Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)
16-18 November 2022, Maldives

Communication Patterns for Evaluating Vehicular E/E Architectures

Elena Lisova
Volvo Construction Equipment
Eskilstuna, Sweden
firstname.lastname@volvo.com

Ruben Broux
University of Antwerp
Antwerp, Belgium
firstname.lastname@uantwerpen.be

Joachim Denil
University of Antwerp
Antwerp, Belgium
firstname.lastname@uantwerpen.be

Alessio Bucaioni
Mälardalen University
Västerås, Sweden
firstname.lastname@mdu.se

Saad Mubeen
Mälardalen University
Västerås, Sweden
firstname.lastname@mdu.se

***Abstract*—The continuous innovation and advancement in vehicle software functionality has driven the evolution of its deployment platforms through several generations of vehicular Electrical and Electronic (E/E) architectures. It is a daunting task to evaluate pros and cons of allocating the new as well as legacy functionality to these architectures. In this paper, we propose a novel approach that uses communication patterns as a metric to evaluate different vehicular E/E architectures and propose a suitable allocation for the functionality. First, we present the characteristics of these patterns in vehicular systems that are derived from the state-of-the-art review, standardized vehicular software architectures, well-known onboard vehicular, communication protocols, industrial requirements and use cases. We leverage the derived communication patterns and their characteristics to propose an evaluation approach for different architectural solutions for the functionality. We utilize a use case from the vehicle industry to demonstrate the applicability and usability of the proposed approach.**

***Index Terms*—Automotive E/E architecture, communication pattern, automotive software.**

I. Introduction

Recent advancement in the vehicle domain can be largely attributed to the advanced software functionality that is deployed on vehicular Electrical and Electronic (E/E) architectures [1]. These E/E architectures are also evolving from the traditional distributed architectures to domain centralized architectures and even to the vehicle centralized architectures that are envisioned for future vehicles [2], [3]. Distributed E/E architectures can include up to 100 in-vehicle computers, also known as Electronic Control Units (ECUs) [1]. Due to the complexity of software distribution and communication among ECUs, distributed E/E architectures are no longer capable of meeting the novel requirements on the advanced vehicle functionality. For example, Advanced Driving Assistance Systems (ADAS) have low-latency and high-bandwidth requirements on the transmission of data that is generated from high-data rate sensors like lidars and radars. In fact, these sensors can generate hundreds of megabytes of data per second. In addition, ADAS have to comply to strict non-functional requirements such as security, functional safety, timing predictability, performance, to mention a few [4].

Domain centralized E/E architectures contain comparatively lower number of ECUs that are more powerful in terms of computation [5]. Furthermore, these architectures allow high-bandwidth and low-latency communication, e.g., based on automotive Ethernet while supporting legacy low-bandwidth communication, e.g., Controller Area Network (CAN) [6]. These architectures are capable of serving the requirements of advanced vehicle functionality more efficiently compared to distributed E/E architectures. Vehicle centralized architectures, regarded as the future architectures, are expected to support high-performance on-board computers, high-bandwidth and low-latency backbone networks like Time-Sensitive Networking (TSN)[1] [7], and over-the-air and cloud services.

Given these generations of vehicular E/E architectures, it is a daunting task, particularly for the industry, to evaluate pros and cons of allocating the new as well as legacy functionality to these architectures. The goals are to lower development cost, allow functionality reuse, and meet non-functional requirements that are specified on the functionality, such as functional safety and security. To address the challenge, this paper proposes an approach to evaluate the vehicle functionality on different E/E architectures based on architectural patterns focusing on communication patterns. The main contributions in this paper are as follows.

- We investigate and present the characteristics of architectural patterns focusing on communication in the vehicular domain. These characteristics are extracted from the state of the art research in the area, standardized automotive

The work in this paper is supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the DESTINE, PROVIDENT, INTERCONNECT, PANORAMA projects; the Swedish Knowledge Foundation via the HERO and DPAC projects; and the Swedish Foundation for Strategic Research (SSF) via Serendipity project. We thank our industrial partner Volvo Construction Equipment.

[1] https://1.ieee802.org/tsn

software architectures, most commonly used onboard communication protocols, industrial requirements and use cases.

- We leverage the patterns and their characteristics to propose an approach to evaluate different E/E architectural solutions for the implementation of the functionality.
- We provide recommendations for efficient allocation of the functionality to the E/E architectures.
- We demonstrate the applicability and usability of the approach on a vehicle industrial use case.

II. Background and Related Work

A. Types of E/E Architecture

There are three different E/E architectures that prescribe three generations of vehicular systems: *Distributed E/E Architectures*, *Domain Centralised E/E Architectures*, and *Vehicle Centralised E/E Architectures* [2].

1) Distributed E/E architectures: Distributed E/E architectures prescribe traditional vehicular systems. These architectures are characterized by a high number of ECUs that are mainly connected via broadcast networks like CAN. These architecture are very modular since every ECU is dedicated to a set of specific functions. However, this results in the software and hardware being highly coupled with each other. In turn, this leads to the problem of vendor lock-in.

