QoS-Predictable SOA on TSN: Insights from a Case-Study

Josetxo VILLANUEVA, Groupe Renault

Jörn MIGGE, RealTime-at-Work (RTaW)

Nicolas NAVET, University of Luxembourg / Cognifyer.ai

RTaW RealTime-at-Work

GROUPE RENAULT

uni.lu UNIVERSITÉ DU LUXEMBOURG

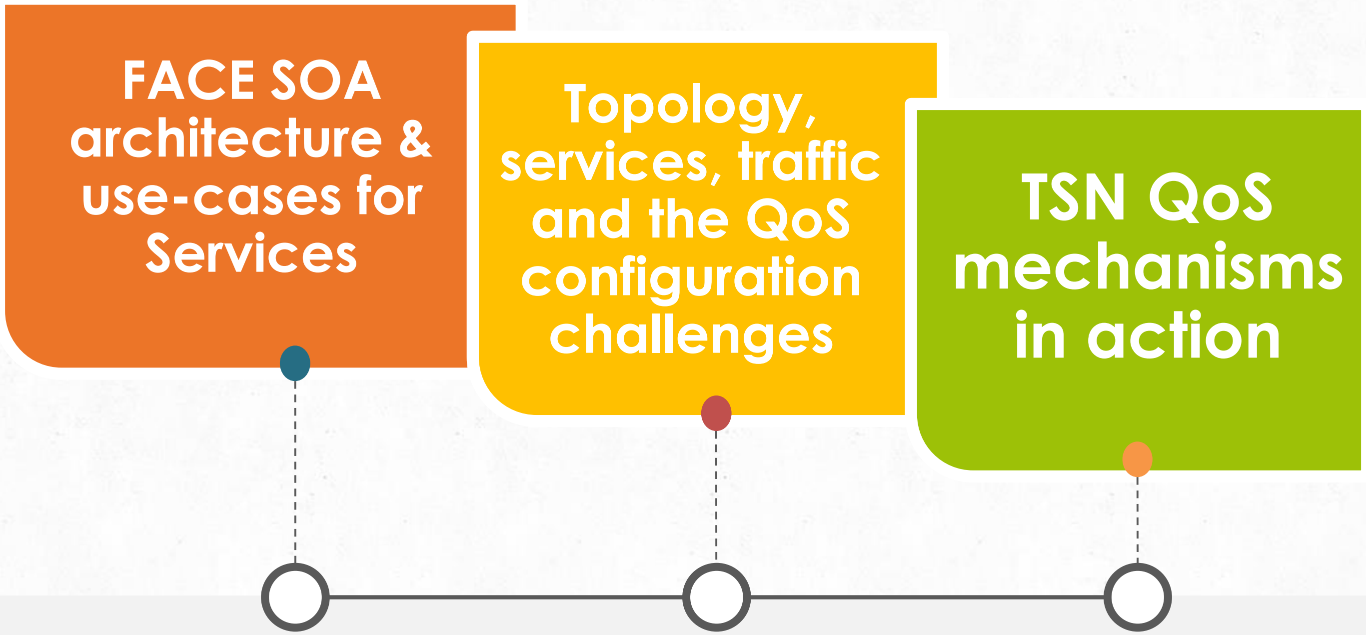
Cognifyer

Automotive Ethernet Congress
February 09-11, 2021| Munich, Germany

Outline & Objectives

- ✓ Takeaways learned in the design and implementation of the Renault FACE Service Oriented Architecture over Ethernet TSN backbone
- ✓ Concrete illustration of the use of services for two QoS-demanding use-cases:

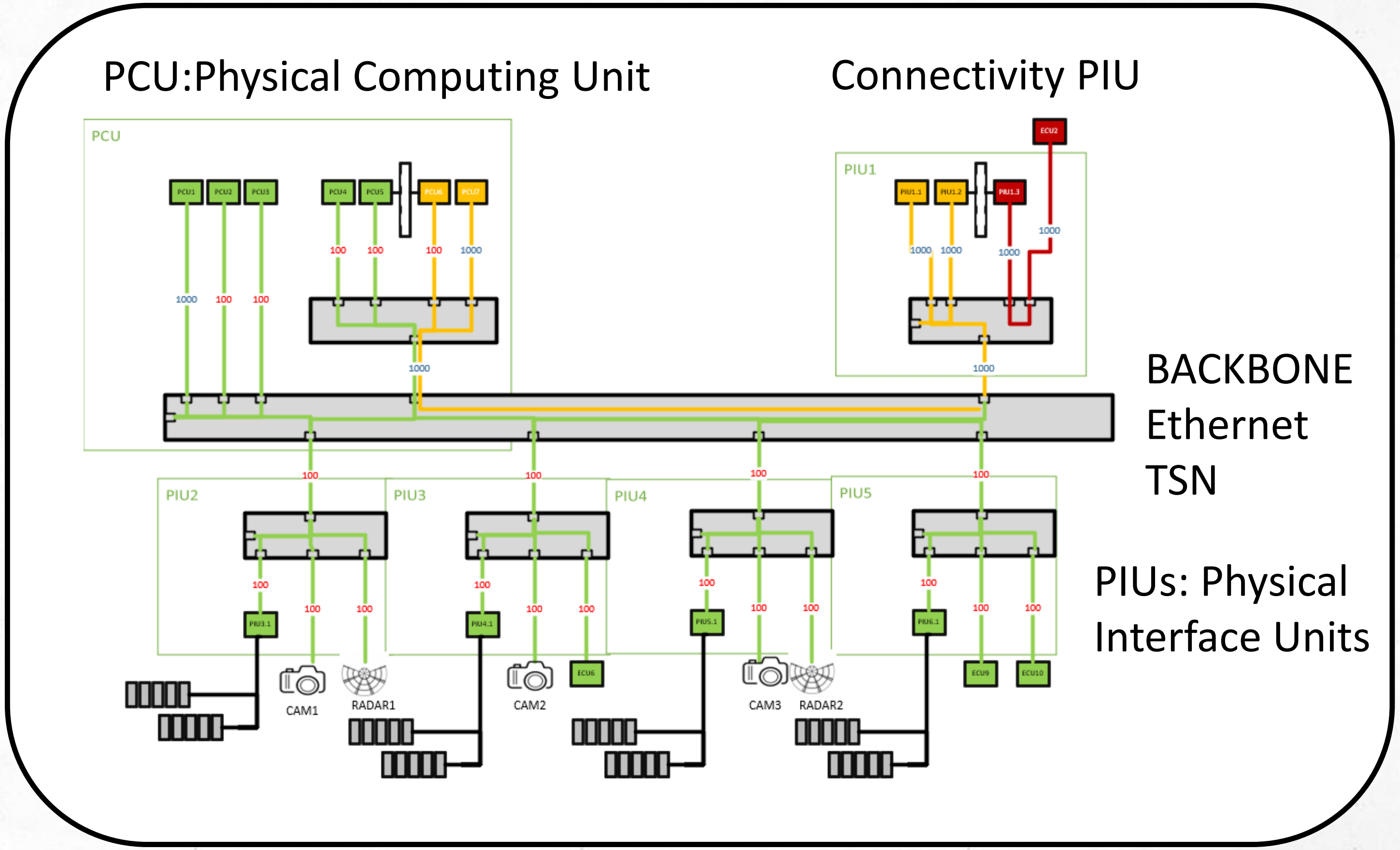
  Light Service Architecture (actuator) & Smart Sensor Fusion Use-Case
- ✓ The challenges in configuring Ethernet TSN for services & possible solutions
- ✓ Experiments: optimizing TSN configuration and the difference it makes in timing & memory



