VOLVO

Signal-Oriented ECUs in a Centralized Service-Oriented Architecture: Scalability of the Layered Software Architecture

Hoai Hoang BENGTSSON & Martin HILLER
Volvo Cars Sweden

Jörn MIGGE
RealTime-at-Work (RTaW)

Nicolas NAVET
Uni. Luxembourg & Cognifyer

Automotive Ethernet Congress 2022, Volvo SoA

2022-06-01

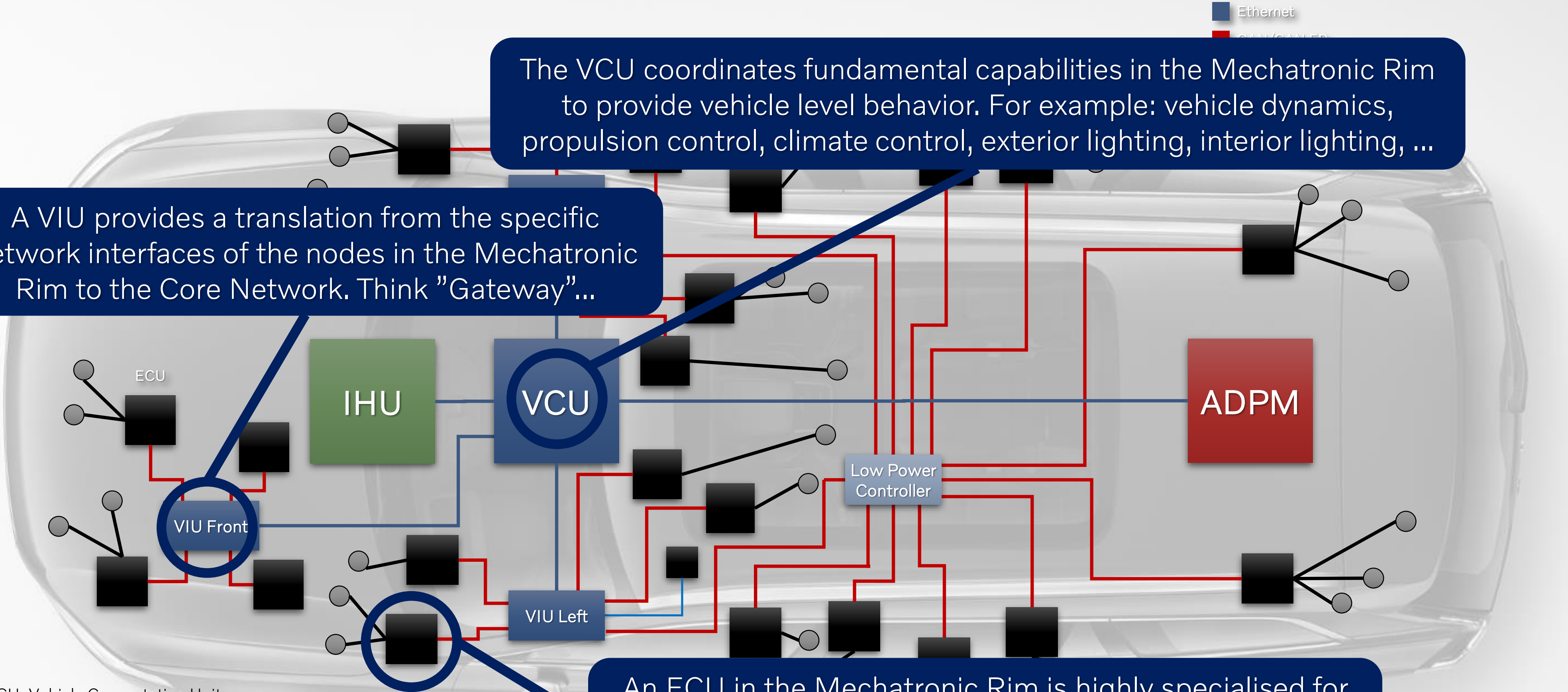
Agenda

- Introduction of SOA architecture – Challenges
- Service interface communication model
- Modelling of the system
- Performance evaluation
- Conclusion

Introduction of SOA in next generation architecture

VOLVO

Centralized E/E Architecture with Ethernet Backbone



The VCU coordinates fundamental capabilities in the Mechatronic Rim to provide vehicle level behavior. For example: vehicle dynamics, propulsion control, climate control, exterior lighting, interior lighting, ...

