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

UNIVERSITÉ DU LUXEMBOURG

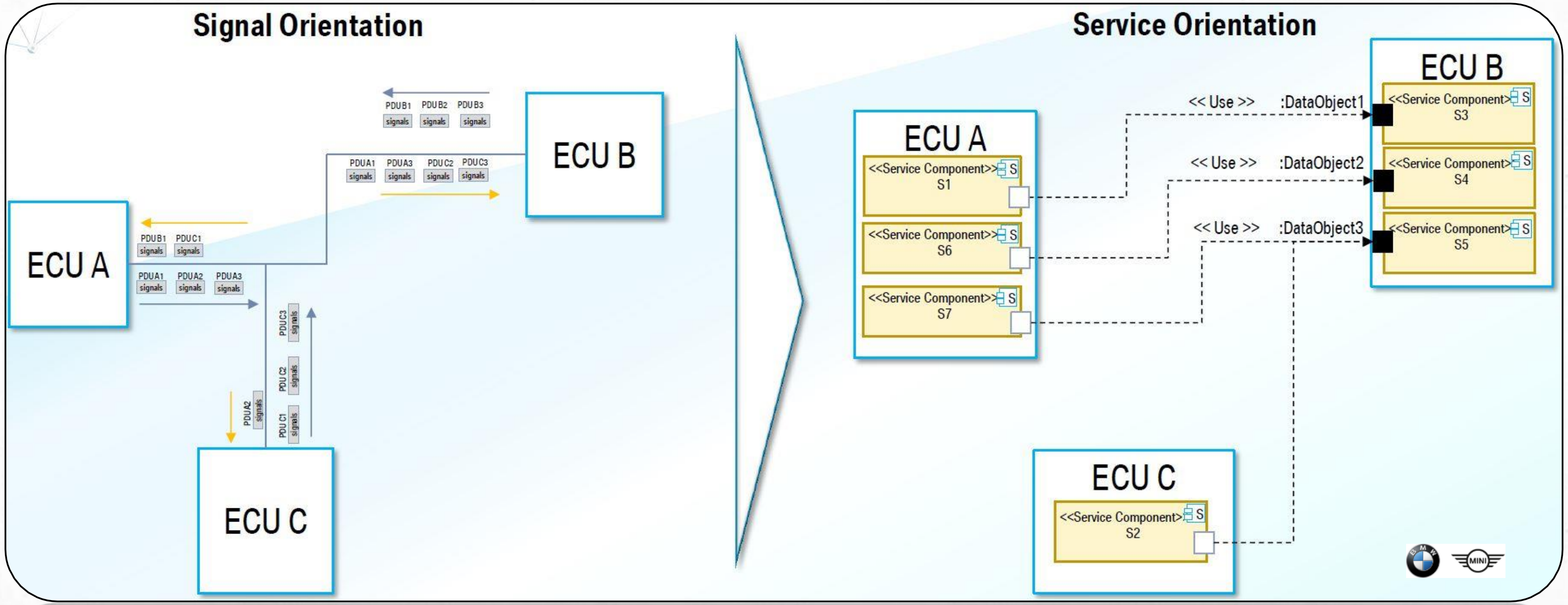
Nicolas NAVET, Uni. Luxembourg / Cognifyer.ai
Patrick KELLER, Uni. Luxembourg

RTaW RealTime-at-Work

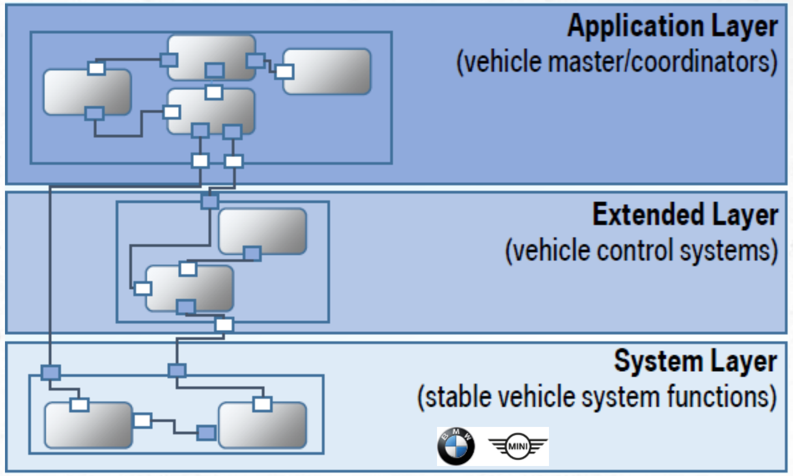
Jörn MIGGE, RealTime-at-Work (RTaW)

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 trusted 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 **"<u>general purpose</u>" and "<u>highly integrated</u>" hardware** that can serve multiple roles, possibly as a software-defined, virtualized infrastructure

