Index	Paper	EC1: Which functional domain(s) does the study analyze and/or modify in relation to SW architecture changes? E.g. ADAS, IVI (In-Vehicle Infotainment), Powertrain, Chassis...	EC2: Which system limitation(s) does the study identify as drivers for SW architecture changes? E.g. Busload, computing power, development costs, development time...	EC3: Which specific technologies does the study identify as enablers or catalysts for changes in the SW architecture? E.g. High-Performance Computing in autmotive, AI & machine learning, electrification, Over-The-Air updates and continuous deployment, connectivity - V2X and 5G...	EC4: How does the study technically address the integration of diverse software requirements (real-time, non-real-time, safety-critical, etc.) within a centralized automotive software architecture? E.g. virtualization via hypervisors, containerization...	EC5: Which architectural patterns or design practices are proposed to systematically support mixed-criticality in centralized automotive software architectures? E.g. Mixed OS environments, Service-Oriented Architectures (SOA), mixed-criticality scheduling, safety island / redundant compute...	Comment
1	Development of vehicle domain controller based on ethernet	ADAS;	Cross-domain communication and vehicular communication that led to backbone ethernet	Electrification, intelligence, network connection and sharing	-	-	DEC1 - Agreement; DEC2 - Agreement after consolidation; DEC3 - Agreement after discussion; DEC4 - Direct agreement; DEC5 - Direct agreement;
2	Contradiction of separation through virtualization and inter virtual machine communication in automotive scenarios	IC - Cabin; IVI - Cabin;	High number of ECUs --> Increases weight, energy, space, costs for expensive hardware components, development and engineerings costs and time; unfeasible function integration due to missing OSs;	Fully Digital Cluster Instruments (FPKs)	Virtualization via type-one/baremetall hypervisors, Virtual Machines (VMs), multi-core, isolated communication channels for shared resources; access control mechanisms for VMs to write/access shared resources; communication data signing; data consistency check;	Multi-OS environment; Design patterns for VMs: Safety relevant parts are separated on VMs to not get compromised; Different OSs to benefit of applications specific to particular OSs 'Limiting hardware resources for a VM' Dedicated hardware resources to isolate VM; Decoupling of development cycles - simplified updates; Architectural approaches - see figures: 1. Clear separation approach - two or more OSs/VMs run without interconnections and dedicated I/Os; 2. Layers of interconnections - various types of interconnections: unidirectional, bi-, App to OS, OS to OS, App to App etc.; 3. Minimalistic Approach - Shared memory partition for each VM, only owner is allowed to write, only trusted OS is allowed to access --> no manipulation by other VMs, one way communication (fire and forget)	DEC1 - Direct agreement; DEC2 - Agreement after consolidation; DEC3 - Direct agreement; DEC4 - Direct agreement; DEC5 - Direct agreement;
3	Autonomy-driven Emerging Directions in Software-defined Vehicles	SDV; Focus on autonomous features - autonomous;	"Embedded HW and SW architectures as bottlenecks - e.g. computing power; Growing demand of software - risk for real-time and FuSa; Zone-based: Delays between sensing and actuation; Service-oriented communication leads to uncertainties; --> 'Pessimistic timing estimates and inefficient implementations' --> Timing analysis challenges - WCET	serivce-oriented communication in SDVs - SOME/IP, DDS	Containerization	SOAFEE SW architecture for SDV;	DEC1 - Direct agreement; DEC2 - Direct agreement; DEC3 - Direct agreement; DEC4 - Direct agreement; DEC5 - Agreement after discussion;
4	Time-sensitive autonomous architectures	Autonomous;	Complexity, computing power, safety/latency, busload/bandwidth, flexibility, scalability, development effort, growing number of ECUs;	High connectivity; Time-Sensitive Networking (TSN); Autonomous driving; V2X; ethernet-based communication; Deep Learning;	VTSN - Virtual TSN as virtual switch as one VM/VS; Jailhouse hypervisors to enable virtual machines; Jailhouse includes IVSHMEM as vPCI with permissions to ensure safety and security; Xen hypervisor with null scheduler to pin each virtual CPU to a physical CPU - one core for each virtual machine; Jailhouse outperformed Xen; TSN - Quality of Service; MPSoC Xilinx Zynq UltraScale+ (A- and R-Cores + FPGA accelerator); VLAN to ensure FFI; TSN scheduler Qav (frame