2) Domain centralized E/E architectures: These architectures characterize the contemporary vehicular systems. Contrary to the distributed E/E architectures, they focus on software qualities such as scalability and maintainability. This is achieved by using a layered architectural style and by introducing the concept of domain, which groups several ECUs. Examples of some commonly used domains in the vehicular industry include chassis, power-train, body electronics, and ADAS [8]. Each domain is controlled by a domain controller, which also acts as a gateway. ECUs can directly communicate with each other within the same domain. Inter-ECU communication between two different domains takes place via the domain controllers. This drastically reduces the vehicle wire harness. Besides broadcast connections, these architectures are capable of connecting to the internet as well. This mainly happens over a secure gateway and a wireless protocol like for example Long Term Evolution (LTE).

3) Vehicle centralized E/E architectures: These architectures represent the future of vehicle systems. Similar to domain centralized architectures, they use a layered style. However, while domain centralized move towards the direction of domains, these architectures move towards services. Centralized architectures partition the vehicle into zones. A single ECU is assigned to each zone. Zones are controlled from a single vehicle control computer or computing cluster. ECUs only act as a gateway between the central computer and the zone. The central computer may exist out of a High-Performance Computing Unit (HPCU) server and acts as the brain of the vehicle [9]. The HPCU opens up a new dimension for emerging technologies such as artificial intelligence, neural networks, cloud, and over-the-air (OTA) updates [10]. Vehicle centralized E/E architectures bring novel and considerable risks related to security and safety. Vehicles will be exposed to the internet and therefore prone to, e.g., trojan attacks.

B. Communication Pattern

A pattern consists of a set of principles, rules, guidelines or solutions that can be used in a recurring manner in solving a problem or developing functionalities [11]. In other words, a pattern can be used as a reusable solution to a commonly occurring problem. Well-known examples of patterns are pipe and filter communication and layered architectures. In this paper, we focus on communication patterns thus patterns that provide solutions for the vehicle communication. Some examples of general-purpose communication patterns include synchronous patterns such as the HTTP-based REST, the publish-subscribe pattern, asynchronous patterns, message-oriented middle-ware, to mention a few. In contrast to the general purpose patterns, we identify communication patterns tailored for vehicular architectures. A communication pattern can be categorized based on its decomposed components [12].

C. Related Work

In recent years, both researchers and practitioners in software engineering have investigated and elicited patterns focusing on different characteristics, e.g., architectural, communication, and for different application domains. Within the vehicular domain, Schoch et al. [12] proposed a set of five communication patterns, namely beaconing, geobroadcast, unicast routing, advance message dissemination and information aggregation. These patterns were defined after analyzing the envisioned use-case scenarios and unique characteristics of vehicular networks. Similar to our work, Schoch et al. proposed communication patterns based on the envisioned applications (e.g., curve speed warning, blind spot warning, highway merge assistant). However, they did not evaluate different vehicular architectures on the elicited patterns.

In the context of system engineering, Amorim et al. [13] proposed a systematic pattern-based approach, which connects safety and security patterns. They also provided guidance for the pattern selection and demonstrated the application of a combined safety and security pattern engineering workflow on a use case. This work can be seen as complementary to our work as we focus on communication patterns and evaluation of different vehicular E/E architectures based on these patterns.

Washizaki et al. [14] presented a literature review on patterns for systems based on the internet of things. The authors categorized these patterns into architecture styles, architecture and design patterns. Although these mostly offer pre-defined solutions for the design of architectures in general, some of them also focus on communication. For instance, pipe and filters pattern is commonly used when realizing resource-constrained real-time vehicular applications [15], [16].

Similarly, Aksakalli et al. [17] proposed a systematic literature review on deployment of communication patterns focusing on micro services architecture. They identified three types of deployment approaches and seven different communication patterns, namely synchronous, asynchronous, publish/subscribe, combination of HTTP and message queue,

TABLE I
CHARACTERISTICS OF COMMUNICATION PATTERNS CLASSIFIED IN DIFFERENT CATEGORIES.

Location	Intra-vehicle	Inter-vehicle		
Medium	Wired	Wireless		
Gateway Mechanism	Homogeneous	Segmented Homogeneous	Heterogeneous	
Data Transmission	Point-to-Point	Unicast	Multicast	Broadcast
Resource Constraints	Time constraint	Memory constraint	Bandwidth constraint	
Transmission Pattern	Sporadic	Periodic	Hybrid	
Data Exchange	Service-oriented	Signal-oriented		
Security	Integrity	Confidentiality	Authentication	
Functional Safety	PL a	PL b	PL c	PL d / PL e

message-oriented middle-ware, point-to-point and binary protocols. Although Aksakalli et al. identified patterns for micro-service architectures, some of these communication patterns are commonly used in vehicular systems. For instance, point to point and asynchronous. Some of these patterns provide useful input to our work.

III. Characteristics of Communication Patterns

This section presents the characteristics of communication patterns in vehicular systems. These patterns are derived from the review of the state of the art and input from the most common network protocols that are used for onboard communication in the vehicular domain (e.g., LIN, CAN, CAN FD, CAN XL, FlexRay, MOST, LVDS and Automotive Ethernet). Furthermore, exploration of the standardized vehicular software architectures like AUTOSAR [18] and AUTOSAR Adaptive [19]. In addition, we considered useful input based on requirements and studying use cases of our industrial partner in the vehicular domain. Based on our findings, we classify the characteristics of communication patterns into nine different categories as shown in Table I.