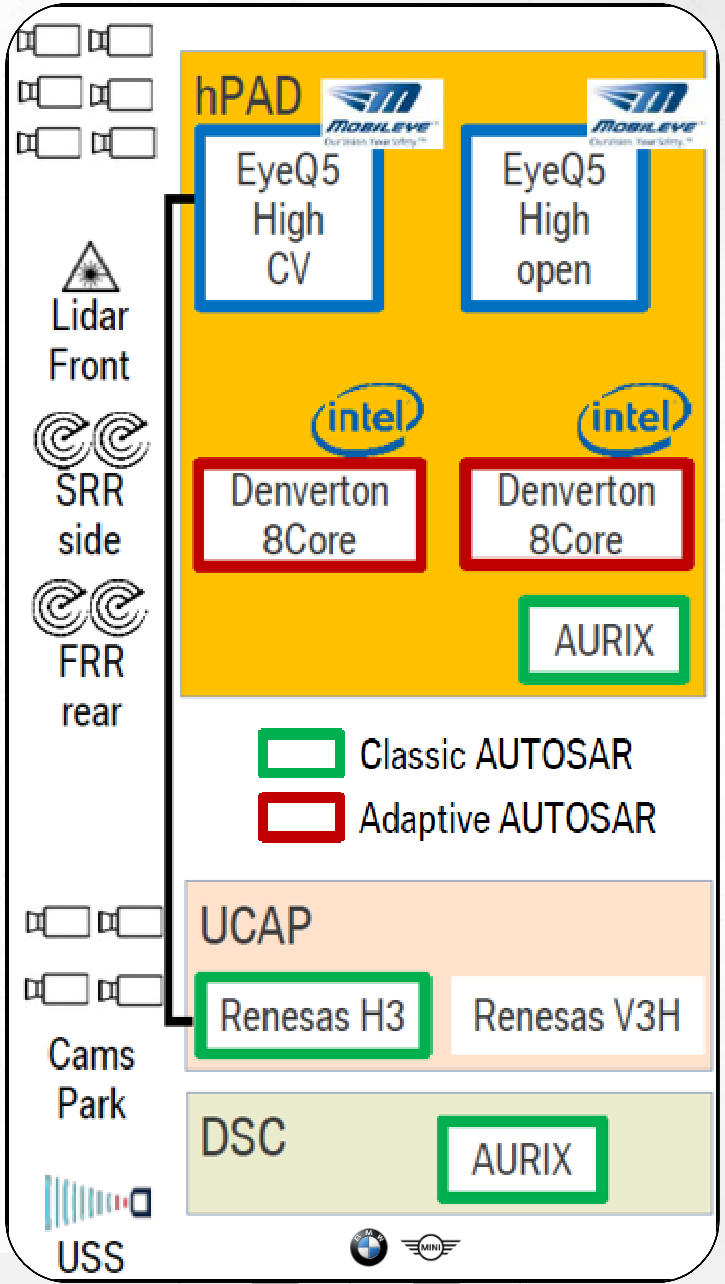
Highway Pilot, L3 primary channel

<u>BMW's Scalable Autonomous Vehicle Architecture uses for Level 3 & 4:</u>

— Infineon's Aurix 3C and Renesas' 9C R-CAR SoCs

— Intel Denverton 8C and Intel Xeon 24C (level 4 only)

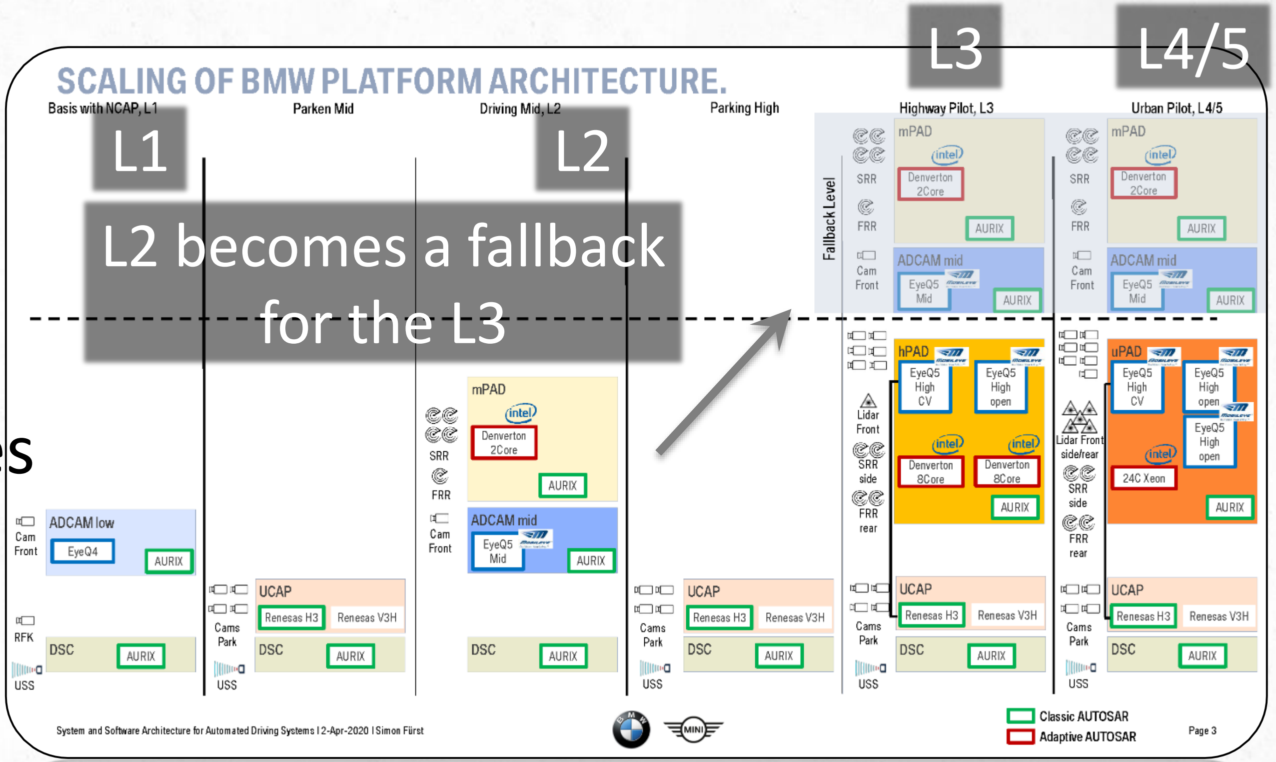
See *"Unveiled: BMW's Scalable AV Architecture"* by Junko Yoshida, EE|Times, 2020/04.



2 Scalability and re-usability of SW and HW through **modularity**

1 <u>Modular privacy and trust</u>:
capabilities, roles, and rights must be centrally manageable, across individual vehicle boundaries

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



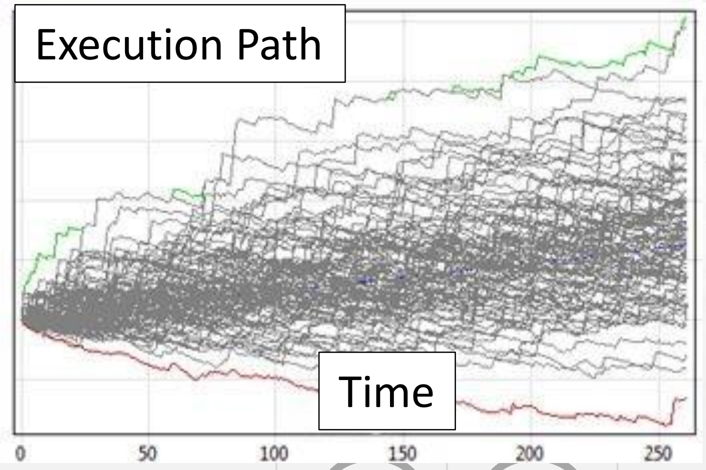
See "System and Software Architecture for Automated Driving Systems" by Simon Fürst (BMW Group), 2020/04.