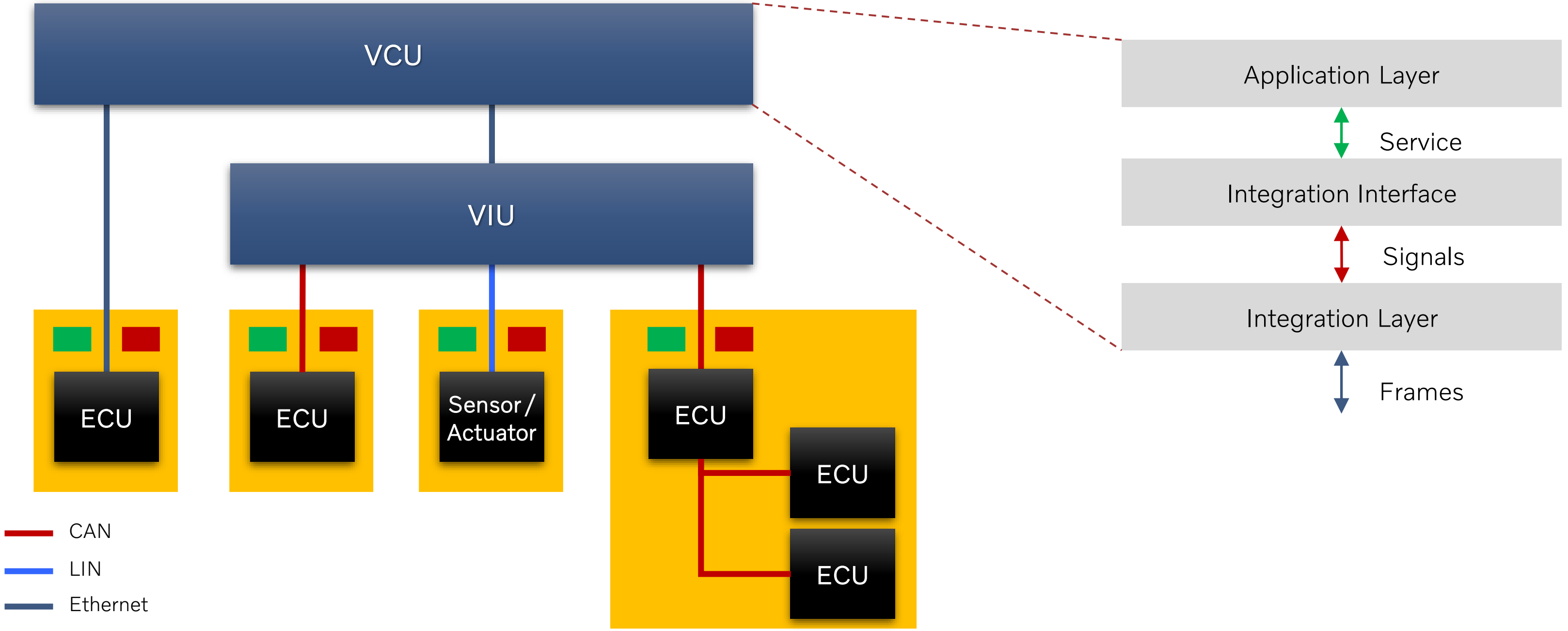
A VIU provides a translation from the specific network interfaces of the nodes in the Mechatronic Rim to the Core Network. Think ”Gateway”...

An ECU in the Mechatronic Rim is highly specialised for controlling its specific device. For example: engine, transmission, brakes, steering, doors, windows, seats, ...

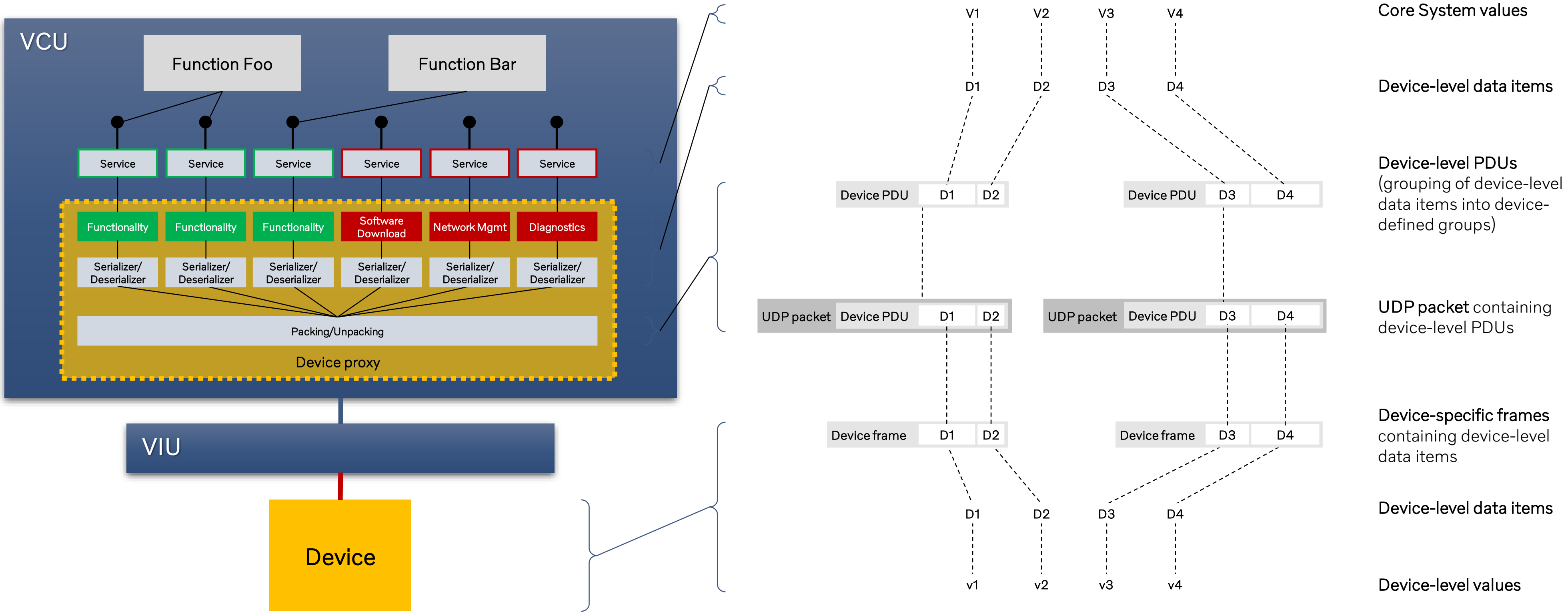
VCU: Vehicle Computation Unit
VIU: Vehicle Integration Unit
ADMP: Autonomous Driving Primary Module
IHU: Infotainment Head Unit

SW Layering Architecture

- **Device:** One or more ECUs, sensor or actuator in the Mechatronic Rim which are connected to the Core System as one unit.
- **Functional interface:** Accessing the (payload) functionality provided by a device.
- **Administrative interface:** Accessing diagnostics, software download, network management, and power management.



Devices and proxies and services, Oh my...