A. Identified Characteristics

*1) **Location**:* This category specifies the location of communication infrastructure from the context of a vehicle. Hence, it is further categorized into intra-vehicle and inter-vehicle communication. The intra-vehicle refers to all on-board communication, e.g., communication between on-board ECUs and communication from sensors to actuators. On the other hand, inter-vehicle communication comprises all the communication that vehicle performs with its external environment, e.g., V2V and V2I. Note that this paper focuses only on intra-vehicle communication.

*2) **Medium**:* This category refers to the medium for network communication, which can be wired or wireless. A large majority of in-vehicle communication takes place via wired medium [1], [7]. However, some in-vehicle functionality requires wireless communication, e.g., tyre pressure monitoring functionality. Note that this paper focuses only on wired intra-vehicle communication.

*3) **Gateway Mechanism**:* The gateway provides an interface and a routing mechanism for the communication between different parts of the vehicle architecture, e.g., a gateway between powertrain domain and body electronics domains. If communication between two parts of the vehicle architecture happens over the same network then the gateway mechanism is referred to as the homogeneous mechanism. For example, a gateway used in a multi-switch TSN network where both domains use the same network. Similarly, if a gateway connects two segments of the same network type then it is regarded as the segmented homogeneous mechanism. For example, a gateway that connects two CAN segments. On the other hand, if a gateway connects two different networks then it is regarded as the heterogeneous mechanism, e.g., connecting CAN and TSN networks. In this case, the gateway is responsible for mapping the messages between the two different networks.

*4) **Data Transmission**:* This category describes what type of connections are established for the data transmission. If two devices in the network (e.g., two ECUs or an ECU and a switch) have a unique connection such that no other device can share the same connection, then the data is transmitted through point-to-point transmission. If a device transmits data to only one other device, to a subset of all devices or to all devices in the network then the transmission is regarded as the unicast, multicast or broadcast transmission respectively.

*5) **Resource Constraints**:* This category refers to different types of constraints that are specified on the vehicle functionality. These constraints are classified as bandwidth, memory and timing constraints. The bandwidth constraint constrains the bandwidth that the network should support in order to meet the needs of the functionality. This, in turn, determines the type of communication technology to be used to support the functionality. For example, TSN is more suitable than CAN or Flexray if high-data rate sensors (e.g., lidars and radars) are utilized. The memory constraint constrains the capacity and type of memory required by the functionality. This constraint can be specified on an on-board ECU or on the cloud if the vehicle functionality requires to offload massive computations to a private (enterprise) cloud via 5G or similar network [20]. Similarly, timing requirements in the functionality are specified by means of timing constraints such as deadline, reaction constraint, age constraint, among others [18], [21].

*6) **Transmission Pattern**:* This category indicates how the network messages are triggered for transmission. Hence, a message can be periodic, sporadic or hybrid (mixed). A hybrid message is both periodic and sporadic and is supported by several higher-level protocols for CAN, e.g., CANopen, HCAN and AUTOSAR Comm [22], [23].

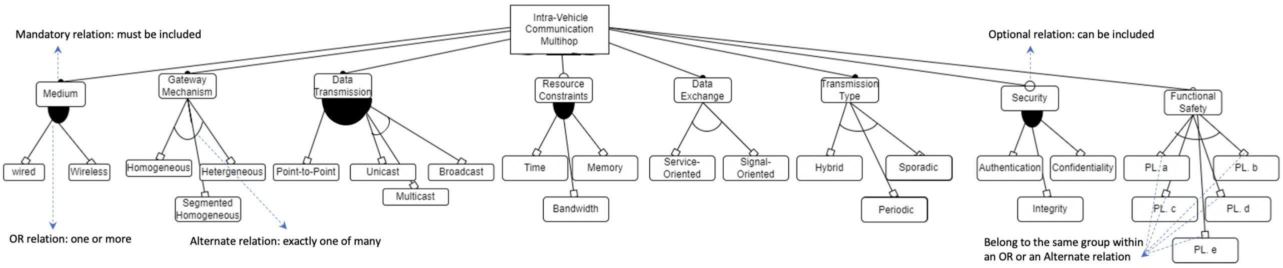

Fig. 1. Feature diagram depicting communication patterns' characteristics and relations.

*7) **Data Exchange Mechanism**:* This category specifies the mechanism that is used to exchange information among various components with the vehicle functionality. This mechanism can be either signal-oriented or service-oriented. Recently, there has been an increasing interest in the service-oriented data exchange mechanism in the vehicular domain [24]. This shift brings a new set of challenges with respect to the vehicle security and safety, especially when getting third-party services.

*8) **Security**:* The category provides an overview of the security requirements and applicable security precautions needed by the vehicle functionality. This category is classified into *Confidentiality*, *Integrity* and *Authentication (CIA)*. Confidentiality is to prevent disclosure of the vehicle functionality's sensitive information from unauthorized people, resources and processes. Integrity refers to the protection of system information or processes from intentional or accidental modification. Whereas, authentication reflects on the assurance that the system is accessible by authorized users when needed [25]. There are several protocols and mechanisms to support confidentiality, data integrity and authentication for the vehicle functionality, e.g., Cyclic Redundancy Check (CRC), parity check, bus guardian, MACsec, CANAuth Protocol, IPsec, TLS, AUTOSAR Secure Onboard Communication, firewalls, honeypots, intrution detection system, to mention a few.