1. Designing next-generation service-oriented E/E architectures

SOA & Central Computing from OEM perspective

- ✓ SOA & Central Computing benefits
  - – Decoupling of HW & SW
    - Service & clients can be instantiated everywhere
  - – Re-use & modularity of Services (Building Blocks)
  - – Ease software-based innovation
  - – Personalization, new business models, by software updates



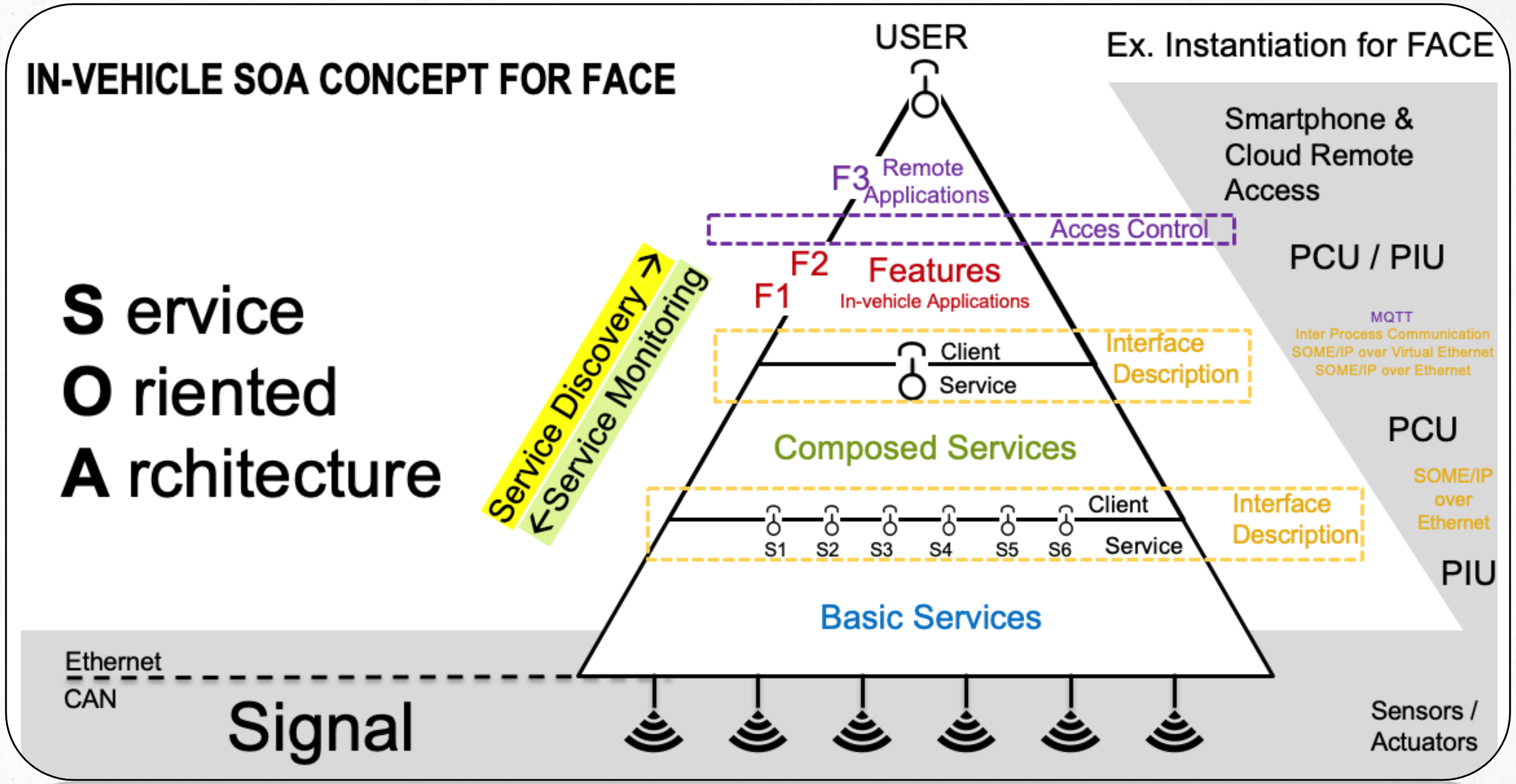
✓ Change of communication paradigms

Multi-platform Middleware (Eg. SOME/IP), flexibility & automation of network configuration, guarantee of network QoS

A hierarchy of services & applications

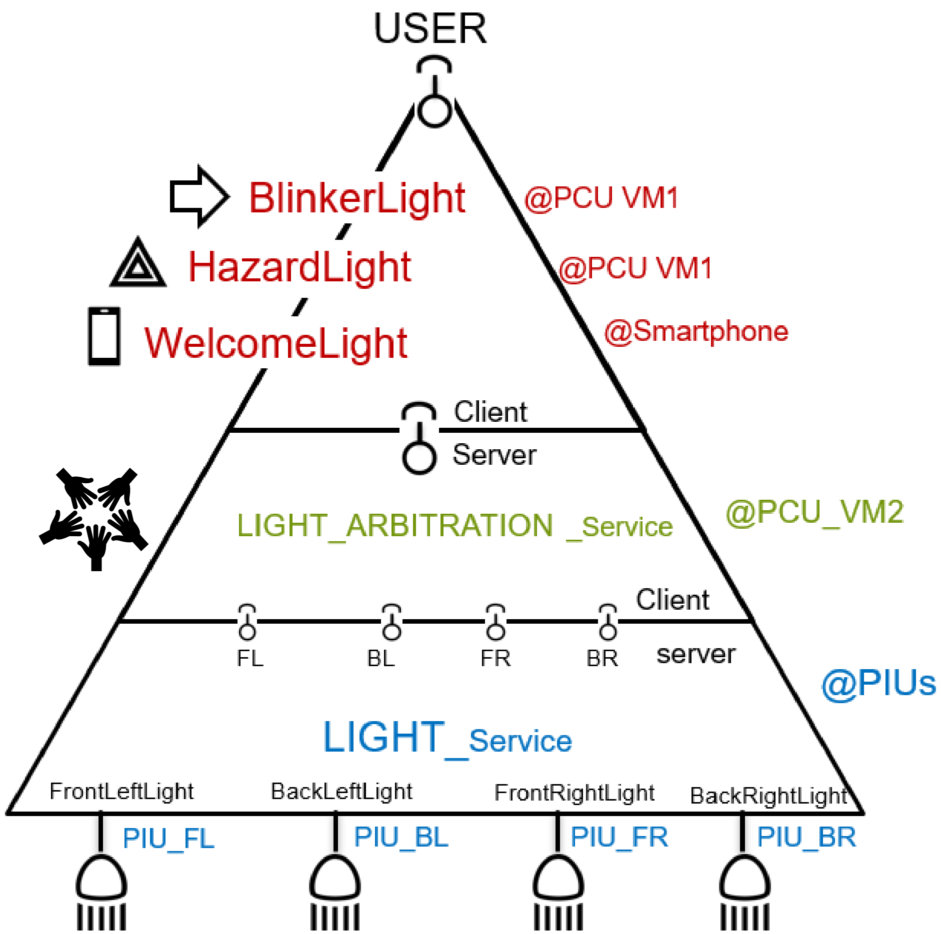
Services are executed:

