Real Life Low Latency applications in Software Centralized Automotive Ethernet Architectures

Atilla Mete Turedi (JLR) &
Eduardo Santos Navarro (Analog Devices)

Summary

Software Defined Vehicle vs Domain Based Architecture

RCP & Software in Central Compute Unit

Suspension Stability Use Case

Suspension Stability Demonstrator

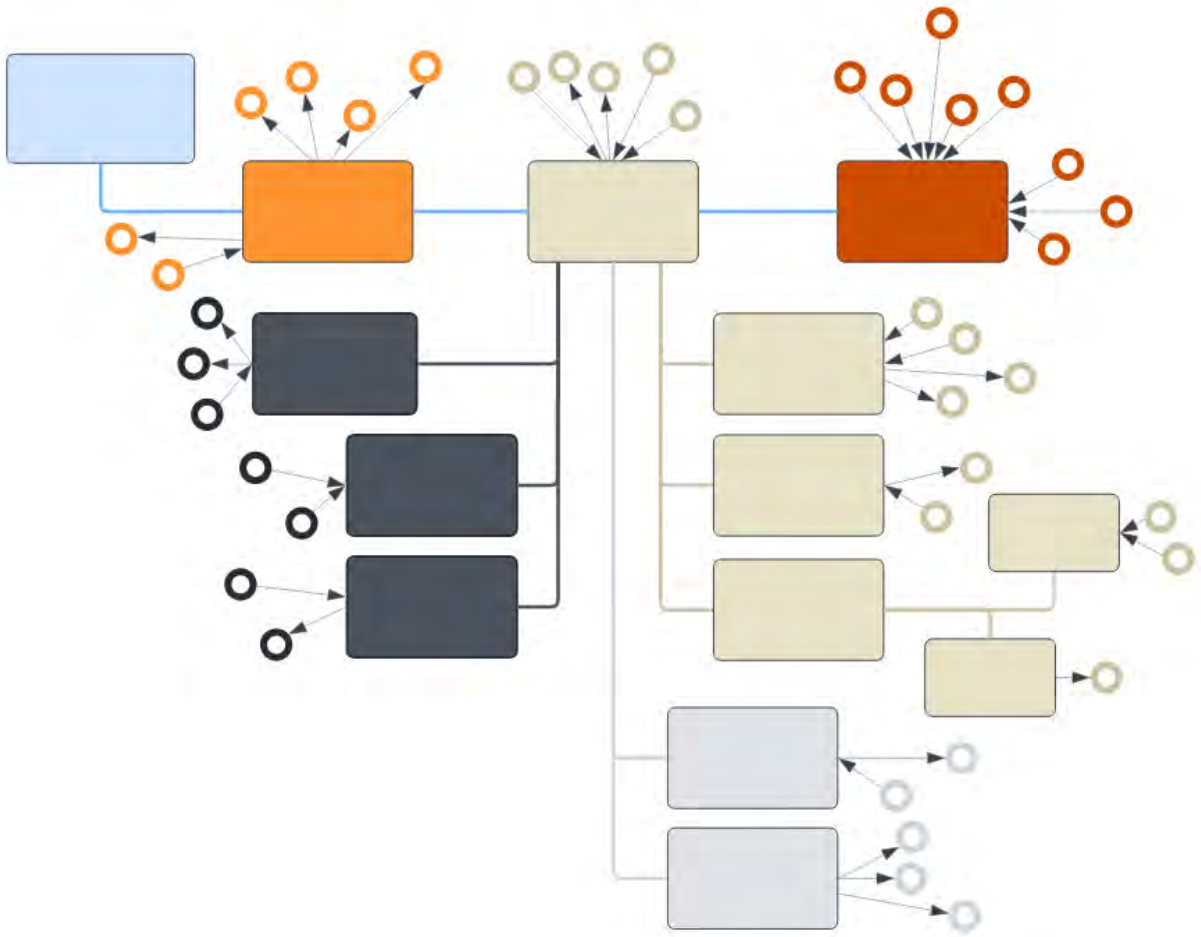
Nodes Synchronization

Effect of Extra Nodes in the Network

Software Defined Vehicle

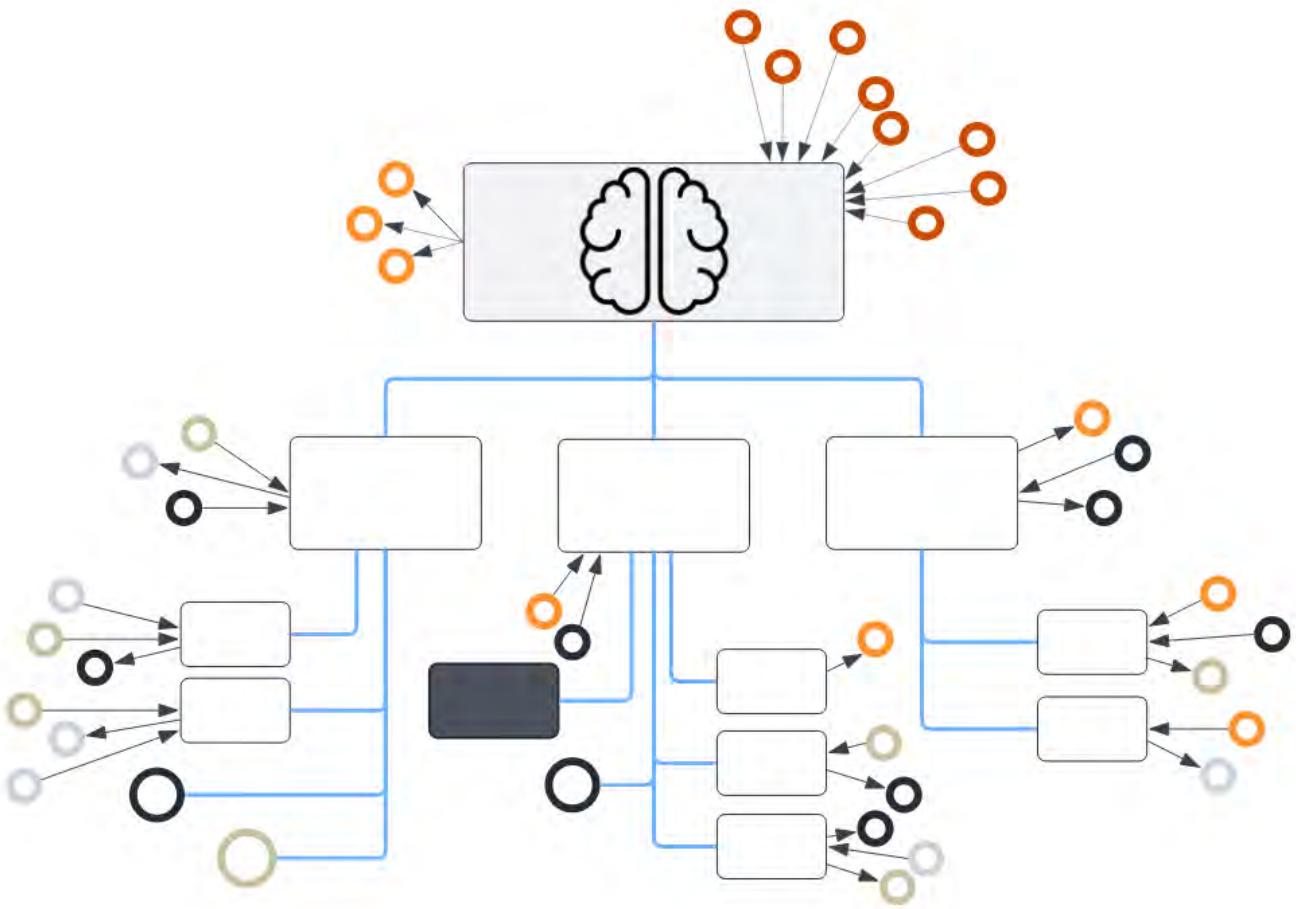
and the role of hardware

Domain Based Architecture



- Domain responsibility with supplier, easy to contain
- Abstracted and locked away data
- Difficult to introduce and improve features
- Security is a challenge
- Limited offboard utilisation