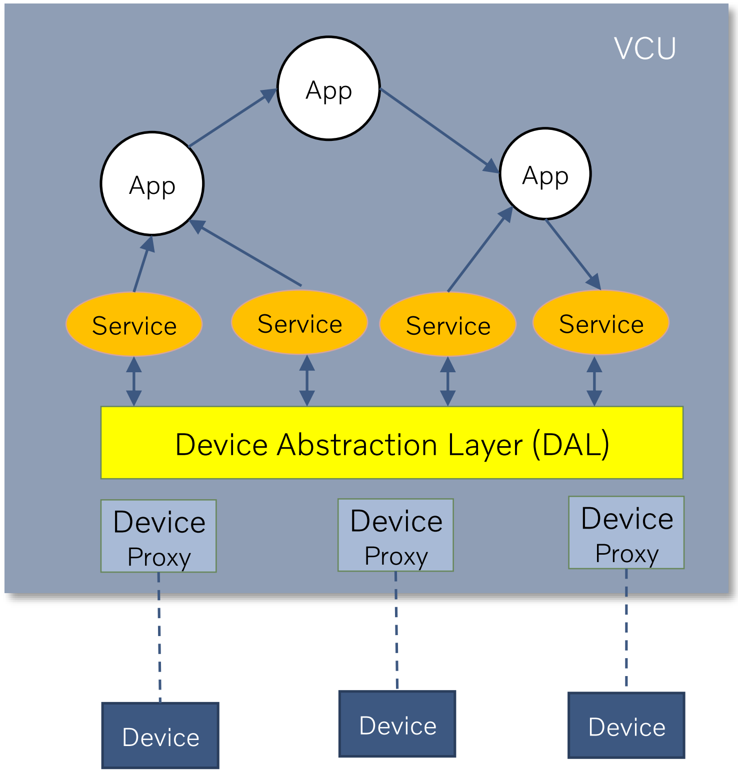


Layering Architecture Concept - Challenges

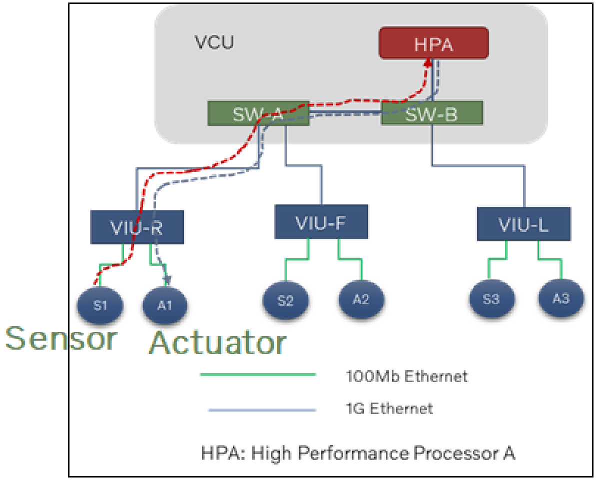
How to efficiently transform signals to services? What are suitable services that a device can provide?

What communication model is suitable for our architecture choice?

How the design choices affect the system performance?

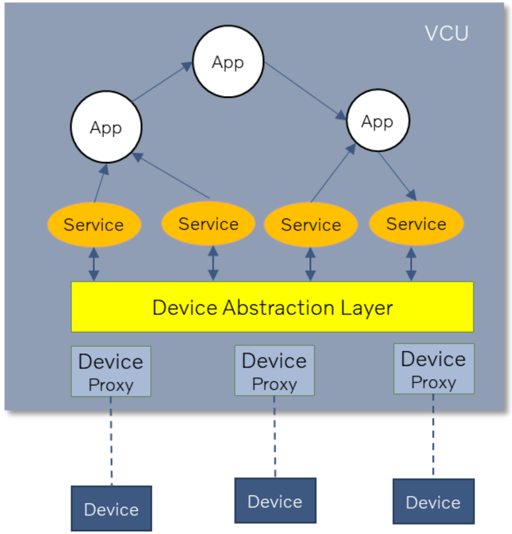


Service interface communication model



<u>Timing chains from sensor to actuator</u> comprise several successive “segments” : data transmission and functions execution

Where do end-to-end delays come from ?



1. Delays come from processors and network loads → f (frequencies, transmission and execution times)
2. Burstiness of the load
3. Waiting times between successive tasks and messages of a timing chain → *our focus*

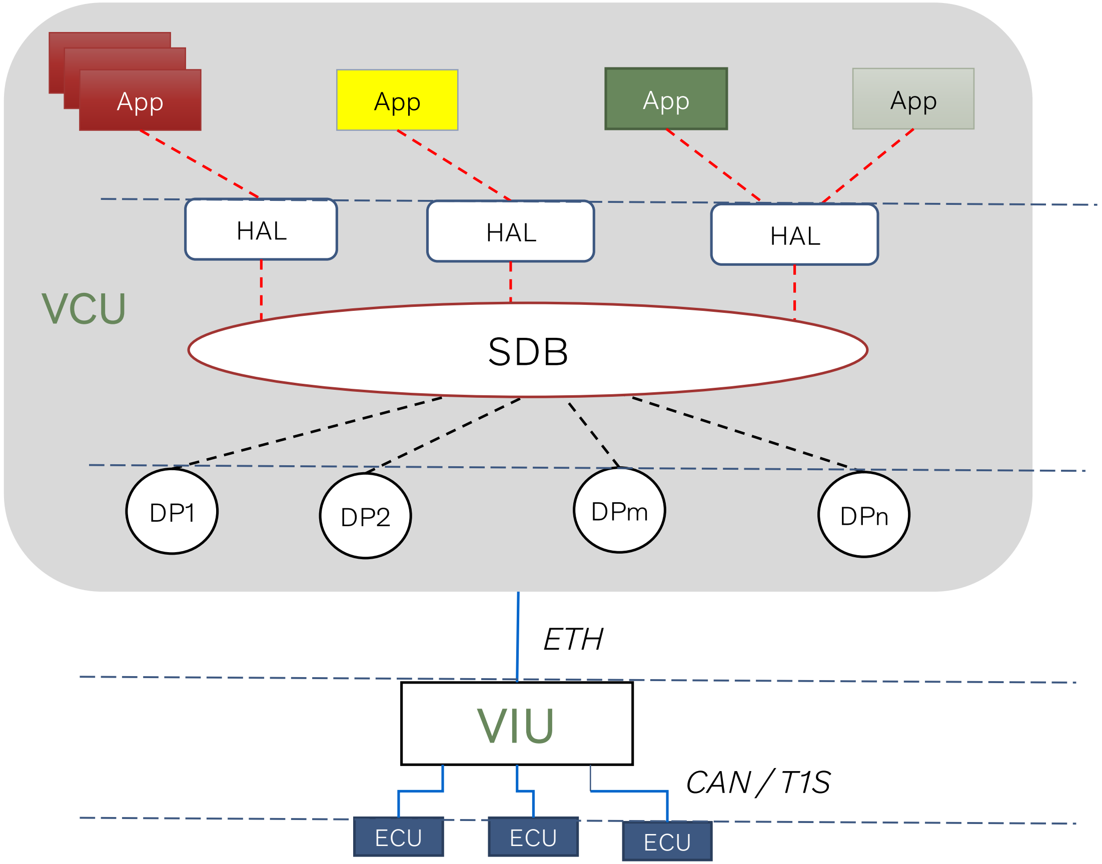
Periodic activations reduce loads!