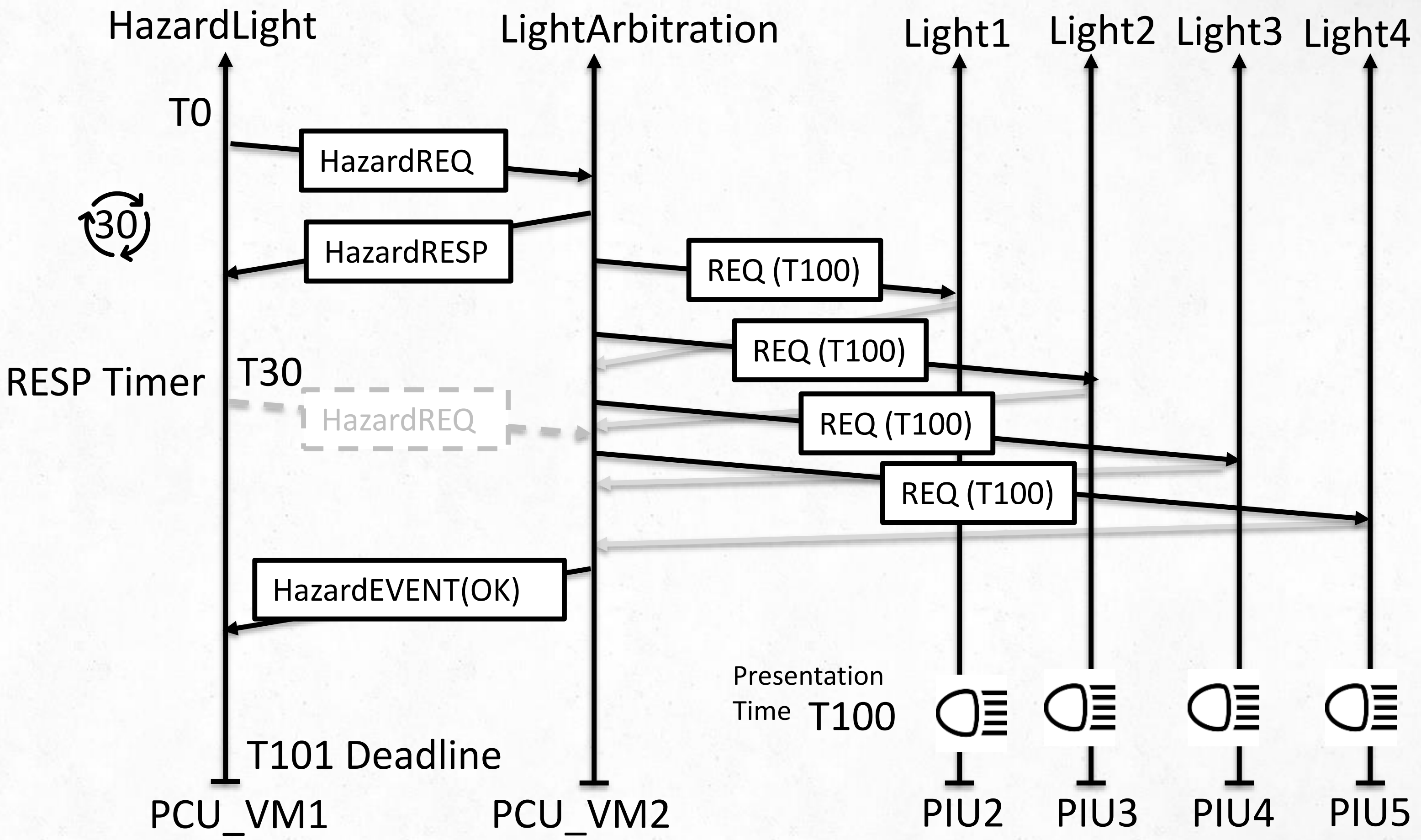
- ✓ In dedicated ECUs, ex: cam. & radars
- ✓ In zone controllers for basic services, ex: sensor data processing
- ✓ In High Performance Computing ECUs for composed services, ex: ADAS
- ✓ In the cloud, ex: infotainment
- ✓ In the infrastructure, ex: speed limit



Services can be grouped to define re-usable Software Components or complete Virtual ECUs

SOA Use-Case #1: Lighting Services

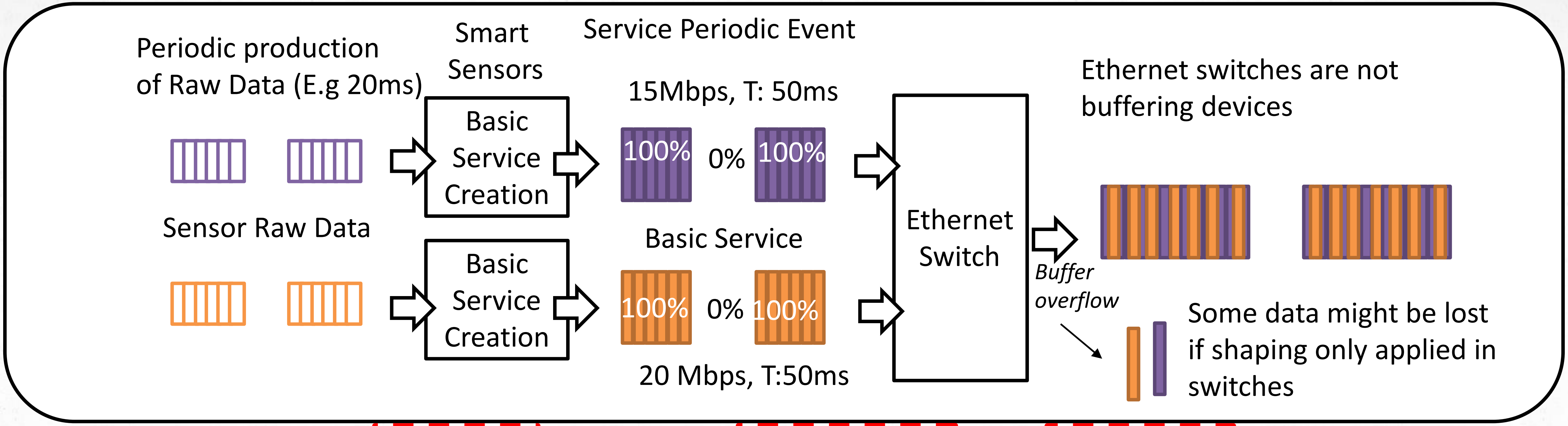




How to configure Service Communication when thousands of flows generated by hundreds of Services are competing for network resources?