Zonal & SDV Architecture



- Increased OEM responsibility, organisational challenges and lack of system boundaries
- Data freedom
- Easy to deploy new features and improvements
- Ethernet has well-known security solutions
- Coherent architecture including offboard

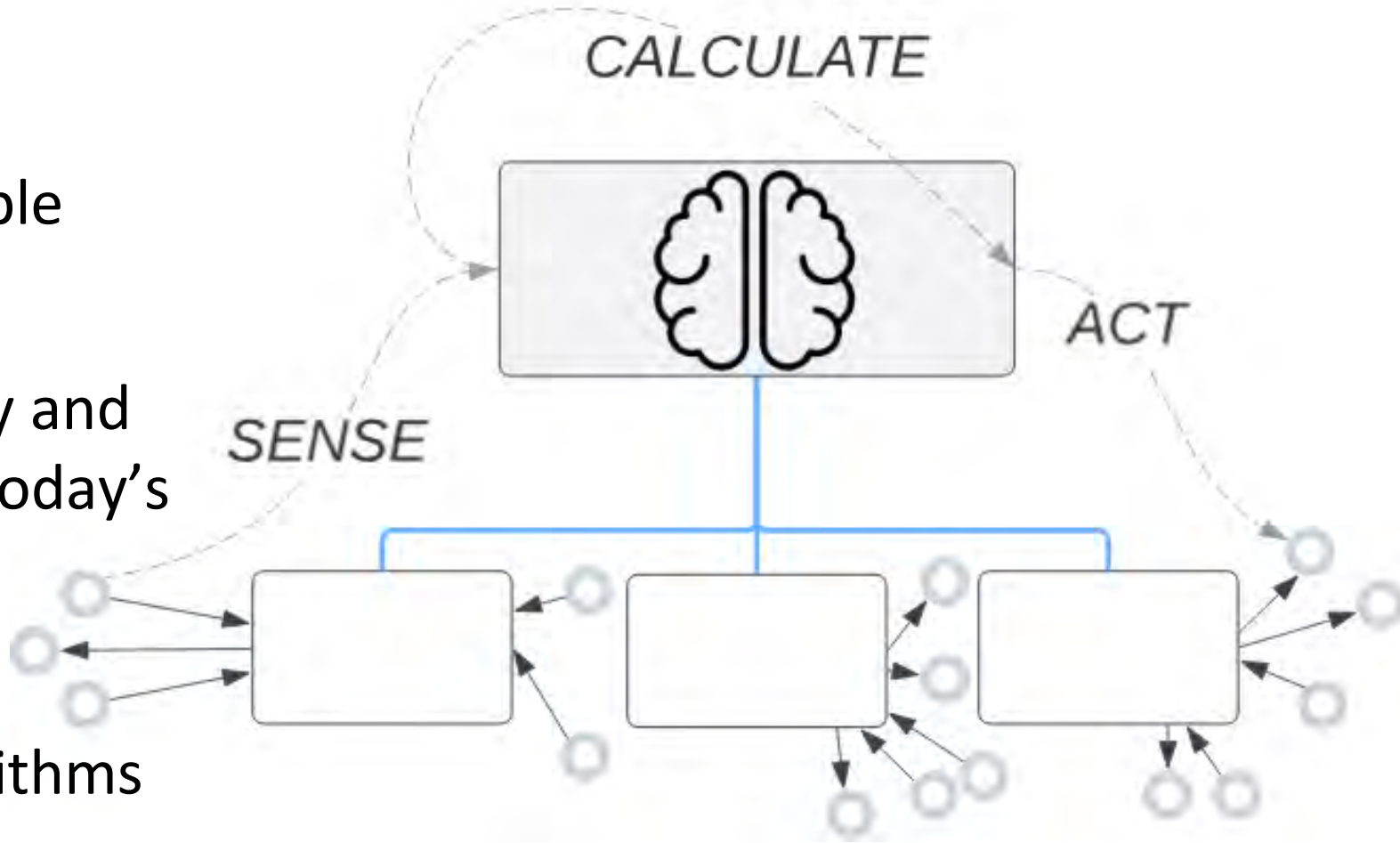
Zonal & SDV Architecture

Tight control loops

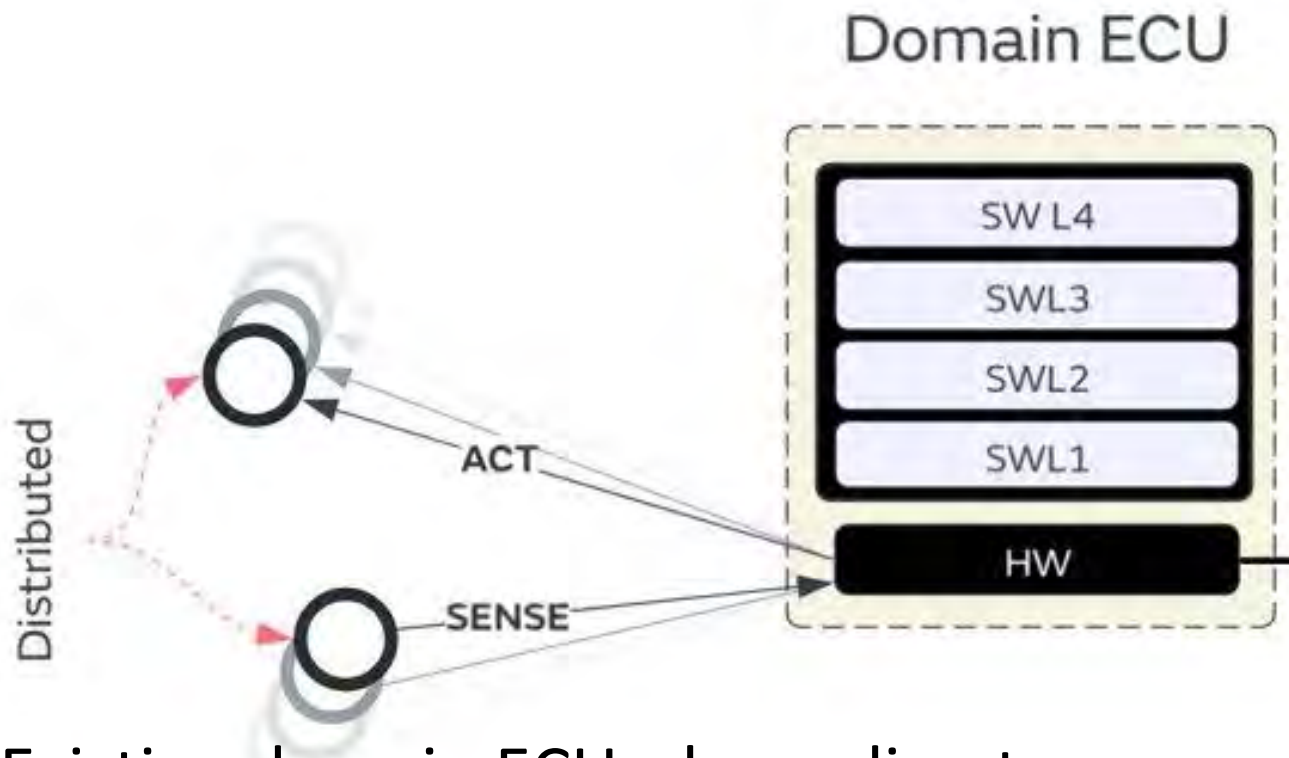
Delays could cause noticeable phase lags