- ✓ Several signals into same frame
- ✓ Periodic tasks and "Publisher-Subscribers" services → execution rate under control

Event-Triggered (ET) activations reduce latencies!

- ✓ No waiting times between segments of a timing chain
- ✓ Apps call services whenever needed and get responses asap (ODS: "On-Demand Service" next)

Choice of activation model at several levels



Service Interface

Publisher-Subscriber model (subscribers are updated e.g. periodically by server) or Client-Server model (ODS) ?

Updating signal database

Device Proxy executed periodically or triggered by ETH packet ?

Forwarding on ETH

Periodic ("snapshots") or ET transmission when CAN frame arrives at VIU ?

Transmission on CAN

Periodic or ET transmission on signal's update?

Design options compared next

Fully periodic model

- ✓ Service level: publisher-subscriber model (“PubSub”) with periodic events
- ✓ Task execution + CAN and ETH frames are periodic

Fully event-triggered model

- ✓ Service level: ODS
- ✓ CAN frame arriving at a VIU triggers ETH packet
- ✓ Signal production triggers CAN frame

Mixed model

- ✓ Periodic model for <u>non-time-critical timing chains</u>
- ✓ Event-triggered model <u>for time-critical timing chains</u>

Simplifications:

- Safety constraints are not discussed
- PDUs are kept out of the picture as far as possible

Metric for performance evaluation : max # of timing chains supported

System Model & Performance evaluation

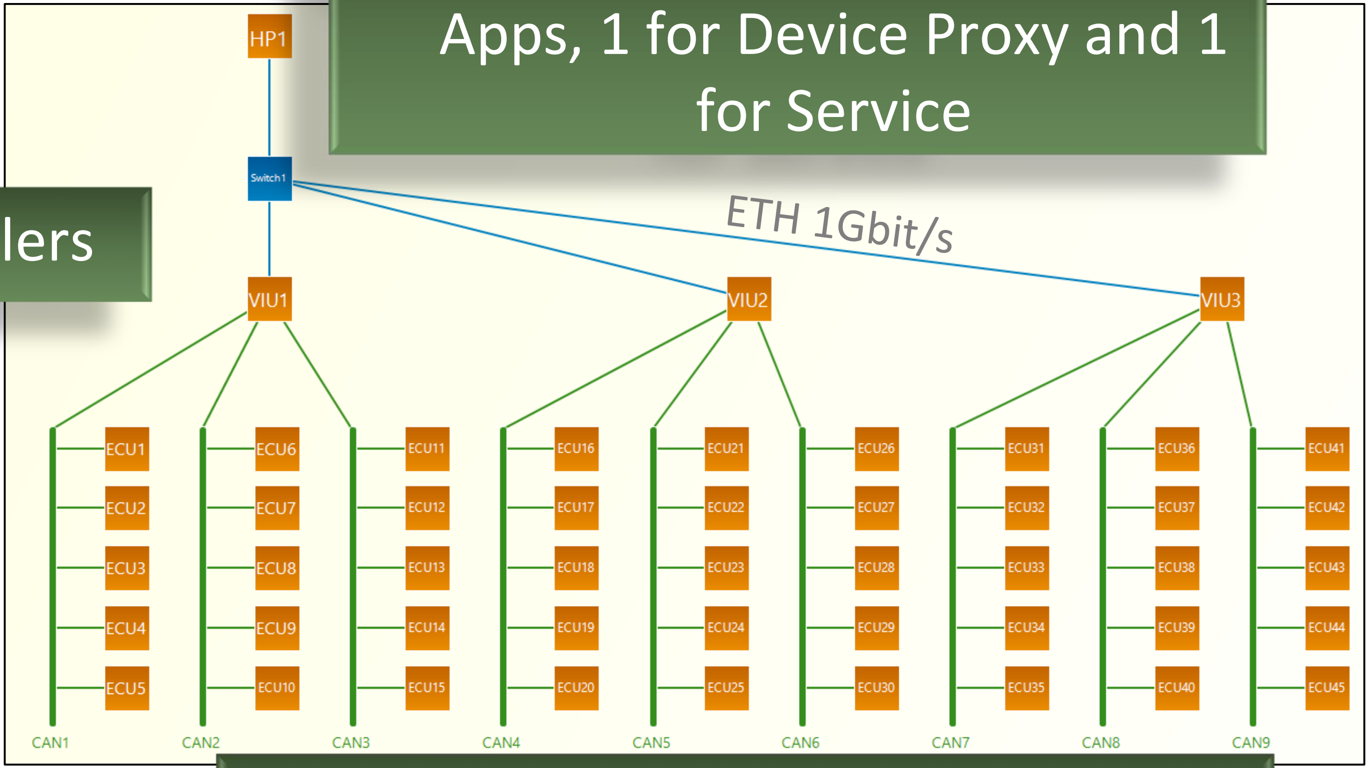
System Model : network

✓ Modelling and simulation with RTaW-Pegase / SDV

3 Zone Controllers

Links Speed	ETH: 1Gbps CAN FD: 2Mbps or T1S: 10Mbps
Sampling Period	10ms and 50ms
Timing Chains	End-to-end deadlines set to 4x (25%) or 10x sampling period

