated with a user-defined cost function f(price, time, risk, weight, …)

Example of a basic cost model

Name*	Cost Model 2	Currency	
EcuBase		SwitchBase	10.00 Unit
EcuPort8021Q	5.00 Unit	SwitchPort8021Q	1.00 Unit
Link100MBits	2.00 Unit	CBS	3.00 Unit
Link1GBits	3.00 Unit	TAS	8.00 Unit
		Preemption	5.00 Unit
ComConfig	2 ComConfig: Fixed Traffic	Costs :	180.00 Unit

Cost on a per port basis for TSN protocols

A cost model is applied to a candidate architecture

[RTaW-Pegase screenshot]

Cost / extensibility trade-offs

Cost model applied

Threshold : 80

Costs model : Cost Model 1

Cost vs Extensibility
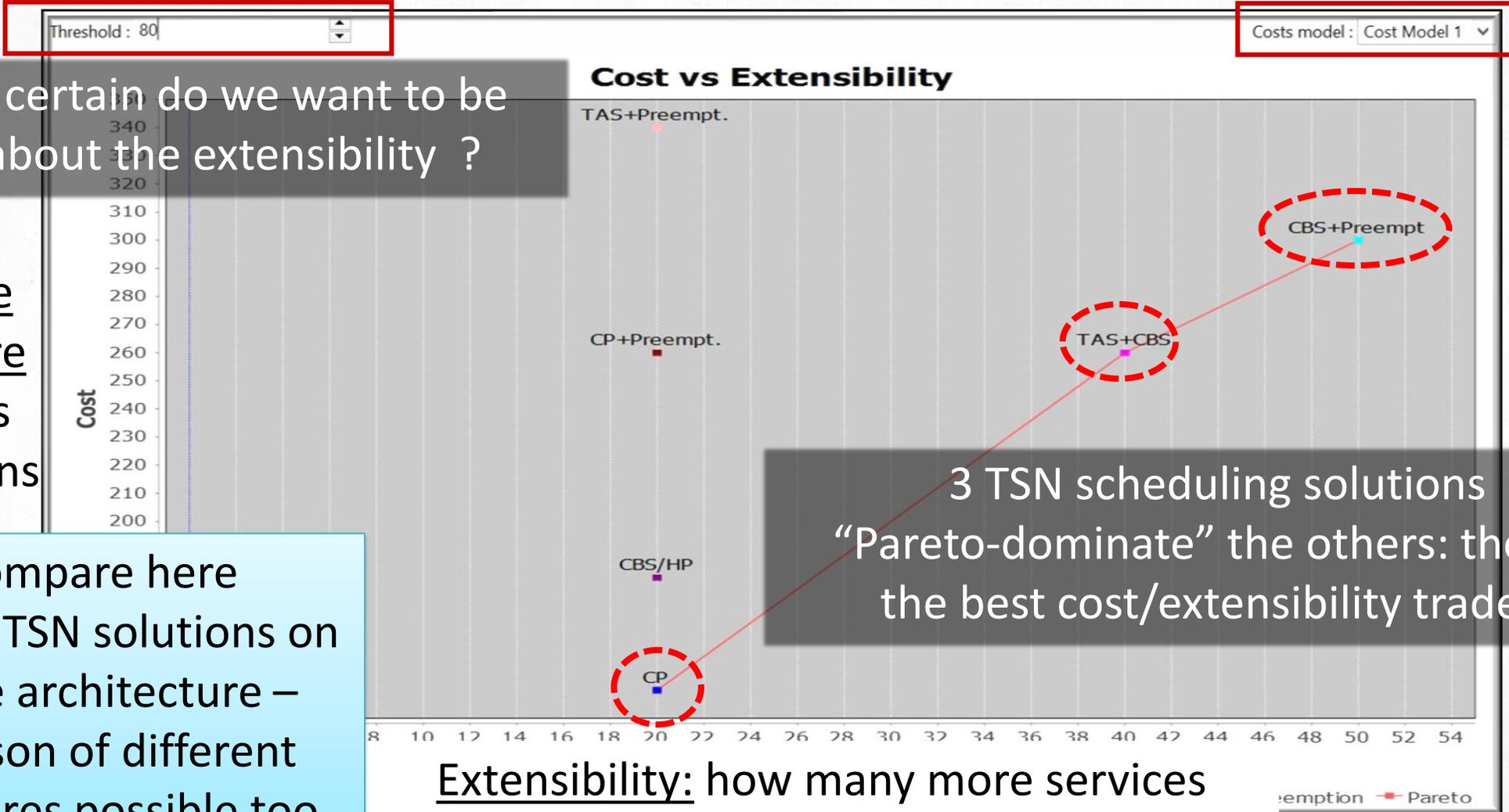
How certain do we want to be about the extensibility ?