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



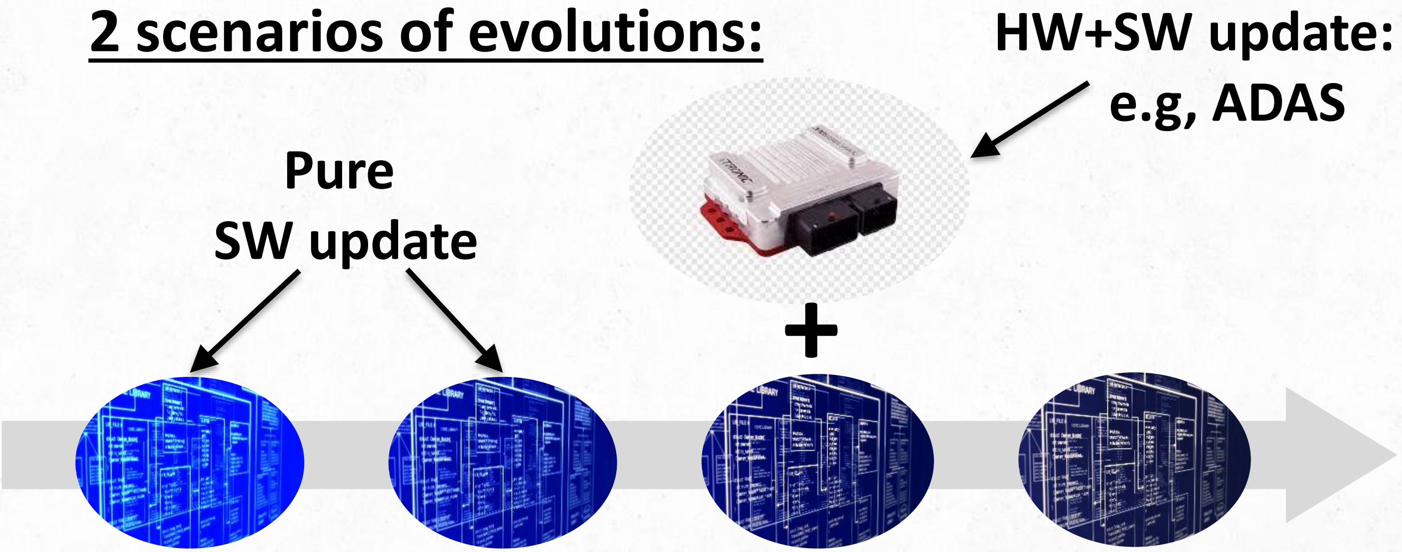
<u>Large variety in methods and tools used in design</u>

a way to intelligently combine their benefits is needed, not replacing them by something more complex

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)

<u>2 scenarios of evolutions:</u>



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

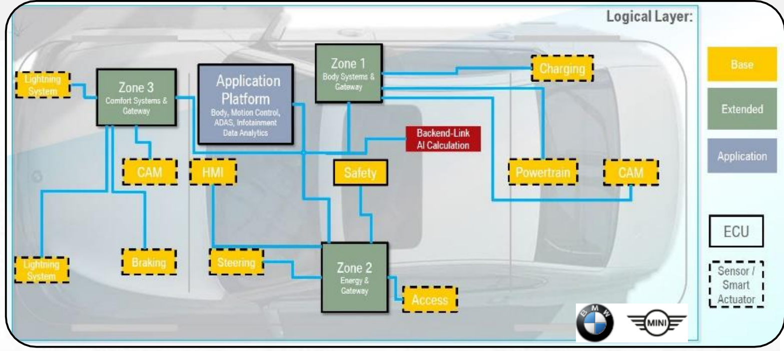
1	Transition to service orientation	**Big data and AI algorithms for correlating many of the various existing design specifications**
2	- Modular privacy and trust - Modular safety case(s) needed	**- Transitive trust algorithms for a centralized security model** **- Mathematical models of fault probabilities 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



Total capacity — A — Quantify network extensibility wrt TSN technological options
Topology Stress Test ®
IEEE SA Ethernet TechDays 2019

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

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

Cost-optimize — D — Cost-optimize by reducing link speeds & # of ECUs
Topology Optimizer ® - AEC2020
Selecting cost-efficient TSN scheduling solutions

Synthesis — E — **E/E architecture synthesis**

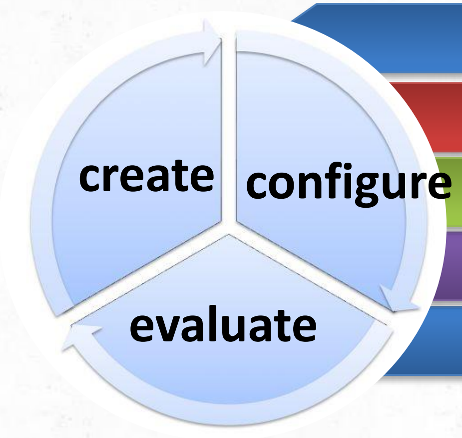
Our focus next

Candidate solution

Solution Refinement

Solution Creation

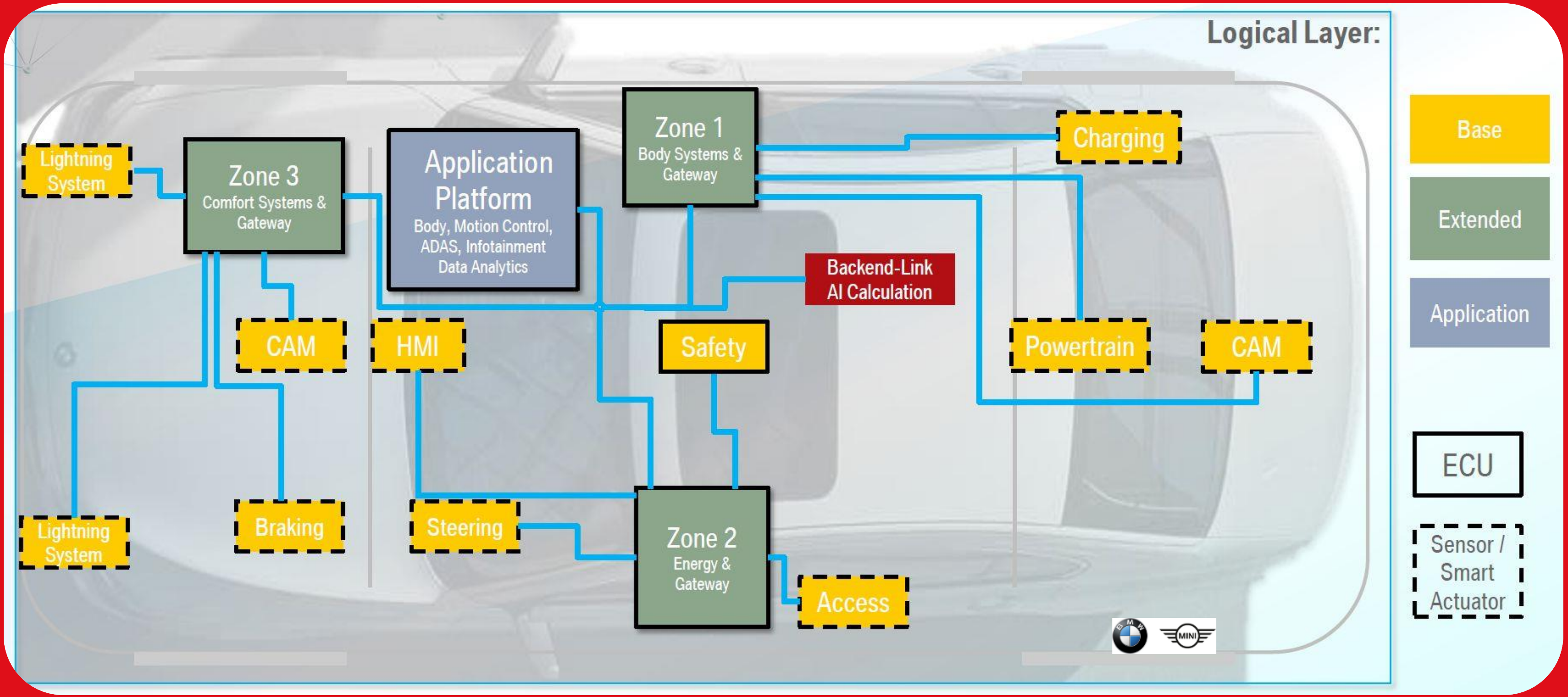
Enabling technologies for E/E Architecture Design Automation