Central Computer with 2 cores for Apps, 1 for Device Proxy and 1 for Service

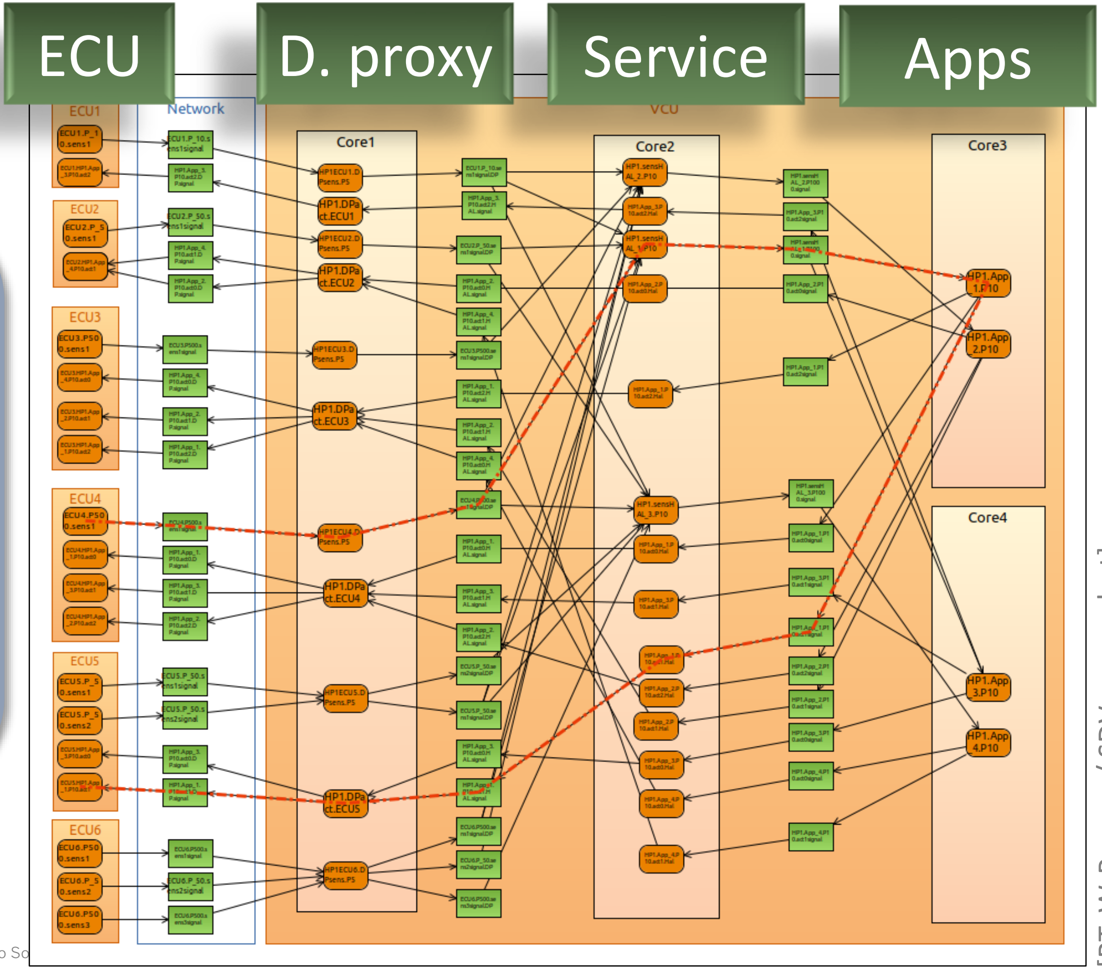


45 ECUs on 9 CAN FD / T1S buses

[RTaW-Pegase screenshot]

System model : flows of data between functions

- Excerpt of the functional architecture with a timing chain highlighted
- Apps are subscribed to several services and thus participate in several timing chains

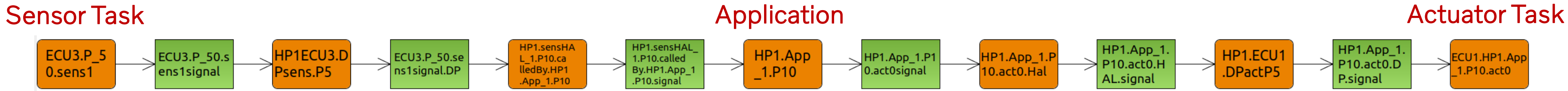


[RTaW-Pegase / SDV screenshot]

Executable
Data exchange
Segment of the timing chain selected

VOLVO

Breakdown of an end-to-end delay: periodic + PubSub service



Step	Activation Mechanism	Max. Waiting Times
Sensor Task	Periodic tasks	none
CAN	Periodic frames	frame period
Ethernet	Buffered PDUs	buffering timeout
Device Proxy	Periodic task	period
Service	Service event (PubSub)	period
Application	Periodic tasks	period
Service	Service method call (ODS)	none
Device Proxy	Periodic tasks	period
Ethernet	Buffered PDUs	buffering timeout
CAN	PDU triggers CAN frame	none
Actuator Task	Periodic tasks	period

Lots of waiting times between successive segments !