Minimum is to meet latency and synchronicity capability of today's systems

Beyond, headroom allow introducing advanced algorithms to enhance features



Domain Based SW/HW

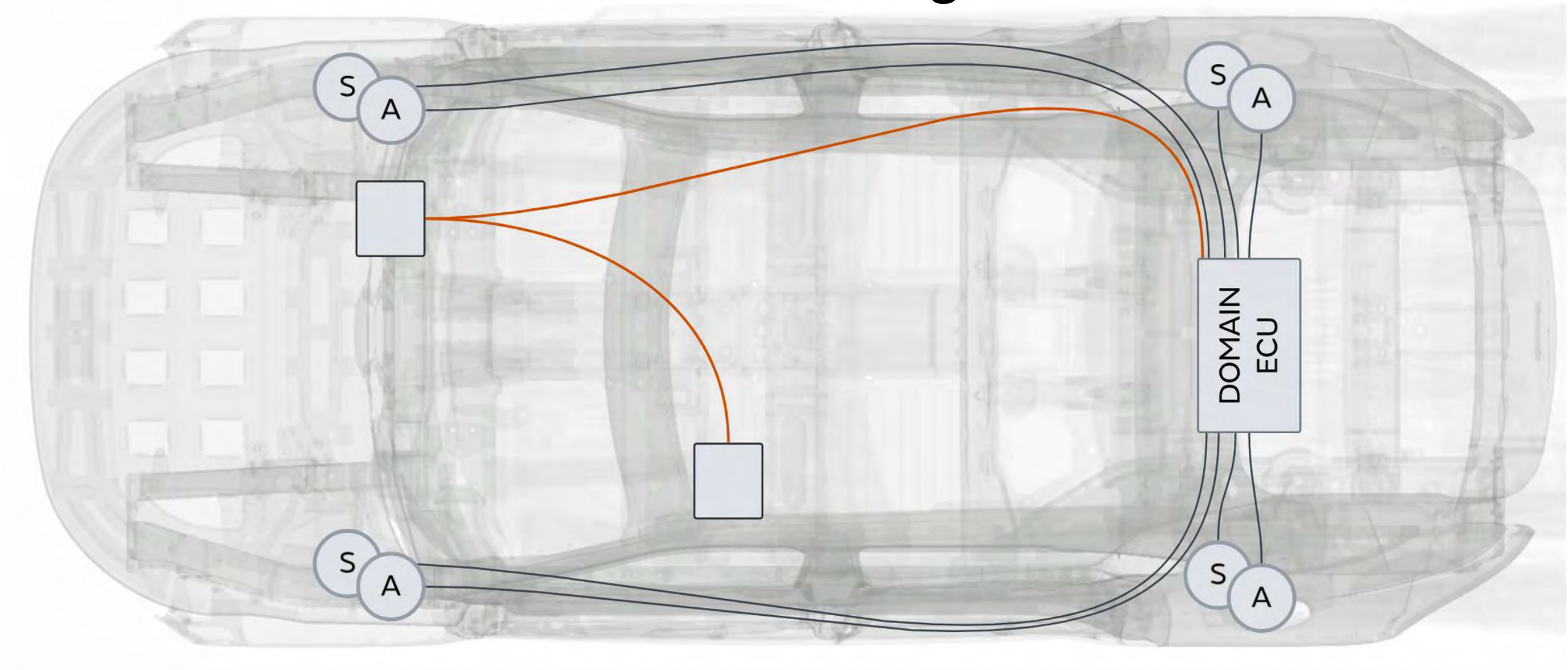


Only abstracted data shared externally

Receive over the network vehicle level signals

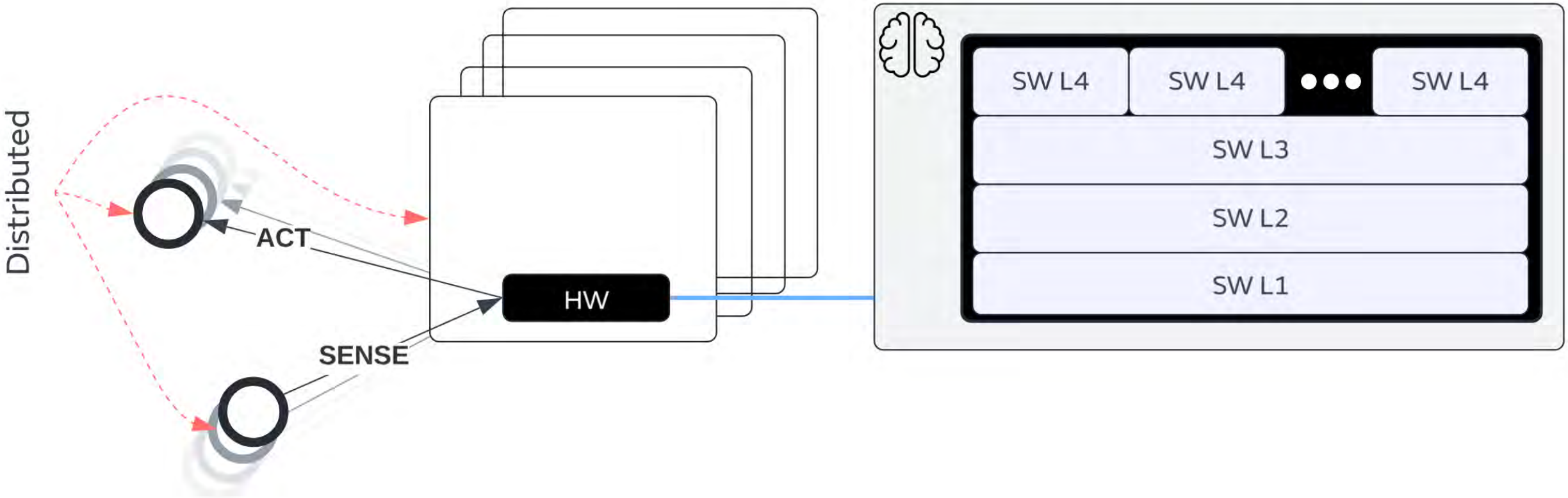
Existing domain ECUs have direct access to all the sensors and actuators they need