- **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

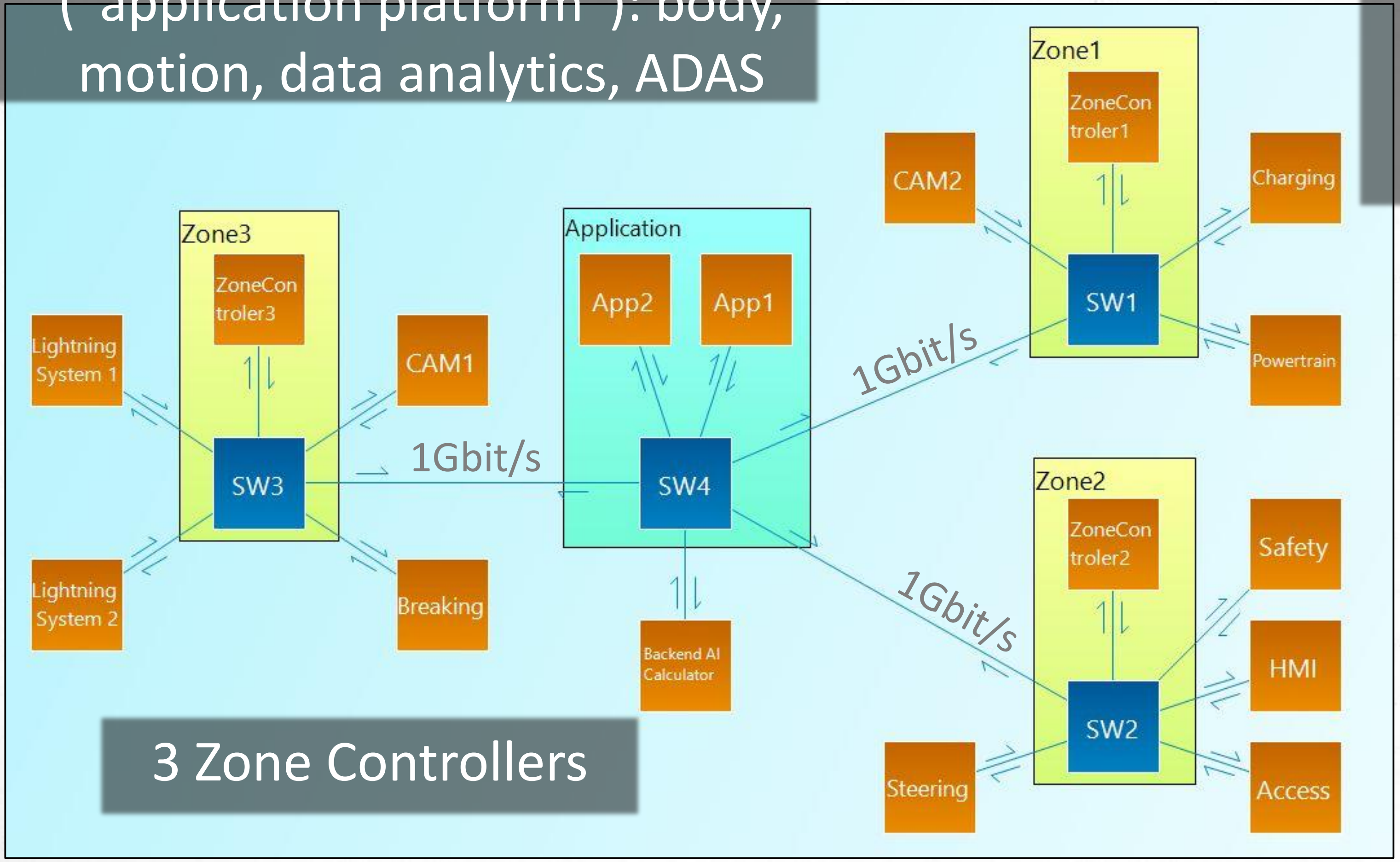


See "Service-oriented architectures as a mindset: Shaping the next EE architecture in a digital age" by Julian BROY (BMW Group), Automotive Networks, Hanser, 11/2019.

Model of the core TSN Network

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

17 ECUs incl. HMI, powertrain, charging, lightning systems, camera, AI backend calculator, access, etc

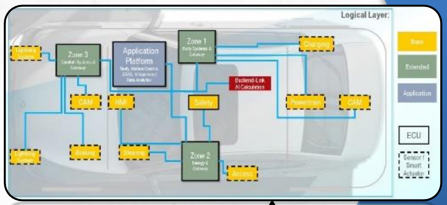


3 Zone Controllers

[RTaW-Pegase screenshot]

# 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%)

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



Overload Analysis

- **Determine upper bound on architecture extensibility**
- Independent of TSN protocols
- Fast, coarse-grained analysis

Next

Total Network Capacity for each TSN solutions

- **Assess the relative ability of TSN scheduling solutions to support additional traffic**
- Allows estimate architecture lifetime
- Precise, compute-intensive analysis

[TechDays 2019]

Cost / Extensibility Analysis

- **Consider the “cost” of the different TSN scheduling solutions**
- Cost can be a function of dev. time price, risk, …