Cost of the architecture for various TSN solutions

TAS+Preempt.

CBS+Preempt

CP+Preempt.

TAS+CBS

CBS/HP

CP

3 TSN scheduling solutions "Pareto-dominate" the others: they offer the best cost/extensibility trade-offs

We compare here competing TSN solutions on the same architecture – comparison of different architectures possible too

Cost

340
320
310
300
290
280
270
260
250
240
230
220
210
200

8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54

eemption — Pareto

Extensibility: how many more services can be "safely" added ?

©2020 - BMW - RTaW - UL - Cognifyer

19

Considering CPU requirements in addition to communication requirements

✓ <u>Assumptions here:</u> each service requires a CPU time proportional the # of flows it processes – all processors have equivalent CPU power

✓ Focus on the best performing TSN solutions: Express at top priority and 2 CBS classes below


% of schedulable configurations

# of flows from 10 to 110

- Without CPU requirements
- Considering CPU requirements
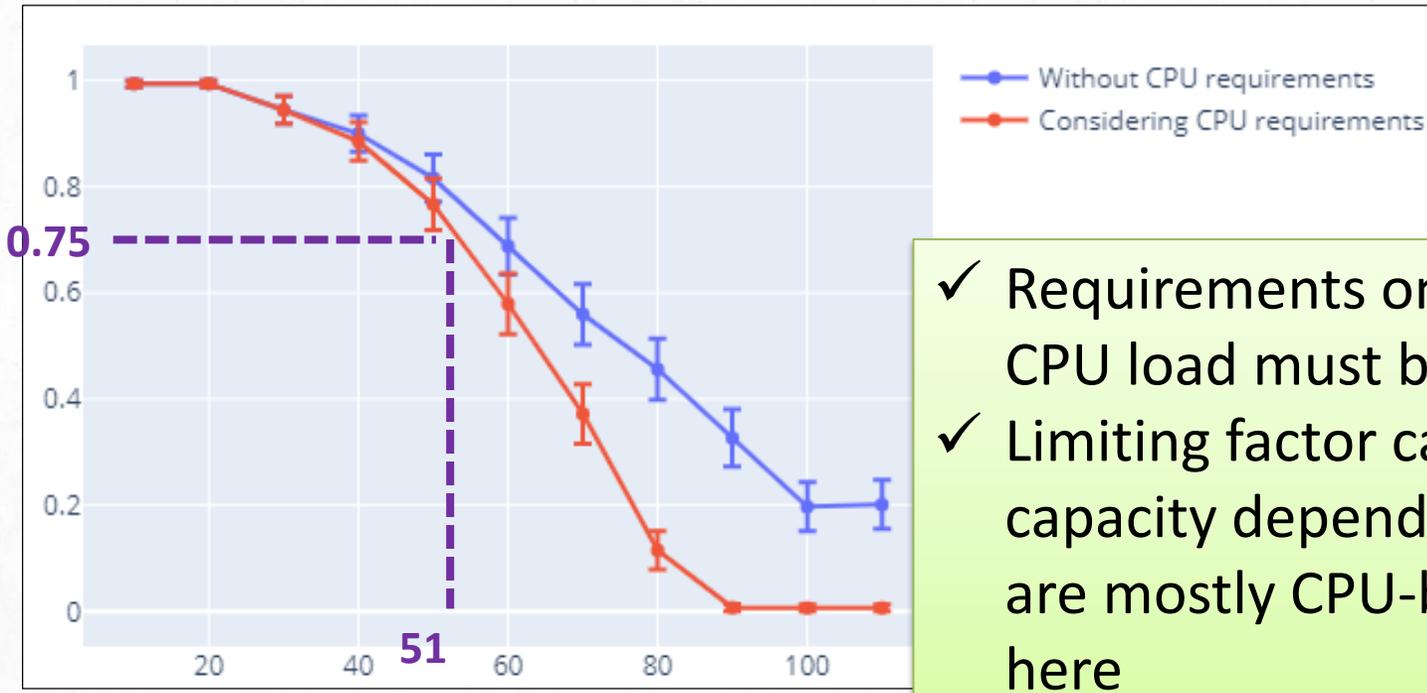
✓ Requirements on both communication and CPU load must be met
✓ Limiting factor can be network or CPU capacity depending on whether services are mostly CPU-bound or I/O-bound like here

Architecture synthesis: extending the core topology

The core topology
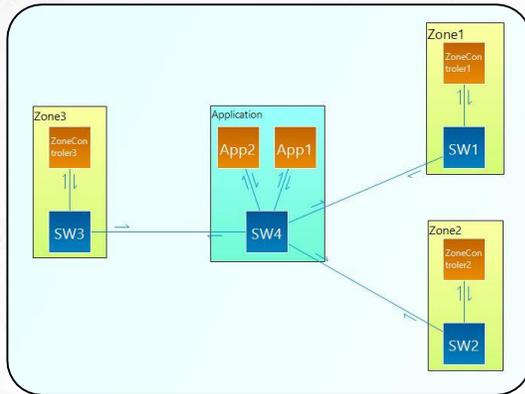
The evolution scenario

- adding SW or HW+SW
- assumptions on the services added (CPU and comm. requirements)
- **HW components that can be added**
- ...