Domain Based ECUs and Wiring

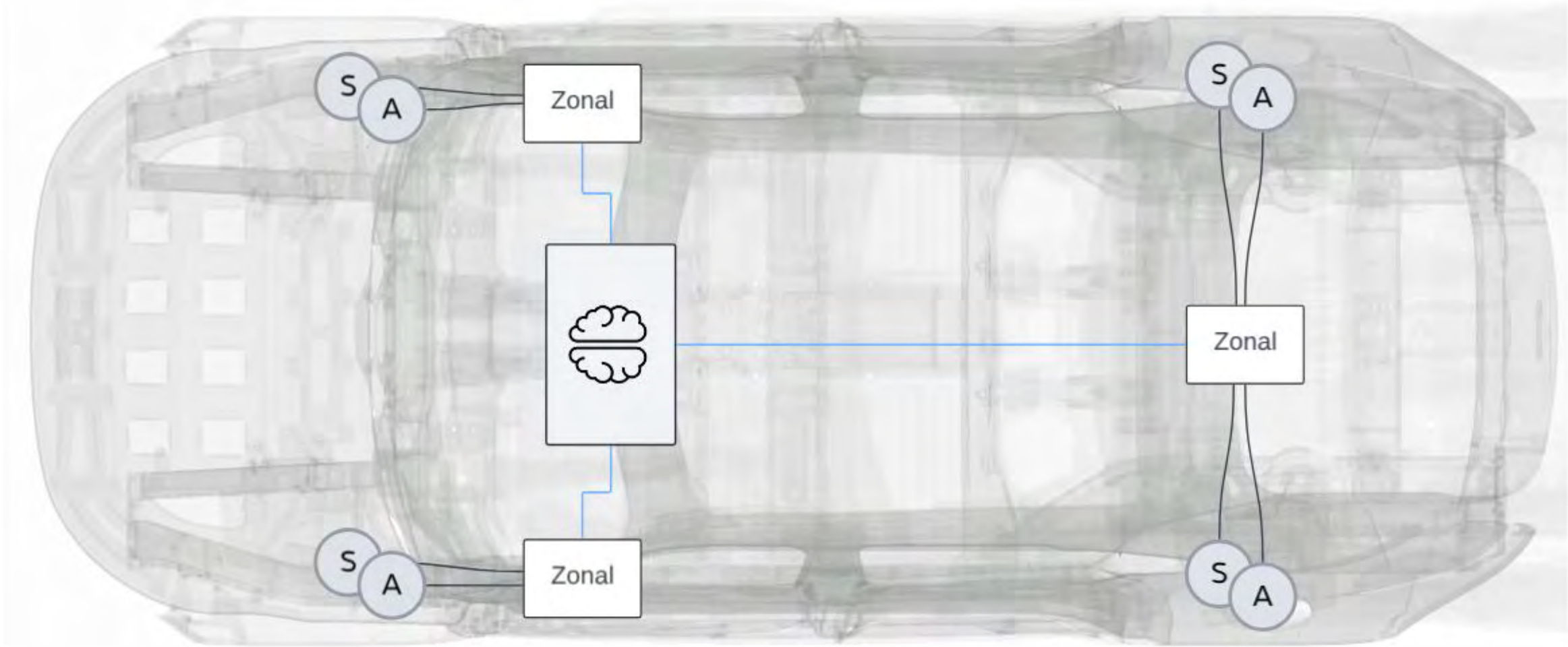


Remote IO Concept

Hardware based remote IO concept enables; lower latency, software reusability, less software in zonals, access to raw IO



Zonal Modules and Wiring



Suspension Use-case

Centralising software makes it easier to improve and deliver new features, eliminates hardware and reduces harness.

Suspension is a good challenge to focus;

- Highly distributed sensors and actuators,
- Tight control loops,
- Constantly evolving capability with increasing demands.

Minimizing software outside compute is critical;

- Less software to maintain,
- Reduced latency.