Next

Cost reduction & Capacity Optimization

- **Remove performance bottlenecks** trough local improvements
- Reduce link speeds
- Reduce # of ECU by relocating functions

[AEC 2020]

Architecture Synthesis based on a Core Topology

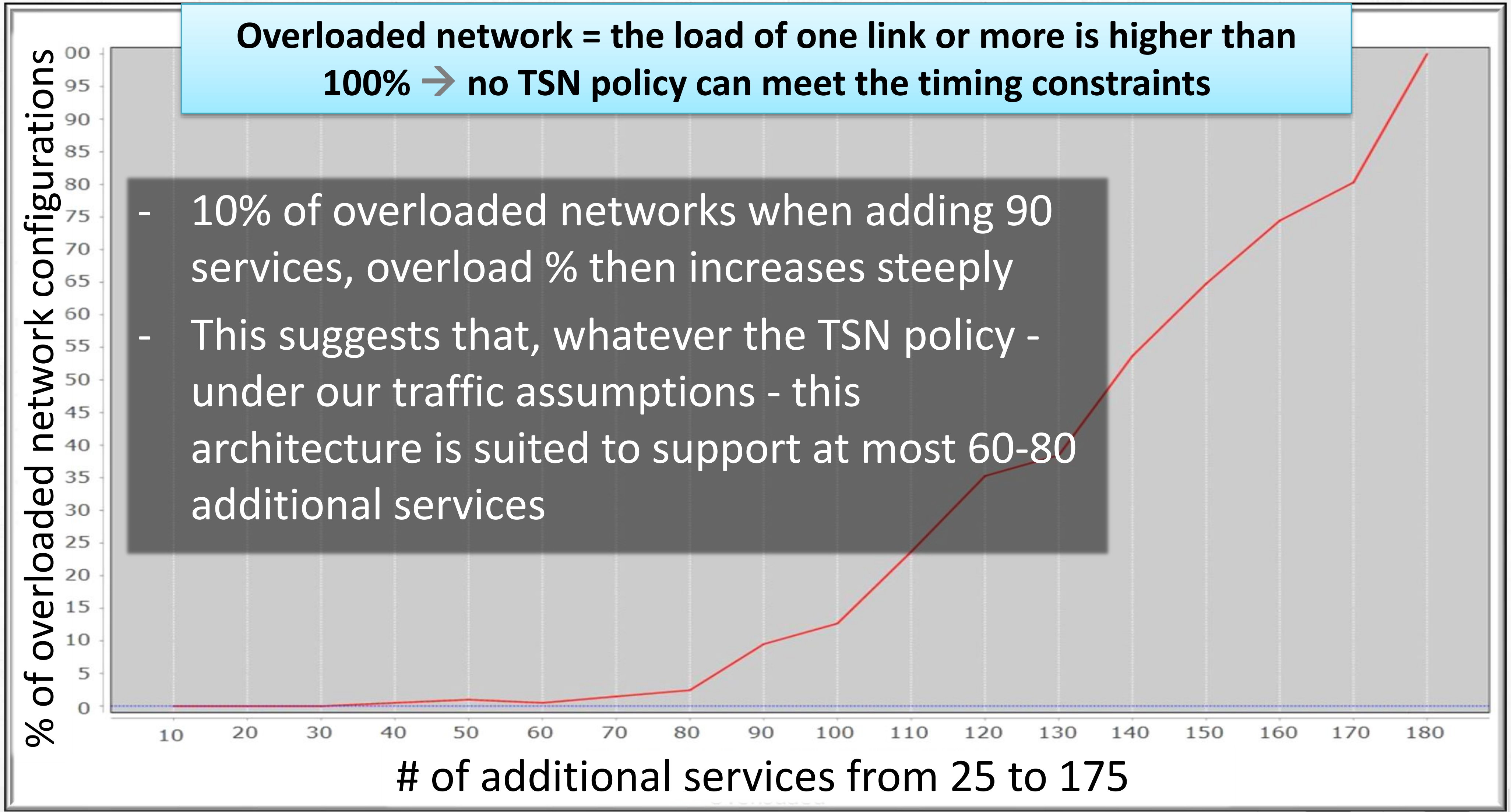
- **Extend core topology by adding HW components** (individual components or “patterns”)
- Benchmark manually-created candidate architectures

Next

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

- 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]

