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
15	A Method for designing and analyzing automotive software architecture: A case study for an autonomous electric vehicle	powertrain, chassis, ADAS, connectivity, infotainment, central computing unit	safety, real-time performance, high computational power, network bandwidth, evolvability, dependability, cost	autonomous driving, electrified vehicles, connected cars, software-defined vehicle, over-the-air updates, agile	component-based sensor-procesor-actuator (CBSPA) architecture, 7 layers of software	microservices, service-oriented architecture, distributed and centralized voter pattern (for safety)	6 ECU types: personal computing unit, out-vehicle computing unit, in-vehicle computing unit, domain control unit and zone control unit, local control unit, sensor/actuator unit. Mentions real-time scheduling. On point paper, could contain more info than extracted.
16	Key Technology and Standardization Route for New Electronic and Electrical Architecture of Intelligent and Connected Vehicles	autonomous driving, powerdrive, chassis, steering, entertainment, information, body electrics	high speed communication limitation, vehicle communication network	intelligent and connected vehicles; vehicle-road-cloud integration	cloud computing, vehicle operating system	service-oriented distributed heterogeneous computing platform; decoupling of hardware and software; AUTOSAR adaptive; Domain Centralized Electronic and Electrical Architecture; Brain Centralized Electronic and Electrical Architecture	The paper discusses E/E archutectures in general, sw architecture is only one part.
17	RACE: A centralized platform computer based architecture for automotive applications	ADAS, autonomous driving, parking, charging	increasing software complexity, higher complexity of development and configuration; high development and testing costs; need for fail-operational behavior, need for Plug&Play after-market extendibility	centralized platform computer, smart sensors and actuators, high-bandwidth communication backbone; X-by-wire	middleware-based approach (Runtime Environment as middleware); time and space partitioning flexible assignment of software components to computers	Publisher-Subcsriber pattern (topics); in related work: AUTOSAR, ARINC 653, SIMATIC	The paper proposes RACE ICT system architecture based on a centralized ocmputer platform with smart sensors and actuators. RACE platform consists of a centralized platform computer, Ethernet, and a runtime environment.
18	E/E Architecture Synthesis: Challenges and Technologies	ADAS, automated driving; infotainment,	computational power, development complexity; communication bandwidth, cost, sw variants, sw complexity;	central high-performance computing unit (multi-system on chips, GPU, RAM, deep learning accelerators); AI-based systems; edge and cloud backend, cloud computing; automotive ethernet; OTA updates, V2V	hardware virtualization (using a hypervisor),	task mapping in multi-core computing units: design-time X run-time, homogeneous and heterogeneous multi-core architectures	-
19	HyFAR: A hypervisor-based fault tolerance approach for heterogeneous automotive real-time systems	autonomous vehicles; thermal management, lane departure warning	fault tolerance, updateability; data throughput, computational power; real-time, safety, performance	user and cloud integration,	migrating software using virtualization; emulation, virtualization, separation	HyFAR: hypervisor-based fault tolerance approach; service-oriented architecture; AUTOSAR Adaptive	Steps of HyFAR: 1. Fault detection 2. Fault containment and Task recovery 3. Communication reconfiguration The paper mainly describes fault tolerance in service-oriented architectures and signal-oriented architectures.
20	Architectural patterns for cross-domain personalised automotive functions	ADAS, comfort, driving	high computational overhead and design complexity; safety requiremenets, customer demand for comfort and new infotainment systems, ADAS;	AI & ML components; intelligent capabilities and personalised functions; context-aware functions	-	Future Dynamic Architectural Model	The authors propose patterns for ML components: concurrent learning, end-to-end learning, user shadow learning. The core of the work is architecture/patterns for AI-based functions.
21	Time-of-Flight 3D imaging for mixed-critical systems	environment perception; interior monitoring, parking assistance, pedestrian detection;	performance and resource constraints	-	AUTRX platform compatible with standards	Time-of-Flight 3D imaginign solution	A candidate for exlusion. It has nothing to do with what we're searchig for.