SOA Use-Case #2: Smart Sensor Fusion Use-Case

- ✓ Smart Sensors: Translate analogue data into Service communication (Eg. CAM, Radar, …)



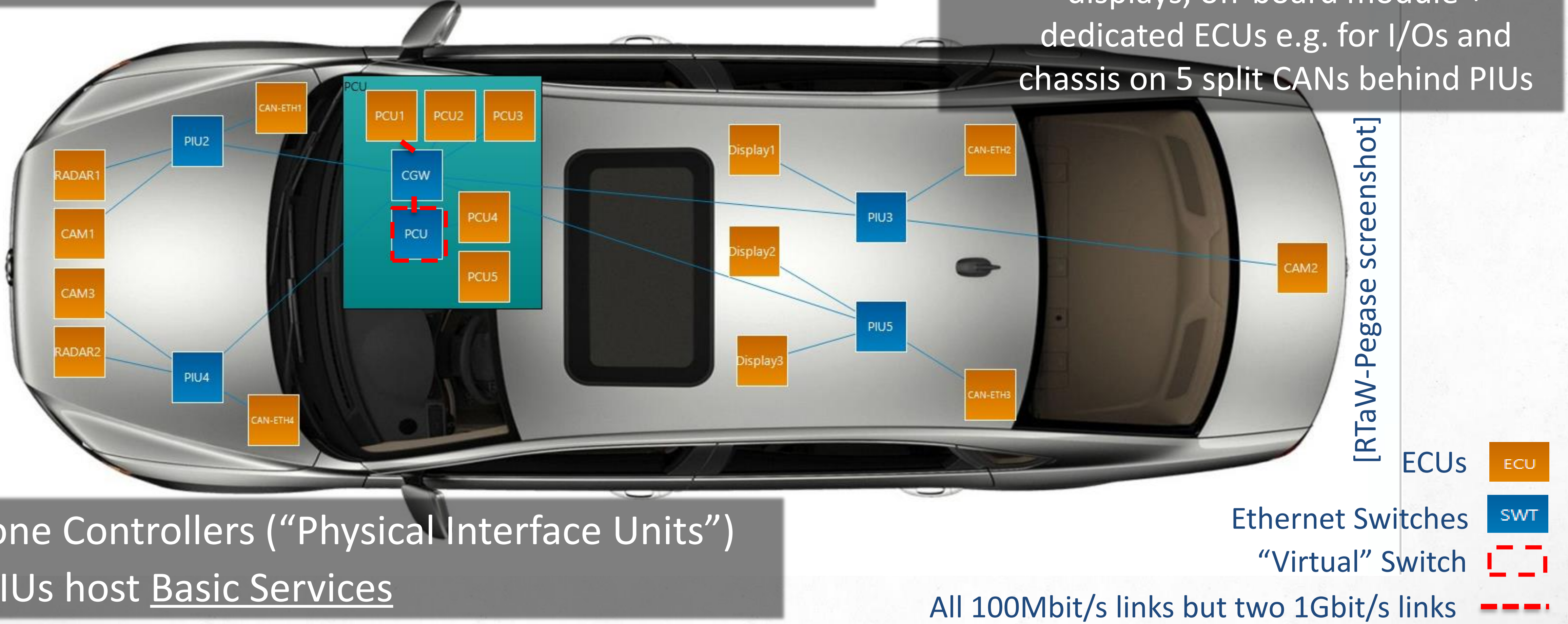
- ✓ Solution shaping with CBS (in HW or SW), pre-shaping (w/o) sub-bursts
- ✓ Which protocol to use, to ease spacement?
  - Upon choice, none either or both request & response can be segmented, and thus
  - Not all services, REQ & RESP of the same service may require the same QoS mechanisms

2. FACE E/E architecture: Topology, Protocol Stack, Services Characteristics and their QoS Requirements

Ethernet Simulation Model of FACE E/E architecture

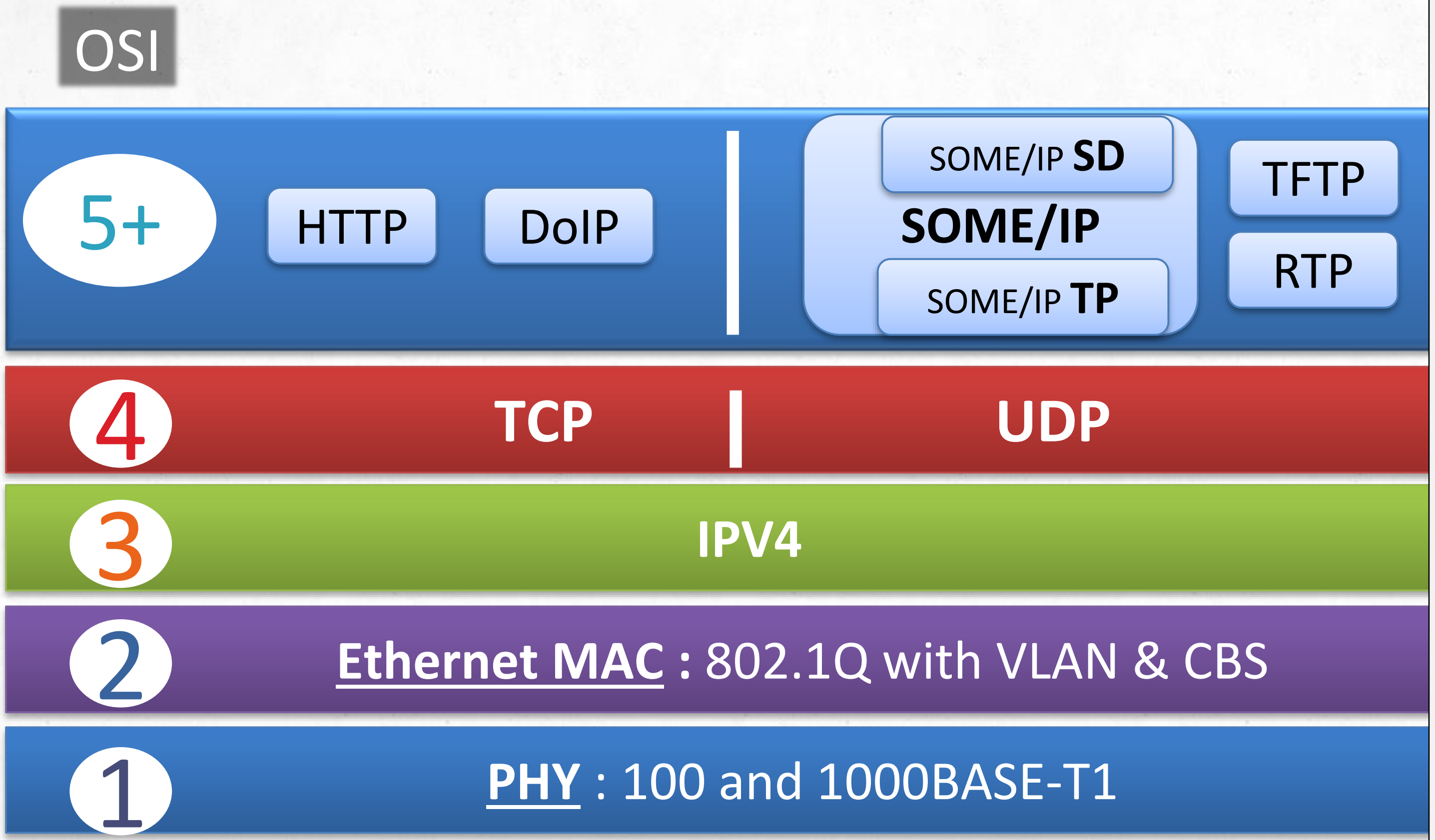
1 Central Computer ("Physical Computing Unit")
→PCU hosts Composed Services & Applications