VOLVO

Breakdown of an End-to-end Delay: periodic + PubSub service

Delays of a timing chain's segments

TimingChain ECU8.P_50.sens1->ECU15.HP1.App_5.P50.act0*

Segment	Min	Average	Max
ECU8.P_50.sens1	0.010 ms	0.045 ms	0.080 ms
'ECU8.P_50.sens1signal' waiting	29.921 ms	29.957 ms	29.992 ms
Tx over 'CAN8'	0.212 ms	0.213 ms	0.341 ms
Gatewaying delay in 'VIU1'	0.000 ms	2.500 ms	5.000 ms
PduUdpTxBuffer	0.000 ms	2.849 ms	5.000 ms
VIU1-> Switch1	0.067 ms	0.177 ms	0.346 ms
Switch1-> HP1	0.067 ms	0.176 ms	0.553 ms
'ECU8.P_50.sens1signal' waiting	0.000 ms	2.496 ms	5.319 ms
HP1ECU8.DPsens.P5	0.020 ms	0.190 ms	0.360 ms
'ECU8.P_50.sens1signal.DP' waiting	2.640 ms	25.606 ms	48.671 ms
HP1.sensHAL33.P50	0.010 ms	0.356 ms	0.700 ms
'HP1.sensHAL33.P50.signal' waiting	0.000 ms	32.630 ms	50.471 ms
HP1.App_5.P50	0.010 ms	0.251 ms	0.500 ms
'HP1.App_5.P50.act0signal' waiting	0.690 ms	0.960 ms	1.180 ms
HP1.App_5.P50.act0.Hal	0.700 ms	0.941 ms	1.190 ms
'HP1.App_5.P50.act0.HAL.signal' waiting	3.310 ms	3.900 ms	4.591 ms
HP1.ECU15.DPactP5	0.020 ms	0.170 ms	0.320 ms
'HP1.App_5.P50.act0.DP.signal' waiting	19.940 ms	49.830 ms	49.981 ms
PduUdpTxBuffer	5.000 ms	5.000 ms	5.001 ms
HP1-> Switch1	0.620 ms	0.650 ms	0.678 ms
Switch1-> VIU2	0.285 ms	0.295 ms	0.312 ms
Gatewaying delay in 'VIU2'	0.000 ms	2.500 ms	5.000 ms
Tx over 'CAN15'	0.091 ms	0.092 ms	0.203 ms
'HP1.App_5.P50.act0.DP.signal' waiting	0.000 ms	24.702 ms	50.028 ms
ECU15.HP1.App_5.P50.act0	0.010 ms	0.030 ms	0.050 ms

[RTaW-Pegase screenshots]

End-to-End delays

4x or 10x sampling period

Name	Min	Average	Max	LatencyConstraint
ECU40.P_10.sens3->ECU33.HP1.App19.P10.act0*	18.493 ms	40.977 ms	59.258 ms	40 ms
HP1.App20.P1000->ECU22.HP1.App20.P1000.act0*	999.703 ms	1141.125 ms	1282.246 ms	4000 ms
ECU6.P_50.sens2->ECU1.HP1.App21.P50.act0	106.003 ms	169.006 ms	207.168 ms	500 ms
ECU14.P_50.sens4->ECU1.HP1.App21.P50.act0*	82.594 ms	159.388 ms	228.761 ms	200 ms
ECU26.P_50.sens1->ECU1.HP1.App21.P50.act0	83.439 ms	135.081 ms	194.531 ms	500 ms
ECU31.P_50.sens2->ECU1.HP1.App21.P50.act0*	81.139 ms	164.284 ms	229.592 ms	200 ms
ECU10.P_10.sens2->ECU24.HP1.App22.P10.act0	32.656 ms	55.191 ms	75.619 ms	100 ms
ECU13.P_10.sens6->ECU24.HP1.App22.P10.act0*	27.804 ms	49.407 ms	66.863 ms	40 ms
ECU27.P_50.sens3->ECU24.HP1.App22.P10.act0	38.017 ms	60.155 ms	79.727 ms	500 ms
ECU30.P_10.sens4->ECU24.HP1.App22.P10.act0	32.446 ms	54.230 ms	71.966 ms	100 ms
ECU9.P_50.sens1->ECU6.HP1.App23.P10.act0	41.766 ms	64.193 ms	81.945 ms	500 ms
ECU15.P_50.sens2->ECU6.HP1.App23.P10.act0	52.499 ms	74.189 ms	92.237 ms	500 ms
ECU27.P_50.sens1->ECU6.HP1.App23.P10.act0	36.500 ms	59.184 ms	79.732 ms	500 ms
ECU30.P_10.sens4->ECU6.HP1.App23.P10.act0*	30.902 ms	53.334 ms	71.631 ms	40 ms
ECU6.P_50.sens3->ECU7.HP1.App24.P10.act0	72.752 ms	93.905 ms	111.702 ms	500 ms

Periodic activations may be a solution only if the end-to-end latency constraints are several/many times larger than the sampling periods

Event Triggered (ET) activation + ODS for **critical** chains

Step	Activation Mechanism	Max. Waiting Times	Detrimental impact of triggering
Sensor Task	Periodic tasks	none	
CAN	Signals trigger CAN frame	none	higher bus and CPU loads
Ethernet	CAN frame trigger ETH frames	none	higher links and CPU loads
DP	Periodic tasks	period	
Service	Service Method call (ODS)	period	higher load and larger response times
App	Periodic tasks	none	
Service	Service Method call	none	
DP	Periodic tasks	period	
Ethernet	PDUs trigger frames	none	higher links loads
CAN	PDU triggers CAN frame	none	higher bus loads
Actuator Task	Periodic tasks	period	