Total Network Capacity

Network extensibility for ≠ TSN QoS options

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

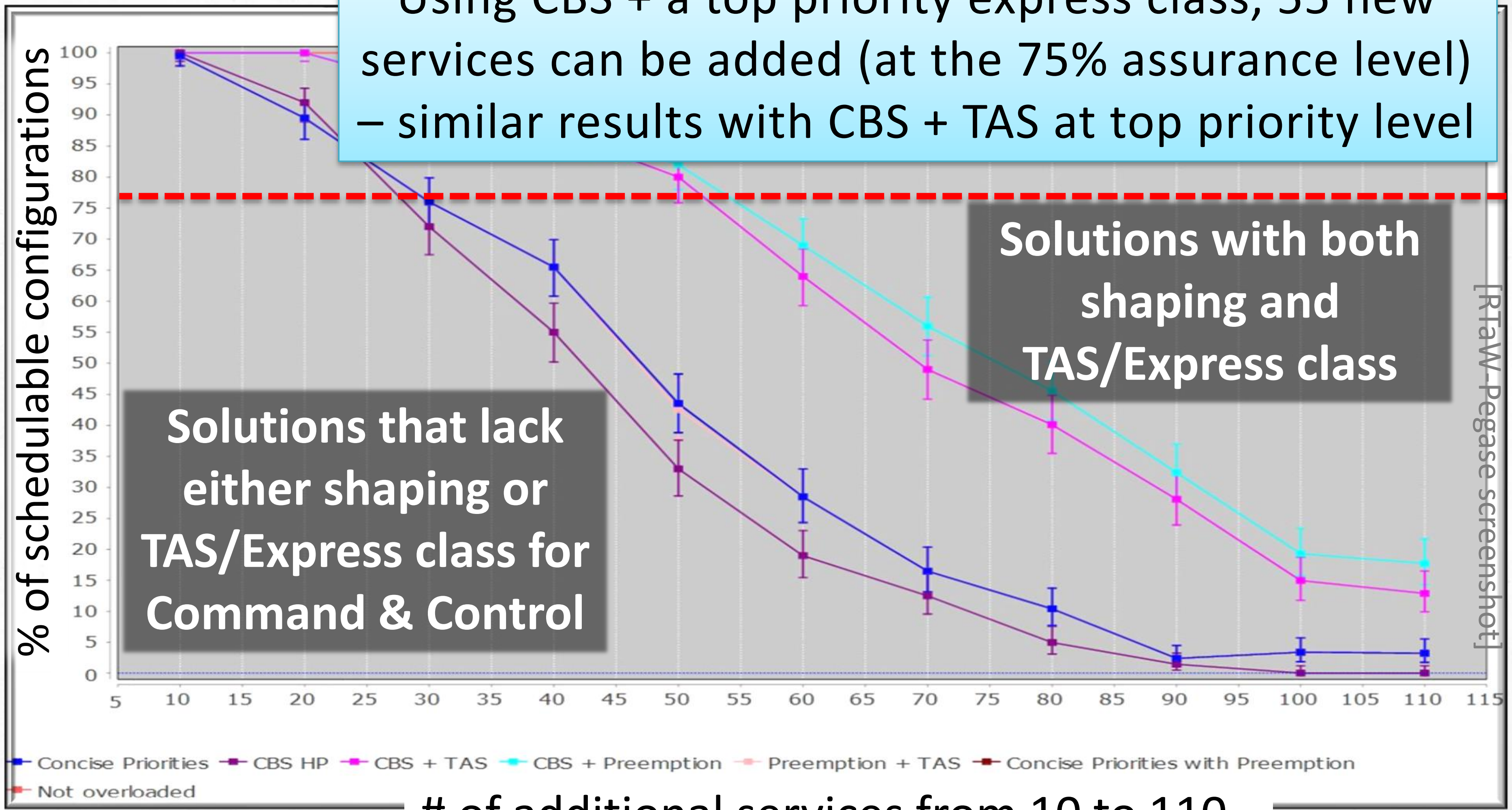
User-defined stream priorities

- [ ] User priorities
- [ ] CBS
- [x] CBS/HP
- [ ] PreShaping

- [ ] Preemption
- [ ] Preemption+PreShaping
- [x] Preemption+CBS
- [ ] TAS
- [x] TAS+CBS
- [ ] TAS+PreShaping
- [x] TAS+Preemption

Stream priorities optimized

- [x] Concise priorities
- [x] Concise priorities+Preemption



Solutions with both shaping and TAS/Express class

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

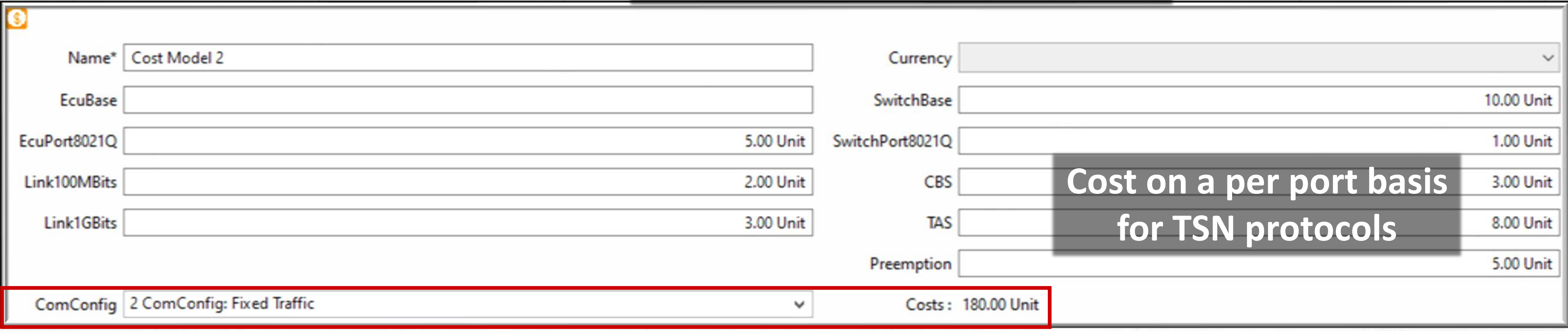
# of additional services from 10 to 110

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 simple cost model



A cost model is applied to a candidate architecture

[RTaW-Pegase screenshot]