*9) **Functional Safety**:* The functional safety category is based on the safety standard ISO 13849 [26], which is used in the segment of safety machines and construction equipment vehicles. This standard provides safety requirements and guidance on the principles of design and integration of safety-related parts in control systems. ISO 13849 provides five Performance Levels (PL), denoted by a, b, c, d and e, where *PL a* is the lowest level and *PL e* is the highest level. This is in line with the other safety standards like ISO 26262 for road vehicles [27]. ISO 26262 defines four Automotive Safety Integrity Levels (ASILs), denoted by A, B, C and D, where A is the lowest ASIL and D is the highest ASIL. The PLs and ASILs can be maped as follows: ASIL D to PL e, ASIL C to PL d, ASIL B to PL c, and ASIL A to PL b [28]. ISO 13849 provides the probability of dangerous failure per hour (PFH$_d$), which indicates the average probability of a dangerous failure happening per hour. Table II depicts the relationship between the PLs and the PFH$_d$s [29].

In order to provide a visual representation of the characteristics of the communication patterns, we model a feature diagram [30] as depicted in Fig. 1. We focus only on the intra-vehicle communication in this paper. Therefore, we consider only "Intra-Vehicle Communication" sub-category in the location category. This sub-category represents the parent object in the feature diagram and has a mandatory relation with the medium, gateway mechanism, data transmission, data exchange and transmission type characteristics of the communication patterns. This means that these characteristics must be included in the intra-vehicle communication. On the other hand, the parent object has an optional relation with resource constraints, security and functional safety characteristics. This means that these constraints and characteristics can be (but not necessarily) included in the intra-vehicle communication. Each of the medium, resource constraints and security categories has an OR relation with its sub-categories. For instance, a medium can be wired, wireless or both. Similarly, the specified resource constraints can be one or more of timing, memory and bandwidth constraint types. Each of the gateway mechanism, data exchange, transmission type and functional safety categories has an alternate relation with its sub-categories. This means, exactly one of the sub-categories can be included in a given communication pattern. For example, the functional safety constraint specified on a part of the functionality could be only one of the PL's shown in Fig. 1. Another example is that the transmission type of a message can be only periodic, sporadic or hybrid. Note that the data transmission category has a mix of an OR and an alternate relation. This implies that the data transmission can be point-to-point and at the same time it can be one of the unicast, multi-case or broadcast.

One of the notable advantages of using a feature diagram is that it can be extended to capture dependencies between different characteristics of a pattern, e.g., using the CANAuth protocol [31] for security requires having one or more CAN buses as the communication medium, which leads to the broadcast nature of communication. This paper focuses on using feature diagrams for design decisions, however combining feature diagram with requirements tree might be used in future work for addressing requirements elicitation for the functionality implementation.

IV. Proposed Approach and Use-Case Illustration

This section presents an approach to evaluate the distributed, domain centralized and vehicle centralized E/E architectures for a given vehicular functionality based on communication patterns. The proposed approach is graphically depicted in Fig. 2. A vehicular functionality (e.g., engine management, vehicle speed calculation, and many more) complemented

TABLE II
RELATIONSHIP BETWEEN PERFORMANCE LEVEL (PL) AND PROBABILITY OF DANGEROUS FAILURE PER HOUR (PFHD).

PL	**PFH**$_d$
PL a	$\geq 10^{-5}$ to $< 10^{-4}$
PL b	$\geq 3.10^{-6}$ to $< 10^{-5}$
PL c	$\geq 10^{-6}$ to $< 3.10^{-6}$
PL d	$\geq 10^{-7}$ to $< 10^{-6}$
PL e	$\geq 10^{-8}$ to $< 10^{-7}$

with non-functional requirements (e.g., functional safety and security) serves as an input to this approach. A catalogue of communication patterns is explored to identify one or more suitable patterns to which the functionality can be mapped. Note that this work does not focus on deriving the communication patterns, but on how to utilize the patterns catalogue to map the given functionality to evaluate the vehicular E/E architectures. A detailed discussion about the derivation of the communication patterns is presented in [28].

Once the functionality is mapped to a communication pattern, the pattern is evaluated against the provided set of vehicular E/E architectures based on the selected evaluation criteria. In this work, security and functional safety (support for redundancy, hardware/software separation and lock-step execution) are chosen for the evaluation as some of the most critical characteristics of vehicular systems. The evaluation provides the advantages and disadvantages of implementing the given functionality to each of these vehicular E/E architectures. Based on the evaluation results, recommendations are made on the selection of the best suited architecture for the functionality. Furthermore, guidelines are provided for further development of the functionality if a particular vehicular E/E architecture is desired due to the industrial needs.

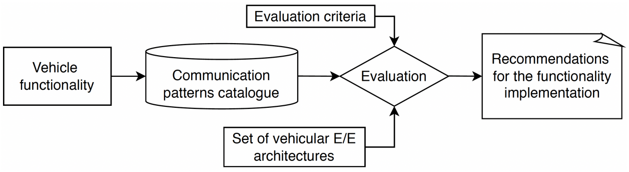
Evaluation criteria
Vehicle functionality
Communication patterns catalogue
Evaluation
Recommendations for the functionality implementation
Set of vehicular E/E architectures

Fig. 2. E/E architecture evaluation for a particular functionality approach.