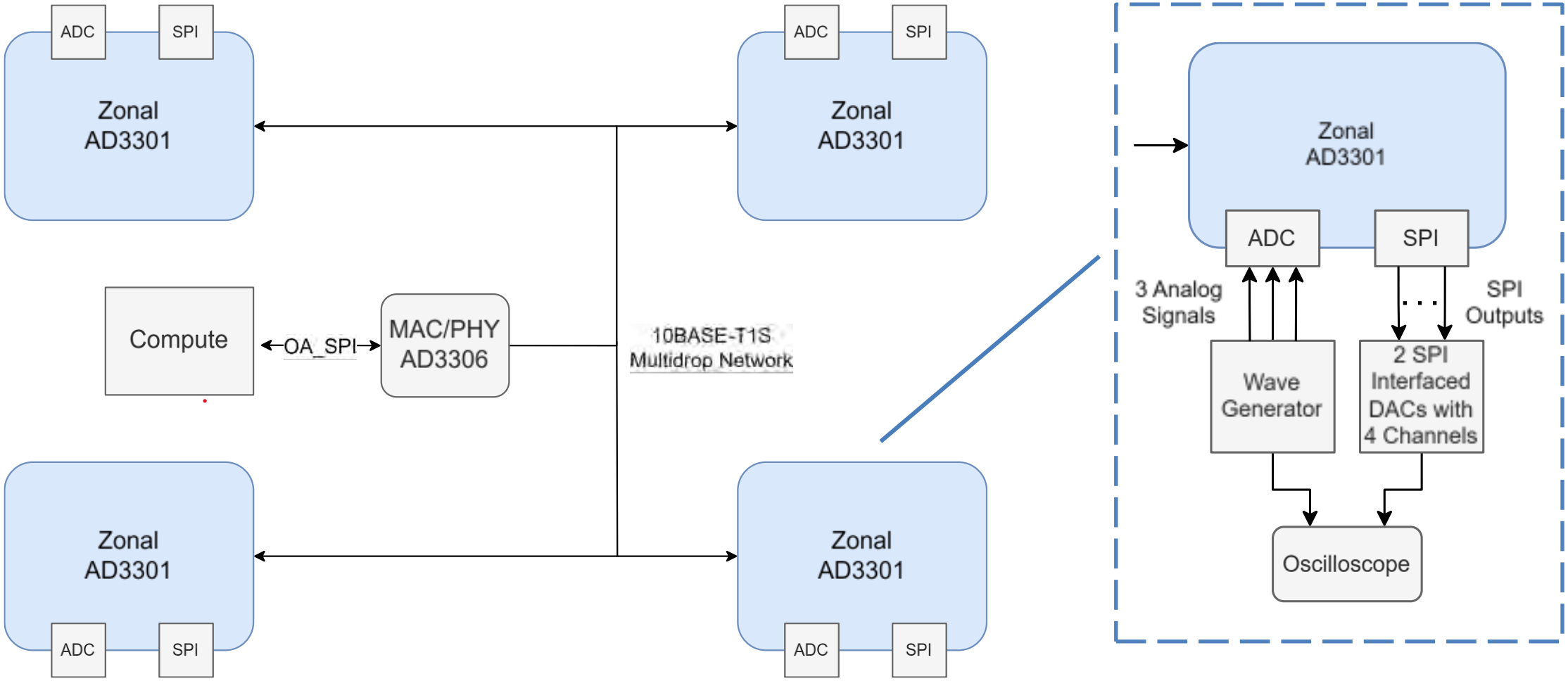
Suspension Proof of Concept

Definition and Results

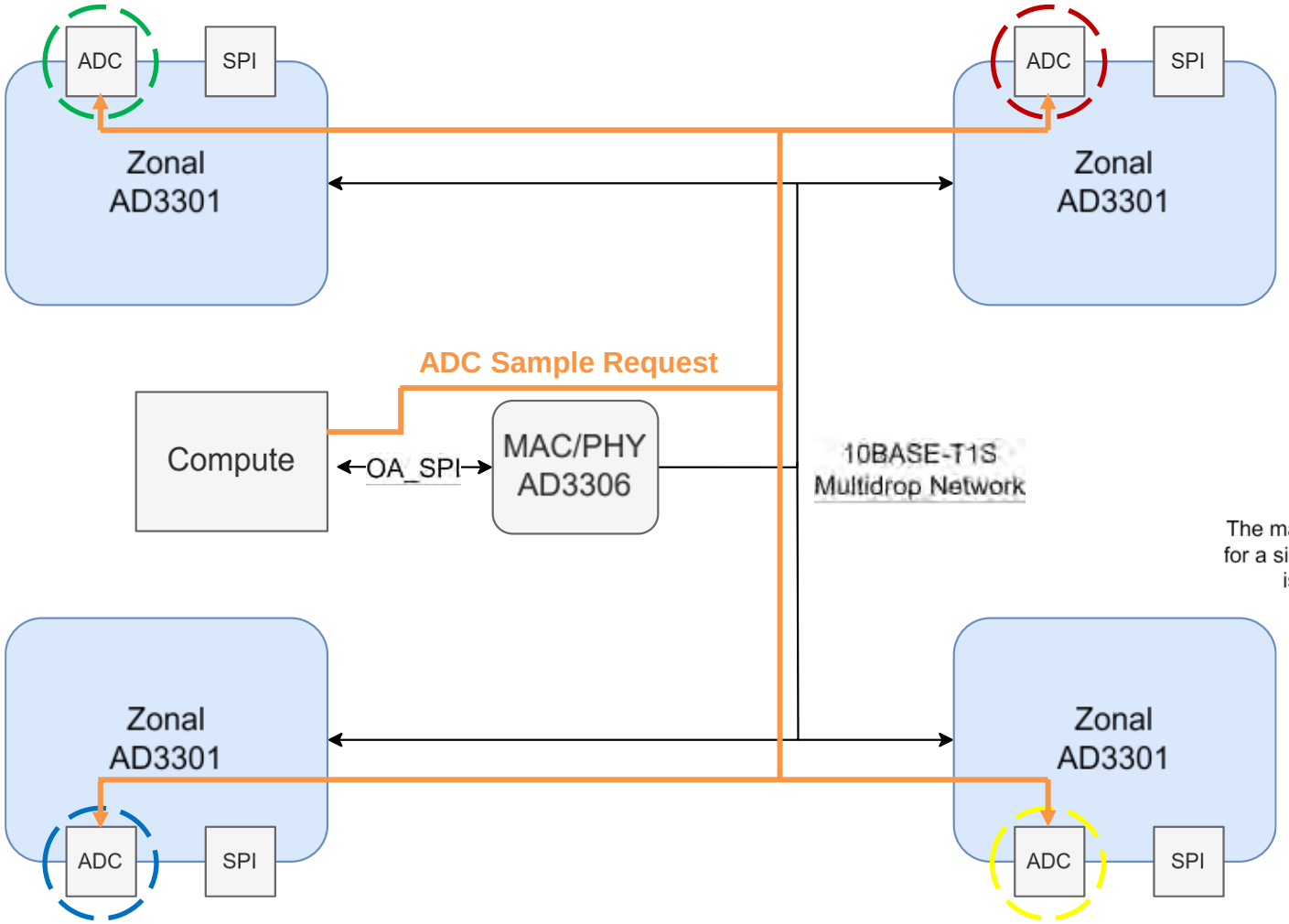
Proof of Concept Scope

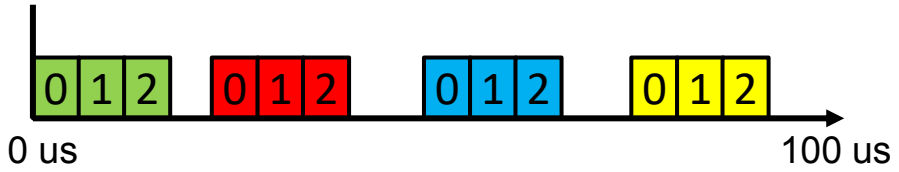
- Demonstrate how Low Latency Control Loops can be achieved when centralizing software and all hardware edge node implemented using the E^2B Remote Control Protocol (RCP) within a 10BASE-T1S network
  - Provide an in-depth timing analysis
    - Detailed timing of all latency contributors to facilitate better network timing estimations for this and other use-cases.
  - Provide worst case closed loop delay to identify time available to allocate for computation in the Compute Unit