prio, idleSlope etc.) and Qbv - Qbv outperforms Qav; Linux (performance, detection) and Erika Enterprise v3 (actuation) as open-source AUTOSAR RTOS; ROS2: DDS, Quality of Service (QoS), highly scalable; Isolation of detection and actuation; Actuation in single-core VM --> enhanced safety and predictability;	TSAA - Time-Sensitive Autonomous Architectures; Service-Oriented Architecture (SOA) --> Enables SDV; FPGA-based switches to enable TSN; Virtualization with advantages related to 'security, cost, reliability, availability, and adaptability'; Paravirtualization (higher performance) vs. full virtualization (all HW emulated); 'time-sharing mechanisms for shared hardware resources'; Freedom from Interferences; Multi OS environment; Temporal and spatial isolation; Domain/Feature/Function isolation; Static partitioning - cells allow 'guaranteed resource access and predictable performance'; TSN scheduling;	DEC1 - Agreement after discussions; DEC2 - Agreement after consolidation; DEC3 - Agreement after consolidation; DEC4 - Direct agreement; DEC5 - Agreement after consolidation;
5	An Enhanced Algorithm for Memory Systematic Faults Detection in Multicore Architectures Suitable for Mixed-Critical Automotive Applications	Autonomous;	functional safety/data integrity; computation power;	HPC; autonomous; multicore processors;	"Spatial memory protection; Multi-layered-cache multi-core --> Dedicated caches; SW algorithms to ensure memory data integrity; Multiple copies, double inverse redundant storage, CRC etc."	Stack-monitoring mechanisms;	DEC1 - Agreement after discussion; DEC2 - Agreement after consolidation; DEC3 - Agreement after discussions; DEC4 - Direct agreement; DEC5 - Agreement after discussion;
6	Software Architecture Modeling of AUTOSAR-Based Multi-Core Mixed-Critical Electric Powertrain Controller	Powertrain;	Increase in efficiency; increasing complexity of software implementations;	Electrification;	-	Implementation of watchdog;	DEC1 - Direct agreement; DEC2 - Agreement after consolidation; DEC3 - Agreement after discussion; DEC4 - Agreement after discussion; DEC5 - Agreement after discussion;
7	Problems and their mitigation in system and software architecting	-	SW complexity;	-	-	-	DEC1 - Agreement after discussion; DEC2 - Agreement after discussion; DEC3 - Direct agreement; DEC4 - Direct agreement; DEC5 - Direct agreement;

8	Design of Criticality-Aware Scheduling for Advanced Driver Assistance Systems	ADAS;	Change of criticality level based on dynamic changes in environment; size, weight, and power constraints; scalability and fault tolerance;	Autonomous driving	Different partitions for different criticality levels	Multi-core partitioned architecture, virtualization using hypervisors, from temporal and spatial isolation to mode changes --> mode based dynamic core allocation;	DEC1 - Direct agreement; DEC2 - Agreement after consolidation; DEC3 - Agreement after discussion; DEC4 - Agreement after alignment; DEC5 - Agreement fter consolidation;
9	A Security Process for the Automotive Service-Oriented Software Architecture	Autonomous; Powertrain;	Cyber-Security; Safety;	Autonomous driving, electric driving	-	ASOA; task scheduling acc. to criticality;	DEC1 - Agreement after discussion; DEC2 - Agreement after discussion; DEC3 - Agreement after discussion; DEC4 - Direct agreement; DEC5 - Direct agreement;
10	Towards the deployment of a centralized ICT architecture in the automotive domain	Driving performance - Powertrain; Comfort - Cabin & Body; passive and active safety - Cabin & Chassis; drive-by-wire - Chassis & Powertrain;	Increasing number of heterogeneous, distributed ECUs; Increasing system complexity; Demand for computing power and data bandwidth; Heterogeneous networks; lack of fail-operational behavior; increasing demand for interconnection, complex system verification; limited flexibility;	Drive-by-wire; autonomous driving assistants; electrification;	-	Redundancy of centralized HPCs; Data-centric paradigm - Consideration of functional and non-functional requirements ensures that runtime system fulfills QoS demands; Centralized ICT architecture will pay off on the long run (lowered barrier for market entrance when standardized SW architecture etc., resource saving, faster development cycles, personalization);	DEC1 - Agreement after discussion; DEC2 - Agreement after consolidation; DEC3 - Agreement after discussion; DEC4 - Direct agreement; DEC5 - Agreement after discussion;