A. Use Case

In order to illustrate the proposed approach, we consider a vehicular use case that comprises a functionality of speed calculation and its display on the screen for the driver [28]. The functionality requires input data from sensors that are located in the powertrain subsystem of the vehicle. The retrieved data is used to calculate the vehicle speed and, finally, the calculated values are sent to be displayed on the driver's screen. The speed needs to be periodically updated and the updated value is used by other connected functionalities, e.g., data logging. The logical decomposition of the functionality is demonstrated in Fig. 3. A particular implementation of the functionality depends on the vehicle architecture, its functional allocation and typology constraints.

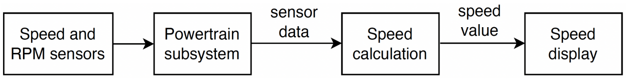

Fig. 3. Logical representation of the speed calculation functionality.

B. Pattern Identification

Given the communication pattern characteristics described in Section III, the speed calculating functionality is characterized as follows. The functionality uses only *intra-vehicle communication* via a *wired medium*. The input and output of the functionality are located in different parts of the vehicle, therefore a *heterogeneous gateway mechanism* is intended. The *multicast data transmission* is required as the calculated speed value is intended to be used by other functionalities in the vehicle, e.g., cruise control functionality. There are *time* and *bandwidth* constraints specified on the functionality. The functionality employs *periodic transmission* and *signal-oriented data exchange mechanism*. The functionality has *security* considerations depending on the threat model used by the vehicular system analyst. The functionality is required to meet safety requirements (PL b) and additional redundancy requirements. The pattern characteristics identified in the functionality are highlighted with green color in Table III.

TABLE III
IDENTIFIED PATTERN CHARACTERISTICS IN THE USE CASE.

Location	Intra-vehicle	Inter-vehicle		
Medium	Wired	Wireless		
Gateway Mechanism	Homogeneous	Segmented Homogeneous	Heterogeneous	
Data Transmission	Point-to-Point	Unicast	Multicast	Broadcast
Resource Constraints	Time constraint	Memory constraint	Bandwidth constraint	
Transmission Pattern	Sporadic	Periodic	Hybrid	
Data Exchange	Service-oriented	Signal-oriented		
Security	Integrity	Confidentiality	Authentication	
Functional Safety	PL a	PL b	PL c	PL d / PL e

We explore the pattern catalogue and identify one communication pattern that is a suitable candidate to map the speed calculation functionality to the vehicular E/E architectures. The pattern consists of three communication entities, namely the sensors and other input, computing platform, and actuators and other output shown in Fig. 4. Fig. 5 depicts the speed calculation functionality mapped to the identified pattern.

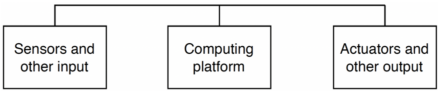

Fig. 4. Communication pattern identified from the pattern catalogue for the speed calculation functionality.

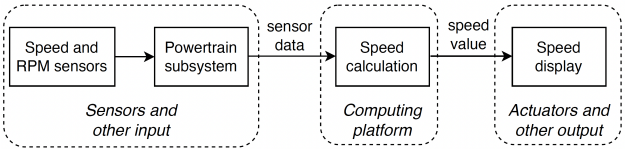

Fig. 5. Speed calculation functionality mapped to the identified pattern.

C. Functionality Allocation to the E/E Architectures

Given the three vehicular E/E architectures considered in this paper, the described functionality could be allocated with each architecture type in a different manner. Fig. 6 demonstrates how the speed calculating functionality can be allocated to a distributed E/E architecture (lines and nodes in bold), one ECU gets the sensor data, provides the calculation of the vehicle speed, and further via a gateway (GW) sends it to another ECU handling the display. Next, in Fig. 7 it is shown how the functionality can be allocated to the domain centralized E/E architecture. In this case, instead of using the gateway node, two domain controllers (DC) communicate the sensor values and corresponding calculated speed values to the ECU that controls the display. Finally, Fig. 8 shows how the speed calculating functionality can be allocated to the vehicle centralized E/E architecture. In this architecture, an input handing node receives the sensors data and then forwards it to the HPCU. The HPCU node, in turn, performs the corresponding calculation and forwards the calculated value to the ECU that controls the display. These allocations reflect the main concepts of the three vehicular E/E architectures. However, these allocations do not represent the only possible solutions for the functionality allocation, and thus further allocations are also possible based on other functional and non-functional requirements of the given functionality.

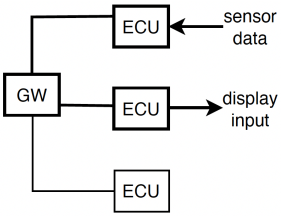

Fig. 6. Functionality allocation to the distributed E/E architecture considering the identified pattern.