Suspension Proof of Concept: Definition

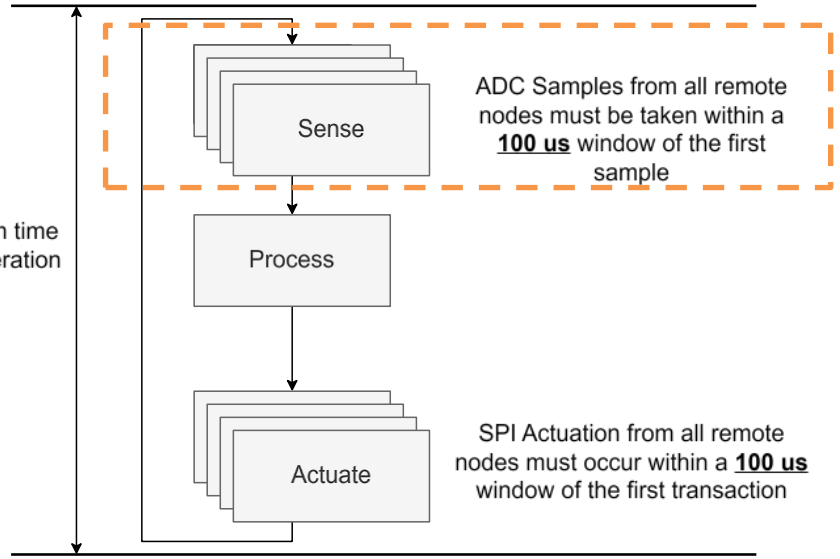


Suspension Proof of Concept: Requirements

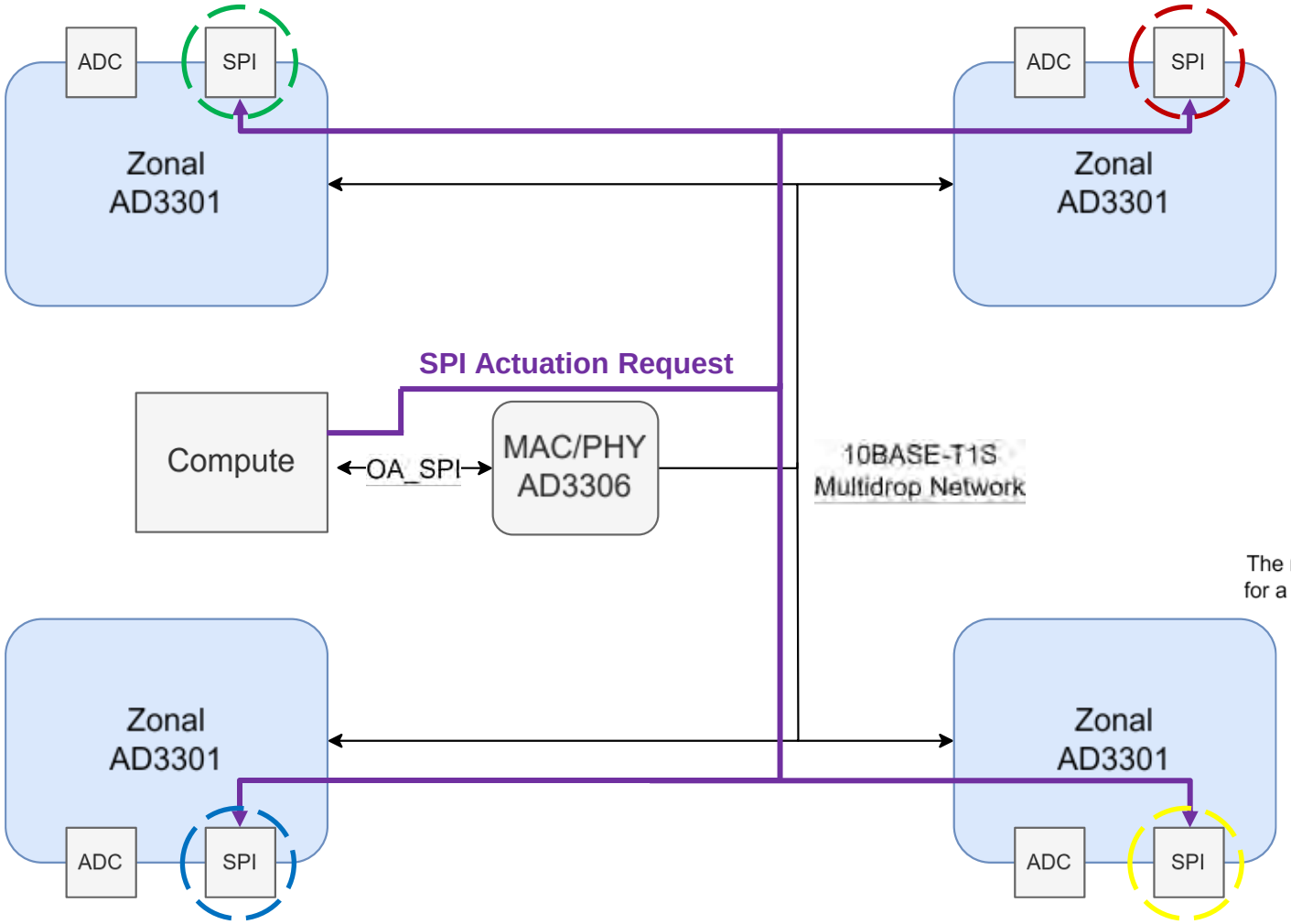


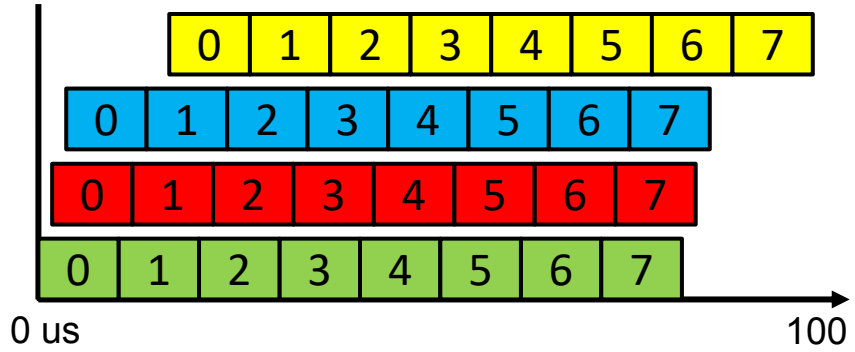


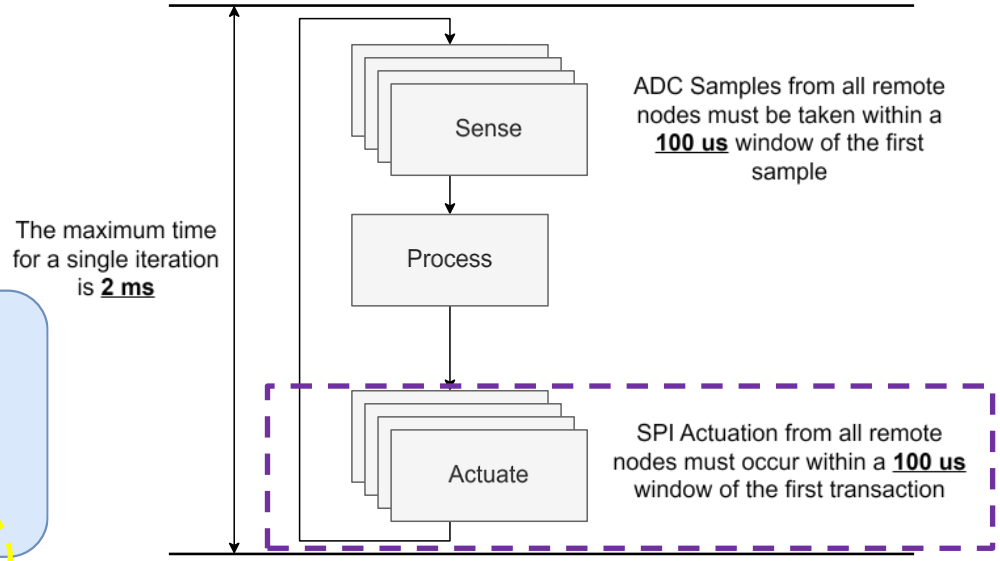
The maximum time for a single iteration is **2 ms**