Improvements in terms of waiting times

Periodic + PubSub *versus* ET + ODS

Triggered *Periodic*

TimingChain ECU8.P_50.sens1->ECU15.HP1.App_5.P50.act0*

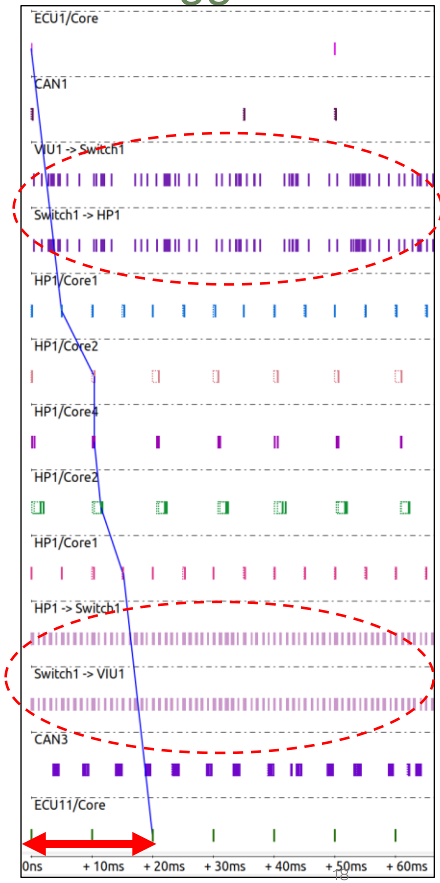
Segment	Min	Average	Max	
ECU8.P_50.sens1	0.010 ms	0.045 ms	0.080 ms	0.080 ms
'ECU8.P_50.sens1signal' waiting	0.000 ms	0.000 ms	0.000 ms	29.992 ms
Tx over 'CAN8'	0.087 ms	0.104 ms	0.298 ms	0.341 ms
Gatewaying delay in 'VIU1'	0.000 ms	2.500 ms	5.000 ms	5.000 ms
PduUdpTxBuffer	0.000 ms	0.000 ms	0.000 ms	5.000 ms
			0.918 ms	0.346 ms
			1.977 ms	0.553 ms
			5.409 ms	5.319 ms
			0.440 ms	0.360 ms
			47.961 ms	48.671 ms
			0.990 ms	0.700 ms
'H			0.000 ms	50.471 ms
			0.010 ms	0.500 ms
			0.970 ms	1.180 ms
			0.980 ms	1.190 ms
			1.341 ms	4.591 ms
			0.360 ms	0.320 ms
			0.000 ms	49.981 ms
PduUdpTxBuffer	0.000 ms	0.000 ms	0.000 ms	5.001 ms
HP1-> Switch1	0.067 ms	1.002 ms	2.884 ms	0.678 ms
Switch1-> VIU2	0.067 ms	0.075 ms	0.219 ms	0.312 ms
Gatewaying delay in 'VIU2'	0.000 ms	2.500 ms	5.000 ms	5.000 ms
Tx over 'CAN15'	0.091 ms	0.092 ms	0.285 ms	0.203 ms
'HP1.App_5.P50.act0.DP.signal' waiting	0.000 ms	25.194 ms	50.018 ms	50.028 ms
ECU15.HP1.App_5.P50.act0	0.010 ms	0.030 ms	0.050 ms	0.050 ms

Waiting times disappear with triggered activations and on-demand service requests … at the expense of CPU and network load

[RTaW-Pegase screenshots]

Periodic

Triggered

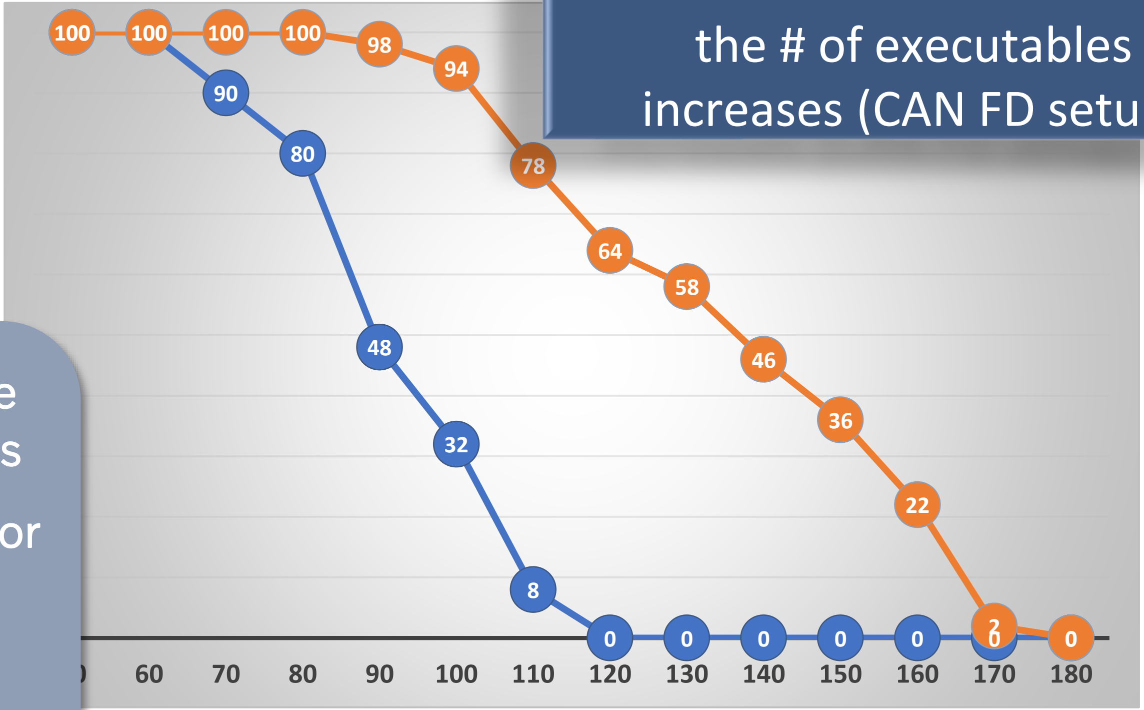


Mixed model is the best option in our case-study

- Periodic + PubSub for non-critical chains
(75%, D=10x sampling period)
- ET + ODS for critical chains
(25%, D=4x sampling period)