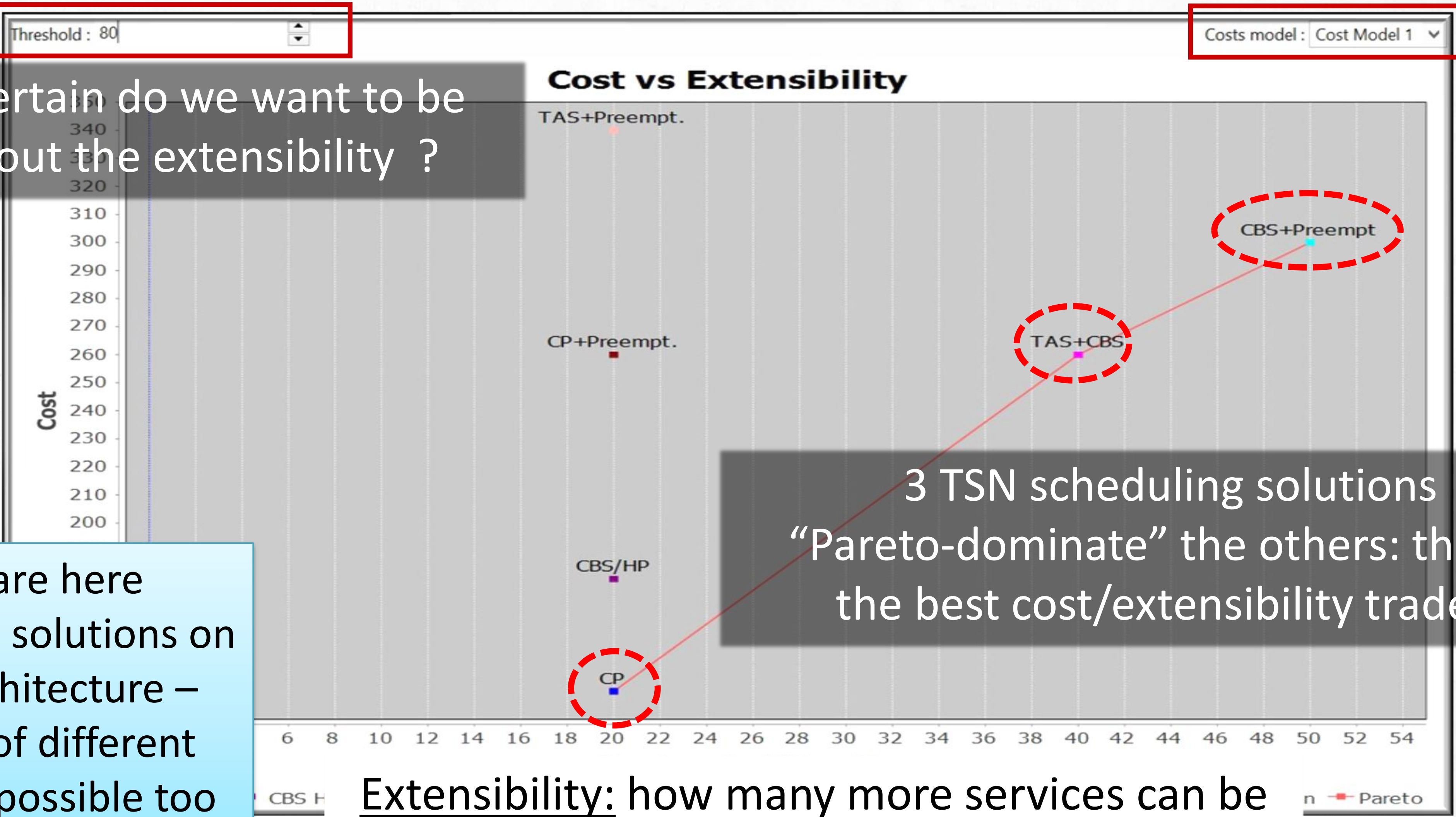
Cost / Extensibilty Analysis

Cost / extensibility trade-offs

Cost model applied

How certain do we want to be about the extensibility ?

Cost of the architecture for various TSN solutions



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

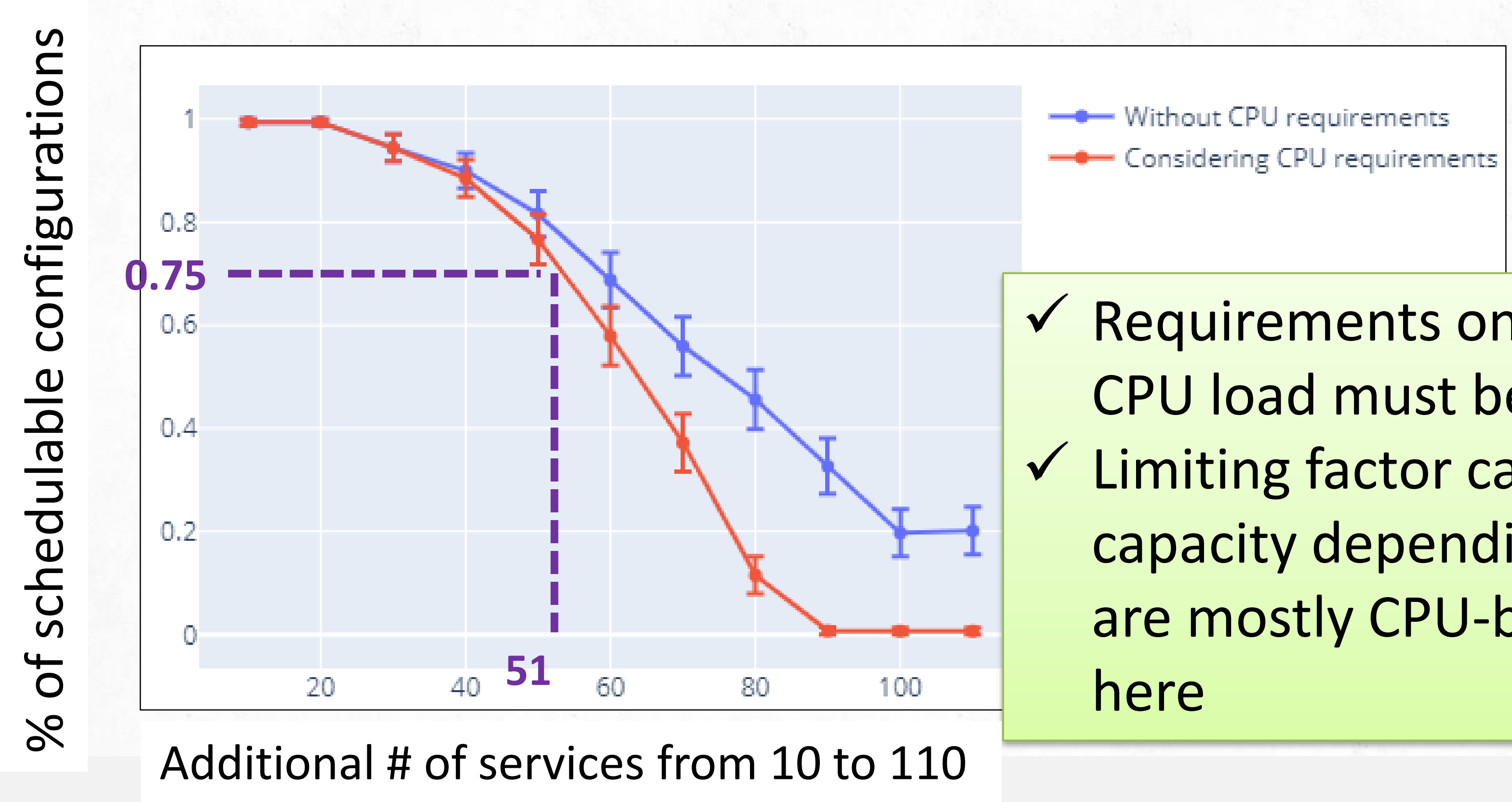
We compare here competing TSN solutions on the same architecture – comparison of different architectures possible too

Extensibility: how many more services can be "safely" added? Lifetime of the platform?

Total Architecture Capacity

Considering CPU requirements in addition to communication requirements

- ✓ Assumptions: each service requires a CPU time proportional the # of flows it processes – all processors have equivalent CPU power in this experiment
- ✓ Focus on the best performing TSN solutions: Express at top priority and 2 CBS classes below

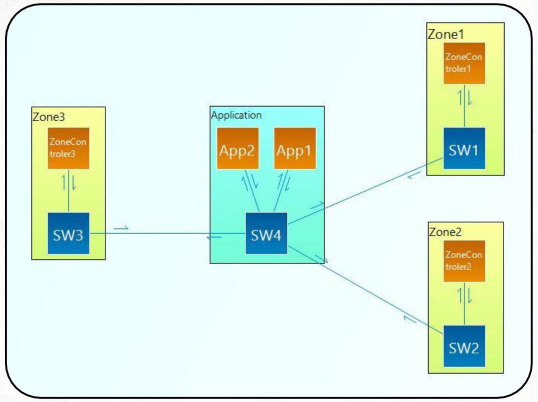


- ✓ Requirements on both communication and CPU load must be met
- ✓ Limiting factor can be network or CPU capacity depending on whether services are mostly CPU-bound or I/O-bound like here