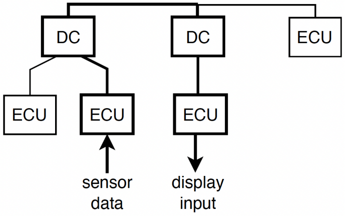

Fig. 7. Functionality allocation to the domain centralized E/E architecture considering the identified pattern.

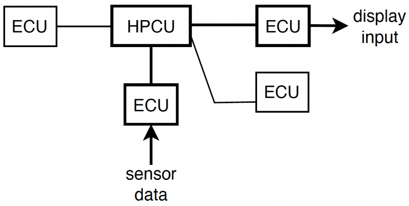

Fig. 8. Functionality allocation to the vehicle centralized E/E architecture considering the identified pattern.

V. Evaluation

This section evaluates the functionality allocation to the three vehicular E/E architectures using the communication pattern presented in Fig. 4. We focus on functional safety and security in the evaluation.

A. Safety Requirements and their Implications

The functionality in the use case has a functional safety requirement specified as PL b according to ISO 13849. Note that PL b is similar to ASIL C in ISO 26262. There are many factors that are required to fulfill the requirements associated to a safety level. We do not consider the process aspects, but focus on the functionality allocation to the E/E architectures. Hence, the main aspects for the required safety level considered in this paper include redundancy in communication and separation of hardware and software resources between functionalities with different criticalities.

In the scenario, where the functionality is allocated to a distributed E/E architecture as shown in Fig. 6, the redundancy requirement refers to a redundant communication mechanism among the ECUs that are connected to the corresponding sensor and display unit. This can be achieved by using two redundant CAN buses to connect all the ECUs in Fig. 6.

When the functionality is allocated to the domain centralized E/E architecture as shown in Fig. 7, the communication redundancy needs to be supported up to the two domain controllers. This means, that both intra-domain networks as well as the inter-domain network should have redundancy. Consequently, this would require redundant CAN buses within each domain and a redundant CAN bus or redundant automotive Ethernet links between the two domain controllers. Hence, a lot more redundancy in the networks is required compared to that of the distributed E/E architecture.

Finally, considering the functionality allocation to the vehicle centralized E/E architecture as shown in Fig. 8, the redundancy is required for communication between the ECUs and HPCU. This type of E/E architecture naturally implies less communication links as the majority of the functionality is allocated to the HPCU. As the vehicle centralized E/E architectures are expected to use high-bandwidth and low-latency networks like TSN, a redundant path in the same network can be used for redundancy. Thanks to the frame-replication mechanism supported by TSN [32], it is also possible to use redundancy in the transmission of network messages. This allows to address temporary faults in the communication network.

Integrity of the network messages is required to support the system's safety as well as system's security. This can be achieved by end-to-end protection protocols, e.g., AUTOSAR supports end-to-end protection mechanism on CAN and has different profiles described for it [33], [34]. These mechanisms are mostly realized with software and have limited implications on the overall topology and architecture. However, these mechanisms cause communication overhead in terms of the amount of extra signals to consider for communication.

Each functionality within the computing platform (ECU, Gaeway node, Domain controller or HPCU) can be associated to a particular criticality level [15]. These applications would require separation of software and hardware resources from each other to prevent a lower-criticality functionality to interfere with a higher-criticality functionality. In order to isolate the speed calculation functionality from other functionalities, each functionality can be allocated to a separate partition that provides separation in time and space from the other partitions on the same core or a set of cores within the same computing platform. Furthermore, lock-step execution that is supported by many modern computing platforms[2] can be utilized to support correctness of execution as well as correctness of the time when the execution is completed for the functionalities that are allocated to different cores within the same computing platform.

B. Security Requirements and their Implications

Traditionally, security concerns have not been in focus for the automotive domain following the assumption of a vehicle being an isolated system that is difficult to breach. However, with the increased outer vehicle connectivity, there are more and more cases of security breaches in the automotive domain [35]. Given that the main vehicle vulnerabilities lie in its connectivity, security architecture is often structured around the telematics unit or domain. The security requirements can be addressed by using *security zones* [36], a firewall placed within the telematics unit and/or using an intrusion detection system (IDS) located in a position that allows extended traffic monitoring. Security zones are zones defined on logical and/or hardware/software levels with a rationale of having different security requirements within the zones. Such structure implies additional security measures on the boarder of the zones. Usually, structuring of zones depends on location of connectivity solutions and allocation of critical functionalities.

A vehicle centralized E/E architecture suggests placing firewall and IDS in the HPCU due to its high computational capability. A zone, in this case, could be structured by subnetworks around the HPCU or centrally around it. In the case of domain centralized E/E architectures, telematics could be separated as a corresponding domain and its domain controller can be the firewall location and possible center of zones structure. In the case of the distributed E/E architectures, a telematics ECU most probably becomes a location for supporting firewall, while the IDS could be placed at other strategical location within the vehicle. From the perspective of the functionality allocation, whether the functionality crosses over different security zones, requires to go through the firewall and/or its location relative to the IDS (i.e., how much IDS can monitor the functionality) need to be considered.

C. Discussion

This subsection provides recommendations for the functionality allocation to the vehicular E/E architectures. The recommendations are based on the above evaluation.