ET + ODS for all timing chains

Probability to meet timing & load constraints when the # of executables increases (CAN FD setup)



# of executables

Each executable involved in 6.5 timing chains on avg

- Fully periodic model not shown here as unable to meet timing constraints
- Load, not timing, is the limiting factor for ET+ODS model
- T1S outperforms CAN FD here but dependent on setup and parameters

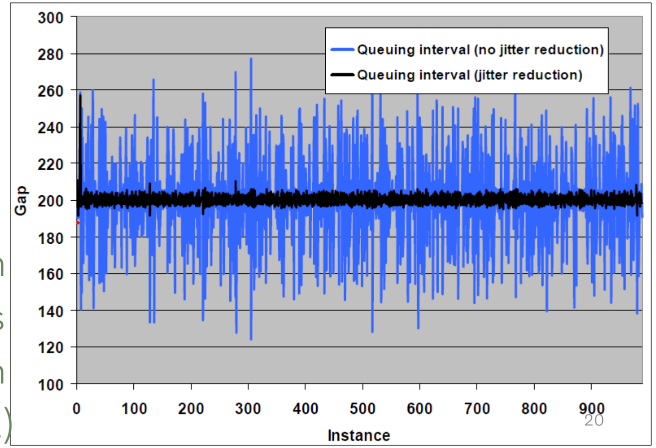
Takeaways

Presented technical solutions explored by Volvo to integrate signal-based ECUs in a zone-based SOA

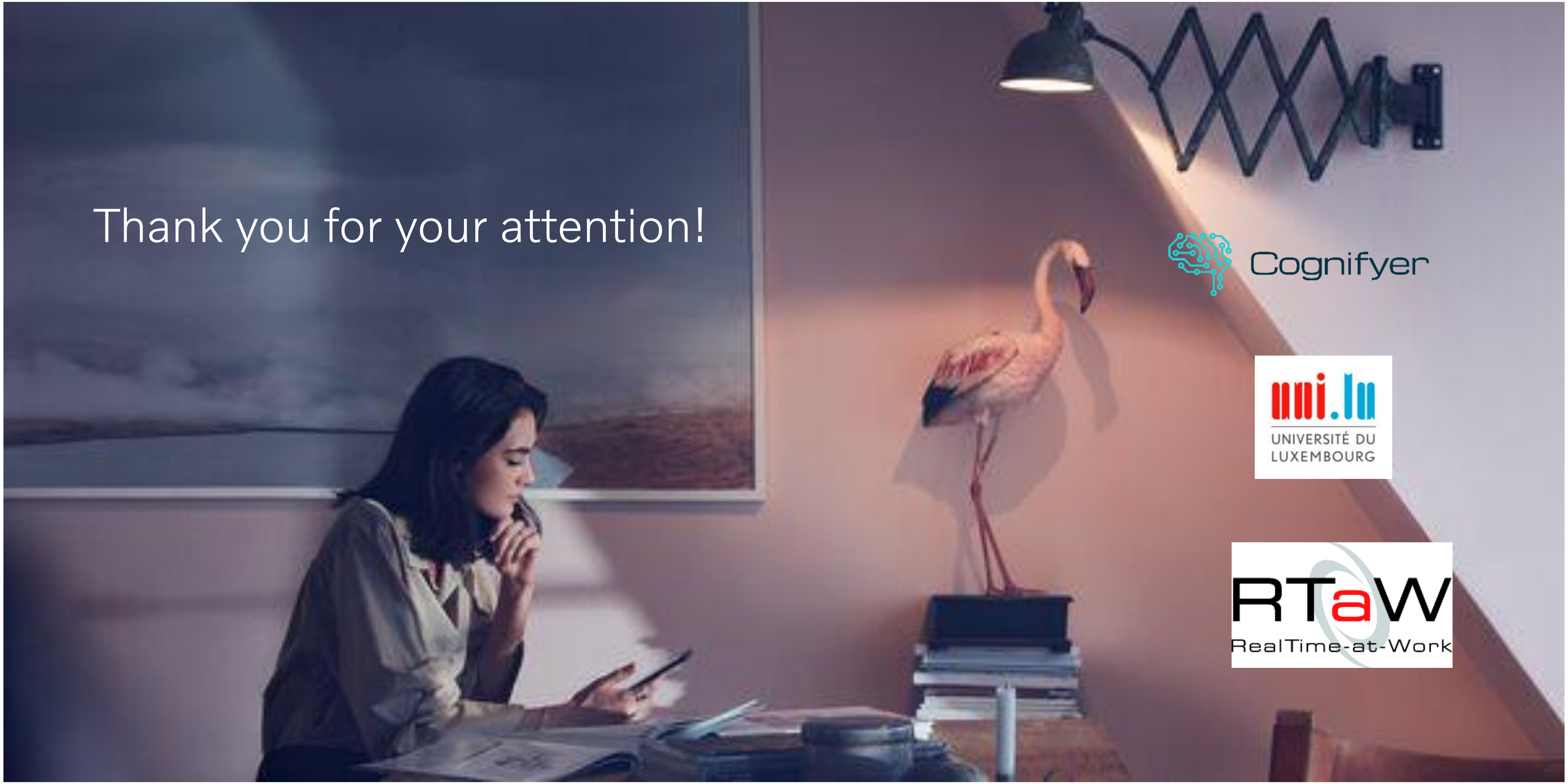
Choice of activation patterns is key to meeting load and timing requirements: mix of both periodic and ET offers best solution → automation through DSE algorithms

ET activation creates coupling between segments as jitters/bursts are propagated → unwanted coupling between sub-networks, between network and SW development!

Local jitter-reduction shaping strategies may allow to preserve independence



Time between forwarding CAN frames with/wo jitter reduction (Davis & Navet, 2012)


Thank you for your attention!
Cognifyer
uni.lu
UNIVERSITÉ DU LUXEMBOURG
RTaW
RealTime-at-Work