Architecture Synthesis

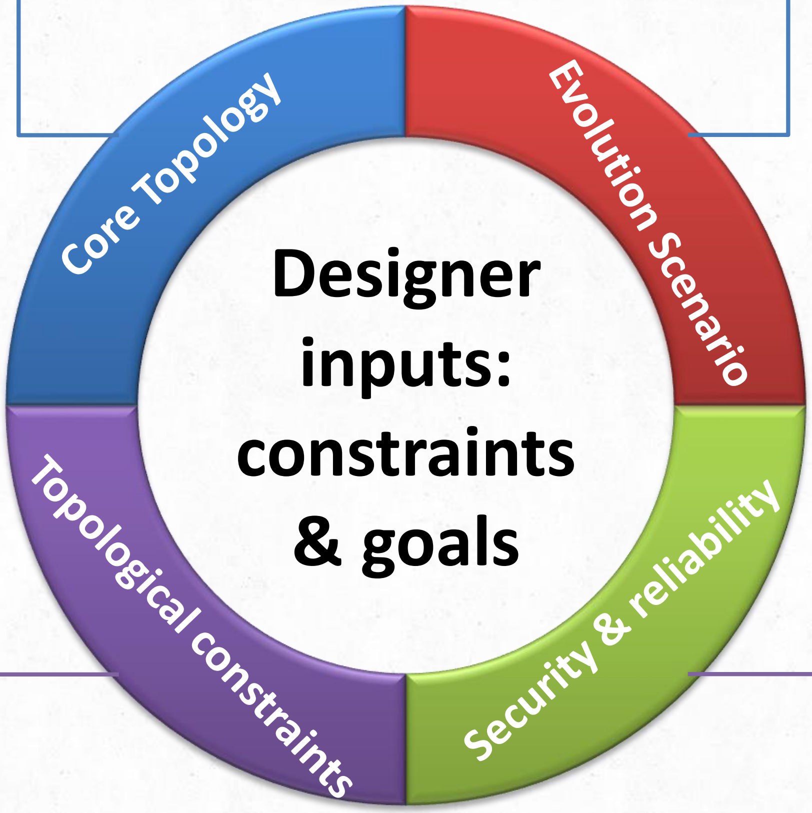
Architecture synthesis: extending a core topology

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**
- …



Topological constraints

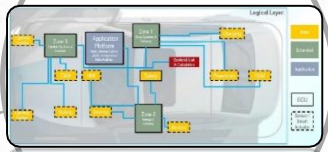
- connection lengths,
- physical location (e.g. vs power & sensors)
- ECU dimension restricting switch sizes, number of pins, power consumption, …

Security and reliability

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

Architecture Synthesis

Extending a topology: HW components that can be added



ECUs / Processors / SoCs

- Computing power
- Reliability & security

Switches

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

ECUs with internal switch

- Space & cost optimization
- Re-use in next generation

Link between switches

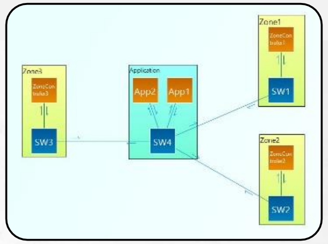
Additional bandwidth e.g. on backbone

Network interface + link ("dual homing")

- Load balancing
- Reliability & Security

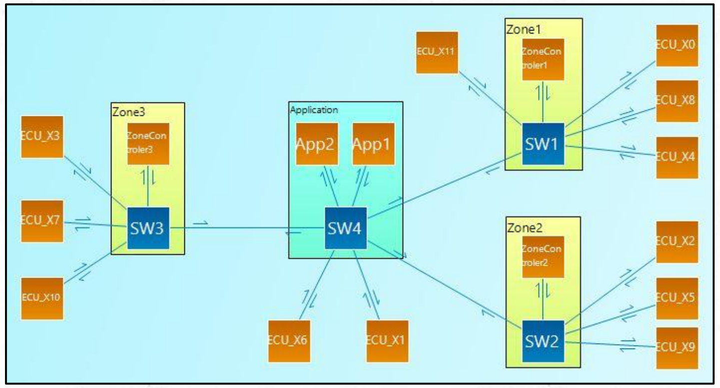
Catalogue of cost-effective "extension patterns" comprised of several HW components

Illustration: computer-generated architectures based on a core topology



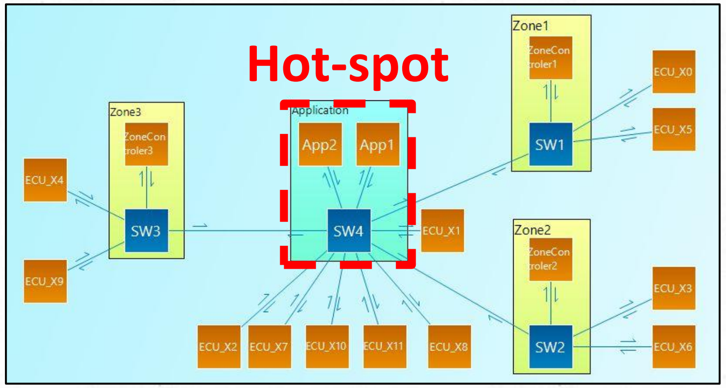
- ✓ <u>Heuristic applied here:</u> additional ECUs close to the "hot-spots", i.e. ECUs subject to max. variability pressure in terms of # future services added
- ✓ Parameter specifies trade-off between topology balance / hot-spots coverage

\- +



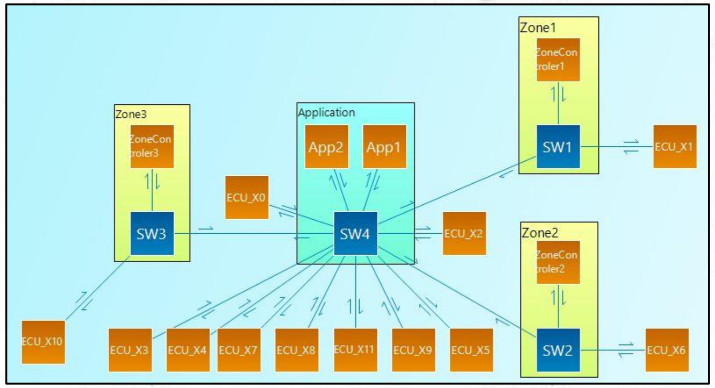
Candidate Sol. A

(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

BMW

UNIVERSITÉ DU LUXEMBOURG

Cognifyer

RTaW
RealTime-at-Work