17 ECUs on Ethernet including
3 front/rear cameras, 2 radars, 3 displays, off-board module + dedicated ECUs e.g. for I/Os and chassis on 5 split CANs behind PIUs



5 Zone Controllers ("Physical Interface Units")
→PIUs host Basic Services

Protocol Stack with a Focus on Segmenting & Shaping



- ✓ TCP: reliable & segmented transmissions but not real-time!
- ✓ SOME/IP TP: timing predictable & segmented transmissions but no shaping capability
- ✓ Segmenting server's messages into several *Events* may be a work-around for not using SOME/IP TP.
  → Defining Event period is not enough. Need to space *Events* transmission.
- ✓ CBS: limited memory in egress ports, additional shaping in SW in sender's comm. stack may be needed, but no standard solution.

Non-functional Requirements on Services

1 **Timing** (→ service colocation constraint)

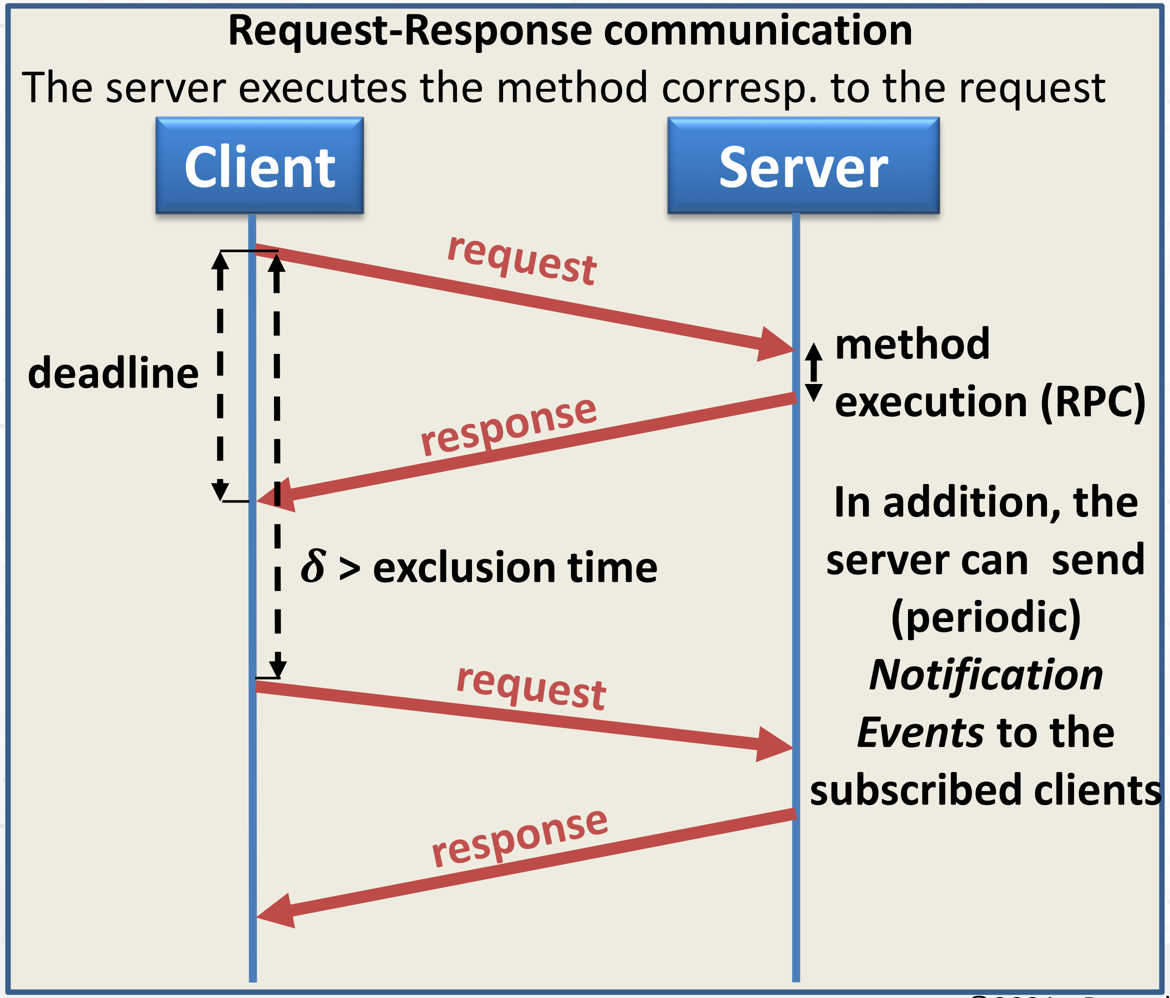
2 **Memory** (SW, HW), **computing power, funct. modes** (power)

3 **Security & Safety** (e.g., redundancy on ≠ cores, core ASIL levels)

- ✓ **Predictable real-time behavior required to suppress dedicated chassis or ADAS ECUs**
- ✓ **Timing depends on the execution platform**: Classic platform on dedicated core more predictable than Adaptive platform (HPC does not mean real-time)
- ✓ **Service allocation is key for optimized resource usage / extensibility** → design-space exploration coupled with timing-accurate simulation can help optimize the allocation
- ✓ **TSN QoS mechanisms such as shaping have an impact on memory usage in HW & SW**

Configuring TSN QoS Mechanisms with Services

- ✓ Configuration should ensure that all streams, not only services, meet their timing constraints
- ✓ How to set priorities and TSN QoS mechanisms ?