Suspension Proof of Concept: Requirements

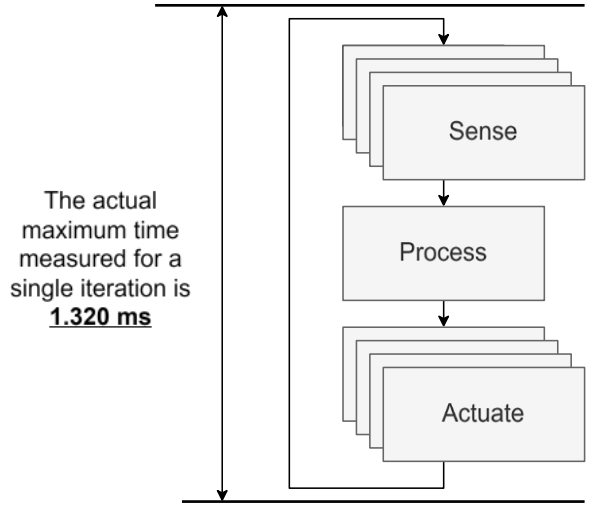






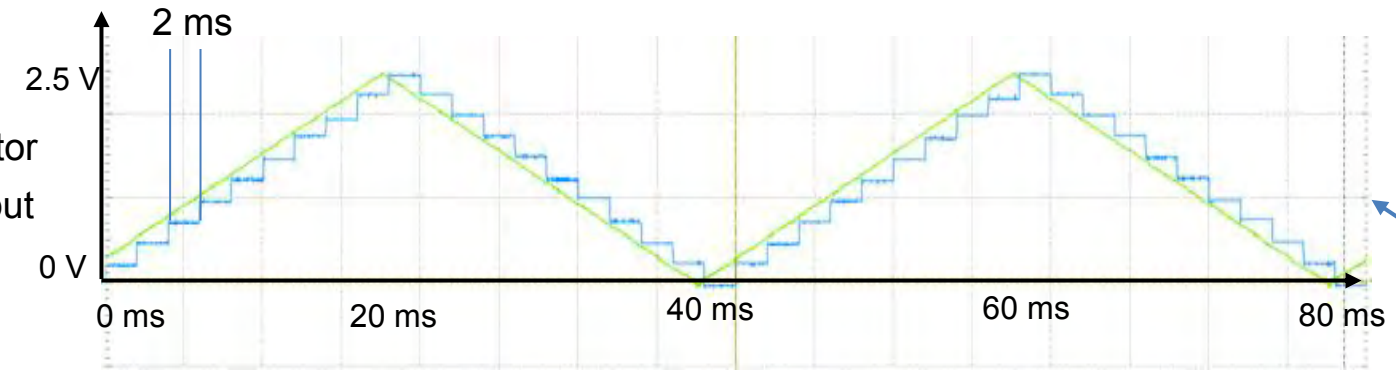
Suspension Proof of Concept : Results

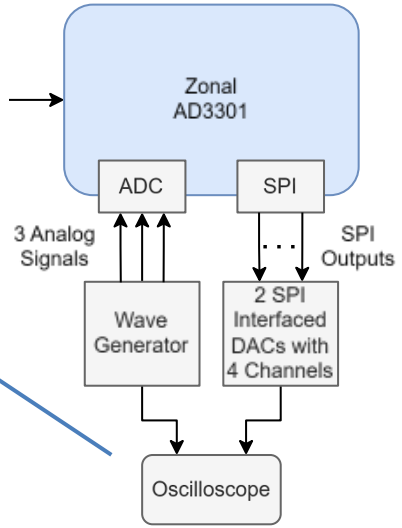
- Satisfied the control loop iteration time requirement of maximum 2 ms.
  - Based on meeting current capability.
  - Remaining time available for computation: 680 us



- Wave generator signal re-generated at DAC output with an actuation period of 2 ms

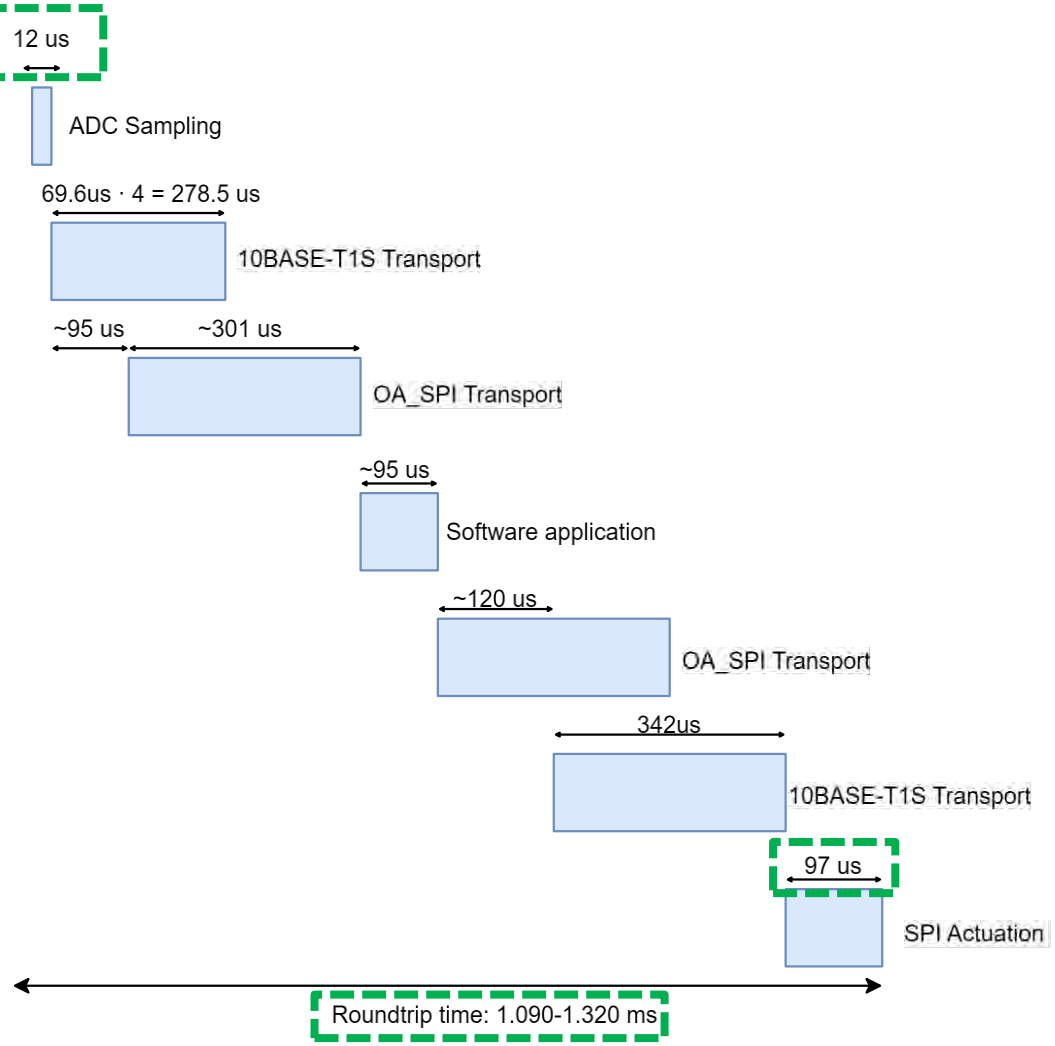
Green – Wave Generator
Blue – DAC Output





Suspension Proof of Concept: Timing Analysis

- Synchronization requirement met of 100 us window for sense and actuation
- OA_SPI Transport time depends on the Central Compute interrupt attendance
- Software application time is consumed by:
  - ADC sample processing
  - SPI Actuation frame generation



Proof of Concept: Closer to a real use-case

- At this point we have only considered where all suspension nodes on a single network addressed using multicast frames

- In a real implementation, suspension sensors and actuators will be located on separate networks within the vehicle
  - Switch latencies must be considered
  - Other data sources may impact latency by interfering either at the zonal or at the ethernet backbone
  - Therefore:
    - Synchronization between networks is required
    - Actuation frame size must be considered