The functionality allocation to the vehicle centralized E/E architecture has a lesser communication overhead as compared to the rest of the architectures. However, this architecture is envisioned as the future vehicle architecture. The functionality allocation to this architecture could be a step-wise process in which the contemporary domain controllers can be first integrated with the HPCU. Later on, the functionalities of the domain controllers can be migrated to the HPCU, thereby eliminating the need for the domain controllers. An example of such a hybrid solution is the Human Machine Interface (HMI) domain controller that could co-exist with the HPCU in the first step in the step-wise allocation process. In this way, the legacy functionality and legacy subsystems in the vehicle can be better supported in gradual evolution of the functionality allocation to the future E/E architectures.

As it was observed in Section IV-B, the input and output of a functionality can be located in different parts of the vehicle, e.g., powertrain and HMI respectively. The HMI unit serves as an interface for the vehicle's connectivity with the driver's nomadic devices. Hence, a dedicated security zone needs to be centered around the HMI unit. Therefore, our assumption is that the use-case functionality is likely to be allocated across more than one security zone. This, in turn, implies that one of the ECUs on which the functionality is allocated serves as the border between two or more security zones. This ECU needs to implement security mechanisms that support authentication and confidentiality.

Apart from the HMI unit, a vehicle is likely to have other points of connectivity with external devices, e.g., a telematic unit that may be located within the HPCU for communication with back office or road infrastructure. In this case, one more security border might be needed between the HPCU and the HMI security zones, which could be realized by the HMI domain controller that implements suitable security mechanisms. In crux, the potential candidates to implement these security mechanisms are gateway ECU in the distributed E/E architecture, domain controller in the domain centralized E/E architecture, and HPCU in the vehicle centralized E/E architecture. These security mechanisms require heavy computations due to which they are more suited to the HPCU because of its higher computational resources compared to ECUs and domain controllers. Hence, the vehicle centralized E/E architectures are inherently more suited to support computationally complex security mechanisms in vehicles.

Although we focused only on the functional safety and security characteristics of the communication patterns in the evaluation, there can be other characteristics such as timing, bandwidth and memory constraints that can also impact on the functionality allocation to various E/E architectures. Evaluation of these characteristics is left for future work.

VI. Conclusions

Allocation of vehicular functionality that comprises new advanced features as well as legacy software functions to a generation of vehicular E/E architectures, while considering non-functional requirements, is a challenging task. It is of

[2] https://www.infineon.com/cms/en/product/microcontroller/32-bit-tricore-microcontroller

utmost importance for the industry to evaluate pros and cons of allocating the functionality to these architectures. In this paper, we leveraged the abstraction of communication patterns to propose an approach to evaluate the functionality allocation to different vehicular E/E architectures. In this regard, we derived the characteristics of these patterns in vehicular systems from the state of the art, standardized vehicular software architectures, most common onboard communication protocols, and industrial requirements and use cases. We used an industrial use case, comprising the speed calculation functionality, to demonstrate the applicability and usability of the proposed approach. The functionality, based on its logical decomposition and dependencies, was mapped to a corresponding communication pattern, which was used to evaluate the the functionality allocation to the three vehicular E/E architectures. The evaluation results indicate that the vehicle centralized E/E architecture allows the most efficient allocation from communication and security points of view. In the case of Human Machine Interface (HMI) applications, a hybrid allocation, where the vehicle centralized E/E architecture is possibly complemented with a domain controller, seems to be a promising solution while taking into account the legacy, functional safety and security considerations.

References

[1] L. Lo Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, 2019.

[2] P. Pelliccione, E. Knauss, R. Heldal, S. M. Ågren, P. Mallozzi, A. Alminger, and D. Borgentun, "Automotive architecture framework: The experience of volvo cars," *Journal of Systems Architecture*, vol. 77, pp. 83 – 100, 2017.

[3] O.Burkacky, J.Deichmann, G. Doll, and C. Knochenhaue, "Rethinking car software and electronics architecture." McKinsey Company, New York, NY, USA, Tech. Rep., 2018. [Online] https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/rethinking-car-software-and-electronics-architecture.

[4] S. Mubeen, E. Lisova, and A. V. Feljan, "Timing predictability and security in safety-critical industrial cyber-physical systems: A position paper," *Applied Sciences*, vol. 11, pp. 1–17, April 2020.

[5] T. Wendt, W. Bernhart, J. Behl, D. Mishoulam, and E. Goldsmith, "consolidation in vehicle electronic architectures," July 2015.

[6] Robert Bosch GmbH, "CAN Specification Version 2.0," Postfach 30 02 40, D-70442 Stuttgart, 1991.

[7] M. Ashjaei, L. Bello, M. Daneshtalab, G. Patti, S. Saponara, and S. Mubeen, "Time-sensitive networking in automotive embedded systems: State of the art and research opportunities," *Journal of Systems Architecture*, vol. 117, p. 102137, 2021.

[8] M. Mody, J. Jones, K. Chitnis, R. Sagar, G. Shurtz, Y. Dutt, M. Koul, M. Biju, and A. Dubey, "Understanding vehicle e/e architecture topologies for automated driving: System partitioning and tradeoff parameters," *Electronic Imaging*, vol. 2018, pp. 1–5, 01 2018.

[9] A. Bucaioni and P. Pelliccione, "Technical architectures for automotive systems," in *2020 IEEE International Conference on Software Architecture (ICSA)*, pp. 46–57, 2020.