Core Topology | **Evolution Scenario**

Designer inputs: constraints & goals

Topological constraints | **Security & reliability**

Topological constraints

- connection lengths,
- physical location (e.g. vs power & sensors)
- ECU dimension restricting switch sizes, number of pins, power consumption, ...
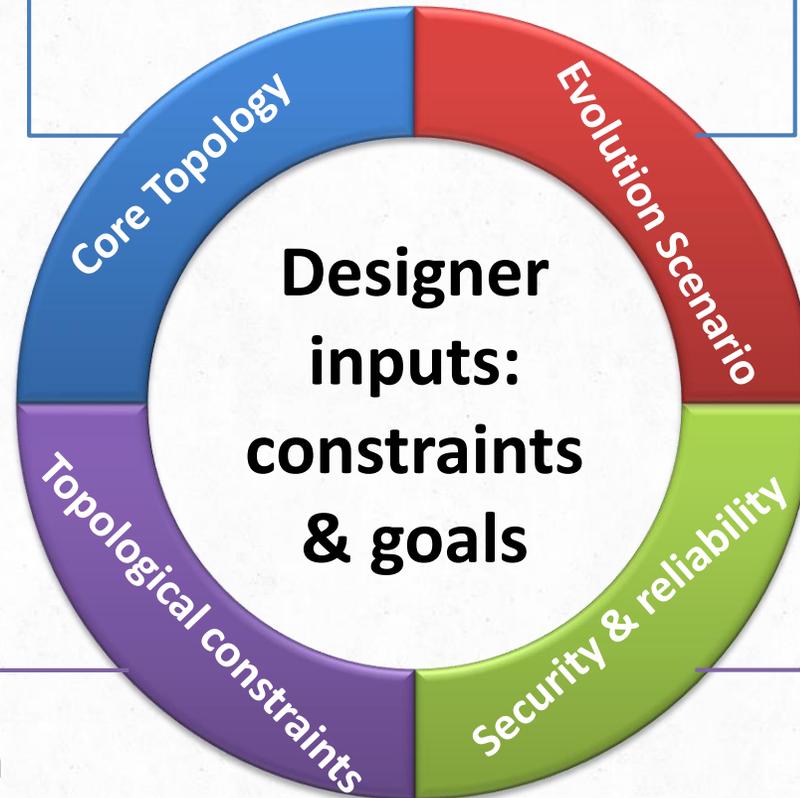
Security and reliability

- stream segregation
- proxy ECUs
- load limits for packet inspection
- multiple paths for reliability
- ...

Extending a topology: HW components that can be added

ECUs / Processors / SoCs

- Computing power
- Reliability & security

Network interface + link ("dual homing")

- Load balancing
- Reliability & Security

Switches

- Additional bandwidth
- Reduce cable lengths e.g. with daisy chains
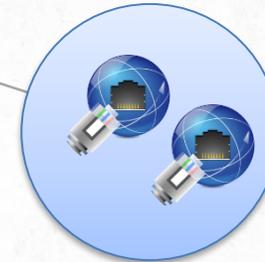
Link between switches

Additional bandwidth e.g. on backbone

ECUs with internal switch

- Space & cost optimization
- Re-use in next generation
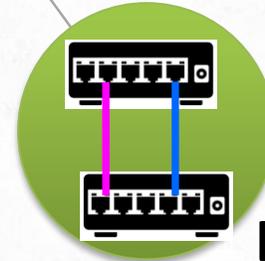
Catalogue of cost-effective "extension patterns" comprised of several HW components
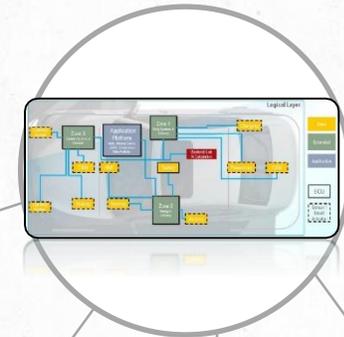
# Illustration: computer-generated architectures based on a core topology
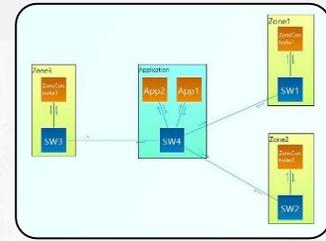
✓ <u>Heuristic applied here:</u> additional ECUs close to the "hot-spots", i.e. ECUs subject to max. variability pressure in terms of # future services added