<u>Configuration challenges with services</u>:
- ∃ deadlines on req.-resp. transactions not only individual transmissions
- Some messages, typ. resp., can require segmenting & shaping
- ≠ timing constraints for each req.-resp. transact. and *Events* of the same service
- Thousands of streams! Which calls for an automated process based on models

Characteristics of the Services

- ✓ Subscribers are SW components not ECUs
- ✓ "period" for calls to methods means their *exclusion time*

		SOME/IP service type
Smart Sensor Basic Services ex: object & infrastructure detection	✓ ≈20 – Typ. period: 50ms ✓ Segmented messages (mostly) ✓ Hard deadlines	Events only subscribed by PCU
Basic Services ex: environment sensing	✓ > 40 – Typ. period: 10-100ms ✓ Non-segmented messages (mostly) ✓ Hard deadlines	1 Event + 3 Methods 1-10 subscribers
Composed Services ex: fusion & resources arbitration	✓ > 60 – Typ. period: 20-100ms ✓ Typ. 5x larger than basic services ✓ Hard deadlines	5 Events + 5-10 Methods 1-10 subscribers
Cloud services ex: heating remote control	✓ 10 < - Typ. period: 5s ✓ Segmented messages ✓ Soft deadlines	1 method and 1 subscriber

Decreasing priorities

A single service can generate a lot of traffic! E.g. 100 unicast streams and 5 multicast streams for a composed service offering 10 methods and 5 events to 10 clients

Non-Service Related Traffic

CAN (FD) Snapshots
Re-forwarded CAN (FD) frames

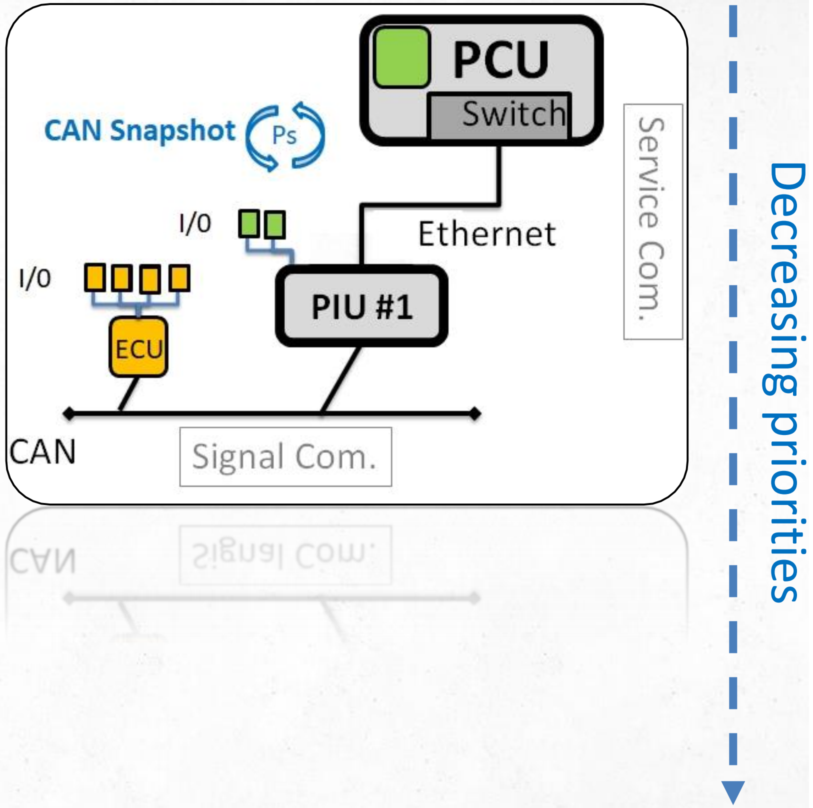
- ✓ ≈30 – period: 10 or 20ms
- ✓ Non-segmented messages
- ✓ Hard deadlines: 5ms

TFTP + RTP Video Streams
ex: infotainment

- ✓ <5 – Period: from 33ms (30FPS) to 1s
- ✓ Segmented messages
- ✓ Mixed deadline and throughput constraints

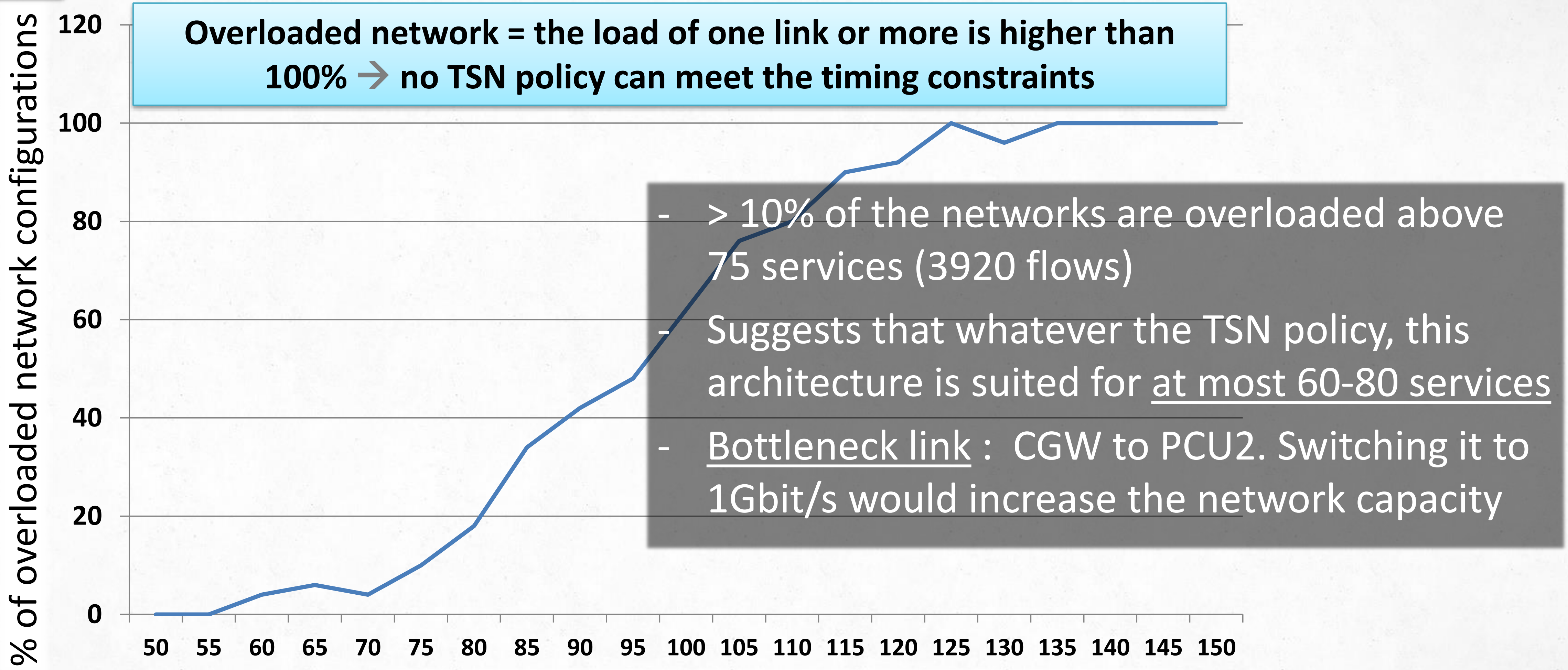
TCP & HTTP streams
ex: off-board comm., DoIP