Node Synchronization with Multicast frames

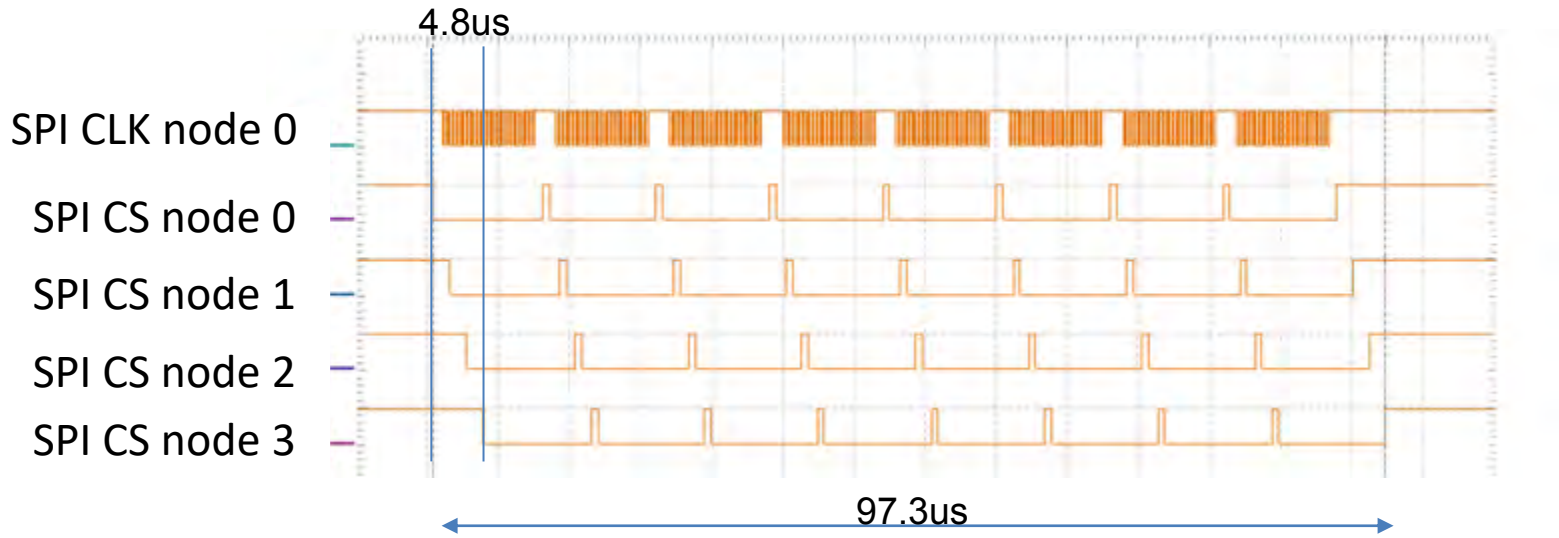
Synchronization of Nodes within the same network

- Multicast frame contains information for all four nodes
  - As each node decodes the frame, the actuation/sampling will start
    - The time delta between actuation/sampling start determined by the hardware
- Pros and Cons
  - 👍 Efficient frame payload with multicast format
  - 👍 Time between start of actuation/sampling across four remote nodes: 4.8 us
  - 👎 Cannot be implemented to sync nodes in different networks

Node Synchronization – Multicast: Results

Synchronization of Nodes in the same network

- Delta between start of SPI Transactions : **1.6 us**
- Total delta between SPI starts: **4.8 us**



Node Synchronization with gPTP

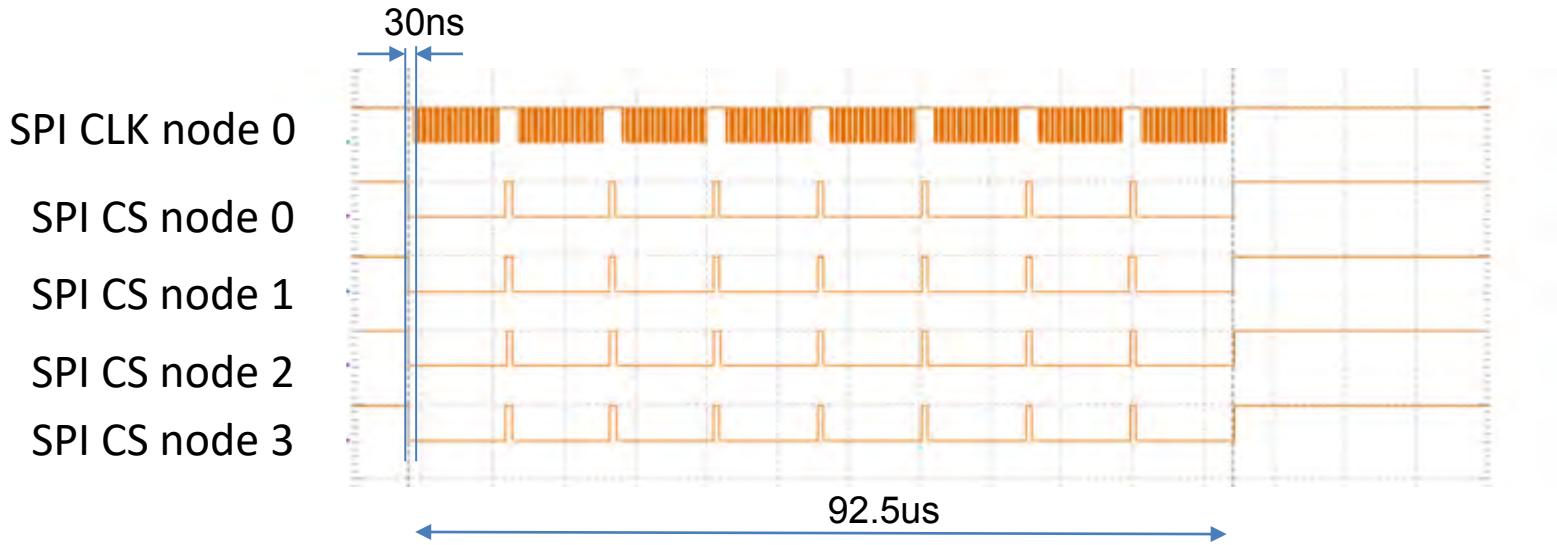
Synchronization of Nodes between different 10BASE-T1S networks

- Using Generalized Precision Time Protocol (gPTP) timestamped actuation
  - The gPTP grandmaster resides in compute unit
  - All nodes in all networks have the possibility to be synced to the gPTP grandmaster timestamp
  - All Interfaces can actuate at a selected gPTP time
- Pros and Cons
  - 👍 Time between start of actuation/sampling in four remote nodes: 0 - 30 ns
  - 👍 Can be implemented to sync nodes in different networks
  - 👎 Requires to add the actuation/sampling time in the packet. Adds extra network complexity.

Node Synchronization – gPTP: Results

Synchronization of Nodes between networks with gPTP

- Delta between start of SPI Transactions: **0 - 10 ns**
- Total delta between SPI starts: **0 - 30 ns**
- This is an improvement over using multicast frames



Effect of Extra Nodes in the Network

10BASE-T1S has a **bounded maximum latency**

- Worst case latency occurs when the node wants to transmit but has just missed its Transmit Opportunity slot
  - A full Physical Layer Collision Avoidance (PLCA) cycle must be completed before the node gets its next Transmit Opportunity (TO)
  - Other nodes transmit their maximum data frame

- Latency from extra nodes can cause difficulties with the low latency control loop timing requirements.
  - Mitigations strategies are available to reduce this effect
    - **Frame size optimization:** the maximum frame size that can be sent in each node is selected. When the payload reaches that maximum frame size it will be automatically packetized and transmitted when the TO permits it

Conclusions

- Architecture exists to deliver customer features, but demanding features continue to increase complexity
- Focus should be in making it easy to deliver the most advanced features rapidly
- JLR and ADI have demonstrated how a challenging low latency use case can be implemented using 10BASE-T1S and E^2B technology

	Requirement	POC Result
Full sense / actuation cycle	2 ms	1.3 ms
Sense synchronization	100 us	12 us
Actuation synchronization	100 us	92.5 us

Thank You

Atilla Mete Turedi (JLR) &
Eduardo Santos Navarro (Analog Devices)