11	A Modular Five-Layered V-Shaped Architecture for Autonomous Vehicles	Autonomous;	Safety - Robustness to unexpected situations, Comfort	Autonomous	-	Safety-critical partitions to be designed redundantly with self-diagnostic capability to hot-switch;	DEC1 - Agreement after alignment; DEC2 - Agreement after consolidation; DEC3 - Agreement after alignment; DEC4 - Direct agreement; DEC5 - Agreement after discussion;
12	Autonomous driving systems hardware and software architecture exploration: optimizing latency and cost under safety constraints	Autonomous;	Latency, safety, cost	Processor technology; computer vision; object recognition; deep learning; cross-domain dependencies;	Usage of multiple processors; ASIL related bundling of functionalities to dedicated uPs to not interfere; Mapping of similar SWCs onto same SoC partition;	Degeneration of tasks (ensure redundancy and fault-tolerance);	DEC1 - Agreement after discussion; DEC2 - Agreement after discussion; DEC3 - Agreement after discussion; DEC4 - Agreement after discussion; DEC5 - Agreement after discussion;
13	MPSoC-Based Platform for FailOperational Control of an Automated Research Vehicle	Autonomous; Powertrain;	Safety	Autonomous, electrification	RPU (real-time processing unit): RTOS for most safety and time critical application; API (application processing unit): Embedded Linux with PREEMPT_RT kernel; Redundant memory partitions;	ASOA; Mixed OS environment; memory redundancy;	DEC1 - Agreement after discussion; DEC2 - Agreement after discussion; DEC3 - Agreement after discussion; DEC4 - Agreement after discussion; DEC5 - Agreement after discussion;
14	Modelling centralised automotive E/E software architectures	Autonomous; Connectivity; Powertrain (Electrification);	Software complexity; scalability, robustness, maintainability;	Electrification, connectivity, autonomous, over-the-air updates, cloud computing, AI, Neural Networks, wireless protocols / 4G&5G, V2X	Multi-core; memory management; resource partitioning; private memory; private cache; manycore processors;	Redundancy, Spatial and temporal isolation to enable FFI; AUTOSAR Adaptive; SOA;	DEC1 - Agreement after discussion; DEC2 - Agreement after discussion; DEC3 - Agreement after discussion; DEC4 - Agreement after discussion; DEC5 - Agreement after discussion;
15	A Method for designing and analyzing automotive software architecture: A case study for an autonomous electric vehicle	Autonomous; Connectivity; Powertrain (Battery Management System);	Increasing complexity, increasing required CPU and memory resources, decreasing evolvability, safety and real-time, time-to-market, development costs, end product cost reduction. Hard real-time vs. soft real-time vs. no real-time (quality of service characteristics - SOME/IP or DDS) - Decreasing HW dependability.	Autonomous driving, electrification, connectivity, software-defined vehicle, over-the-air updates	CBSPA: 7 layers / hierarchies of automotive software - Sensor/actuator types vs. process types - Hardware dependency on lower layers - Update frequency, computational complexity and agility on higher layers - Serveral combined OSs and platforms with hypervisor - Decouple software functions from hardware	SOA, microservices, software-defined vehicle, centralization/consolidation For FuSa focus: Redundancy, FuSa decomp --> Reduce development costs, timing and memory partitioning for FFI, real-time scheduling with timing partition for mixed-ASIL tasks Good safety-related SW architecture: Higher number of lower or equal rated SWCs than C, low number of connections between FuSa related SWCs, low number of low-ASIL SWCs assessing high ASIL-SWCs Protected dual channel pattern vs. decomposed channel with centralized voter with monitor pattern	DEC1 - Agreement after discussion (only domains used in case study for validation are mentioned); DEC2 - Agreement after consolidation; DEC3 - Agreement after consolidation (agile removed as not a technology); DEC4 - Direct agreement; DEC5 - Agreement after consolidation (SOA, microservices and voter pattern on both sides);
16	Key Technology and Standardization Route for New Electronic and Electrical Architecture of Intelligent and Connected Vehicles	Connected - Connectivity; Intelligence - No functional domain;	Busload / data transmission / bandwidth; complexity of network communication; intelligence; computing performance;	Cloud/Edge computing; connectivity; autonomous driving; V2X; intelligence - interpreted as AI;	multi-core; decoupling of hardware and software to enable cloud computing;	SOA; BCEA (Brain Centralized Electronic and Electrical Architecture); AUTOSAR Adaptive; mixed OS environments; decoupling of hardware and software;	DEC1 - Agreement; Steering as Chassis; DEC2 - Agreement ('intelligent' interpreted as a system limitation related to AI); DEC3 - Agreement ('intelligence' interpreted as technology related to AI); DEC4 - Agreement (no cloud computing as it does not support centralized SW architecture in-vehicle); DEC5 - Agreement after convergence;