- ✓ ≈10 – sporadic: typ. 1s
- ✓ Segmented messages
- ✓ Throughput constraints



3. Optimized TSN configuration for services to maximise network capacity and reduce memory consumption

% of overloaded networks as a function of the number of services : KPI of Evolutivity



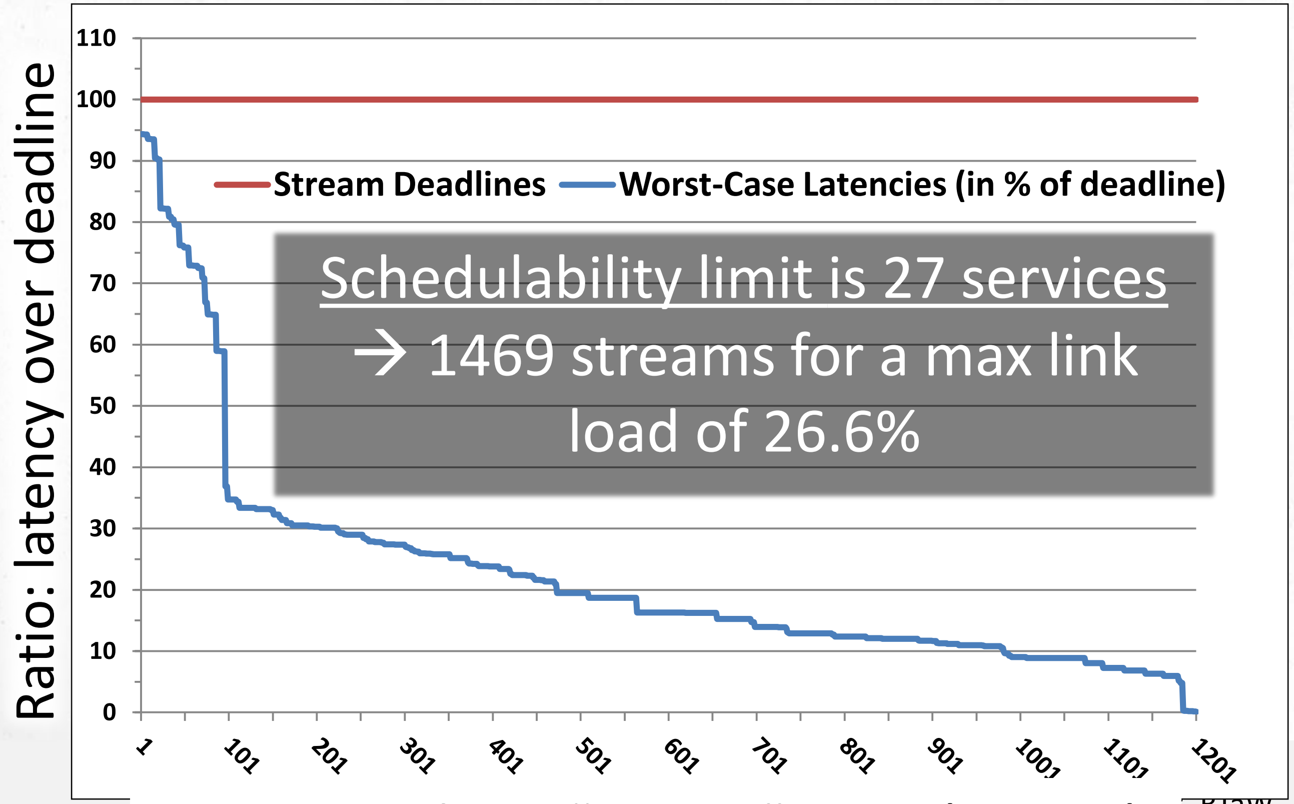
# of services from 50 to 160 – each service requires a bandwidth up to 1Mbit/s

Manual Traffic Prioritization

- ✓ TSN mechanism: priority, no shaping
- ✓ Traffic classification based on deadlines with manual tuning:
  - – Urgent Services : deadlines < 10ms

ClassConfigurations + −

TrafficClass	Priority	Express	SchedulingPolicy
CAN SNAPSHOTS	7	false	FIFO
SERVICES (urgent)	6	false	FIFO
SERVICES	5	false	FIFO
ADAS-SERVICES	4	false	FIFO
DISPLAY	3	false	FIFO
TFTP+TCP	2	false	FIFO



Streams of the “critical” class (prio. 5)

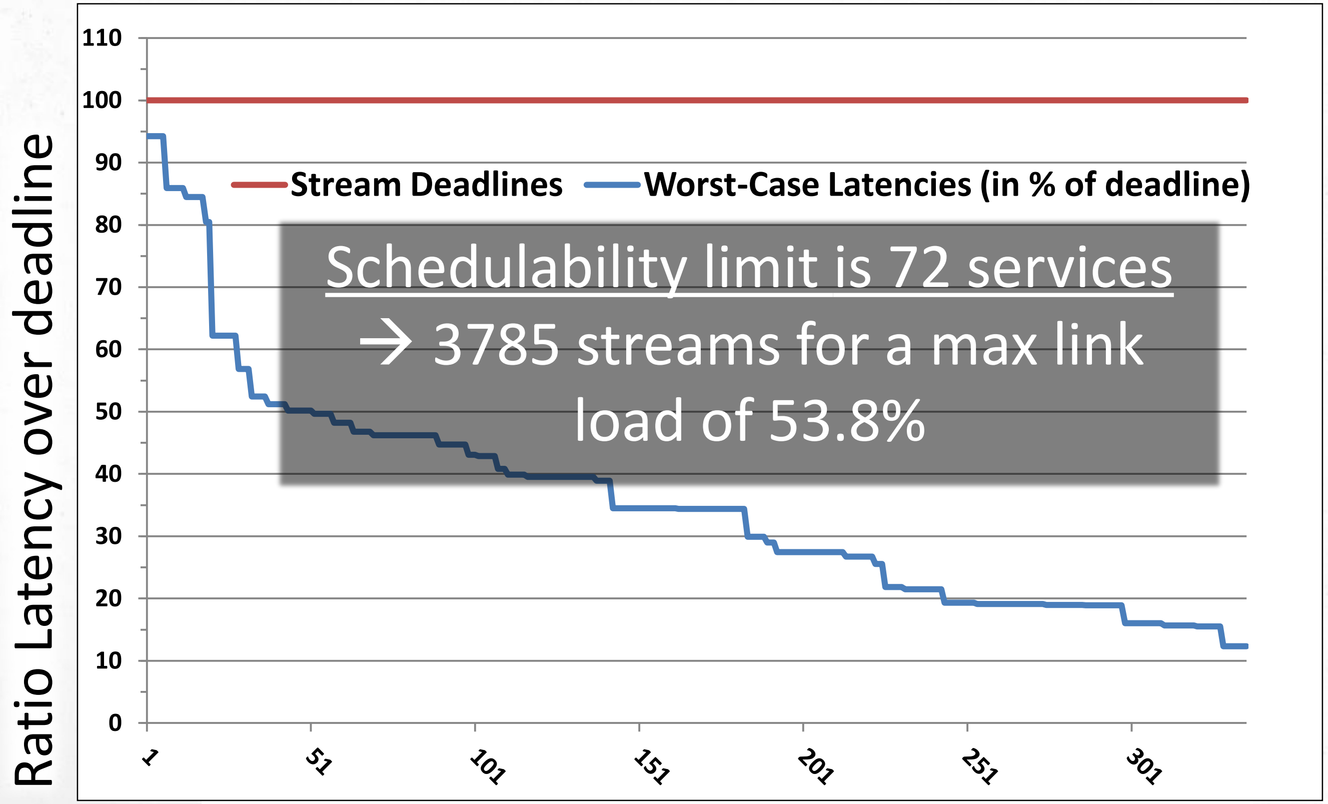
- – Shaping not helpful: ADAS-Services are at low priority level & shaping non-segm. packets of limited use
- – Preemption not helpful as deadline misses do not occur at top priority