[10] V. Bandur, V. Pantelic, M. Dawson, A. Schaap, B. Wasacz, and M. Lawford, "A domain-centralized automotive powertrain e/e architecture," in *SAE WCX Digital Summit*, April 2021.

[11] L. Bass, P. Clements, and R. Kazman, *Software architecture in practice*. Addison-Wesley Professional, 2003.

[12] E. Schoch, F. Kargl, M. Weber, and T. Leinmüller, "Communication patterns in vanets," *Communications Magazine, IEEE*, vol. 46, pp. 119 – 125, December 2008.

[13] T. Amorim, H. Martin, Z. Ma, C. Schmittner, D. Schneider, G. Macher, B. Winkler, M. Krammer, and C. Kreiner, "Systematic pattern approach for safety and security co-engineering in the automotive domain," in *International Conference on Computer Safety, Reliability, and Security*, Springer, 2017.

[14] H. Washizaki, S. Ogata, A. Hazeyama, T. Okubo, E. B. Fernandez, and N. Yoshioka, "Landscape of architecture and design patterns for iot systems," *IEEE Internet of Things Journal*, 2020.

[15] A. Bucaioni, S. Mubeen, F. Ciccozzi, A. Cicchetti, and M. Sjödin, "Modelling multi-criticality vehicular software systems: evolution of an industrial component model," *Software and Systems Modeling*, vol. 19, no. 5, pp. 1283–1302, 2020.

[16] A. Bucaioni, S. Mubeen, J. Lundbäck, K.-L. Lundbäck, J. Mäki-Turja, and M. Sjödin, "From modeling to deployment of component-based vehicular distributed real-time systems," in *2014 11th International Conference on Information Technology: New Generations*, pp. 649–654, IEEE, 2014.

[17] I. K. Aksakalli, T. Çelik, A. B. Can, and B. Tekinerdoğan, "Deployment and communication patterns in microservice architectures: A systematic literature review," *Journal of Systems and Software*, 2021.

[18] The AUTOSAR Consortium, "Autosar technical overview, version 4.3.," (2016). http://autosar.org.

[19] The AUTOSAR Consortium, "Autosar apadtive platform, release r21-11," (2021). https://www.autosar.org/standards/adaptive-platform/.

[20] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold, K. Sandström, and M. Behnam, "Delay mitigation in offloaded cloud controllers in industrial iot," *IEEE Access*, vol. 5, pp. 4418–4430, 2017.

[21] S. Mubeen, T. Nolte, M. Sjödin, J. Lundbäck, and K.-L. Lundbäck, "Supporting timing analysis of vehicular embedded systems through the refinement of timing constraints," *Softw. Syst. Model.*, vol. 18, 2019.

[22] S. Mubeen, M.-T. Jukka, and S. Mikael, "Extending schedulability analysis of controller area network (can) for mixed (periodic/sporadic) messages," in *ETFA2011*, pp. 1–10, 2011.

[23] S. Mubeen, J. Mäki-Turja, and M. Sjödin, "Integrating mixed transmission and practical limitations with the worst-case response-time analysis for controller area network," *Journal of Systems and Software*, vol. 99, pp. 66–84, 2015.

[24] M. Rumez, D. Grimm, R. Kriesten, and E. Sax, "An overview of automotive service-oriented architectures and implications for security countermeasures," *IEEE Access*, vol. 8, pp. 221852–221870, 2020.

[25] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[26] International Organization for Standardization (ISO), "ISO 13849-1:2015: Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design," 2015.

[27] International Organization for Standardization (ISO), "ISO 26262: Road Vehicles - Functional Safety," 2011.

[28] R. Broux, E. Lisova, and S. Mubeen, "Communication patterns in automotive systems," tech. rep., March 2022.

[29] A. Porras-Vázquez and J.-A. Romero-Pérez, "A new methodology for facilitating the design of safety-related parts of control systems in machines according to iso 13849:2006 standard," *Reliability Engineering and System Safety*, vol. 174, pp. 60–70, June 2018.

[30] K. Czarnecki and A. Wasowski, "Feature diagrams and logics: There and back again," in *11th International Software Product Line Conference (SPLC 2007)*, pp. 23–34, 2007.

[31] A. V. Herrewege, D. Singelée, and I. M. R. Verbauwhede, "CANAuth - A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus," in *ECRYPT Workshop on Lightweight Cryptography*, 2011.

[32] IEEE TSN Task Group, P802.1CB – Frame Replication and Elimination for Reliability, https://1.ieee802.org/tsn/802-1cb/.

[33] AUTOSAR, "E2E Protocol Specification," 2017.

[34] The AUTOSAR Consortium, "Specification of secure onboard communication, version 4.3.1," (2017). http://autosar.org.

[35] R. E. Haas and D. P. F. Möller, "Automotive connectivity, cyber attack scenarios and automotive cyber security," in *IEEE International Conference on Electro Information Technology*, pp. 635–639, 2017.

[36] M. Kern, E. Taspolatoglu, F. Scheytt, T. Glock, B. Liu, V. P. Betancourt, J. Becker, and E. Sax, "An architecture-based modeling approach using data flows for zone concepts in industry 4.0," in *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pp. 1–8, 2020.