17	RACE: A centralized platform computer based architecture for automotive applications	ADAS; autonomous;	Increasing lines of code / SW complexity; Cross-domain communication; high development and testing costs; missing fail-operational behavior; after-market extendibility;	Autonomous driving; electrification; After-market extendibility; centralized platform computer; X-by-wire; high-bandwidth communication backbone; smart I/Os;	Time and space partitioning of applications; flexible assignment of SWCs to computers; middleware-based approach;	SOC - Publisher/Subscriber to decouple communication between applications; IMA; ARINC --> Partitioning to segregate mixed-criticality applications; AUTOSAR also supports partitioning; SIMATIC;	DEC1 - Agreement after alignment (charging as powertrain; parking as ADAS/powertrain); DEC2 - Agreement after consolidation; DEC3 - Agreement after consolidation; DEC4 - Agreement after consolidation; DEC5 - Direct agreement (qualitative discussion of reference patterns);
18	E/E Architecture Synthesis: Challenges and Technologies	ADAS; Autonomous; IVI - Cabin;	Computational power; safety (low latency, safe persistency, safe data transmission); software/application complexity; growing number of ECUs, wiring harness, communication bandwidth, cost, software variants; security;	Central multi SoC HPCs; ADAS and autonomous driving features; AI-based applications/systems; deep and machine learning; gaming in IVI; automotive ethernet; edge and cloud computing; OTA updates; V2V; perception, mapping, planning;	Multi-core; SoCs; middleware; hypervisor; TSN; Type 1 hypervisor - 'direct control and access to hardware resources'; One core per partition to isolate resp. avoid FFI; Type 1 better than Type 2 with host OS; static versus dynamic task mapping;	Software-defined vehicle; Virtualization to integrate safety and non-safety; Task mapping in Multi SoC units: design-time X run-time; Heterogeneous multi-core architectures;	DEC1 - Direct agreement; DEC2 - Agreement after consolidation; DEC3 - Agreement after consolidation; DEC4 - Direct agreement; DEC5 - Agreement after consolidation;

19	HyFAR: A hypervisor-based fault tolerance approach for heterogeneous automotive real-time systems	Autonomous;	Safety / real time / Fault tolerance; updateability; busload / data throughput/bandwidth, computing power;	Autonomous; battery cooling; lane departure warning; cloud connectivity, user connectivity;	Virtual Machines via hypervisors; virtual address spaces; Type 1 (bare metal) versus Type 2 hypervisors - Type 2: If host OS fails, hypervisor fails and all VMs; Full (HW emulation leads to overhead) versus Para Virtualization (highest level of FFI); Cache coloring, memory throttling or Arm's Memory Partitioning and Monitoring (MPAM); Type 1: Exclusive/static resource assignment / partitioning to virtual machines - Highest level of FFI; Type 1: Better performance and efficiency due to direct hardware resources access;	HyFAR; SOA; mixed OS environments; virtualization; freedom from interference; AUTOSAR Adaptive;	DEC1 - Agreement (thermal management as powertrain; lane departure warning as ADAS); DEC2 - Agreement after consolidation; DEC3 - Direct agreement; DEC4 - Direct agreement; DEC5 - Agreement after consolidation;
20	Architectural patterns for cross-domain personalised automotive functions	Comfort - Cabin; IVI - Cabin; ADAS; Intelligence - No functional domain;	Computing power; design complexity; safety; personalization of features;	Integration of machine learning (ML) for e.g. computer vision, automated driving, customer personalization; context-aware functions;	-	-	DEC1 - Direct agreement; DEC2 - Direct agreement; DEC3 - Direct agreement; DEC4 - Direct agreement; DEC5 - Agreement after discussion (future dynamic architecture model is not enabling mixed-criticality systems);
21	Time-of-Flight 3D imaging for mixed-critical systems	ADAS; IC - Cabin; IVI - Cabin;	Computing performance, RAM/Memory	Time-of-Flight 3D image processing	Multi-core; lock-step operation mode;	Redundant compute; mixed-criticality scheduling;	DEC1 - Agreement after discussion; DEC2 - Agreement after consolidation; DEC3 - Agreement after discussion; DEC4 - Agreement after discussion; DEC5 - Agreement after discussion;