✓ Parameter specifies trade-off between topology balance / hot-spots coverage

- ⟶ **+**
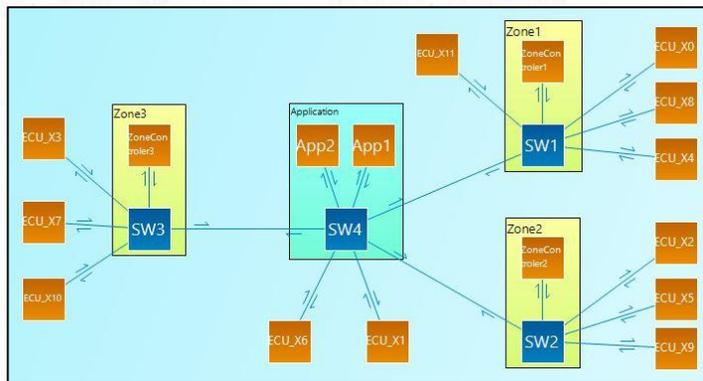
Candidate Sol. A
(3 or 4 ECUs per zone)
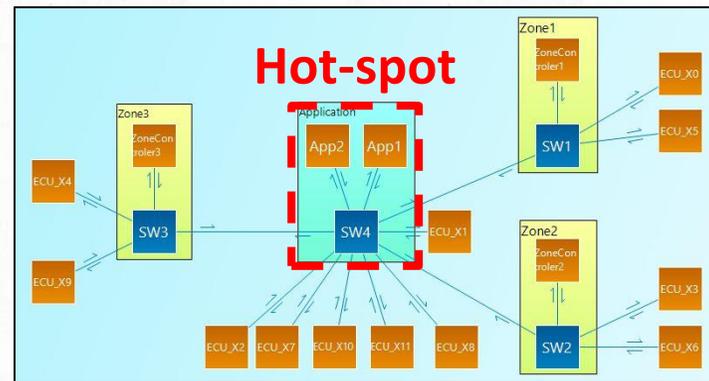
Candidate Sol. B
(2 ECUs per zone)
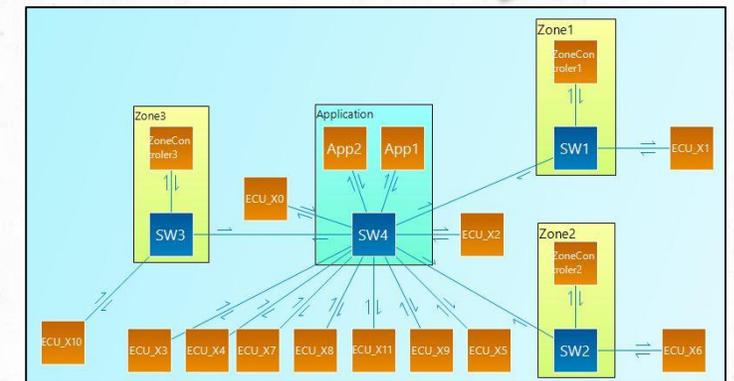
Candidate Sol. C
(1 ECU per zone)

Daisy-chains & bus topology using 10BASE-T1S, and different types of CPUs open up many more design options that can be systematically explored
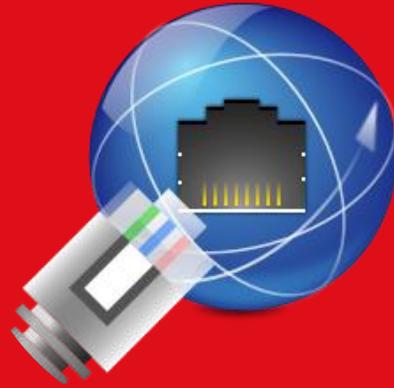
Conclusion and a look forward

Ignore, challenge, or embrace it ?

The state of technology enables computer-aided E/E architecture design, incl. evidence-supported TSN architectural & technological choices

Complexity, time & cost effectiveness, extensibility requirements are key drivers

Is it just a convenient tool or will it ultimately reshapes the innovation process & the organization of R&D ?

How such a novel approach fits into the existing design flow at BMW? Which timeline, limitations and risks, what to expect and not expect ?

THANK YOU FOR YOUR ATTENTION.

THE NEXT
100 YEARS