Traffic prioritization with *Concise Priorities* algorithm

- ✓ TSN mechanism: priority, no shaping
- ✓ Automated traffic classification
  - – Streams of different types will be mixed at all priority levels

ClassConfigurations

TrafficClass	Priority	Express	Scheduling...
Class 0	7	false	FIFO
Class 1	6	false	FIFO
Class 2	5	false	FIFO
Class 3	4	false	FIFO
Class 4	3	false	FIFO
Class 5	2	false	FIFO
Class 6	1	false	FIFO
TCP+TFTP	0	false	FIFO



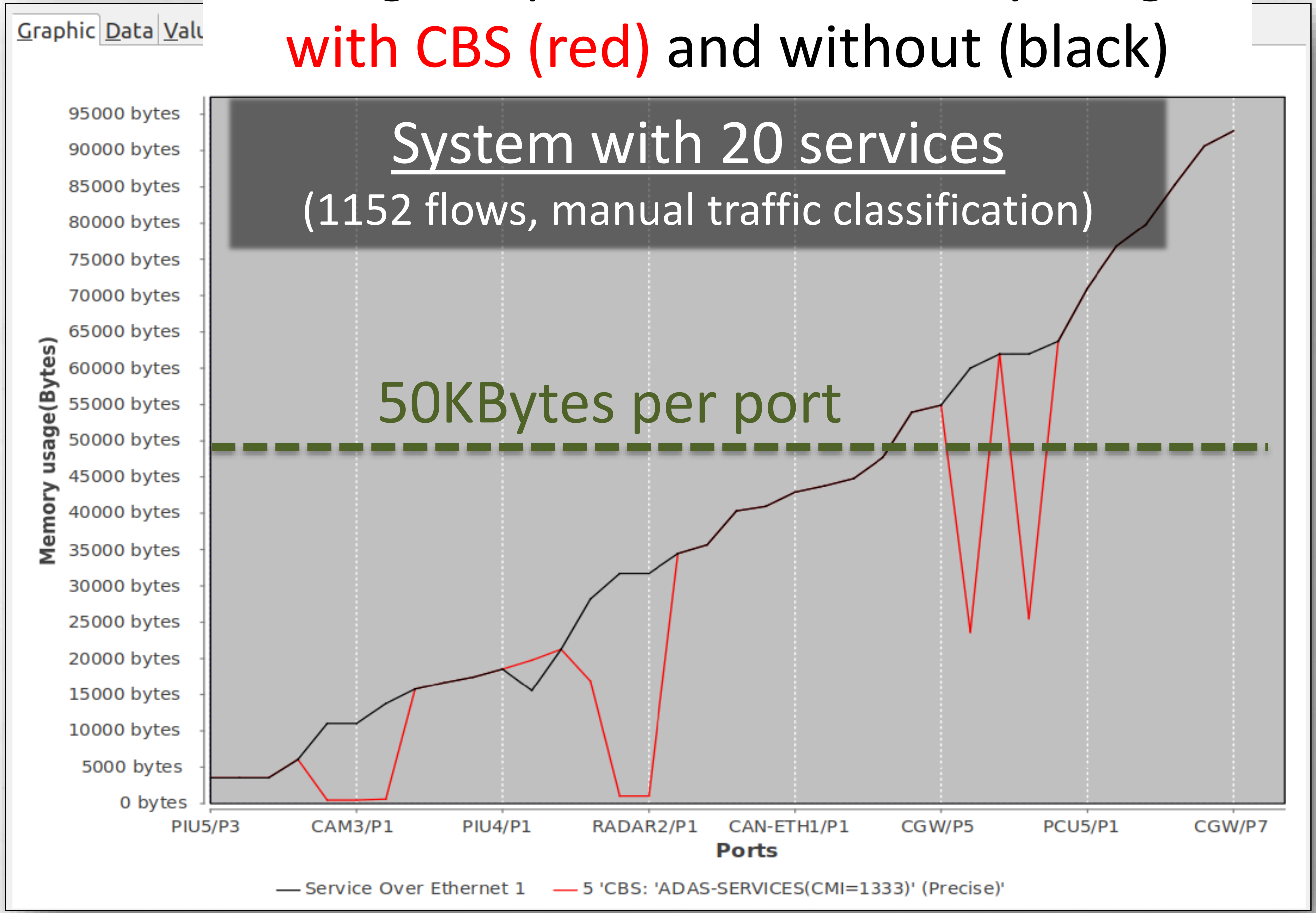
Streams of the "critical" class (prio. 6)

- – All 8 priority levels are used
- – Deadline misses are top priority
- – Preemption not helpful as blocking from low prio. packets limited in delays
- – Shaping only marginally effective for memory as there is little slack time

4. Algorithm–based Configuration

Reducing Memory Usage with Shaping

Per-egress ports max. memory usage:
with CBS (red) and without (black)



- ✓ In practice, ∃ constraints not only on latencies but memory & CPU usage → ≠ TSN sched. sol. lead ≠ tradeoffs
- ✓ Shaping improves delays for lower priority packets but not always memory usage

- – CBS CMI: 1333us, comparable gains with CMI 250us or SW shaping on senders
- – Per device memory usage reduction with CBS – max: 97%, average: 12.3%
- – No gain in switch ports where video & radar streams are merged
- – Memory in egress ports can be exceeded even with moderate load!

Conclusion and a look forward

Takeaways

Finding the <u>right granularity for services definition</u> is fundamental

– “big” services, generating each up to 1Mbps of data, have been used in this work but is it always the right choice?

Correct service execution relies on the QoS provided by Ethernet TSN, which requires proper protocols selection and configuration

<u>Challenges in ensuring network QoS</u>:

- Tight collaboration between OEM and Tier1/2 needed during integration phase due to limited maturity in SOA/Central Architecture (COTS, tools, …)
- Shaping in Autosar comm. stacks ?
- Non-deterministic execution platforms place additional constraints on network (e.g., retransmissions)
    - System complexity & subtle interactions between QoS mechanisms calls for model-based configuration and verification for optimized resource usage

Thank you for your attention!